Developing Design Recommendations for Computer Interfaces Accessible to Illiterate Users

Matthew Paul Huenerfauth, M.S., H.B.S.

National University of Ireland
University College Dublin
Faculty of Science

Master of Science (M.Sc.)
August 2002

Department of Computer Science
and
Media Lab Europe

Head of Department: Gregory O'Hare
Supervisor: Fred Cummins, Ph.D.

Table of Contents

Acknowledgements 4

Abstract 5

Introduction: Overview and Methodology 6
- Purpose of the URSULA project 6
- Scope of this Thesis Work 7
- Research and Design Methodology 7
  - Previous Research Areas Consulted 7
  - Timeline of Thesis Research 8
- Structure of this Document 9

Chapter 1: Users and Environment 11
- The Users: Digital Divide and Technological Literacy 11
  - Digital Divide 11
  - Telecenter Movement 12
  - Challenges to Technology Use 12
  - Previous Technology and Communication Experience 13
  - Promise of the Simputer Project 14
- The Users: Written Language Literacy 15
  - Definitions and Statistics 15
  - Failings of Current Interfaces for This Population 16
  - Effects of Adult Literacy in Developing Regions 16
- The Environment: Rural Indian Community Life 17
  - Way of Life in Rural Communities 18
  - Diversity of Indian Languages 19

Chapter 2: Technology and Work 21
- The Technology: The Simputer 21
  - Form Factor and Component Technologies 21
  - Information Markup Language 22
- The Work: Important Applications 23
  - Health Information Distribution 24
  - Employment Listings 25
  - Online Government Services 26
  - Agriculture Information Application 27

Chapter 3: Previous User Interface Work 28
- Previous Work in Accessible Interfaces 28
- Interfaces without Visual Displays 29
- Previous Work on User Interfaces for Illiterate Users 30
  - Goetze & Strothottle: Illiterate Web Browser 30
  - EZ Access Touch Screen Kiosks 32
  - Other Software for Developing Countries 33

Chapter 4: Initial Interface Considerations 35
- Building Interfaces for Novice Users 35
  - Memory and the Interface 36
  - Encouraging Exploration 37
  - The Role of Feedback 38

Input and Output Channels ... 38
Touch Screen Input ... 39
Speech Input and Output ... 40
Audio Output Channel ... 46
Visual Output Channel ... 48
Dialogue Structuring Approaches ... 53
Question and Answer ... 54
Menus ... 55
Form-filling ... 56
Command Language ... 58
Direct Manipulation ... 58
Natural Language ... 59

Chapter 5: HUDS Introduction and Scenario 1 ... 61
Approaches to User-Interface Design ... 61
Experimental Approaches ... 62
Non-Experimental Approaches ... 63
Hypothetical User Design Scenarios ... 65
HUDS 1: Distributing Health Information ... 67
Synopsis ... 67
Entities and Characters in the Scenario ... 67
The Scenario Script ... 69
Analysis of Issues in the Scenario ... 74
Areas of Future Research Highlighted ... 78

Chapter 6: Hypothetical User Design Scenario 2 ... 80
Issues Not Explored in HUDS 1 ... 80
HUDS 2: Finding Employment ... 81
Synopsis ... 81
Entities and Characters in the Scenario ... 81
The Scenario ... 83
Analysis of Issues in the Scenario ... 88
Future Areas of Research ... 97

Chapter 7: Conclusions and Recommendations ... 98
General Conclusions ... 98
Advice for Designers ... 101
Structure of the Recommendations ... 103
Command and Control ... 104
Start Page / Portal Screen ... 104
Command Option List ... 105
Users Entering Information ... 106
Forms ... 106
Yes/No and Short Multiple Choice ... 107
Long Multiple Choice ... 107
Geographic Information ... 108
Input of Free Text ... 108
Displaying Information to Users ... 109
Index View: Presenting Lists ... 110
Detail View: Navigating Multiple Pages ... 111
Images and Icons ... 112

Continuous System Status ... 112
Event Notifications ... 113
Assistance to Users ... 114
Help with Reading ... 114
Educating Users ... 115
Resources and Capabilities ... 116
Personalization and Identification ... 116
Email and Bulletin Boards ... 117
Libraries of Resources for Developers ... 118
Storing Information for Later Use ... 119
Offline Mode ... 120
Chapter 8: Experimentation and Future Directions ... 121
Reasons for Experimentation ... 121
Planning for Future Development Work ... 122
Stage 1: Initial Experiments and Design Specification ... 123
Stage 2: Building, Laboratory Testing, Re-Design of Prototype ... 124
Stage 3: Field-Testing of Prototype System ... 126
Stage 4: Toolkit Production and Distribution ... 127
Final Comments ... 127
Bibliography: Works Cited ... 129
Bibliography: Works Consulted ... 134

Acknowledgements

I would like to thank my advisor Dr. Fred Cummins for his assistance during the research and thesis-writing process. In addition to making himself available for any of my questions or concerns, he provided me with resources and opportunities to further my research and publish my findings.

My family and friends in the United States deserve thanks for their support and encouragement while I spent this year abroad to perform my research. My fellow Mitchell Scholars and friends in Ireland have helped me find roots in Dublin and make my year the exciting and challenging adventure I had hoped it would be.

Funding for this research has been provided by a George J. Mitchell Scholarship administered by the US-Ireland Alliance (www.us-irelandalliance.org) and by an Irish government HEA grant to Dr. Fred Cummins, Department of Computer Science, University College Dublin.

Abstract

User-Interface Recommendations Supporting Universal Literacy Accessibility (URSULA) is a project to create user-interface guidelines for developers who are writing applications and websites that may be used by illiterate users. In particular, these guidelines would be useful for developers of devices for developing communities, such as the Simputer, a small information access device to be distributed in India. Because of low literacy levels among the target population for this device and the diversity of languages used throughout India, special consideration must be exercised when designing applications such that they are understandable by users who lack written language literacy. Interfaces that use speech and dialogue interaction, display non-linguistic graphics, adapt to the literacy level of the user, and accept input by microphone and touch screen would be well-suited to the domain. This thesis discusses how previous work in user interface design can be applied to this population, and it traces how a conceptual interface design approach, Hypothetical User Design Scenarios (HUDS) can be used to drive the design process.

Introduction: Overview and Methodology

This chapter introduces the reader to the URSULA project and explains the content of this thesis. The research goals of the URSULA project are outlined, and the scope the entire project and of this initial stage are differentiated. This chapter goes on to explain the methodology used during this UI design process – highlighting which areas of previous research were consulted, what design approaches were considered, and how the guidelines and recommendations in this thesis were produced. Finally, this introduction presents an overview of the remaining chapters of the thesis to help the reader understand their organization and locate sections of particular interest.

Purpose of the URSULA project

The purpose of the User-Interface Recommendations in Support of Universal Literacy Accessibility (URSULA) project is to create a set of guidelines for developers of computer user interfaces who are interested in making systems accessible to illiterate users [Huenerfauth 2002]. This work is of particular interest to developers for the Simputer [Simputer 2001] [Encore 2002] platform (further explained in Chapter 1), and much of the design considerations in this thesis focus on building accessible interfaces for this device. While the Simputer project is used as a research vehicle, the recommendations made in this project are meant to be generalizable to other form-factors and devices. The output of this thesis research is an easy-to-follow set of guidelines for producing an illiterately accessible user interface, a set of recommendations and advice to application and web developers for the Simputer platform, and a discussion of the future directions of the URSULA project.

Scope of this Thesis Work

The research done as part of this thesis has highlighted the complexities of designing user interfaces for illiterate populations and making these interfaces compatible with the Simputer form-factor and usage environment. Because of this complexity, this thesis comprises only a preliminary investigation into the issues and user-interface approaches that need to be considered for the URSULA project's successful completion. For this reason, this work emphasizes areas of inquiry which will need to be explored further, highlights issues that are not yet settled in the design of the interface, and lays out a plan for future work on the URSULA project after the 2001-2002 academic year. It is anticipated that URSULA will grow beyond this initial research project into a working group at University College Dublin exploring the theoretical and implementation issues involved in the production of speech user interfaces.

Research and Design Methodology

Previous Research Areas Consulted

The initial goal of this project was to survey previous work applicable to designing user-interfaces for illiterate users. Human computer interaction literature suggested approaches that could be taken during the design and evaluation process of the user interface. Multimodal and mixed-initiative interface projects suggested challenges which users typically face when navigating an interface incorporating speech and dialogue elements. Research work specifically focused on natural language based interfaces helped highlight the promises and current limitations of this technology for users unfamiliar with direct manipulation GUIs. Current results in speech synthesis and recognition technology were reviewed to understand the challenges of incorporating these capabilities into an interface and to understand the performance and accuracy which can be expected. The growing field of accessible interface design provided a foundation to understanding how to design interfaces for users with special needs, and previous work in screen reading technology for the blind was particularly applicable to this project.

Because the Simputer project involves a target user population much different than that of most HCI research (office workers or home computer users), materials from a diverse set of other disciplines were also consulted during the initial stages of this project. The findings of other technological development projects for developing communities helped focus the core usage applications of the Simputer device and help to narrow the user interface concerns of particular interest. Information on the language diversity of Indian provinces, the cultural and economic situation of the Simputer target population, the current status of the Simputer project, and the special concerns of adults with literacy challenges were also important when conceptualizing the situation of future users of this URSULA-inspired interface.

Timeline of Thesis Research

This research project into illiterately accessible interfaces began with a review of relevant articles in the HCI, design, and speech interfaces literature during the autumn of 2001. Previous information technology work in developing countries was also consulted to find which projects were most successful in these countries and which applications of this new technology were of the highest priority. By the end of 2001, an analysis of the users, technology, applications, and working environment of the device was conducted. A survey of previous work in speech-based interfaces, dialogue structuring, and multimodal issues was used to produce an initial set of design recommendations in early 2002.

After identifying general interface guidelines from this literature survey, two Hypothetical User Design Scenarios (described in chapters 5 and 6) were employed in the spring 2002 to create more specific recommendations for the interface. In the summer of 2002, high-level planning for the future directions of URSULA after this thesis were created, including plans for user interface experiments to test alternative user interface designs and inform specific critical design decisions.

Structure of this Document

The remainder of this paper highlights important issues in the creation of an illiterately accessible user interface and gives an initial set of guidelines for this work. This thesis approaches this task from a theoretical to a progressively more experimental manner.

Chapters one to three consider interface design issues using a survey of previous work, an analysis of the situation of the users and their information needs, and a consideration of the available design technologies. Chapter one discusses the users of the interface and their living environment. This chapter explores literacy-related issues, the culture and language of the Indian communities, and issues in its deployment. Chapter two discusses the technology involved in the Simputer project and the tasks it should enable for its users. Chapter three is a survey of previous work in the field of UI design which impinges on URSULA; accessible interfaces, speech-based interaction, and previous tools for illiterate users are all discussed.

Chapters four to six begin to create guidelines for an illiterately accessible interface; in these chapters, non-experimental methods are employed to produce these recommendations. In chapter four, the elements of an interface input and output are decomposed and considered in light of the special needs of illiterate users. Previous work in speech UI, dialogue-based interaction, and pictorial interfaces is also incorporated into this analysis. Chapter five and six contain two narrative scenarios that consider a hypothetical interaction between a user with a particular information need and the Simputer device.

Chapters seven and eight summarize the findings of the design work to date and propose directions that the project can take in the future. Chapter seven contains conclusions about the work and an organized summary of the recommendations for the illiterately accessible user interface. Whereas previous chapters of this thesis will mention these recommendations as they arise in the course of discussion, this chapter makes an effort to organize and index these findings into a manner useful for a Simputer application designer. Chapter eight discusses the future directions of the URSULA project, including how to use user-interface experiments to make design decisions for URSULA.

Chapter 1: Users and Environment

In the model of Human Computer Interaction design laid out by [Eason 1991] and adapted by [Preece 1995], there are four sets of project-specific issues that must be carefully considered at the start of a successful design process: the users of the system, the environment in which it will exist, the work which it will help accomplish, and the technology which it will employ. This conceptual framework will be used to structure the initial discussion of the Simputer interface design. This chapter will discuss issues relating to the users and the environment, and chapter two, the work and the technology.

The Users: Digital Divide and Technological Literacy

By studying the special needs of the target population, the designer can better anticipate the behavior of users when writing Hypothetical User Design Scenarios (HUDS) interaction scripts later in the design process. While URSULA's focus is combating language literacy barriers to technology, the typical Simputer user also faces technological literacy challenges.

Digital Divide

The digital divide is a recent term used to describe the increasing disparity between the amount of access members of the developed and developing world have to information technology. In most cases, this amount of access is many times greater than the differences in economics between these populations. Many statistics show that although there are over 1.5 billion web pages, 2 million being added each day, and there are 200,000 devices being added to the Internet each day, 95% of the world population is not participating in this new information technology growth [Colle 2001b].

> "The important point about the digital divide is not that some people have the technology and others do not, but that not having it puts people at a disadvantage and cuts them off from participation in important economic, social, cultural, and political activities." [Cooper 2000]

In a world in which an increasing number of employment opportunities, commerce, health information, government services, and communications are available through electronic channels, this "digital divide" means that some of the most disadvantaged members of the worlds population are being left even further behind.

Telecenter Movement

To combat the growth of the digital divide in the 1990s, many international organizations funded efforts to expand the reach of information technology to developing communities who might not have otherwise been able to afford such access. A major approach the International Telecommunications Union, the United Nations, the World Bank, and other groups have taken to counteracting this problem is to establish "telecenters," small office-like spaces with telephones, computers, and a good Internet connection in developing communities throughout the world [Colle 2001b]. While the Simputer Trust is using a different approach to promote information technology access (one based on small devices owned by end-users), studies of the success of previous telecenter projects has helped illuminate the nature of the communities into which this new technology will be introduced.

Challenges to Technology Use

One of the major challenges of these telecenter projects has been educating the communities in which they exist of the value of information and communication technologies [Colle 2001a]. When there are many other pressing basic needs such as hygiene, sanitation, and safe drinking water that are threatened in such communities, convincing community members that their lives can be enriched and improved through greater access to information and communication can be a challenge. Another important issue affecting technology uptake can be technophobia. With some studies showing that over half the world's population has never made a telephone call [Colle 2000], the idea of using a computer or handheld computing device can understandably be an intimidating prospect. Even when users in developing areas gain access to Internet resources, they may become frustrated at the lack of information in local dialects or pertinent to their region.

Previous Technology and Communication Experience

One aspect of modern technology has already infiltrated rural Indian communities – the television. The Indian government began a campaign to provide television sets located in central public locations in many towns and villages to increase access to news and information [Singh 1993]. This is useful to know because it means that the target users already have some comfort level with three aspects of the Simputer interface paradigm:

- Going to a central community location to receive information from a shared electronic device.
- Viewing an electronic screen with visual information.
- Listening to audio information from an electronic device that compliments the visual on-screen information.

Even in those areas where television has not yet infiltrated, other forms of technology and communication channels have made inroads. Radio is available throughout several regions of the developing world where television has not yet reached. While lacking a screen, users of a radio would be acclimated to listening to information from an electronic device.

Although postal services may be available only sporadically throughout some countries, often rural residents can travel to a regional postal center to send written messages to people in other areas. At a basic level, this experience has prepared users for the e-mail paradigm which may be part of an URSULA interface.

Promise of the Simputer Project

The Simputer Trust is an organization founded by several Indian researchers to develop and market a low-cost hand-held Internet-accessible computing device to combat the growing digital divide in rural communities in India [Simputer 2001] [Encore 2002]. The vision of the project is to produce affordable Internet access clients that would be personalizable and sharable among large groups of rural Indians such as extended families or small villages. Major foci of the project include taking advantage of the communication capabilities of the device and the many commercial and government services it could help users access [Encore 2002]. Since the Trust is licensing its design to various companies who will produce and market Simputer devices, much of the advertisement of the device to the Indian population has yet to occur. (Sale of the devices is planned to begin in the summer or autumn of 2002.) As the marketing for the devices is begun in earnest, application and interface designers should take note of these advertisements to understand the expectations users will have before ever operating their first Simputer.

The Users: Written Language Literacy

The most significant feature of the URSULA project which distinguishes it from other interface design projects is its focus on users who lack written language literacy skills. Consequently, it's only natural that an analysis of these users begins with a discussion of literacy issues. Not only is it important to define what skills users of this interface will or will not possess, but the implications of illiteracy on various other aspects of the users' lives will help the designer gain an appreciation for the context in which they will be interacting with the device.

Definitions and Statistics

Modern definitions of literacy include such skills as reading, writing, and counting in one's primary language, taking advantage of information sources in this language, and being able to compose short meaningful statements [Chlebowska 1990] [Street 1984] [Baynham 1995]. These definitions of literacy typically extend the colloquial definition of literacy: the ability to read text in one's language in a written form. No matter exactly which skill sets a particular definition of literacy includes, in practice, individuals who lack the ability to read text generally lack these other associated skills. So, when designing an illiterately accessible interface, the designer must be careful not to assume abilities beyond those of the target population. For example, many users with literacy challenges would also have difficulty with counting and numerals. So, designing the audio portion of an interface such that it makes reference to screen elements according to their numerical order could pose challenges for these users.

Because of economic challenges and reduced access to educational opportunities, literacy is a major concern in India. According to the 2001 Indian census, 45% of the adult population is illiterate, increasing to 51% in rural areas [India 2001]. Rates of illiteracy for rural women are more alarming, with reports varying from 61% to 82% [Chlebowska 1990] [India 2001]. Some of this gender difference is caused by a cultural tendency to emphasize education and resources for male children; consequently, even if a girl has some access to educational opportunities, she may be removed from school at an earlier age.

Failings of Current Interfaces for This Population

With such a large portion of the population unable to read, previous telecenter approaches have been limited in their success at bringing information to communities. With standard PCs and Internet access, an illiterate user would not have the skills to read text on the screen, type required input, or use the keyboard to control the system. Most of these telecenters provide standard operating systems and software applications designed for literate users. Even if the user possesses some partial literacy, they may still be unable to navigate the system; sometimes this software has not been localized to the language or dialect of the region. When telecenters provide literate volunteers to help these users navigate online, many may be too intimidated to ask for help. There is also potential for intermediaries to control or filter the information to which users would be given access [Aral 2001].

Effects of Adult Literacy in Developing Regions

Literacy challenges pose many problems outside of the immediate realm of information technology access. A lack of literacy skills can affect a rural community member's confidence when asking health questions, dealing with government bureaucracy, breaking from harmful cultural customs and conventions, locating aid services, adopting advances in childcare practices and nutrition, discovering financial opportunities, understanding reproductive issues, preventing HIV infection, and communicating to family who have moved to urban centers for work [Chlebowska 1990].

Information technology and literacy education must be tied to immediate way-of-life benefits in order to make them enticing to these communities. The URSULA interface should encourage literacy learning while the user performs other desired tasks, and it should help users see how interface tasks might be made easier with additional literacy skills. For example, by including written text on the display and by highlighting this text as it is read aloud by a text-to-speech system, the user is able to improve their literacy skills by reading along with the device. If text is included on the screen, then users with partial literacy skills will be able to navigate the interface without waiting for the text-to-speech system to read all of the information to them. By making the interface experience more engaging for users who improve their literacy skills, then the device can provide a concrete motivation for the user to develop these skills (while still facilitating use of the device before these skills have been acquired).

The Environment: Rural Indian Community Life

To better anticipate how users of Simputer devices would interact with the system, the setting in which the device is used is important to consider. Major features of the Indian communities in which this interaction will be situated include their rural agricultural way of life and their great linguistic diversity. Economic and employment activities will influence our understanding of what applications will be attractive to users when design scenarios are created later in this thesis. The wide variety of languages spoken in Indian adds complexity to the design of the interface:

the system will need to be easily localizable to many languages. Another concern is how the linguistic background of the users may affect how they interpret elements of the interface.

Way of Life in Rural Communities

Rural Indian communities have traditionally led an agrarian life with time spent on the seasonal demands of subsistence and small-crop farming. Shrinking agricultural profits and unpredictable weather changes have forced many males to leave farms seeking employment in urban centers [Chlebowska 1990]. Many women are left to manage the household and farm, making them important target users of the Simputer. Given less access to education, rural women have lower literacy rates, making technology access a challenge. In addition to gender issues, sensitivity in the interface design will also be needed because of the diverse religious and cultural traditions, separatist political movements, and social caste hierarchies which are part of Indian life. For example, if geographic information is communicated to users through maps with audio captioning, the political borders and place names used to refer to map elements may be under dispute.

Other cultural aspects of the Indian population will affect how users will adopt the new technology. Ways in which people deal with confidence, blame, and frustration will affect how comfortable new users will be learning the interface. If users will have access to other, more experienced users while they are interacting with the device, and they are willing to ask these other users for assistance, then knowledge of the interface can be transmitted by these social interactions. If users are less willing to ask others for help, then the interface should encourage more browsing, self-learning, and exploration. Another concern is how the use of politeness and formality in the various Indian cultures will affect how users expect the device to speak to them and what ways they will be comfortable speaking to it.

Diversity of Indian Languages

The diversity of languages in use throughout India also poses challenges to users of information access devices. With dozens of major languages and hundreds of dialects, localizing the device into all of them would be impractical. Many people will need to interact with the Simputer using their non-primary language; fortunately, most of the Indian population is multilingual. However, even if they are comfortable speaking another dialect, they may not be able to read text written in it or use complex vocabulary. So, even users who may be fully literate in one language will need to take advantage of the illiteracy accessibility features of the Simputer.

While most Indian languages run left to right and use one of a small set of scripts, there are languages which run the other direction and many that use scripts not yet included in Unicode or other character set standards -- thus complicating the design problem. Since not all of these languages have keyboard input standards, there is an additional incentive to employ speech-based input even for literate users. Developing speech recognition and synthesis solutions for these diverse languages and dialects is another area of research of the Adaptive Speech Interfaces Group at UCD [Olinsky 2002].

Another linguistic complication which would limit a Simputer users ability to access information on some websites is that since encoding and input standards for many Indian languages have only recently been developed (if yet at all), there are many web pages that represent textual information in Indian languages as GIF images of script. Since text-to-speech screen reading technology cannot access this information, these web pages would be inaccessible to illiterate users. Other sites may use a variety of romanization schemes to represent the text of Indian languages in the ASCII character set. While it might be possible to develop screen reader technology that could take this type of encoding as input, it adds more technical complexity to the interface implementation.

Chapter 2: Technology and Work

By better understanding the goals, specifications, and technology underlying the Simputer, the designer will be better able to anticipate how a user might interact with the physical device. Likewise, by considering which applications of information technology would be most valuable to the target population, the designer can ensure that the interface that is developed supports the needs and requirements of these applications. Several of the applications identified in the "Work" section below will be used as starting points for developing hypothetical situations for HUDS user-interaction scenarios in chapters 5 and 6.

The Technology: The Simputer

The following section will discuss the technical details of the Simputer device as they impact the user interface experience. While the purpose of the URSULA project is to develop user-interface guidelines for providing illiterate access to a variety of devices, it is useful to ground the discussion of the design process traced in this thesis to a particular system and technology. In addition, some level of technical specification is required to successfully produce the detailed HUDS user interaction scenarios in later chapters.

Form Factor and Component Technologies

Simputer devices will be produced by companies that license the design from the Simputer Trust; therefore, the specifics of the device may vary by manufacturer. Also, the quality of the technical specifications of the device may increase over time as components improve and become less expensive. In general, the device would have a form factor similar to a Palm Pilot or Pocket PC with a touch-screen, microphone, and speaker [Simputer 2001]. The device may also have some physical buttons on the case which could be used to interact with the interface, but it would not possess a keyboard. The current Simputer screen specification is black and white with a resolution of 320x240 pixels; so, there the interface design will need to reflect the device's capabilities and its limitations on the complexity of images and animations.

The device will contain Smartcard reader technology such that a single Simputer could be shared by multiple users who each possess a personal SIM card (much like those used for mobile telephones). Inserting the card into the device would give the user access to their personal information and preferences. A telephone modem and USB port will allow the device to connect to the Internet and other peripherals. The V.34/V.17 data/fax modem technology of the device and the quality of telephone lines in India will limit the connection speeds the device can achieve; this connection rate may have consequences for the complexity of the user interface which is produced. For example, while pages with lots of animations may be appealing to illiterate users, they could take too long to download over the modem connection.

Information Markup Language

The Simputer Trust has developed a mark-up language for Simputer-readable web pages, the Information Markup Language. IML would store pages with phonetic information that would facilitate the device's ability to read information aloud. By encoding phonetic information into the markup that is delivered to the device, the size of the text-to-speech software and processing demands on the hand-held system are reduced. Pages that were not originally encoded in IML could be converted to IML by intermediary servers that would perform the initial portions of TTS processing and add phonetic information to the page before transmitting it to the Simputer. The IML standard also includes interface elements for the pages that can be viewed by the device which are appropriate to the small screens and touch screen input.

The Work: Important Applications

Sustainable Access in Rural India (SARI), a recent survey-based study on the feasibility and sustainability of telecenters, is the most recent and comprehensive work into information technology applications in rural India [Aral 2001]. The research team used focus groups and written surveys to identify a set of realistic applications for rural Indian information technology. Their goal was to find applications that would “enhance the social and economic development and be most valuable to the rural poor.” [Aral 2001]

Some applications were deemed unsustainable, impractical, or not of interest to the populations of the target communities. For example, although education delivery applications might seem well-suited to an Internet connected telecenter, the difficulties of establishing relationships with schools in India willing to create online content and resources makes this application difficult to implement. While an application which would provide farmers with pricing information on agricultural products would empower them when negotiating prices for goods from suppliers, updating and ensuring the accuracy of this information could prove quite difficult. The level of localization of this information would be critical as prices for agricultural goods could vary greatly by region; also, it can be difficult to select a correct selection of agricultural products to track whose prices fluctuate and are important to farmers' livelihood. [Aral 2001]

After surveying the potential users of Indian telecenters and investigating implementation challenges for various information technology applications, the SARI project identified the following four high-priority telecenter goals:

- Health Information Distribution
- Employment Listing Bulletin Boards
- Online Government Services
- Agriculture Reference and Seasonal Information

Health Information Distribution

The SARI study determined that very technologically ambitious online health care applications such as remote medical consultations and diagnoses were impractical and ill-suited to the health care realities of modern India. Instead, the study recommended informational health portals and training materials that would address common medical ailments and day-to-day health issues for a general audience. Applications for an online health information channel like epidemic announcements, sanitation practices advice, and nutritional recommendations were also discussed [Aral 2001]. The often resource-strapped and under-trained health care providers at local clinics for low-income villages are another potential target audience. These doctors and nurses could take advantage of up-to-date online reference materials and training without having to travel to cities and leave the local health centers unmanned in their absence.

In addition to the findings of the SARI project, there are other indications that a health information application for rural communities would receive sufficient support and resources. There are several international health organizations who are working on projects that would develop localized health information that would be suitable for distribution over the Simputer channel [Colle 2001b]. Health Information for Development (HID) began a project in 2000 to create localized health information content for developing nations that could be disseminated in health-information resources centers it would establish. The World Health Organization will complete a project in 2007 entitled the "Health InterNetwork Project" which would be a localized health information portal accessible through tens of thousands of web-access points to be established in developing communities. If content from both of these projects could be accessible through an illiterate-accessible interface, then the reach of this health information would be increased to include populations who may need it most.

Employment Listings

Since many of the villagers in the region studied by the SARI team worked as day agricultural laborers, they found a lot of interest in an application that could facilitate the process by which these workers found employment. Currently, work can be difficult to find in the dry season, and landowners can have great difficulty finding workers with the appropriate skills for various farm tasks during the wet season. An application which combined the labor markets for nearby villages and allowed workers to be matched to the skills of employment that was well suited to their wage needs and location would increase productivity and benefit all participants in the economic system. For laborers, life is full of uncertainty as to the availability and stability of agricultural labor income since employment is generally arranged with short notice and through word-of-mouth channels. An online service could bring stability to their employment plans and help them to avoid losing a day's wage due to difficulties in arranging work. Landowners could avoid having their fields go idle in peak season by scheduling work ahead of time and finding employees with the needed skills to work their farms. The SARI study indicated that an online bulletin board system which allowed employers to post job opportunities and allow workers or their work-group leaders to respond to postings would be the best structure for this system.

Online Government Services

The Indian government, particularly at the local level, has a great potential to improve the quality of life in rural India through various services and initiatives. Unfortunately, due to inefficiencies in the manner in which these services are delivered and the difficulty in advertising the benefits villagers could receive through various programs, many communities are not benefiting from these opportunities. For example, according the SARI survey findings, a request for a birth certificate not only required a great amount of time, effort, and fees to acquire, but in many cases a bribe may be needed to ensure the service is promptly provided [Aral 2001]. A government services portal that could advertise new initiatives and provide an automated channel for villagers to submit requests for government documents and assistance could address many of these problems. For assistance with more individual queries or difficulties, villagers may benefit from an e-mail service which could allow them to contact the offices of government officials that may be located in cities quite far away.

Agriculture Information Application

Most of the rural Indian economy is based on agricultural production, and the portion of many villagers time which is spent on agricultural work is significant. For this reason, an online agricultural information tool that could help farmers access information on regional weather forecasts, learn about best practices for their particular crops, and to coordinate purchases of agricultural goods with nearby farmers to negotiate better prices would be a great asset to these communities [Aral 2001]. SARI project surveys and focus groups with farmers indicated that there was sufficient interest in such applications that villagers would be willing to pay a fee for such information services if they brought about increased productivity in their work -- making this information technology application particularly promising and sustainable. The SARI study also discussed how this agricultural application could help farmers establish communication with agricultural extension offices, government agencies designed to provide farmers with assistance in crises and training on more effective production techniques.

Chapter 3: Previous User Interface Work

Previous Work in Accessible Interfaces

In the field of Human-Computer Interaction, much recent attention has been paid to users with sensory or motor challenges. Some modern interfaces concentrate on finding ways to reformulate the output of the system to make it accessible to users with sensory difficulties. For users with hearing impairments, there has been research in captioning audio information or producing visual screen displays for audio events in the interface. A slightly more developed research area is interfaces that are accessible to users with visual disabilities.

Perhaps one of the most challenging user interface scenarios has been developing non-visual computer interfaces for blind or partially sighted users. There have been various approaches to adapting an interface for a user with visual impairments. Simple approaches merely read all portions of text on the screen, while more sophisticated techniques reorganize the interface into a hierarchal menu structure [Mynatt 1992] or help the user understand the graphical layout of the screen [Mynatt 1994]. Both of these interaction methods allow the user to develop a mental representation of the interface structure.

Other accessible interface researchers have considered input-related issues. User interfaces for the blind not only need to reformulate the system output into audio, but they also need to provide tools which allow the user to operate the machine predominately from the keyboard (since aiming an on-screen pointer with a mouse would prove difficult). Other work produced special typing or pointing hardware [Law 2000] or speech input methods [Yankelovich 1995] [Stifelman 1993] for users with motor difficulties. The market for speech technology is broadening; people in hands-busy environments or with poor typing skills have become the new target markets for voice command and control and speech dictation software. Some popular products in this market include Dragon's Naturally Speaking and IBM's Viavoice.

Interfaces without Visual Displays

Since illiterate users would be unable to read textual information on the screen of their device, it is also useful to look at previous research into applications where the user does not have access to a screen or is unable to read it. Interfaces for small devices without visual displays could also help to inform the URSULA design process, such as those for telephone-based applications or eyes-busy situations (like driving a car). Users of such devices often experience a common set of difficulties: remembering what commands they can say to the system, knowing what mode or states the system is in, understanding what silence from the system means, struggling to recover when the system makes a speech recognition error, and navigating through audio menu hierarchies (see Chapter Four).

While the designers of these interfaces have attempted to address these problems, many users of non-visual interfaces still experience frustration. The difference between these interfaces and the Simputer is that although an illiterate user may not be able to read information on the screen, they can still see the interface and interpret images on it. In this way, the URSULA project has an additional modality at its disposal that these other interfaces lack: the non-linguistic graphic/visual information channel. By understanding the particular difficulties users experience with non-visual interfaces, URSULA can structure the graphical portion of the Simputer interface to address these problem areas.

Previous Work on User Interfaces for Illiterate Users

Unfortunately, there has been quite little work in the area of interface design with illiterate users in mind. Two significant projects which come close to the URSULA design goals of illiteracy accessibility include a web-browsing application for illiterate adults and an accessible government services kiosk. Both of these projects are discussed in detail below.

Some software on the periphery of this research area includes reading educational products like "Reader Rabbit" or other forms of visual storybooks that teach literacy skills to their users. While these products are designed for a population with partial literacy skills, the amount of interface guidance that can be distilled from them for the URSULA project is limited. These programs are rarely designed with an adult population in mind, and they are not focused on helping the user accomplish non-educational life tasks.

Goetze & Strothottle: Illiterate Web Browser

Goetze and Strothottle [Goetze 2001] conducted research into constructing a web-browser that replaced individual words on the page with pictures representing their meaning. The system also allowed the user to hover on a word to display an image representing its meaning or to listen to it read aloud. A highlight of the approach was that the web pages needed no special encoding; so, users were not restricted to a subset of the web. They could navigate freely and choose to view pages anywhere on the Internet; the browser would simply convert any text on the pages into picture-words. Another interesting aspect of this browser is that it attempted to hide some of the complexity of the windowed operating system interface behind the browser. Instead, the system used a touch screen and stylus to allow the user to point to words on the screen or to drag across them to ask the system to read them aloud. In this way, the user who would likely be a novice computer user could feel more comfortable using the browser since it would mimic a familiar interface paradigm, pen and paper.

The Goetze and Strothottle approach was impractical for several reasons. With the large lexicons and morphological productivity of many natural languages, it is doubtful that the picture dictionary would ever be sufficient to represent all words the system may encounter. Even for a limited subset of the vocabulary of a particular natural language, it is unlikely that intuitive, salient pictures could be chosen for all words in the user's vocabulary. If the system were to be employed in multiple locations with different cultural traditions and metaphors, then the images chosen for the picture language would need to be sensitive to these variations.

If a user of the Goetz and Strothottle approach relied heavily on the picture language representations of the text on the screen, then they may attempt to memorize the images of the picture vocabulary in order to increase the speed with which they use the system. The cognitive complexity of learning this visual vocabulary could be similar to that of learning to read written text. Unfortunately, by learning the picture language, the user of this browser would not have acquired a skill useful beyond the browser. If the interface had instead encouraged the user to develop written language reading skills through its use, then a positive educational goal would also have been accomplished.

Another difficulty with the picture language browser is that it does not address the input side of the interaction. While users can use the picture icons and text-to-speech capabilities of the system to understand the content on the screen, they have no way of entering text into the system because of their lack of literacy skills. Applications like search and e-mail would therefore not be supported.

EZ Access Touch Screen Kiosks

The EZ Access Touch Screen Kiosk [Law 2000] is a device for delivering government services designed to provide access to the widest possible user base. The kiosk consists of a touch-screen video console, speakers, keyboard, and other input devices integrated into a podium which could be located in a public location to allow access to a set of services online. The designers of the system concentrated on providing access to users with a variety of sensory and motor capabilities. The podium is wheelchair accessible, and users with limited motor skills could either use some of the oversized controls on the podium or could link the podium to a specially designed input device of their own. Deaf users can operate the kiosk because no information was transmitted only along the audio channel. The designers of the kiosk also considered the needs of seeing-impaired users, and this work is of interest to the URSULA project.

The EZ Access group addressed the some of the needs of illiterate users of their kiosk when they developed a blind-accessible version of their interface. The kiosk included an easily identified button which would trigger the contents of the screen to be read to the user by text-to-speech technology. Text was highlighted as read, and the user could also point to portions of the screen to have them read and explained. Because users may sometimes wish to point to items on the screen in order make selections and other times may wish to point to items in order to have them read aloud, the system needed two operating modes. The interface indicated when the kiosk was in press-to-activate or press-to-read mode. Even when in press-to-activate mode, the kiosk asked the user for confirmation before performing any action. While this was mostly designed as a safeguard for blind users, such confirmation may also be desirable for illiterate users who would likely have little computer experience.

For the designers of this system, who were mostly concerned with sensory or motor disabilities, making the kiosk accessible to illiterate users was only ever a secondary goal. So, the interface which was developed for illiterate users was the same as that for blind users. Therefore, illiterate users could theoretically use the device without looking at the screen; so, the designers took little advantage of the non-linguistic visual/graphic modality to communicate information to illiterate users and allow them to interact with the system. While their literacy-access approach may be adequate enough to allow users to operate the kiosk, it is certainly not an optimal interface for this population.

Other Software for Developing Countries

Jim Youll of the MIT Media Lab has outlined a proposal for a communication system accessible to illiterate users in which several standard conventions of literate e-mail systems have been questioned and modified [Youll 2000]. Instead of listing users of the system by name with written e-mail addresses, the directory of users of the system is presented as a visual network structure of images of people connected by social relationships. All users of the system are also assigned a serial number that is storable as a printed bar code that users can physically carry to the access device and uniquely refer to other users and entities in the system. Messages sent to other users would not contain any "meta-information" such as subject lines; so, messages only required a destination and the message itself. While e-mail may be an included capability, the Simputer project will also focus on users being consumers of information posted online by agencies or other users.

The IDRC Study/Acacia Initiative has proposed an initial design for a web access system for illiterate users [IDRC 1997]. The proposal mostly explores the technical background to such a system including speech synthesis technology for African languages, the design of server or client side speech synthesis, and audio streaming formats. No effort would be made in this project to modify the content of the individual pages to make them more easily used by users, but instead, the system would be designed to give users as broad access to Internet resources as possible. While giving users broad access to web pages seems like an attractive idea, it ignores the fact that few pages on the Internet are written in African languages or address issues of local interest to African people.

This system did address the written language literacy challenges of its users, but not their technological fears. “When the user passes the mouse pointer over the words on the page (or touches them using a touch screen), they will be audibly pronounced by the computer." [IDRC 1997] While users would be able to understand the words on the screen in this manner, they may still find the interface too difficult to navigate. As in the Goetz and Strothottle system, this accessibility approach also fails to address the input side of the interface, thus many interesting IT applications are not enabled, such as Internet searching or e-mail.

Chapter 4: Initial Interface Considerations

One of the most important steps in the design of a new interface is to consult the relevant user-interface design literature for recommendations on various aspects of the interface. While chapter three included a discussion of previous work by discussing individual systems that address the needs of special needs populations, this chapter will focus on interface design recommendations and take an issue-by-issue approach. After a discussion on ways to build interfaces that are understandable by novice users, this chapter will consider two ways to dissect a user interface to make it easier to analyze. In one section, the input and output channels of the interface will be considered in light of the special needs of illiterate users. In the final section, the various ways to structure the dialogue between user and computer will be discussed, and the particular advantages of each approach for illiterate users will be considered.

Building Interfaces for Novice Users

An important aspect of the Simputer user population is that there will be a low level of previous computer experience. Therefore elements of the user interface design literature that make recommendations for interfaces for novice users are particularly applicable to this demographic. Novice users pose a special challenge to interface designers; unlike traditional users, they lack a familiarity with common computer interface paradigms. Encountering a new interface, they cannot draw from previous experience to guide how they interpret the system. Novice users are often concerned that they will break the computer or accidentally tell it to do something undesirable. For this reason, they need to be reassured while using the interface that their interaction with the system is progressing correctly. When novices do make a mistake, they can be easily confused by the error state of the device and may have particular difficulty returning to the system to normal operation. Interfaces for novices need to take this into account and construct the interaction in a robust, conservative manner that focuses on how the user can recover from errors.

While providing a training course to users of the system could alleviate many of the concerns of novice users discussed below, it is not certain that the resources would be available to provide personal time and attention for each new user. For this reason, the interface should be constructed in such a way that it provides tutorial services to its users and encourages their exploration and learning of the system. Making the interaction as intuitive as possible will also allow users to figure out the operation of the system if they do not have access to training and help services.

Memory and the Interface

If the interface requires a user to remember a lot of information in order to use it successfully, then it will prove particularly challenging to novices and those learning how to operate it. The user should be able to accomplish their goals without thinking much about the interface. For this reason, designers should reduce the amount of information that has to be remembered to help the user carry out work-related tasks [Marshall 1987]. Poorly designed interfaces often require users to tax their memory by not making state information available on the interface and by forcing the user to recall how to make the interface perform particular operations.

Designers should use the visual organization of the interface to indicate to the user the current state of the device [Preece 1995]. Users should be able to determine the state of a device by quickly checking its UI; they should not need to remember where they currently are in the sequential steps of the interface. For the handheld Simputer domain, this is a particular concern; since the device is so small and portable, a user's attention may easily be diverted from the device in mid-operation. The user should be able to return their attention to the device and infer from the interface where they left off in the previous interaction.

Designers can also tap into users' recognition memory to make interfaces easier to use [Preece 1995]. Salient screen or interface elements can act as clues to the user which may trigger their memory as to how the interface functions. For example, a menu structure requires users to remember less information than a blinking command prompt; a user may recognize a command on a menu that he or she would be unable to recall without such prompting. With users new to information technology that may only use the Simputer infrequently, expecting users to recall complex methods of interaction would be unreasonable.

Encouraging Exploration

Since a novice user population may be hesitant to approach a new piece of information technology, it is very important that the interface for the system be visually engaging and attractive to users [Preece 1995]. In addition to the superficial appeal of the interface, by designing it such that a new user can feel confident exploring it, a sense of trust can be developed between the user and the device. A way to make users feel more comfortable exploring an interface is to offer a powerful and consistent 'undo' key [Marshall 1987] [Rudnicky 1996]. In this way, the user will feel it is less likely they will commit to an undesired operation.

Interfaces should be avoided that have the potential for a user to accidentally perform a 'fatal' operation, i.e. one that leads to drastic and unrecoverable result such as deleting a file, damaging the operating system software, or losing information [Marshall 1987]. In particular, on interfaces with icons that can be clicked by the user to initiate actions, the selection of an icon should never immediately initiate a 'fatal' operation; the user should always be asked for additional confirmation. Such drastic consequences from merely clicking elements of the screen interface could train the user to be afraid to touch the device.

The Role of Feedback

While the feedback that an interface provides is important to all users, it has special significance to those who are new to information technology. While feedback designed for expert users should communicate the effect and consequences of a user action and inform the user of the new system state and location in the interface [Marshall 1987], the feedback goals for novice users can be much more ambitious. When a user is unfamiliar with a task, it is particularly important that they are given feedback as they progress through the various stages of its completion so that their level of confidence in their actions is maintained [Marshall 1987]. Feedback can also be used to educate the novice user about the interface and to explain any metaphors which may have been used when the designer constructed the interaction.

Input and Output Channels

Since the purpose of this chapter is to consider the elements of an illiterately accessible interface and discuss how previous user interface literature can inform its design, it is natural to break down the interface into its most basic units. The next two portions of this chapter will perform this deconstruction from two different perspectives. In this section, the channels by which information flows in and out of the system to the user are considered in light of the special needs of an illiterate user population. The final section will discuss the various ways in which a dialogue interaction between the user and the device can be structured.

Touch Screen Input

With the small form-factor and potential portability of the Simputer device, input hardware such as keyboards, joysticks, and mice would seem difficult to use. Also, considering the literacy challenges of the user population, expecting users to successfully enter data to the system using a keyboard makes little sense. For this reason, a touch-sensitive display screen is a well-suited input modality. In addition to its space advantages, touch screens are generally considered one of the most intuitive and easy to learn input methods for novice computer users; it is for this reason that public kiosks and information displays often make use of these screens.

One of the challenges of a touch screen interface is that aiming the stylus to select screen elements can be difficult if the images are too small. Especially for users with little IT experience, the hand dexterity needed to operate a small screen device may take time to acquire. Another concern is that with touch-screen interfaces, the stylus is both the mouse and the keyboard. On some touch screen systems, the screen is able to accept both point-and-click style interaction and a form of handwriting recognition. This technology might not be desirable for illiterate users who would not know how to draw the script characters for their languages. If any form of linguistic input will be done with the stylus, an on-screen keyboard approach which allows users with partial literacy to recognize characters already drawn on the screen would be preferable.

Speech Input and Output

Considering the difficulties illiterate users would have using touch screens and keyboards for linguistic input, choosing to employ speech technology in an illiterate interface is a natural decision. Speech systems provide an input and interaction method which can feel more comfortable and natural to novice users because it can mimic the conversations and vocal requests people have everyday. On a small device such as the Simputer, speech can compensate for a lack of screen space for complex GUI elements. Because of the portable and hand-held nature of the device, speech is advantageous because a user can maintain their interaction with the device even if their eyes or hands are momentarily distracted elsewhere.

Aside from the technological challenges with speech recognition discussed in the section below, there are other reasons why using speech in an interface can pose special difficulties. Speech is also a temporary mode of communication in which the user cannot review or reference information that has been conveyed [Grasso 1996]. Since this ephemeral quality of speech output poses challenges to some interface applications, special care and ingenuity will be needed to make these speech interfaces user-friendly. For example, it would be quite easy for a user of a speech-only application to become overwhelmed by a long list of options being read to him or her by the computer.

Speech interfaces can also frustrate users by changing their expectations of the computer system. While users may be familiar with talking about their wishes and intentions, by allowing them to talk to a computer, they may anthropomorphize the device and expect the computer to have capabilities beyond what is reasonable [Grasso 1996]. Another issue is that communicating information to the user through the speech channel can be ten times slower than if users read written text for themselves [Maberly 1966]. Therefore users may become frustrated by the slowness of the speech interface; so, designers will need to reduce the amount of information to be communicated to the user.

The novice computer users of the Simputer system may also feel frustrated when talking to the device because it is not conveying the same conversational cues that humans do when speaking. Humans use sophisticated modifications of their vocal prosody to indicate when they are finished speaking and it is time for the other person to participate in the conversation [Davis 1987]. Other studies indicate that humans use facial expressions and eye gaze during conversation to indicate the meaning of pauses and silences in the dialogue [Levow 1997]. Unless the interface is carefully constructed, the lack of these prosodic and visual clues in the speech system may lead Simputer users to become confused as to when they should speak to the device.

Speech Recognition Errors

One of the biggest challenges with using speech technology is the frequency of speech recognition mistakes. While researchers have been developing speech recognizers for languages such as English or Japanese for some time, other language families have received little or no attention. Many of the popular languages and dialects of India have only recently been studied by the speech research community. Initially, there would be a high recognition error rate for these new languages as their speech models are developed.

While the technology itself is steadily improving, there are many external factors which can reduce the accuracy of speech recognizers. The noise levels of the setting in which the device is situated and the quality of the microphone used can all degrade the quality of the sound input to the speech recognizer. If the user of the system is speaking in their non-native language or has a speech difficulty or cold, then the language models of the system can be confused by these individual voice variations. Since the Simputer may not be localized into all the languages of India, then it is very likely that some users of the system will be interacting with the device in their non-preferred language. The resulting accent that they may have in their speech could reduce the system's recognition accuracy.

Because of the likelihood of speech recognition errors, a well-designed interface will need to cope with situations in which there has been a speech recognition error. Some designers suggest that interfaces which use speech always provide a back-off approach involving pointing-and-clicking in case the recognizer cannot understand someone's voice [Preece 1995]. While this may sound like good advice, it may not be practical for all interface scenarios. When the user needs to input a large block of text, pointing-and-clicking may not be practical (unless an on-screen keyboard is provided, which still would not be a reasonable approach for illiterate users). For these reasons, an illiterately accessible interface will need to provide users with a way to identify and cope with recognition errors.

As the first step to fixing a speech recognition error is to identify it, an illiterately accessible interface will need to make users aware of when it was unable to accurately transcribe their spoken text. In a traditional speech application, when text is dictated, it is displayed on the screen and a literate user is able to read it as it is inputted. Since illiterate users would not be able to spot errors in this fashion, the system would have to locate these mistakes for them or find another way for them to identify them. The system could use some type of language/grammar checking software to look for unusual words in the input string and then highlight the questionable text. Otherwise, the user could ask the system to read back the text which was entered and listen for places in which the text diverges from their intended input.

The process of correcting an error is also more challenging for illiterate users. When the user notices a recognition error, they can either stop the dictation to fix it or wait until the end of the dictation to correct the errors. [Roselli 1998] identifies five possible approaches to correcting errors made by speech recognition systems: (1) vocal spelling, (2) write the information, (3) type the information, (4) choose from a set of options, or (5) just fail and put something that sounds right. Because of the literacy challenges of the target users of the Simputer, options 1 through 3 are not practical. Option 4 may only be appropriate for situations in which the user is selecting one of a small set of possible options, and option 5 is merely a failure case in which the incorrect text is left in place.

Two other approaches that might hold potential for illiterate users would be for the user to attempt to re-dictate the words in question, re-dictate the individual components of those words, or to use a form of voice recording input instead of converting the input to text. This final option would only work for applications that could accept audio file input instead of text. Another concern is that some countries, including India, have legal restrictions on the transfer of recorded voice over computer networks. While such laws are designed to protect the interest of telephone companies, they can provide a stumbling block to interface designers for information technology. A revision of Indian law in March of 2002 does allow PC-to-PC voice communications over computer networks within India and international transmission of voice data from PCs in India to telephones abroad [Ribeiro 2002].

However, the implications of this new law on the technologies of the Simputer system will need to be carefully considered.

Speech Synthesis Errors

The technological sophistication of speech technology is also a concern for the text-to-speech (TTS) output direction of the interface, and in fact, the UCD Adaptive Speech Interfaces group and other researchers are working on ways to facilitate the creation of TTS systems for new languages in developing countries [Olinsky 2002] [Donovan 1996]. From an interface interaction point of view, failures in the TTS direction are less of a concern than recognition errors because humans are generally skilled at figuring out the meaning of incorrect utterances from their surrounding context. In the case that the system is unable to pronounce a particular word correctly and the user cannot discern its meaning from the context, then there are a few options for how the word could be communicated: the system could display the word on the screen, it could spell the word aloud, or it could use a thesaurus/dictionary to replace the word with one that it can successfully pronounce.

Speech Interface and Interaction Issues

Most empirical studies of users talking to computers indicate that people speak differently when they know that they are talking to a computer [Dey 1997]. These studies indicate that people tend to be less polite and speak in short commands; they also tend to mimic the terminology used by the computer to describe itself. What makes the findings of these studies difficult to generalize to the Simputer domain is that they were carried out in Western cultures in which most people possessed a general knowledge of what a computer is and what its limitations might be. For illiterate users in villages with little exposure to information technology, the expectations they will have for the device and the cultural forces that will affect how they speak to it may be quite different.

A common complaint of users of speech-only interfaces is that the system frequently overloads the user with information; in particular, the presentation of lists can be overwhelming without a visual reference [Davis 1987]. When users are confused by the arrangement of a list or it is too long to be held in their working memory, they will commonly ask for the list to be repeated or will write down the options [Dey 1997]. As illiterate users would not be able to take written notes as they use the interface, the system will need to make lists more memorable or to provide easy to use list-repeat capabilities. One approach is to add visual clues or mnemonics on the screen associated with each item in the list to aid the user's memory. While illiterate users could not read linguistic on-screen clues, icons or other images could be used.

Another approach to making lists easier for users to understand is to organize the elements of the list in a more memorable or salient fashion. Cognitive psychology has demonstrated that items at the beginning and the end of lists are the most easily remembered. So, a good strategy might be to place frequently accessed or important items at the start of the end of lists [Marshall 1987]. Since the lists are being presented aurally, putting popular items at the beginning would be more desirable since the user would need to wait for all of the items in a list to be read. The items of the list could also be arranged into categories or groups according to a scheme that is understandable and transparent to the user [Marshall 1987].

Users of speech-only interfaces typically get nervous during silences. They don't know whether or not the system is processing their input, is waiting for a command, or has entered an error state [Levow 1997]. When a visual interface is present, it would be best for the interface to indicate to users when they need to wait for durations of more than about five seconds [Dey 1997]. For the illiterately accessible interface for the URSULA project, an appropriate approach to silences might be for the system to have a consistent 'wait' icon that would be displayed on the screen to tell the user that the system is processing and that it is not waiting for them to do something.

Just as users can find unexpected silences unsettling, times when the interface is continuing to speak when they wish to enter a voice command are equally frustrating. Users of speech-only interfaces like to feel that they are in control of the interface, and allowing users to interrupt the system's speech output is an important way to convey this sense of control [Roselli 1998]. Another control issue arises when speech interfaces require the user to navigate very deep menu hierarchies or remember complex states -- making them feel like the interface is telling them what to do [Roselli 1998]. It will be important to structure the URSULA interface in such a way that users feel like it is a useful tool and an extension of their capabilities, not a puzzle for them to master.

Audio Output Channel

Having discussed the potential of an illiterately accessible interface to use linguistic speech input and output in the section above, this section will focus on the ways in which non-linguistic audio output can enhance the user experience. The Simputer device will include a speaker for transmitting aural information to the user. The quality of the speaker on the device, the bandwidth of the network connection, and the processing power of the Simputer will all affect the quality of the sounds that the device will make. Because of the size of the device, the speaker will likely be very small in size and will be monaural. The Simputer's size and portability will place other limitations on the audio channel of the URSULA interface; because of the setting in which the device will be used, the audio channel may be very susceptible to noise and may not be private to the user of the device [Grasso 1996].

Despite these challenges, there are many ways in which the audio channel of an illiterately accessible interface can convey important information to the user. The audio channel is ideal for conveying messages that are simple, short, and will not need to be referred to again [Deatherage 1972]. In particular, audio works well for indicating when events have occurred in time or to call for the user to take an immediate action. Along these lines, non-speech audio seems appropriate for alarms, warnings, and indicators that monitor the status of an operation. In situations where the audio channel is already crowded with output, the message being conveyed is complex, or the output deals with spatial relationships or locations, using non-speech audio can be confusing [Deatherage 1972].

Unlike speech output, which can be interpretable by new users to a computer system, non-speech audio comprises a learned vocabulary of interaction. The users of the device will have to be taught how to interpret any of the non-linguistic sounds made by the interface. For this reason, actual linguistic output may be preferable to random sounds. For example, if an operation is completed, it might be better for the system to say "operation complete" then to merely make a bell sound or tone.

There has been some research into using "natural sounds" to represent interface events that tap into users' intuitive world knowledge [Gaver 1986]. For example, when a user asks the system to empty the trashcan, it may make a 'clunk' sound with a volume level corresponding to the size of the files being deleted.

Although this approach is interesting, it can be difficult to generalize to varied interface situations, and it may sometimes rely on cultural associations between with sounds and their meanings. For these reasons, it is doubtful that "natural sounds" would be used extensively in the Simputer interface.

Augmenting the Speech Interface with Audio Information

Linguistic and non-linguistic auditory information can also be used simultaneously. The speech output of the interface can signal additional information to the user through various non-lexical auditory features. For example, the interface can vary the prosody of the speech output, add beeps or other noises in the background behind speech audio, or change the gender, timbre, or other features of the synthesis speaker [Shriver 2000]. Researchers have built interfaces using these approaches to express when the system is unsure of its output, has reached the end of a list, or wants to indicate the 'type' of information of particular words in the output stream. In the case of illiterate users who are unable to read linguistic information on the screen, having this additional modality of output may be useful.

Visual Output Channel

The visual channel is the most traditional modality of computer interfaces; users of computer systems generally think of their interaction with the device in terms of the screen elements they see and manipulate. The visual channel is also rich in the type of information it can convey; visual information can vary according to its components: alphanumeric characters, typography, color, geometric shape, angle of inclination, size, visual number, and brightness [Baecker 1987]. Unfortunately, due to the literacy challenges of the target population, information conveyed with alphanumerics (and variations in typography) will not be useful for this interface.

The technological specifications of the Simputer device's screen will also make variations in color and brightness difficult to display, and its small size will reduce the variations of size and visual number of screen elements that can be conveyed.

Fortunately, symbolic information can be displayed via the visual channel by means of non-linguistic representations such as charts, maps, graphs, and diagrams. Using such approaches, information about quantitative data, geographic locations, complex relationships, and navigation and progress through the interface can be communicated to illiterate users [Baecker 1987]. Unfortunately, there may be a special form of literacy connected to many of these graphical representations, and the user base of the Simputer may have as much difficulty interpreting information in these forms as they would textual data.

Oftentimes, symbolic representations such as charts, diagrams, and maps use textual captioning to explain elements of the images. In an illiterately accessible system, captioning could be provided through the audio channel, but issues of user attention and memory will need to be explored to discover whether such representations are understandable without visual captions that users can browse at will. For example, to an illiterate user, a bar graph can be very confusing if the text labels on each bar are not readable; a map with locations names written as text could also be inaccessible to an illiterate user.

Augmenting the Speech Interface with Visual Information

One of the great strengths of a graphical interface for this device is the way in which it can be used to augment the shortcomings of a speech-based interface. Common problems users have with speech-only interfaces include feeling unsure of the current state of the system, not knowing if the system is waiting for more input, getting lost in the navigation structure of the interface, and forgetting the options of what can be said.

An important contribution an output screen can make to a visual interface would be to let the user know the state of the system, whether it be ready for input, currently listening, recognizing input, processing, or blocked with an error state [Rudnicky 1996]. By using on-screen diagrams or progress indicators, the interface can also convey to the user their position in the map of possible interactions with the system to help prevent them getting lost in the interface or forcing them to remember a large amount of state information.

A screen can also be used to trigger the memory of a user who is trying to remember the possible options he or she has of what to say to the computer. If there are screen elements visible that represent each of these possibilities, then the user may more easily navigate the spoken commands needed to accomplish their goals [Rudnicky 1996]. The visual portion of an interface that includes speech input should allow the user to 'say what they see' on the screen [Grasso 1997].

When a speech interface needs to communicate feedback to the user, it could risk crowding the input audio channel with noise if it communicates too much of these feedback aurally. A visual display can be used to accomplish important feedback tasks without interrupting the user's speech or distracting them from their task. In particular, an on-screen feedback display with consistent position and appearance is well-suited for continuous forms of information such as whether the system is still live, the input volume of the user's voice, and the volume of background noise [Rudnicky 1996].

Conveying Meaning through Icons

The visual channel can also be used to convey information through icons, animations, and photographs whose interpretations are more intuitive than the graphics and diagrams discussed above. Icons are commonly used in situations where the written literacy of individuals in a particular language is in question. For example, the signage for arriving flights, exits, baggage check, and restrooms in most international airports use representative icon images [AIGA 1981]. An important determinant of the success of these icons is that they represent items that are namable, memorable, easily discussed, and somewhat concrete. The icons map to concepts that the user is familiar with in their day-to-day existence [Marshall 1987]. Icons that represent concepts in the internal mechanism of a system or the interaction domain but are devoid of meaning to the user in their life outside of the interaction are generally less successful. Especially for the users of the Simputer system who will have very little previous information technology experience, icons such as the "floppy disk" image often used to represent "save" would be alien and confusing.

Since some of the most communicative icons take advantage of the common life experience and archetypes of a particular user base, they are often rooted in assumptions about the culture and way of life of their viewers. These images need to be localized to each new community in which they are deployed, and they must be sensitive to their culture and sensibilities. For example, in a Muslim community, the use of alcohol-related images could be taken as offensive. In many African countries, the complex symbolism surrounding animal images can make them too dangerous and potentially confusing for use in computer interfaces [Murrell 1998].

Selecting meaningful and appropriate images for the interface that are interpretable across cultures will be a challenge. Even the most seemingly innocuous differences in an image can cause great cultural confusion. For example, Andrews [Andrews 1994] describes how the color of a snake in an animation could completely change the interpretation of the image by some African users. Another interesting example involves the common use of a red circle with a diagonal crossbar to indicate "no" or "not permitted" when superimposed atop another image, such the "no smoking" signs on many airplanes. Even such a seemingly non-linguistic symbol is open to cultural reinterpretation. In one experiment, an illiterate Zulu-speaking person interpreted a no-smoking sign as "you may only smoke half a cigarette" and a similar "no exit" sign as "persons who go through this door will be cut in half" [Andrews 1994].

Considering the high risk of icon misunderstanding in the illiterately accessible interface domain, icons should provide captioning or other ways of explaining their meaning; the image itself should never be the only way to understand its purpose. Another way to make images more meaningful to Simputer users would be to incorporate animation into some images to allow communication of temporal information, processes, storylines, or cause/effect relationships. However, the memory and bandwidth limitations of the device would certainly affect the amount of animated images that would be possible on the interface.

Facilitating Navigation through Icons

Because of the small size of the Simputer visual display, many portions of the user interface and information content will need to extend onto multiple screens. Users of the device will need to understand how to move between these pages of information and how they relate to one another. Icons can play an important role in communicating this information to an illiterate audience, and the user interface literature offers suggestions as to how these icons should appear. Some work [Barron 1998] suggests that interface designers include ‘previous’, ‘next’, and ‘main menu’ links on every page of their interface in order to encourage users to view the pages in the intended sequence. A study of South African students [Amory 1994] suggested how these links should appear. Nearly all of the students in the study were familiar with the interface of tape decks or VCRs which often use arrows (play), reverse arrows (rewind), and squares (stop) on their control panels. These standard icons could prove successful for indicating navigation controls in interfaces designed for the developing world.

Dialogue Structuring Approaches

Nearly all of the interaction between the user and a computing device can be thought of as a dialogue between them. The user communicates their needs and intentions, and the systems informs the user of their options, provides feedback, and asks for additional information. Interfaces generally vary according to how the initiative of the dialogue is decided. Passive systems simply wait for the user to input commands and follow instructions; directive systems guide the user through the steps of a particular operation [Cole 1995]. Those dialogue structures that include both forms of initiative are called variable or mixed initiative interfaces. Generally, computer systems will used a mixed approach with directive at first and switching to passive when waiting for the user to read screen elements or perform other time-delayed tasks [Cole 1995] [Marshall 1987].

In addition to classifying interfaces according to how they determine initiative, many authors categorize them according to common patterns in the structure of the interaction. Early text-only interfaces included question and answer, menu structure, form structure, and command language approaches to allow the user to control the system [Coats 1987]. As graphical displays became more commonplace, direct manipulation interfaces increased in popularity. With advancements in work in computational linguistics, interfaces that understand commands and information expressed in traditional human languages (called natural language interfaces) have also become possible.

Question and Answer

The question and answer interface is simplest form of dialogue structuring; it consists of the system querying the user for information and the user replying to these specific requests. This is an interface in which the computer has the initiative. For novice users in the Simputer domain, the simplicity of this interface is an important advantage. From conversing with other people, humans are very comfortable to answering questions; the Q&A computer interface should be very intuitive. Question and answer interfaces are best for sets of short questions that are all mandatory for the user to answer; this interface is appropriate for situations in which the user is already on a set path or has already decided on a direction to move through the interface.

The selection of prompts and queries for a Q&A approach is a very important determinant of the success of the system at eliciting the correct information from the user. Stylized or overly polite language can be confusing for the user to understand; instead, the questions should be as concise and clear as possible [Coats 1987].

Approaches such as incremental or tapered prompting can help the system be clear without being too verbose. Such approaches give longer prompts to the user the first time a question is asked or if an incorrect input has been received; when the user is used to the question or has entered a correct response previously, then shorter queries are employed [Grasso 1997].

Since question and answer is such a simple interface, it can become repetitive or tiring for users if not varied with other approaches. In situations where the next question does not rely on the answer to the previous one, a form-based interface may be more appropriate. When the range of inputs to respond to the question is small, then a menu-based approach should also be considered.

Menus

In a menu-based interface, the user is presented a list of alternatives and chooses an option from this set. Generally menu options are displayed on the screen as text, but for an illiterately accessible interface, this text could be replaced by icon images and speech output. If icons on the screen are highlighted as they are read aloud, then the user could watch the screen and press the icon that corresponds to their selection. Unlike the Q&A approach, menu-based interfaces do not require explicit prompting for input. By presenting icons on the screen and reading options aloud, the system implicitly prompts the user to make a selection. Menu interfaces are ideal for situations where there is a small set of options [Coats 1987] or the user is at a decision point [Downton 1993], but they are generally awkward to use for scenarios requiring significant data entry. Menus can be more intuitive for novice users because they provide the user with a list of options, thus requiring the user to recall fewer specifics of the interface. Q&A interfaces can also have problems when users enter data in the wrong format; menus avoid this potential source of error.

When the set of options that a user can choose from is extensive, the options are broken down into a hierarchy of submenus to make them more manageable for the user [Downton 1993]. While the use of submenus doesn't generally confuse users of visual interfaces, individuals using speech interfaces are more likely to get lost in the structure of the menu tree. To avoid such confusion, menus presented aurally should avoid becoming too deep, and the visual channel of the interface should be used to help the user understand where they are in the menu structure. While this concern with depth may encourage designers to use very flat menu structures, the limits of users working memory cause menus with more six or seven options at a particular level to be confusing to listeners [Coats 1987].

Form-filling

Electronic form interfaces mimic their paper counterparts. The computer displays a set of questions on the screen, and users can enter responses for each and then submit all of the responses simultaneously when completed. In a form-based interface, the computer controls much of the initiative since it is asking questions of the user – making it a good approach to novice Simputer users. When a user is entering information in a standard sequence or initiating a transaction request, a form-based interface may be a good alternative [Coats 1987].

Form-filling interfaces differ from the Q&A approach because they require multiple responses to a single multi-part query. However, for illiterate users, this distinction may be blurred since the user would be guided though the form in a linear fashion by the screen reader. The difference would be that the form filling interface would give the user the option of skipping some questions, filling them out in a different order, seeing progress through the whole set of questions [Downton 1993], and reviewing/changing responses before submitting them to the application.

For traditional computer users, forms have the added advantage of being a familiar interface; many people have seen paper forms filled out even if they have never done one for themselves [Downton 1993]. For illiterate users, the advantage of the form interface being familiar may be less clear. The proliferation of paper forms in the rural Indian communities targeted by the Simputer would need to be investigated.

For illiterate users, form-based interfaces have the advantage of having a consistent appearance. Whenever input or output follows a consistent, memorable sequence, then illiterate users will have an easier time of understanding the system because the data will require less interpretation each time. A user may remember the sequence or layout of questions on the page and be able to fill out the form after having been taken through it a few times by the screen reader.

The ordering and arrangement of the form's questions should follow a natural, logical progression. It should also correspond to the local customs and language as much as possible. For example, in some cultures the surname precedes the given name when both are used; forms for such a community should follow this convention. In addition to sequencing, a logical clustering of the questions can make the form more intuitive for the user, and allow the prompts to be shorter in length since they naturally follow in progression from the previous question. This would be a great advantage for speech interfaces in which the reading of information to the user can slow down the interface.

Command Language

In a command language interface, the initiative is given entirely to the user to drive the operation of the system; the device would simply respond to and obey commands that the user submits. (A common example of such an interface is the DOS or UNIX command line.) Although concise and fast, this interaction type is error-prone and hard to learn [Downton 1993]. Command language interfaces are difficult to learn because the user must memorize the syntax of a language of input commands – perhaps an unreasonable expectation for illiterate users. It is also difficult to provide useful help messages on such systems since there is no context of a currently engaged task or dialogue. For these reasons, it is generally the least ideal approach for novice users.

Despite its problems, a command language dialogue may be a sensible approach for situations in which a particular set of commands are always valid on the device no matter the form or menu the user is currently on. For example, when navigating through information content, a command-language approach to processing the verbal inputs 'quit', 'home', 'next', and 'back' may be appropriate. In limited domains such as this, even novice users may be able to navigate such an interface.

Direct Manipulation

Direct manipulation interfaces emerged on the computer scene with the advent of graphical display interfaces. The file management interfaces of popular operating systems such as Windows or Macintosh OS use this graphical approach. The premise of a direct manipulation interface is that it closely follows an interface metaphor that can be communicated visually on the device's screen. Interactions with the objects in the virtual space correspond to actions on the actual computer [Downton 1993]. For example, in the Windows domain, an office and desktop paradigm is adopted to represent the information on the computer as files and folders.

While many users find direct manipulation interfaces intuitive and easy to use, there is a learning curve to this interface approach. The user must become comfortable with the display conventions and metaphors of the system. Small screens, lack of familiarity with the office metaphor, lack of experience with the windowing interface standard make this less attractive for the Simputer domain. With much of the content of the Simputer device being transmitted via IML (a simple form of HTML adapted for this domain), some of the drag-and-drop interactivity of direct manipulation systems may not be possible.

However, many of the graphical input and output conventions of modern direct manipulation GUI interfaces can be adopted for an illiterately accessible interface. The non-linguistic mode of graphical interaction provided by highlighting selections or clicking on icons would be very desirable for this domain. An attractive graphical interface would also be important for the novice users of the Simputer; by making the interaction engaging and appealing, users of the system would be encouraged to use the system and explore its features [Downton 1993].

Natural Language

In a natural language interface, the computer system can understand free-form user input written or spoken in a human language. The system could follow commands given in human language, and it could provide information and feedback in that same language. Such a system would mimic the dialogue interactions between humans, and would therefore be an ideal solution for novice computer users. Little training would be required to operate such a computer system [Downton 1993].

An optimal dialogue should exploit the natural strengths of each participant. Unfortunately, the areas at which a computer excels are not those surrounding the complexities and ambiguities of human language [Downton 1993]. This makes programming a computer to understand human texts and interact in a dialogue using natural inputs quite a programming and linguistic challenge. Unfortunately, the current state of computational linguistic technology and the overhead in customizing such a system to a particular application prevents high-quality natural language systems from being developed for most domains. Especially considering the diversity of languages used throughout India and the fact that only a small fraction of them have received significant attention from the computational linguistic community, the accuracy of natural language interfaces for understanding these languages would be very poor.

Because of the limitations of current linguistic technology, natural language interfaces can be built for specific domains in languages such as English or German, which have developed computational linguistic communities. However, even these systems will suffer from difficulties in processing ambiguities and managing imprecise input [Downton 1993]. Users can have difficulty determining the limits of the system's vocabulary, and often the error states they encounter after a misunderstanding can be too confusing to recover from successfully [Baldwin 1997]. As with speech interfaces, users of natural language systems are lured into attributing too much intelligence to the system [Downton 1993]. This is a particular risk among users with little or no previous experience with computer technology that do not understand the limits and possibilities of such systems.

Chapter 5: HUDS Introduction and Scenario 1

Previous chapters have used user-interface design literature and analysis of the system's users, environment, work, and technology to shape the recommendations for an illiterately accessible interface. The next two chapters will demonstrate the use of a design tool called Hypothetical User Design Scenarios (HUDS) to explore new issues in the interface and spark new ideas for its design. In the following sections, the advantages of the various user-interface design approaches are briefly discussed, the reasons for choosing the HUDS design tool are outlined, and the first HUDS scenario is presented.

Approaches to User-Interface Design

The Human-Computer Interaction and User-Interface Design literature have outlined several approaches designers can use when working on a project, many of which have been considered while designing the URSULA guidelines. Broadly, these approaches vary according to how much empirical information they require. Some design methods involve interaction with or observation of users of the system – a costly yet often worthwhile investment of resources. Other approaches are conceptual design processes which designers can follow to focus their ideas for an interface, think through potential problems, and harness their previous design expertise. For the URSULA project, such a conceptual tool was developed in order to drive development, namely the Hypothetical User Design Scenarios (HUDS) approach.

Experimental Approaches

Some user interface design approaches discussed in the literature require experimental or observational work in order to guide the interface design process: holding focus groups with potential users of the system, observing users of the system working in their current environment, organizing Wizard-of-Oz tests, or running usability experiments to find out if an interface prototype meets an established usability metric. While information gained from such studies is extremely compelling because it grounds design decisions in empirical findings, they can be costly and time-consuming to run. For a design project that is small in scope, has a limited number of specific issues to be decided, or whose intended users are easily accessible for experimentation, such empirical approaches are well suited. Such testing is poorly suited to interfaces for the developing world. Gaining access to target users, rural Indian villagers with literacy challenges, for experimentation would be challenging and costly. Also, the scope of URSULA is quite broad -- to establish guidelines for building interfaces for multiple handheld applications -- experimentally testing interface designs for such diverse applications would be impractical.

While gaining empirical data to guide the entire design process is not possible for URSULA, empirical methods can be employed if used in a limited context. Instead of running a trial experiment of every possible interface, limited experiments can be run after the broad shape of the interface has already been determined by less time-consuming approaches. Empirically based design methods can be employed to make fine-grained distinctions in the final interface or to help make choices at critical points in the project's development. Empirical methods can also play an important part in the usability evaluation process of an URSULA-produced user-interface. (Such possible uses of experimentation are discussed in the Future Directions chapter at the end of this thesis.)

Non-Experimental Approaches

Fortunately, the HCI and Design literature also offer many tools and approaches to designers who are looking for non-experimental approaches to decide upon an interface. Many of the methods discussed below have been incorporated into larger HCI design frameworks, such as the User-Centered Design process of Gould and Lewis [Gould 1985] or the Star Model of Hartson and Hix [Hartson 1989] [Hix 1993]. Such broad frameworks help organize critical design tasks such as requirements gathering, conceptual design, functional analysis, prototyping, and implementation into an ordered process that can be followed by the creator of an interface.

In one approach, Task Analysis, the designer identifies and traces the cognitive tasks and subtasks which the user needs to accomplish in order to successfully interact with the interface [Johnson 1992]. As this analysis is performed, the designer enumerates the individual "objects" and "procedures" in the domain which the user will need to interact with conceptually. Important cognitive challenges that the user may face in a particular design are also brought to light.

In Structured Design, the designer composes a grammar for the interaction between the user and the interface [Preece 1995]. These grammars which are created can take into account progressively finer levels of detail: the task itself, the semantics of the portions of the interaction, and syntax of these interaction steps, and the exact interaction steps performed. This design methodology often employs formalized grammar notations to capture the interaction in format that can be reused in later portions of the design process and easily communicated to others.

These first two approaches share some common disadvantages. They can lure the designer into making design commitments too quickly by forcing him or her to specify the interface at too fine level of detail early in the interface-creation process. This tendency may encourage the designer to make arbitrary or poorly motivated design choices merely to satisfy the degree of specification required by the methodology. Both of these techniques also require the designer to put forth a large amount of specification and cognitive analysis effort in order to develop a small portion of the final interface. For large projects, this ratio of effort to output might not be practical.

A third design approach often discussed in the literature is perhaps the most intuitive; in Visual or Holistic Design, the designer creates an initial mock-up, image, or idea of how the system will appear through visual sketches and brainstorming [Preece 1995]. Several alternative designs are constructed and compared, and the designs are made available for others to examine and critique. Through this creative process and cycles of criticism, a final design will emerge. While some of the previous design approaches could be criticized for approaching the design task too obliquely, Holistic Design may tend too far in the other direction. Although it can be appealing to jump into the design process by creating some possible interfaces, without careful forethought poor initial decisions could become enshrined in the final design and important user issues could be ignored.

Hypothetical User Design Scenarios

In a HUDS, the design tool used for the URSULA project [Huenerfauth 2002], the designer specifies a potential user of the system with a detailed information/application need and a particular set of environmental factors. Prior to writing the first HDUS, the designer investigates the users, technology, applications, and usage environment of the system, and he or she performs a survey of previous user-interface research work relevant to his or her users. Next, all locations, organizations, individuals, and important issues that might impact how the designer should think about the interaction are identified and described.

For each scenario, the designer describes a situation that would motivate the user to use the system and constructs a script for the interaction between this user and the interface. As the designer crafts this interaction, he or she notes any issues which arise or open research questions which would need to be solved in order to create a high-quality interface. After the script is completed, the designer enumerates these open issues, discusses the approach he or she took to addressing them in the script, and then proposes a set of future research questions to be explored that would give greater clarity to the interface. This script-writing process can be iterated to consider a diverse set of applications and issues for the user-interface.

The structure of the HUDS makes it easy for the designer to record the progress made during its creation and apply it to future stages of the design work. While there are many design techniques which incorporate scenario-writing approaches [Benner 1993] [Carroll 1995] [Redmond-Pyle 1995] [Nardi 1992], the value of the HUDS is the template it provides the designer for specifying information beyond that included in the scenario itself. The designer analyzes the problem space before writing the first scenario using a 'users, technology, work, &

environment' structure. Insights into the user and the system environment are recorded at the beginning of each scenario in the "Entities and Characters" section. During the course of writing the interaction script, important design issues which arise are noted and recorded in the "Analysis of Issues in the Scenario" section. Most importantly to the future stages of the design process, the HUDS includes a section where the designer can record open research questions -- these can spark ideas for future experimental work to guide the design.

The HUDS is a working tool which allows the designer to flesh out potential interface issues that may have been previously overlooked, to encourage the designer to think about the interface in increasing detail, and to prompt discussion and comparison of ideas. The level of detail the scenario specifies about the user and his or her environment and situation may surprise readers unfamiliar with scenario-based design approaches; for example, the HUDS example in this chapter carefully details the geography and economics of the town in which the hypothetical user lives. By making these aspects of the user’s experience as concrete as possible, it's harder for the designer to gloss over potential problem areas in the interface and it is easier to discuss the design later with others. A HUDS-facilitated discussion with other designers would be more grounded in specifics, making it easier to critique and compare candidate approaches.

While a visual Holistic Design (HD) mock-up could also have facilitated sharing and criticism of a candidate design, the HUDS accomplishes this while overcoming some of the disadvantages of HD. Both HD and HUDS allow a designer to propose an approach to a user-interface problem and communicate this proposal to others. Both techniques encourage designers to think of their design with increasing degrees of detail as the proposal is drafted. However, the typically visual medium of HD may lure the designer into making some interface commitments too soon, may trap him or her into one view of how the system should operate, and may discourage criticism from others if the mock-up is presented too attractively. (It can be difficult to criticize something that looks visually appealing, especially when one does not have an equally attractive alternative design to offer in its place.) Unlike HD, HUDS encourages the designer to think about the elements of the user interface as individual tasks or as an interaction script; in this way, it can incorporate thinking from a Task Analysis or Structured Design approach.

HUDS 1: Distributing Health Information

Synopsis

The World Health Organization has detected the early indications of a cholera outbreak in the Mahbubnagar district of Andhra Pradesh. The WHO needs to communicate important information about the disease to residents of this area so that they can take steps to avoid infection or to recognize symptoms of the disease to prevent dehydration.

Entities and Characters in the Scenario

World Health Organization

The World Health Organization is a non-governmental body that coordinates international health initiatives. The organization works to eradicate various contagious diseases, particularly in developing countries. In epidemic situations, the WHO coordinates with local government agencies and the broader health community to combat the spread of disease. One of the most valuable responses to these crises is to disseminate timely health information to the affected communities. The improved access to networked information afforded by the Simputer can provide another channel for this information.

Geography of Mahbubnagar

Mahbubnagar is a district in the west of Andhra Pradesh, a state on the east coast of the India peninsula. The district is on the boundary between a mountainous area to the north and a coastal plain with many rivers and creeks. The climate is typically hot and humid with an annual rainfall of over 125cm. The population is mostly Telegu-speaking with some Kanarese and Kannada speakers on the western edge of the district. Agriculture is the most important sector of the economy with rice as the most significant crop. The political status of the region is somewhat under dispute; there is a political movement within Andhra Pradesh for a separate Telegu state.

Cholera

Cholera is a severe intestinal infection caused by a bacterium called Vibrio cholerae often found in infected water or food. Large outbreaks of the disease can occur from a contaminated water supply; young children are particularly sensitive to infection. Symptoms of infection include diarrhea and vomiting which can lead to severe dehydration and death without prompt treatment.

Jainder

Jainder is a farmer near the town of Gadwel in southern Mahbubnagar. His wife and three daughters live with him near a river. One of Jainder's daughters has become ill and it is important for him to learn of the epidemic. Also, his family's drinking water comes from this river and it may be contaminated. Jainder is able to speak but not read Telegu.

The Scenario Script

Options in This Scenario

There are three different types of proposed interaction styles discussed in parallel in this one scenario script. The script will jump between the three possibilities, and sections that correspond to each option (or to all of them) will be marked:

- Option A: Notification to User
- Option B: Advertisement on Portal Screen
- Option C: Information Available for User Query

Step 1: Background to the Scenario

For All Options

Jainder finishes his work for the day just before dusk, and he returns home to collect his SIM card. He travels to the local school building to use the community-owned Simputer. He arrives at the school, the attendant signs him into the logbook, and he inserts his SIM card into the Simputer. He turns on the device, and it connects to the Internet using a phone line and its internal modem. The Internet service knows that Jainder is a resident of Gadwel Township in the Mahbubnagar district because this information is stored on his SIM.

Step 2: Using the Simputer

Option A: Notification to User

Jainder is using the Simputer this evening because he needs to find an appropriate market to sell his goods next Saturday.

As soon as the device is connected, a loud tone sounds and a text box flashes onto the screen. The box fills the screen and it has a flashing border to draw attention. A voice reads the content of the message to Jainder: "Important health alert for your town!" There are two buttons on the screen – a circular one which says "More Information" and a square button which says "Ignore." The voice reads the content of each button, telling Jainder to press the circle if he wants to learn more and the square to continue with his work. As they are mentioned by the voice, the circle and square buttons are highlighted. Jainder chooses to touch the circle.

Option B: Advertisement on the Portal Screen

Jainder is using the Simputer this evening because he needs to find an appropriate market to sell his goods next Saturday.

When the device is connected, the usual "start screen" is displayed. This screen presents Jainder with a list of options he can perform using the Simputer and its Internet connection. The screen is filled with picture/icon buttons that have small text captions below them. As Jainder watches the screen, each of the button options is highlighted in turn and the Simputer voice reads aloud the name of the button and brief description of what it will do. Since Jainder has used the device several times before, he is already familiar with what the various icon buttons signify. By pressing on a button on the screen, Jainder can trigger a menu of options or a particular information retrieval task. Jainder can also point to a special place on each button to ask the system to read the button aloud instead of triggering an action.

At the top of the portal/start screen, new buttons, advertisements, or announcements are displayed. This portion of the screen is highlighted and read first since it contains previously unseen information. An announcement about the Cholera epidemic is currently at the top of Jainder's screen. This announcement has a special high-priority highlighting, and there is a special urgent tone / voice which is used for this announcement. Since the announcement sounds important, Jainder clicks on its button to learn more information.

Option C: Information Available on Query

Jainder is using the Simputer this evening because his daughter is ill. There are rumors in the town that a dangerous disease is spreading to young children. Jainder wants to find out if this is true, and he wants to find out how he can help his daughter.

After the Simputer connects to the Internet, Jainder realizes that he is unsure of how to find out the information he needs. He doesn't know if he should search for a doctor, e-mail someone in a nearby town, read a local news report, or find out first-aid information for his daughter. As the options on the start portal page are read aloud, the "Health and Medicine" option catches Jainder's attention. He decides to look in this area. There are several options within: (1) "someone is sick," (2) "where is a doctor?," (3) "how do I stay healthy?," (4) "health news for my area," and several others. Jainder chooses "health news" so that he can find out if the rumors are true.

In the health news section, the Simputer reads the headlines to Jainder. There is a story about a disease alert for Mahbubnagar; so, Jainder clicks on this option to hear more information.

Step 3: Reading the Health Information

For All Options

The screen fills with an animated image of a person drinking from a river, feeling ill, and then going to a doctor. On the screen, the text of an epidemic alert issued by the WHO about the cholera outbreak is displayed. The Simputer voice reads the text of the alert aloud to Jainder. The information says that there is a dangerous disease in his town, that it may spread through contaminated water, that people should go to a special center for treatment, and what the symptoms of the disease are.

Since there is a lot of information, it is broken into pages. At the end of each page being read, Jainder is presented with the option of going back, repeating, going forward, quitting, or asking for help. The device uses standard icons at the bottom of the screen for navigating the information; as each navigation option is read aloud, the corresponding button is highlighted on the screen. Whenever the Simputer presents more than one page of information to a user, these same navigation icons are used. Jainder can either press the buttons on the screen, or he can say verbal commands such as "go back."

Sometimes, instead of progressing through the entire article, the system will ask the user a question to decide if it should read the next section of text or jump to another. The system asks Jainder if he might know someone with the symptoms of cholera. Since Jainder knows his daughter is sick, he answers "yes."

The system presents Jainder with a set of brief first aid directions for the sick person. It tells him to keep them hydrated and to bring them to the emergency cholera treatment center write away. As each of the steps of the directions is read aloud, a corresponding icon is highlighted on the screen. These icons represent the steps of the operation, and can help Jainder remember the sequence. Jainder can also point to any of the numbered items in the directions to have it repeated for him. For the directions to the health center, the system displays a map. Since the system knows that Jainder is from Gadwel, then it can give him personalized directions from where he is starting.

Step 4: Leaving the Device

For All Options

The system offers to print a copy of the directions and map for Jainder to keep; he agrees. Although he will not be able to read all the instructions on the page, he can still be reminded of the steps by the icon images. Also, he may be able to show the instructions to someone else that can read them aloud for him later. When he tells the Simputer that he would like to "quit" reading the cholera information, it offers to bookmark the information so that Jainder can easily find it again later. He agrees.

Map and instructions in hand, Jainder turns off the Simputer and quickly returns home to his daughter.

Analysis of Issues in the Scenario

As the scenario script above was developed, various issues came to light which are critical to the successful design of this interface. Each of these issues is discussed as a topic below.

Making the User Aware of Critical Information

When the system needs to communicate critical information to the user, there are several possible approaches. The system could send the user a notification which could appear on the screen when he or she signs on to the Internet, the system could advertise this information as a link on a starting Internet portal screen, or the system could allow the user to initiate a query and return information only if requested.

If information distribution techniques such as notification or portal advertisement will be used (as in options 1 and 2), then standards will be need to be established about what information is critical enough to be conveyed in these obtrusive manners. While advanced users might be comfortable with notifications and frequently changing portal pages, novice users could become confused. Whether personal information and user preferences will help determine announcement policies is an open question. Whether notification or prominent portal placement would also be used for commercial advertisements is another.

The design of the opening screen should anticipate as many common information retrieval tasks as possible. Expecting the user to navigate deeply and search for information (as done in option 3) may require an unreasonable level of computer literacy or luck. Allowing the device to control the initiative of the interaction would be best for most information retrieval tasks.

Expression of Urgent Information

URSULA will need to decide how a literacy-accessible interface should demarcate some information as more important than others. Forms of visual and aural highlighting that would be most effective for a population with limited computer and written-language literacy will be needed. Some standard forms of highlighting rely on cultural or reading conventions that would not have meaning to the target population of this device.

Highlighting or flashing is one way of drawing the user's attention to important information, but this type of marking might confuse the user. If another part of the screen is also highlighted because it is the current text being read, then the user may not understand what the device is doing. An aural form of importance-marking may be more appropriate. Perhaps tone of voice or an audio signal can be used to convey urgency.

Specifying Geographic Information to the System

Users will need a way to communicate their geographic location to the device. While a home location could be stored on a SIM card, an approach is needed for other locations. If typing the names of locations is not possible, then using maps seems like a likely input option, but we might not be able to assume map literacy among the target population. If names of geographic locations were in dispute for cultural or political reasons, then the device would need to be sensitive to this.

Communication of Conditional Advice to the User

The system may need to recommend different medical information according to the symptoms or questions of the user. This is a specific example of a general issue: the system will need to express different types of information to the user based on the user's current state or information needs. There are several ways in which this communication of state/needs could take place: the system could initiate a set of queries and await responses, the user could make one long query containing lots of state information, the system could present the user with a form, or the system could give the user a menu-like listing of all of the anticipated queries or information needs.

Communication of Sequenced Instructions to the User

When a computer system presents sequenced instructions to a user in a written form, the numbered list format of the information helps to communicate the sequence of the items and acts as a reference for the user while enacting the instructions. The designer will need to decide how to provide the user with mnemonics to help him or her remember the sequence when the information is presented aurally. Perhaps non-linguistic graphical screen elements could be used as visual landmarks of the numbered instructional items.

Organization and Presentation of Health Information

If a high-priority information goal of the Simputer will be to distribute health-related information to the user, then the typical format of this information and its accessibility to an illiterate population needs to be considered. If pictures or diagrams are to be used to facilitate this presentation, then these images must be sensitive to the cultural attitudes of the target population. Facilities should also be provided for the user to ask for medical terminology to be defined and explained.

Taking Information Away From the Simputer

While illiterate users may be able to access information while it is presented on the device because of its reading capabilities, these users will have difficulty taking information away from the device. The ability to print information, store results, take notes, and bookmark pages would help address this problem. Pictures and graphics that can be copied or printed would also allow the user to take information away from the device.

Literacy Accessibility for Web Resources

Since many of the most exciting information access scenarios for the Simputer would entail internet connectivity, the URSULA project may need to consider the user interface design process as not just involving developers of applications but also websites. If so, then the user interface experience will be constantly developed and modified as new web sites are developed.

While application developers may be willing to invest large portions of time creating highly accessible, dialogue based, or speech friendly interfaces, web developers may be more concerned with generating content. While it will be important to consider innovative new interaction styles in the URSULA project, the guidelines produced should also offer simple screen-reading standards for accessing web resources that have not been designed with illiterate users in mind.

There is also potential for shared development of common web resources required by multiple services. Generating maps and directions for areas of India may be one of these potentially sharable capabilities. Many of the high-priority scenarios for the system involve the user looking for a geographic location at which to get medical attention, sell goods, or ask for work.

Advantages for Partially Literate and Experienced Users

Since speech-based user interfaces can sometimes lead to slower user interactions (because portions of the screen need to be read), the interface should be designed to take advantage of the skill of returning or partially literate users. The interface should have the ability to detect and adapt to the literacy level of the user. If the user is partially literate, then simple text captions could be used on screen to help the user jump to the right choice without waiting for a screen to be read. If the user already knows what icon to press to initiate an action, then he or she should be able to interrupt the reading of a list of options in order to trigger the desired behavior.

Screen-Reading Mode Interface Issues

While elements of the screen are being read or described, the user should be aware of what element the system is currently talking about. There will need to be some form of visual highlighting to help the user follow the audio captioning. This focus tracking will make it easier for the user to understand what element of the screen needs to be clicked in order to select an option that the system reads aloud. There should also be a facility for the user to click on objects in order to request that they be read or described – instead of triggering their default behavior.

Areas of Future Research Highlighted

The user-interaction script and the discussion of critical design issues in this Hypothetical User Design Scenario have highlighted several open research questions. These issues could be explored through user-interface experimentation to further illuminate the design process for the URSULA interface.

- Do novice computer users understand how to deal with pop-up notifications? Does the wording of the notification affect their success?
- How do various forms of visual highlighting affect a user's success as identifying the intended piece of most important information? How long do various forms of highlighting take to draw a user's attention to important information?
- How successful are various forms of aural highlighting at helping a user find the intended piece of most important information? Are special sounds or voices effective at communicating importance or urgency?
- How successful are users at following aural geographic directions presented in various forms: with map, with turn-by-turn directions, by place names, by geographic features?
- Do illiterate users prefer to have written text appear on the screen and be highlighted as read? Or is it better for no text to appear and for the interface to be entirely speech based?
- How successful are various approaches for allowing the user to indicate portions of the screen that he or she would like to be read or explained by the device's voice?
- What set of visual navigation icons (go back, repeat, go forward, quit, help) are most intuitive for illiterate users?
- What forms of screen printout are most effective at allowing users to remember long or complex directions? Numbered items with pictures? Maps and diagrams? Written text that can be read by a literate acquaintance?
- Which forms of user-querying are most effective for getting information from illiterate users? Open-ended queries? Yes/no or short-answer questions? Forms? Menu Options?

Chapter 6: Hypothetical User Design Scenario 2

This chapter explains how a HUDS-based design process continues after the initial scenario has been prepared. The HUDS approach provides a framework for continuing to drive the design process after the initial HUDS script is developed. In this chapter, the reader is taken through another iteration of the process by identifying unexplored issues in the design, constructing new hypothetical user situations that intersect these issues, and performing additional HUDS analyses to address them.

By creating new scenarios that explore diverse users, applications, and technologies, a clearer picture of the critical issues and requirements of the interface can be developed. In this way, the HUDS process can be iterated until the designer is satisfied with the amount of design guidance and analysis the approach has facilitated. The first section in this chapter discusses a set of issues not covered in the HUDS scenario presented in the previous chapter. After identifying these topics, a new hypothetical user and situation are constructed which incorporate them, and a second HUDS analysis is performed at the end of the chapter.

Issues Not Explored in HUDS 1

Aspects of the situation and user presented in HUDS 1 prevented that scenario from exploring several important issues for the Simputer interface design. Before constructing a second HUDS, it is useful to consider which of these issues need to be included in a follow-up scenario. There was only one user in HUDS 1, Jainder, a farmer with negligible written language literacy skills; so, applications of the Simputer which might have involved multiple users, communication between users, and interface elements aimed toward partially literate users were not explored.

In scenario 1, the user was merely a consumer of online information, not a producer; in this HUDS, users will post information to online depositories. This change will allow us to investigate the complexities which arise from online information management and written text composition by illiterate users. With e-mail one of the most popular applications of Internet technology and with some studies of information technology projects for developing countries showing e-mail as one of the most sustainable user applications [Aral 2001], it will also be important for this scenario to address how e-mailing might be accomplished in the Simputer domain.

HUDS 2: Finding Employment

Synopsis

A farm worker uses the Simputer to locate employment in a nearby district. He uses online bulletin board services and e-mail to interact with his potential employer.

Entities and Characters in the Scenario

Farm Laborers in Indian Agriculture

Because of its warm climate, India supports a variety of agricultural production year-round. Most of this production is done by small farmers who rely on hiring temporary farm help for labor-intensive planting and harvesting work. These workers often need to travel some distance to follow the seasonal and regional crop cycles. Instances of drought can also dramatically reduce the employment opportunities in a particular area by decreasing crop yield.

Shetimitra

Shetimitra is a farm worker living in Sarkoli, a small village near Pandharpur in Solapur District. His village is experiencing a severe drought, and he will need to travel in order to find agricultural work. He is unsure of where employment opportunities exist outside of his village. Shetimitra is able to speak but not read Marathi.

Geography of Solapur

Solapur is a district in Maharashtra, a state in the west of central India. The state is on the western coast of the country on the Arabian Sea; however, Solapur is on the inland side of the state and plateaus and river valleys characterize its geography. The region has a tropical monsoon climate. Floods caused by heavy rains as well as droughts caused by long dry breaks negatively affect the agricultural industry. While this district is particularly industrialized, agriculture is still an important sector of the economy — wheat being a significant crop near Solapur. The population is primarily Marathi-speaking.

Patil

Patil is a farmer in village near the riverside town of Kilarni, in the Latur district of Maharashtra. Harvest time is approaching in two weeks, and he will need several workers to help reap his wheat crop. Patil is partially literate in written Marathi.

Geography of Latur

Latur is another district in Maharashtra; Latur is northwest of Solapur in a region of the state called Marathwada. Marathwada is a slightly drier portion of the state; however, there is fertile agricultural soil on the banks of several important rivers. This region is also Marathi-speaking.

The Scenario

Step 1: Placing the Advertisement

Patil finished checking the wheat crop in the early afternoon; it looked like there should be about two more weeks to harvest. Unfortunately, two of his more reliable farmhands had moved west to get jobs in Bombay several months before. Patil had been making due since then, but he knew that he needed to find a new worker before the harvest. He had overheard one of his neighbors talking about how he had placed a “help wanted” ad last month using the Simputer device at the town center. Patil had used the Simputer a few times before to check on prices for seed, but he had never placed his own advertisement before. Deciding that it couldn’t hurt to try, he found his SIM card and headed into town.

Inserting the SIM into the device, Patil turned on the machine and waited for it to sign on to the Internet service. Patil saw the word for “employment” on the start page and decided that sounded appropriate. (He knew that he could back up to the start page if he had guessed incorrectly.) Within this section, he listened to the explanation of the options and selected the link that was highlighted when the Simputer read, “Post a job listing here.”

The Simputer first explained to Patil what it meant to post a job listing on the Internet, who would be able to read his posting, that he would only be allowed to post a job listing for a maximum of a three month period, and that he should include a way to contact him when he filled out the posting form. The Simputer voice told Patil that it would ask him a series of questions about the job he would like to advertise, that the voice would read back to Patil what it thought he had said, and that he could tell the Simputer if it had understood him correctly. Many of the fields in the job posting form were multiple-choice; so, for these entries the Simputer voice read the options to Patil. For some of the multiple choice fields that had too many options to read aloud, Patil was asked to say what he wanted to fill into the field, and the Simputer read him a list of those options that were similar to what he had said. Some fields that asked for a geographic location offered to insert Patil's current location, show him a clickable map, or simply allow him to say the name of the place he intended. There were similar graphical data entry features for several of the fields for which a graphical interface would be appropriate.

A few of the fields were simply places where Patil could say several sentences describing the job. For these fields, the Simputer typed the text on the screen as he spoke it, gave Patil the option of bringing a "tap" keyboard on the screen, offered to read the text aloud to Patil whenever he paused talking, and highlighted each word as it read. Since Patil wasn't comfortable with his spelling, he didn't turn on this on-screen keyboard.

As he spoke, words appeared on the screen. At all times, a button was on the screen that would cause the Simputer to read all of the text, highlighting each word as it went. This button was a standard feature of many Simputer applications, and Patil knew to look for it when he wanted the device to talk. If there were words that were incorrect, Patil could tap them. Then the Simputer would ask him if he wanted to read, delete, or replace that word. If he wanted to replace, then Patil could then say the word again, and the Simputer would give him a list of options as to what it thought that he had said. Patil could select among these options or he could type out the word for himself. Patil could click the spaces between words to insert additional words into the text. At the end of the writing process, there were still a few mistakes, but Patil decided that the write-up was understandable enough to submit. When he was finished entering the information, the Simputer asked him if he was sure that he wanted to post the job, and he said yes.

He had left his Simputer e-mail address as his contact information; so, he knew that he would have to stop into the town center every couple days to see if anyone was interested. The time for harvest was approaching quickly. Patil hoped that someone would see the posting soon.

Step 2: Looking for an Employment Listing

Shetimitra had never had any trouble finding work during harvest season. The farmers in the village knew that he was a good worker, and there were several farms that he helped at regularly. But ever since the drought, work had been difficult to find in Sarkoli, or any of Solapur for that matter. One of the farmers who had usually been able to hire him suggested that he check the Simputer for any jobs in villages nearby. Shetimitra headed for the local school to sign on.

After connecting to the Internet, Shetimitra listened to the Simputer read the options he could do. The Simputer told him to click on the text that flashed when he heard the option he wanted or he could repeat the name of the command that the Simputer read to him. When Shetimitra heard the "Employment" option, he decided that was what he needed; so, he said "employment." Eventually the Simputer read to him the "look for a job" option; Shetimitra repeated this as well. The Simputer told him that it would ask a series of questions about the type of job he is looking for. The forms he had to fill out included yes/no, multiple choice, text, and geographic entries (much like the form filled out by Patil). When Shetimitra didn't have a preference for a particular field, or when their were multiple values for a field that he would find acceptable, the Simputer gave him that option by allowing him to say the names of all the values he would like to search for. Shetimitra was willing to travel to several districts around his location in search of work; so, he told the Simputer this when he was asked the location of the job. At the end of the form-filling session, Shetimitra was able to submit the search for a job.

On his screen appeared a table of information. The Simputer told him that this was a list of jobs, and it asked if he would first like to hear what each column of the table meant before reading the jobs individually. Shetimitra said that this was ok. The Simputer highlighted each column heading and read a description of each value type. Icons on the screen were used where appropriate to illustrate the meaning of some fields. Then, the Simputer read through each job listing. The Simputer voice asked him if he would like to look at the entire list before deciding which employers to contact, he agreed.

At the end of each listing, the Simputer asked Shetimitra how interesting this job sounded (on a scale of 1 to 10); so, Shetimitra guessed a rating for each. The Simputer voice told him that he didn't need to be exact about this – that this information was just to help him remember his favorite jobs when he reviewed this list later. When the Simputer got to the end of the list, it asked Shetimitra if he wanted to look at the list again, this time listing the most interesting jobs first.

Shetimitra agreed. This time, after each listing was read, the Simputer asked him if he would like to look at this listing in more detail and find out how to contact this employer.

Shetimitra selected a farming opportunity in Latur district, and he heard the farmer's description of the job. At the end of the listing, Shetimitra said, "yes," when asked if he would like to contact this farmer. The Simputer asked if he would like to send the farmer an e-mail message now since the only contact information he provided was an e-mail address. Shetimitra said yes. The Simputer created an e-mail, addressed it to the farmer, and told Shetimitra that he could begin to say his message. Words appeared on his screen as he spoke, and the text entry interface was identical to the one that Patil had used the day before. When Shetimitra was finished with his e-mail, he printed out this job advertisement and the listing of others to show them to his friend.

Step 3: Reading the E-mail Response

The following day, Patil signed on to the Simputer. One of the notifications at the top of his start screen was that he had an e-mail message. Patil selected this option after hearing it read by the Simputer voice. Instead of starting on the standard e-mail screen, which would give the user several e-mail related options, the Simputer opened up his e-mail inbox since he had clicked on a new-e-mail notification link.

A table of incoming e-mails was listed on the screen. The Simputer offered to explain each column, but Patil did not accept. He had read e-mail many times before and he was able to understand the headings on the columns. The Simputer began to read the name and subject of the first e-mail message, and Patil told the device to "open" it. The message was from someone named Shetimitra, and since this was the first person to reply to the message, Patil replied to his e-mail and told him the job was still available. He told Shetimitra the name and location of his town was on the original job listing, and he asked him when he could arrive.

After replying to the e-mail and turning off his Simputer, Patil left the town center. He was relieved to have found help so quickly.

Analysis of Issues in the Scenario

Posting Information to Servers

Many peer-peer applications of the Simputer will involve users contributing to online collections of information by writing to message boards, uploading news articles, sending out e-mail messages, or posting employment opportunities. The interface for such posting/uploading operations should provide a consistent experience that is understandable and easily navigable for illiterate users. Form-like interfaces or simple spoken dialogue interfaces where the Simputer guides a user through filling out a form are two likely approaches to this information-posting task. For illiterate users, speech understanding technology would be necessary for this content creation process, and the Simputer may need to provide the user an off-line text entry/editing capability for longer pieces of text input (if the Simputer is capable of operating in an offline fashion).

We should not expect Simputer users to return to online depositories where they have posted information and to remove or edit their contents. While web communities could provide this capability, materials that are posted online should be automatically managed. Expecting users to remove or otherwise manage materials after they have been posted may assume too much computer literacy from the user base; so, techniques for encouraging the user to express dates by which postings "expire" automatically would be desirable.

Expressing the User's Geographic Location

Because the target population for the Simputer has poor written-language literacy skills, application designers may not be able to assume geographic map literacy. One solution to this problem presented in scenario 1 was to encode the village of residence of a user on his or her SIM card. Unfortunately, for traveling workers or those without a permanent address, such an approach will not suffice. These users may still need a facility for communicating their current location to the device in order to benefit from various Simputer applications. Entering the geographic location of other entities would pose similar challenges.

Selection from a Long List of Options or Categories

Many searching or querying applications for the Simputer may require the user to choose one element in a pre-defined set of options or categories. When this list is small, the system can merely read all of the options to the user and allow them to select one. When the list is longer, such as in types of employment as in this scenario, the user is encountering the "golden/yellow pages" problem. As in a business directory listing, there is an extremely long list of categories (types of businesses) that the user must select among to receive a list of information they desire (a listing of phone numbers).

In a literate user scenario, alphabetizing of the categories and allowing the user to browse them is sufficient to facilitate successful information retrieval. When the user is not written-language literate, task becomes more complex. One approach would be for the Simputer to ask the user to say the name of a category that they expect to exist, the system would return and read aloud a short list of possible categories that are close to the user's input, and the user would need to select among them. Identifying the situations in which Simputer users encounter the "yellow pages" problem and providing a consistent interface for handling this task will be an important step in the URSULA design process. The interface will need to guide the user through repeating their initial guess, trying a different item, and eventually allowing the user to listen to the entire set of possible inputs for a particular field if the previous approaches prove unsuccessful.

Helping the User Manage a Long List of Returned Hits

Navigating and making sense of a long list of returned hits from a search query can be a challenging task even for users with good written-language and computer literacy skills. Understandably, the target population for the Simputer will have additional difficulties with this task. When literate users are shown a long listing of possible "hits" from a search query, they are able to visually browse through them quickly, scroll up and down to read new entries, turn their attention to particular entries on the page at will, and remind themselves of previously read items by returning their gaze to those entries and rereading them.

In this way, literate users can rate, compare, and select hits asequentially. Illiterate users would need to rely on the Simputer to read hits aloud to them; the asequentiality and spontaneity of the interaction is therefore reduced. Because users will need to make judgments about each link as it is read aloud, they will not have heard all of the links before needing to decide if a particular page is what they need to see. Instead of forcing the user to select a link based on incomplete information, it may be useful to give the user a mechanism for "rating" each hit as it is read and storing these rating so that the user can see them on the screen. Then the user could select the link that he or she gave the highest rating or have the system reread the list sorted according to the ratings he or she assigned. In this way the list can be prioritized and reduced in size as it is being read aloud, and the user can get a sense of the entire "hit-space" before having to make a particular selection.

Unlike a written interaction, the Simputer voice interface will need to make a more difficult trade-off between the level of detail to present about each entry and speed of the user's hearing all of the returned hits. Whereas a text system can include both titles and descriptions of possible hits on the page simultaneously and rely on the user's visual skimming abilities to reduce the amount of information they are exposed to, the Simputer voice system will need to perform some of this information filtering process for them.

In every information retrieval application, the quality of the algorithm used to rank the resulting hits for the user is a major determinant of the user's happiness with the system. Since an illiterate user will be employing the Simputer's screen-reading technology, the presentation of results will be slower than with a text system; therefore, the ranking of these hits becomes even more critical. Advances in information retrieval search ranking algorithms will have a significant impact in the effectiveness of a Simputer hits-presentation interface.

As done in the scenario script, when listings of information are presented to users, whether they be e-mail messages, search hits, or job listings, the system should present the list in a table so that users with partial literacy skills can gain some information from seeing the text on the screen or as a print out. The Simputer screen-reading interface may need to include special handling routines for reading table contents to a user. The system may need to explain the table header captions to the user so that they understand what type of information is stored in each cell.

Contacting Individuals to Gain More Information

Although the Simputer may be able to put a user in touch with important information, it would be doubtful that the device would be able to satisfy a user's complete information needs automatically. A user will typically need to contact an actual person to gain more specific or up-to-date information than what can be located online. The interface will need to provide opportunities for the user to e-mail other individuals, or if this capability is not available, then it will need to provide ways for the user to print out or record a mailing address and a telephone number.

Taking Information Away from the Simputer

While information presented on the Simputer device can be read to illiterate users, they will have difficulty taking a copy of that information with them away from the device (aside from using some type of audio recording technology). Since one of the least expensive methods of producing a hard copy of information is to print it out, in scenario 1, techniques were suggested for helping illiterate users print out information that could be used to remind them of information heard on the Simputer. These techniques included printing out icons or other graphical items with each portion of text in an ordered set of instructions to remind the user of the steps to perform. The user could also have a literate acquaintance read the information to them at a later time.

For extremely long, complex, unordered, or non-specially prepared information, this icon-printing technique would not be appropriate. This type of information would be more difficult to memorize – “reminders” would not suffice. For example, if the user in this scenario wished to print out a page full of employment listings, the user needs more than simple reminders, he or she would need detailed information on each. The print out would be a reference tool, not a mnemonic.

Aside from labor-intensive approach of creating meaningful icons to represent every aspect of the detailed information about each employment opportunity, the Simputer interface will need a more practical way to allow a user to print out detailed information listings. One approach is to present information in a table format, so that at least the user could understand what “type” of information might appear in each column of the table. (For example, the user may know that a particular column stores street addresses, even if he or she can’t read them. Or another column may store wage information, and the user could them reread the information if he or she were number-literate.) In this way, partially literate users could reap some advantage from printing out the information.

If the user rated the elements in a list, as in the information retrieval interface discussed above, then printing these ratings in another column with each entry would also be useful. In this case, the user may only need a literate acquaintance to read those hits that he or she had already determined were the most important while online.

Composing Text without Literacy

There are likely to be many Simputer applications in which the user will need to author a block of text. Considering the form factor and target audience of the device, facilitating text entry using speech recognition would be a natural choice.

However, it's important to realize that many factors can contribute to errors being made by speech recognizers. In a literate speech input scenario the user can recognize these errors as they occur; he or she can see words as they appear on the screen and can see misrecognized words. The user can then use word processing skills to go back, correct the text, and continue the dictation.

Illiterate users would not be able identify errors as readily. Instead of expecting the user to read the text as it is recognized, the system will have to read the text back to the user, they will need to listen for mistakes, and then correct them. Being unable to read the text on the screen, the system will need to be able to read words as the user points to them in order for the user to correct the text. Illiterate users would also be unfamiliar with the computer word processing paradigm; so, they may need a novel approach to edit the text they have entered.

User Notification

As in scenario 1, a notification technique was employed in this situation: when Patil received an e-mail message, the Simputer made him aware of this as soon as he signed on. Notifications are particularly important for incoming communications because this technology will be for novice users who initially may not know how to check e-mail or frequently remember to do it. Since telephone connectivity may not be pervasive in some of the target regions for this device, the Simputer e-mail inbox may be the only possible means of contacting individuals encountered while using the Simputer. Providing individuals with a reliable communication channel would be particularly valuable for distributing critical information or for facilitating commerce.

Portal Screen Metaphor

The portal screen metaphor used in scenario 1 seems applicable to this situation as well. Having notifications, new entries, or temporary information at the top of the screen, and the remainder of the Simputer capabilities listed in categories at the bottom allows for a simple list-like interface that is well suited to screen reading. As many push-button telephone-based systems have demonstrated, nested menus with logical categories are effective for an interface over an auditory channel.

Offline Simputer Capabilities

This scenario reinforces another way in which the true value of the Simputer is dependent upon its ability to connect to the Internet, and thus to information and other online individuals. How/whether to use the Simputer in an offline mode, how to express this difference to users, and how to help them manage local tools and file space would also be challenging future research questions.

Advantages to Partially Literate and Experienced Users

Interactions with Patil and the Simputer have highlighted some ways in which partially literate users can opt out of some assistive screen reading features when they are not necessary for understanding. This demographic should be kept in mind during the user interface development and testing scenario. A possible way to make the device more useful to partially literate users is to store their individual preferences for various levels of reading assistance by the Simputer voice on their SIM card.

Advertising the Applications of the Simputer

When Patil was unsure of where to find help for his farm, he didn't think of using the Simputer until one of his colleagues suggested it. Patil didn't know the Simputer was able to help him post a job listing. Obviously, the word-of-mouth approach will be an insufficient way to help Simputer users learn the capabilities of the device and how to operate it. An interesting research question that arises is how to advertise features to the user in a way that will not be obtrusive or confusing. Perhaps the interface should include pop-up advertisements of new features, or there could be a rotating set of "featured" capabilities that are positioned at the top of the portal screen each day.

Identifying and Authenticating Users

Being able to uniquely identify users is critical to simplifying many Simputer scenarios; for example, individual identification and authentication could assist in: logging into e-mail, authenticating for returning to websites where content was created, personalizing start screens, and knowing personal information about a user such as geographic location or language. Whether personal information about the user information should be stored on the SIM card itself or on the server is an open question. Potentially, the SIM card could store merely a username and password for an individual. Storing information only on the card has the disadvantage of a lost or theft of the card means lost of the data as well. While even if a Simputer card with just a username and password were stolen from a user, there would be a danger of the thief being able to impersonate them online, at least the user could change their password later and make that card unusable. There should be a way for a user to log in without their SIM card or allow an identity to be copied to a new SIM card in case one is lost.

Future Areas of Research

The scenario script and the discussion of critical design issues in this Hypothetical User Design Scenario have highlighted several open research questions. These issues could be explored through user-interface experimentation to further illuminate the design process for the Simputer interface.

- How do we communicate to users the type of an entry field in a form? (i.e. whether it is a yes/no, multiple choice, or free text response) Is this something best done through visual or auditory clues? How explicit should the interface be about this difference?
- How would a computer illiterate person expect items to be sorted initially? How familiar is this population with sorting results of lists? What types of sorting or resorting do novice computer users need from a listing? Is it ok to remove this capability if it is confusing?
- How familiar is this population with filling out forms? Is there a metaphor to use to help the user understand the helper role the Simputer voice plays in the form filling process?
- What cultural paradigms are appropriate for e-mail? Inserting the SIM card? Signing on to the Internet?
- Which methods of advertising the capabilities of the Simputer do users find understandable, unobtrusive, and not confusing?
- When the input a user provides to a form is incorrect or unacceptable, when and how should the system communicate this to the user? Do approaches to this problem for literate populations work for illiterate users?

Chapter 7: Conclusions and Recommendations

The first section of this chapter will discuss general conclusions and insights developed during the year spent researching this thesis. What follows is an outline of recommendations drawn from the work in previous chapters: the 'users, environment, work, applications' analysis, the review of previous research projects in this area, the analysis of user interface recommendations made by other authors, and the two HUDS scenarios. Whereas design suggestions elsewhere in the thesis were made as they arose in the course of discussion, this chapter organizes these recommendations into a form that is more easily consumed by designers of illiterately accessible interfaces.

General Conclusions

Helping illiterate users operate a computing device is a difficult problem that unravels more layers of complexity as it is explored. During the design process, it was discovered how profoundly the various types of literacy challenges complicate a user’s ability to interact with a traditional interface. Aside from the most obvious problem that users would have reading pages of textual information for themselves, users would encounter difficulties interacting with interface buttons and menus, recognizing and understanding numerical information, browsing lists of options to make a selection, determining the status of the device from inspecting the screen, taking information away from the device, and interacting with labeled graphical elements like maps, diagrams, and charts. While the initial issue of users not being able to read paragraphs of text can be addressed via the addition of speech synthesis screen reading software, the other challenges mentioned above require more profound modifications to the user-interface. The fact that these users would also have little or no familiarity with computer systems places an even larger burden on the interface designer.

The human computer interaction and user-interface design literature provided little specific resources for designers interested in building software for illiterate users or members of the developing world. Many of the resources used in this thesis are drawn from papers discussing screen readers for the blind or speech-based interfaces for users working over the telephone or without a visual display screen. While the specific skills and needs of these users are not identical to illiterate users, the situations bear enough similarity to make these findings of interest to this project. Considering the efforts of several international aid organizations and small start-up businesses to provide information technology services to users in the developing world, the lack of current research in the area was disappointing. Hopefully, the continuing work of the UCD Adaptive Speech Interfaces group and the Simputer Trust will encourage more interest in this challenging realm of human computer interaction research.

While the special needs of illiterate user might best be met through redesigns of the computer user-interface to make it suited to speech interaction and animated images, the recommendations in this thesis take into account the technological and economic realities of the Simputer system when suggesting potential interface designs. The practical limitations on the bandwidth users of the device will achieve when connecting to a network limits the richness of animations, sounds, and voice data that can be transmitted to the interface – as does the processing and hardware capabilities of the system. The extensive use of animations and well-chosen icons to aid illiterate users might be a desirable way to enrich the interface, but it also places an artistic and design burden on the application and web designers. While some developers would be able to put forth this effort to increase the accessibility of their illiterate users, others may not; therefore, the interface should be designed so that it can help illiterate users navigate web content and applications that have not been specifically adapted to their needs. Because users in the developing world would have little time and money to spend on computer training, the will need to make special efforts to be understandable to novice users and to provide some of that training electronically.

Despite these challenges, this thesis has shown that it is possible to design an interface for a computing system that illiterate users would be able to operate successfully. By taking careful steps to arrange onscreen text such that it is easily understood and navigated when read aloud, illiterate users can navigate an information device with the help of screen reader technology. Non-linguistic graphical images can also be used to communicate system status, navigation features, and informational content to illiterate users. When the user is required to perform tasks that are ill suited to an aural interface, such as browsing a long list of onscreen options, then the interface designer will need to recognize these situations and take advantage of one of the approaches discussed in the recommendations section (later in this chapter) to make the interaction more comfortable for illiterate users. When users need to input linguistic information, designers can use the recommendations below to find ways to simplify the input options, help the user enter text through speech recognition software, record the user's voice, or make the interface flexible enough so that users do not need to perform complex text input in order to operate the system.

As discussed in the introduction of this thesis, this research project represents only an initial effort at designing an illiterately accessible interface. Because of the challenges and required resources of using experimental methods to determine aspects of the interface, this thesis employed problem analysis, literature survey, and HUDS scenarios to produce design recommendations non-experimentally. While these approaches have produced good insight into the challenges and needs of illiterate users and have helped to suggest an initial shape for a user interface, experimental trials will be needed to answer some design questions, validate the design process, check the usability of a prototype of the interface, and uncover unanticipated difficulties with the approach. For this reason, the content of the recommendations below may be modified or further specified as future experimental design work is performed. While performing experimental trials of the interface falls outside the scope of this thesis, the future directions for this research – including experimentation – will be described in the final chapter.

Advice for Designers

The objective of the Simputer and other devices like it should be to make all users, regardless of their literacy and computer experience, able to take full advantage of information technology. When designing a user interface with ambitious goals for the capabilities and services to be provided, it can be tempting to rush ahead with new applications and leave users with special needs behind. After designing interfaces for illiterate users as part of this research, I have an appreciation for how easy it is to take shortcuts in the interface that may make some tasks too challenging for illiterate users. However, it is important for the designer to maintain high standards of usability for special needs populations because these groups are rarely able to lobby companies for computer accessibility improvements for themselves. Illiterate users should not be treated like second-class or limited users of the system; instead, they should be thought of as the primary target population during the design process. The Simputer and other devices should be used to alleviate inequalities of information and resources in the developing world, not add another dimension to the problem.

Designers of websites and applications for the Simputer will need to be sensitive to the special needs and way of life of their users. As discussed above, a lack of written language literacy skills rarely occurs without associated difficulties using numbers, charts, graphs, maps, and other information artifacts of the literate world. These users will be unlikely to have familiarity with the operation or capabilities of computer systems since most devices would be inaccessible for them to use. Even when interface accommodations are made for these users, the types of mistakes and errors they are likely to make may differ from those of literate users; for example, errors in speech recognition input may produce strings of text which sound like but do not appear similar to words the user spoke. Web and application designers will need to anticipate these new types of errors. The information content, organizational structure, images, and other content of the interface should also be sensitive to the cultural taboos, political controversies, gender identities, religious values, and social conventions of the users in order to avoid offending or alienating users. When envisioning how the device will be employed, designers should bear in mind the social and family structures which influence the context in which the device is used and the cultural and economic factors which will affect the interests and demand for various services on the device.

As they work, designers of illiterately accessible interfaces should be conscious of where the initiative of the dialogue interaction is, what dialogue structure is being employed, and how common elements are combined to produce the interface for a particular application. By thinking about the user interface in this way, it will be easier to break it down into standard components and schemata. With this modular conception of the interface, designers will find it easier to use the outline of recommendations below to guide their work.

Structure of the Recommendations

The recommendations in this chapter are organized into short paragraphs and bullet points on particular aspects of the user interface design. The interface of a typical application is decomposed into three major parts (command and control features, ways in which users input information, and ways information is displayed to users) and recommendations applicable to each are discussed. By breaking apart the discussion of the interface into these parts and their components, it should be easier for developers to locate information of interest when working on particular portion of an application or website.

In addition to recommendations for developers, this chapter contains recommendations of interest to the designers of the Simputer device. This section also suggests how system-wide features should operate on the device, such as screen reading assistance, the start page portal, system pop-up notifications, and user help and tutorials. The recommendations also outline component technologies and capabilities such as user authentication via SIM cards, printing of information to hard copy, producing online libraries of common resources for application developers, and enabling posting of messages over e-mail and bulletin board services.

This chapter actually contains a mix of three different kinds of recommendations: principles, guidelines, and initial design specifications. Principles take the form of broad suggestions to application and web developers of ways to approach the design problem and important considerations when building the interface [Norman 1983]. An instance of a principle would be "illiterate users should not be second-class or limited users of the system." Guidelines are more specific and testable than principles; they consist of requirements that a particular system should satisfy in order to be successful [Norman 1983]. An example of a guideline is "users should be able to quickly return to the start page from anywhere in the interface." Finally, some of the recommendations below go beyond guidelines to specify an initial design for the system. These points are actually early design specifications of how the illiterately accessible interface should operate and are perhaps the most concrete form of guidance in this thesis for future developers.

Command and Control

The device will need to indicate to the user the set of possible commands that are available, and assist the user in making their selection. This section of the recommendations will discuss how this command interface could be structured for illiterate novice users.

For novice users who may only intend on using the device infrequently, navigating a windows-based graphical interface may be too complex and demanding. Such interfaces are also ill suited to speech input and output. An interface based on textual dialogue interaction or text-based hypertext navigation could be a more desirable way to guide users through the interaction.

Start Page / Portal Screen

- The opening screen of the device should be a start page or a portal to the services of the device. The screen should anticipate as many common tasks and requests of the user as possible. Notifications of e-mail, system messages, or responses from previously requested services should be saliently displayed on this page.

- The screen should function as a top-level menu for the operations and information available on the device, and it should be easy for users to return to this page at any time during their session.
- The page should include common commands and information requests organized into categories that are understandable to the novice users of the device. The structure of these commands in each category will be discussed in the next section.

Command Option List

- When the device is displaying a list of possible commands or information requests to the user, the menu of choices should follow a consistent interaction format. The system should display on the screen picture buttons with short text captions for each category or command.
- The audio portion of the interface should cycle through each of the options – reading each aloud with a synthesized speech explanation. As each item is read, it should be highlighted in some way so that the user knows what the audio is referring to. The user should be able to interrupt the audio presentation to select and item at any time.
- The user should also be able to request the audio presenter to move ahead to the next item or return to describing a previous one. When the end of the list of options is encountered, the user should be given the option of having it repeated, going back one level in the menu structure, or requesting some other service (such as a help page) from the device.
- As the user progresses deeper into the hierarchy of menu options, the screen should in some way display a trail of the branches of the menu tree that were followed to get to that point.
- In addition to structuring the categories of this menu in a manner that seems logical to the user, the list of options can be made more understandable through careful ordering. Alphabetical listings of options will have no significance to

illiterate users, instead newly available or popular menu options should appear at the top of each list.

- While users should be able to input their selection through a voice command, touch screen input should be encouraged because it typically leads to more accurate interaction. To facilitate the clicking of onscreen buttons with a touch screen stylus by novice users, screen elements should not be too small.

Users Entering Information

When the user has selected a command or operation to perform, oftentimes, he or she will need to input information to the device so that it knows what to do. How this information input should be structured is the subject of this section.

Forms

- Many of the information entry operations of the device should be conducted via a standard form-based interface. Forms have the advantage of being an input metaphor familiar to many novice users, and they simplify the dialogue between the device and the user. The user tells the system all the information it needs to know, and then it is the system's turn to provide a response to the user.
- The use of forms also benefits partially literate users. Whenever the input and output of the system comes in a consistent, memorable sequence, then users with partial language and computer literacy will have easier time remembering how it is structured.
- Forms are also easily adapted to a speech-based interface because the computer voice can lead the user through each field of the form sequentially. Each of the input fields can be of a different type, and the voice system can enter a subroutine appropriate to each (as discussed in the following sections).
- When the system can get personal information about the user from his or her SIM card, it should do so to fill in some fields of the form automatically.

Yes/No and Short Multiple Choice

- Yes-and-no questions are the simplest form of input to a form, and are easily adapted to a speech-based interface. The computer can ask the user each question in turn, and the user can respond yes or no with a voice command or button press.
- The system can use special icons, images, or shapes to distinguish the yes and no buttons, and the audio interface can make reference to this distinguishing characteristic. For example, it could say "press the square for yes or the circle for no."
- For multiple choice questions that have a short set of possible responses, the system can simply read all of the options to the user and allow him or her to select the desired option.
- Users can indicate their choice using some of the approaches discussed in the 'command option list' section above.

Long Multiple Choice

- Literate users would not have difficulty with a multiple-choice question with a long list of options if it were presented visually and sorted in a logical manner (perhaps alphabetically).
- For an illiterate user, when list of options for a multiple-choice question is too long to read all of the choices to the user, then the audio interface will need to take a more efficient approach.
- One solution is to arrange the options on the list into a menu-structure of nested categories so that the user can be presented with short lists of options at each branch in the tree until the final option is selected.
- If the number of categories is too large or the list is not well suited to such organization, then the user can be prompted to guess an entry or category on the list and say the name of it to the speech recognizer. The system can use sounds-like and thesaurus algorithms to show the user a subset of the list items that

sound like their input or are similar to it. Then the user can select an option from this shorter list.

- There will need to be a failure case for long multiple-choice situations. Either the system will be forced to read all of the options aloud to the user, or the particular field of the form should be made optional so that the form can be submitted without it.

Geographic Information

- When the user needs to input a geographic location to the system (or is presented with a geographical location as output), then an onscreen map may facilitate this interaction. The user can be provided with a text box in which they can enter the name of the location via free text (discussed below), but a map-based interface may be easier for most users.
- Since illiterate users may not be familiar with using maps and would be unable to read the place names on the map, the audio interface will need to give them an audio tour of the locations on the map. The system should also be able to read aloud the name of map locations pointed to by the user.

Input of Free Text

- The most challenging type of data input for illiterate users will be to enter words, sentences, or paragraphs of text into the system. Such unstructured blocks of textual information will be referred to as ‘free text’ in this section.
- Since it is such a difficult form of input to facilitate with a speech input, other forms of input discussed above should be used in favor of free text whenever possible. One reason for this is that all of the other forms of input can be entered using the touch screen; for many users, free text would require the microphone and speech recognition software.
- Even literate users who might have been able to enter text via an onscreen keyboard or handwriting recognition may be forced to use speech recognition software if keyboarding standards or handwriting technology is not available for their language.

- The major user-interface difficulty with facilitating the entry of free text via a microphone is how to help the user detect and fix errors made by the speech recognition software.
- Literate users would be able to see the text as it appears on the screen of the device. If a particular word or words are incorrect, then the user could click on that text and either re-dictate it, spell the words aloud, or type them using an onscreen keyboard if available.
- Illiterate users would need the device to read their text back to them via the speech synthesis software. These users would need to listen for errors in the text and watch the onscreen highlighting to see where the incorrect words are on the page.
- After the illiterate users select the incorrect text, the device would need to allow them to re-dictate the words, show them a list of replacement text that sounds like the misrecognized portion, or simply record the user's voice as they say the words.
- If recorded portions of human voice are to be used, then the application and web designers will need to structure their forms so that text input boxes can also accept recorded sound input.

Displaying Information to Users

After the user has requested a service from the device and filled out the required forms to submit their information, the system will process the request and display the results to the user. Sometimes this presentation of results consists of a simple message or a block of information on a particular topic; in this case, its presentation could follow the recommendations of the 'detail view' section below. If the result of the device's processing is a list of options or relevant items to the user, then the 'index view' section recommends how such information should be displayed. This portion of the recommendations also discusses how images can be used to facilitate communication to illiterate users, and how continuous and event-based information should be displayed.

Index View: Presenting Lists

- When search results, options, information topics, or other lists of items are presented to the user, such lists should be presented in a table format.
- Tables can aid the understanding of onscreen information by partially literate users because paragraphs of textual data can be converted into a short-hand representation where numbers or brief pieces of text located in table cells indicate information.
- If the device can provide an audio description of what each column of the table indicates, then the user may be able to take advantage of the onscreen data even with limited literacy skills. The use of numbers or picture icons in cells of the table can make them even more accessible.
- Since illiterate users will not be able to quickly scan a list of choices visually, the audio interface will need to help them make an informed selection.
- The interface will need to read through each of the options (the rows of the table) and give a concise description of the item. Since conveying information by audio output can be slow, the audio interface may need to provide only a subset of the onscreen information.
- Because of this slowness, the quality of the algorithm that initially sorts the results given to the user is important. Sorting the results initially in manner in which the most desirable choices appear at the beginning would make the interaction more efficient.
- If a user wants to hear all of the choices before making their decision, they should not be expected to remember all of the items. Therefore, after each item is read aloud, the system should allow users to rate how desirable that option is. If the user is listening to the description of an item and knows how they wish to rate it in the middle of the audio output, he or she should be able to interrupt the audio presentation.

- At the end of the list, the system can resort the list of options according to the user's ratings and filter out the items with low scores. At this time, the user can again listen to the options and make their final selection.

Detail View: Navigating Multiple Pages

- If a user has selected an item from a list described above or navigates to a page in the interface containing information that extends beyond one screenful, the system should present a consistent interface for managing multiple pages of information on a single topic.
- Such informational pages would contain text that could be read aloud by speech synthesis software. This text could be interspersed with icons, images, animations, maps, charts, diagrams, and other visual communication media discussed in the next section.
- When the page describes a sequenced set of instructions or directions, the interface should present an icon or image associated with each step of the process to make the information more memorable. If an illiterate user chooses to print such a page of information, then the images can be used as a mnemonic for remembering the information, even if the user cannot read the text.
- When information extends beyond one screenful, then the system should use a consistent set of navigation icons for allowing the user to go forward one page, back one page, back to the beginning of the text, or to return to a main menu of choices. Simple arrow and square inspired icons like those on VCRs or tape decks could be appropriate for this purpose.
- If the user is reading information that is in the form of hypertext in which the user is presented with links that create decision points in the direction of the content to be displayed, then these options should also be presented in a consistent manner throughout the interface. Hypertext links embedded in the middle of text may be confusing for illiterate users and difficult for an audio interface to present. Placing links at the end of a screenful of text would be less confusing.

Images and Icons

- The use of icons should not be considered an easy solve-all for the problem of illiteracy in a user interface. Finding an effective icon to communicate a concept can require great care and effort.
- An effective icon should tap into the culture and common experience of the user base, and it should make reference to concepts in the users' lives, not to internal concepts in the designers interface. Icons should be memorable, nameable, and concrete so that users can discuss them with each other.
- Animation can be used to add another dimension of meaning to images in the interface. Animated icons can be an effective way to communicate concepts involving time, processes, stories, and cause/effect relationships.
- It will not always be possible to select meaningful icons or images that express ideas contained within text. For this reason an icon should never be used on its own to express meaning, a form of visual/audio text captioning should always be available.
- While it may be reasonable to expect users to learn a small set of icons for page navigation or other basic system commands; in general, designers should not expect users to memorize a large vocabulary of icon images.
- While maps, charts, graphs, diagrams, photographs, and tables can also communicate information in a non-linguistic fashion, they still require careful audio captioning in order to be understandable by illiterate users.

Continuous System Status

- When the interface needs to communicate the continuous aspects of the device's status, it is most reasonable to communicate this information in a visual fashion. Constant audio updates would be a distracting way to convey whether the system is still live, if the volume level is sufficient, or if there is too much background noise.
- When the system is performing processing or is waiting for input, the user may experience a period of silence. When this silence lasts for more than a few

seconds, then the system should communicate to the user the state of the device and whether it is waiting for input from the user.

Event Notifications

- When users receive an incoming message, an important announcement or alert, non-immediate results or feedback from a previously initiated operation, or an error message, they must be notified of such an event.
- For simple short messages that contain information that the user will likely not need to refer to again, an audio announcement would be a sufficient way to convey a notification. Audio messages would be less distracting and invasive during the user's operation of the device.
- While some interfaces make use of tones to communicate information announcements in a concise manner, such output requires the user to memorize a vocabulary of non-linguistic sounds. For novice users, audio announcements in the form of short sentences could be more understandable. (For example, it would be more understandable to say, "you have e-mail," instead of making a ringing bell noise.)
- If a message is particularly complex, urgent, or requires an immediate response from the user, then a visual notification would be preferable. A text box could appear on the screen containing the message and the speech synthesizer could read it aloud.
- If a response to the message is required, then the input techniques discussed in the 'users entering information' section above could be used to solicit it.
- If the message is urgent, then the system could use flashing visual effects or special loud tones/volume to express the message. The user should not have to learn a convention for the communication of urgent information, whatever way in which the information is highlighted visually or aurally; it should attract the user's attention naturally.

Assistance to Users

This section will discuss how the interface can provide special help to users with literacy challenges or who are unfamiliar with operating computers.

Help with Reading

- The most important characteristic of the illiterately accessible interface is that it should not require users to read text in order to fully operate the device. When a user encounters a problem using the interface because of a reading difficulty, it should read text to the user or provide alternative methods of input and output.
- By always including text on the screen, the device can encourage literacy by providing an advantage to users who have partial literacy skills. It can motivate learning by linking literacy to the immediate goal of operating the device more quickly -- the ultimate goal being that the user can accomplish important tasks more efficiently.
- When including speech synthesis and audio captioning in the interface, designers should remember that a lack of literacy skills can affect more than just the reading text. The user's familiarity with numbers, charts, maps, and some vocabulary may also be reduced.
- The interface should make it very easy for the user to request that the entire contents of the screen be read aloud; there should be a simple 'read button' on the device to initiate this action. In most cases, the device should read its screen contents to the user automatically without being asked.
- The interface should also include the capability for the user to request that a specific screen element, piece of text, or item be read aloud. Simply clicking on the item to request it be read would be an undesirable because clicking on objects should be used to indicate that an action should be performed or a selection is made.
- There would need to be a special form of clicking used to indicate that an item should be read - perhaps double-clicking or click-and-hold. Another approach would be for the interface to require users to hold down the 'read button'

(mentioned above) and simultaneously click on an item to request that it be read aloud.

- In order for users to understand what portion of the screen is currently being read, the interface should highlight the currently read item in some consistent fashion.
- The device would need to highlight screen elements both to indicate urgency/importance and to indicate what item on the screen is currently being read. It should perform this highlighting in a distinct and consistent manner for each purpose. For example, the device could reserve flashing and reverse video to indicate urgency, and it could use underlining or a moving arrow to indicate the item currently being read aloud.
- Users should easily be able to interrupt a reading assistance feature if they are partially literate or remember how a portion of the interface operates without having it read aloud. The device should remember when a user has requested to bypass a reading assistance feature but make it easy for the user to request reading help if needed again.

Educating Users

- The system should advertise its features to users by advertising a feature-of-the-day directly on the portal screen and by logically arranging commands within portal screen categories so that users can anticipate where to find them. Newly available services and features can also be advertised on the portal screen.
- The device should have a brief built-in tutorial for users that informs them how to operate the interface and tells them about the basic features of the system. It can also educate users about interface metaphors, terminology, and the reasonable expectations of what the device can accomplish.
- At other locations in the interface, the user should be able to request a more detailed "tell me what I can do from here" tutorial from the device. For example, the device should be able to describe its health capabilities if the user asks for a tour inside a "Health" category.

- If the device is about to lead a user through the steps of a complicated operation or form, then users should be able to request directions, explanation, and a brief tour of what they can expect before the session begins.
- The device should encourage users to explore the interface to become familiar with the system's capabilities. The interface should therefore be visually engaging and attractive, and it should always provide reassuring and encouraging feedback to novice users to overcome their technophobia.
- By having a powerful 'undo' capability and by always asking for confirmation before fatal operations, the device can make users feel less nervous about exploring portions of the interface.
- While users should be able to learn how to use the device on their own, it is also possible that friends and neighbors of the user could also provide tutorial assistance and help when the user encounters a problem. In this way, capabilities of the device and tips for navigating the interface may spread by word of mouth between users of the system. To facilitate this important form of learning, screen elements and navigation directions on the interface should be memorable and easy to refer to by users when away from the device.

Resources and Capabilities

This section will discuss how specific features of the device should be implemented in the interface. It also suggests technologies and resources that would enable additional capabilities of the device.

Personalization and Identification

- Users of the device will each own a personal SIM card that stores their name, address, telephone number, email address, and a voice recording of each of these pieces of information. Other personal information like Internet bookmarks and an address book may also be stored on the card. The user’s level of literacy skills and their desired amount of speech support and menu complexity could also be recorded on the card.

- All of the information on the SIM card should also be stored in a central online location so that a fresh SIM card can be created for a user whose card was lost or stolen.
- Users should be able to change their personal information on the SIM card and online.
- The user's password should not be stored on the SIM card so that if it is stolen, another individual cannot gain access to a user's personal information or impersonate them online.
- The SIM card should also store a unique identifier for each user to make it easier for them to access personalized services on the system and to be contacted by other users. This identifier should be more than a secret electronic signal, it should be representable and communicated by users when not online, e.g. an e-mail address or username that can be told to others.
- The unique identifier for each user should allow them to universally log into the services on the network without remembering additional usernames and passwords. Perhaps the Simputer system could partner with other universal login services on the Internet (like the Microsoft Passport system) to extend this capability to websites beyond those specially designed for the Simputer.

Email and Bulletin Boards

- E-mail and bulletin board applications will be integral to the success of the Simputer system because they facilitate communication between users. E-mail can act as a primary point of contact for specific individuals who may not have a permanent telephone number or postal address. Bulletin boards allow communication to a large number of individuals that are not specifically known by the author of the message.
- E-mails and bulletin board applications can share a common interface for composing new messages and reading messages in an online folder or in the user's inbox.
- The text and voice recording of a user's name from the SIM card can be added to the header of a message automatically. The original date of the message can also

be added by the system. The user could use the free text composition methods discussed in the 'user entering information' section above to compose the subject line and body of the message.

- If the e-mail or bulletin board posting is being created as part of the user performing a standard operation, the system could fill in the subject line for the user or use a template to create the body of the e-mail automatically. Since entering free text can be challenging for illiterate users, composing portions of e-mail or bulletin board postings automatically would be desirable.
- Since managing messages in online bulletin boards or cleaning out old e-mail messages can be confusing for novice users, the system should help users perform these operations. Users should be encouraged to set an expiration date for messages posted to online folders, and there should be a facility for guiding users through their old e-mail messages and asking them if they wish to delete them.
- The index and detail views discussed in the 'displaying information to users' section above can be used to display e-mail and bulletin board messages to users. The index view can display all messages in a folder, and the detail view can be used to display each screenful of a message to the user.
- When displaying a single e-mail message in the detail view, the system can display navigation icons to allow the user to jump to the subsequent or previous e-mail message or return to the folder view. Common commands for replying, forwarding, and deleting the message can also be available on the page as hyperlinks or command buttons. The audio interface can read these options to the user when it has finished reading the message aloud.

Libraries of Resources for Developers

- The various applications that the Simputer will need to enable can be grouped according to common themes (health issues, agricultural work) and information resource requirements (maps, thesauri). Such resources could be developed as libraries of resources that application and web developers could take advantage of during their work.

- Catalogues of images, terminology, and reference materials relating to health and agricultural applications could be useful to the developers of a wide variety of applications for the Simputer. If these resources could be developed and made available online, then the production of content and services for this new platform would be encouraged.
- Many user interface elements could take advantage of online maps of India labeled textually and phonetically in multiple languages. Likewise semantic and sounds-alike thesauri for Indian languages could facilitate the display and input of text on the device. Online pronunciation dictionaries of proper nouns could also increase the speech synthesis accuracy of the device.
- Another advantage of having a central library of icons, help facilities, navigations tools, and other interface elements is that it will foster consistency across applications of the device. This consistency will make the device easier to learn by novice users.

Storing Information for Later Use

- If a user has found information using the Simputer and wishes to use it at a later time, he or she will need a way to take this information with them after leaving the device. While literate users can simply take notes from what they read on the screen, illiterate users would have no way to record the information.
- Users should be able to print information, store bookmarks of pages of interest, take voice or speech-dictation notes, and add a person's contact information to their personal address book.
- If a printing device were connected to the Simputer, then a hard copy of screen information could be produced, but illiterate users would be unable to read text on the page. Users would have to bring the page to a literate acquaintance in order to have it read.
- To be useful users with minimal literacy skills, the printout would need to display information in the form of images, diagrams, maps, and tables. (The advantages that tables provide to partially literate users are discussed in the 'index view' section above.)

- If a user is printing out contact information for another person, then the print out should use icons next to the various forms of personal data to indicate their meaning. For example, an envelope image could appear next to a mailing address, a telephone next to the digits of the telephone number, and a Simputer/computer next to an e-mail address.
- If the user is printing out a set of instructions from the system, then the onscreen icons associated with each bullet point of the instructions could act as mnemonics to aid their memory away from the device.

Offline Mode

- Many of the most exciting applications of the Simputer require the device to be connected to a computer network. For this reason, the device may attempt to connect to a network as soon as it is powered on.
- If the device will enable some tasks such as composing text, filling out long forms, or reading e-mail messages to occur offline, then the interface will need to indicate the connectivity status of the device. As this is a continuous form of status, it would be desirable to communicate whether the device is online in a visual manner.
- If the user requests a service that requires a network connection, then the interface will need to offer to make a new connection and display the status of the device while it attempts to go online.

Chapter 8: Experimentation and Future Directions

The preceding chapters of this thesis have discussed and analyzed the research work performed to date while producing design recommendations for an illiterately accessible user-interface. This final chapter will describe how the research of the URSULA project can continue beyond this thesis and, in particular, how it can employ experimental approaches to advance the interface design. This chapter will discuss how areas of future research identified during the HUDS scenarios and design issues not yet anticipated can be explored by following a cyclic design methodology suggested by the user-interface design literature. It will also propose how the URSULA project can provide designers with more than just recommendations by producing a user-interface development toolkit for illiterately accessible application developers.

Reasons for Experimentation

While this thesis has demonstrated how non-experimental approaches can be used to drive a user-interface design process, experimental techniques can oftentimes provide additional insight into a project. Experimental studies can help identify problems with a system that designers may have failed to anticipate during scenario-based methodologies, or particular components of an interface will sometimes be too unusual for a designer to confidently approach without information from specially designed experimentation. User trials can also help designers understand their target population: what they expect from the system, how they will want to use it, and how it will be adopted into their life and work [Preece 1995].

Even when the results of experimental studies merely confirm the design recommendations made through analytical or scenario-based approaches, they contribute to the design process by validating the results of the non-experimental methods. Many application and web developers find recommendations based on empirical findings more compelling than those derived from other sources. While it is true that experimental results can help to ground design decisions in actual data, it is important to remember that user-interface experiments are difficult to design and can often have many uncontrolled variables. Despite the potential problems with this data, developers can be tempted to draw conclusions that are too strong from experimental design methods because of their empirical basis.

Planning for Future Development Work

As the URSULA project grows beyond this initial thesis research, it will develop into a team within the Adaptive Speech Research group studying issues of speech-based user-interface design. For the reasons discussed above, this team will need to move beyond the analysis and scenario-based approaches of this thesis and begin to employ user-interface experimental testing. The following section will propose a four-stage plan for this future line of research.

The user-interface design literature contains many models of how a design process can be organized [Eason 1991] [Avison 1990] [Greenbaum 1991], but there is disagreement of how to order the stages of this process. Some authors even suggest that strictly ordering the elements of a design process is inappropriate [Hartson 1989] [Hix 1993]. Since such loosely ordered design models are difficult to use for planning purposes, a more structured method was chosen for the future stages of the URSULA design process - User-Centered Design [Eason 1991]. In this model, stages of planning, designing, building, and testing are iterated to slowly improve a system; the development work outlined below (particularly stages 2 to 4) follows this four-part cycle.

Stage 1: Initial Experiments and Design Specification

Each of the HUDS in chapters 5 and 6 includes a section entitled 'Future Areas of Research,' which consists of general user-interface questions that could be settled through experimental studies. Some examples of issues that arose during these scenarios include how to visually indicate the current piece of text being read by a screen reader, whether novice users were confused by pop-up notifications, and whether illiterate users understand the concept of sorting lists. Many of these issues would have broad repercussions on the design of the interface, and most could also be tested experimentally without producing a complete mock-up of the system. Such issues would therefore be appropriate to test by conducting experiments prior to creating an initial prototype version of the interface. In particular, issues that would affect the design direction taken at critical decision points in the system's development should be explored in this stage.

Because the experiments during this initial stage of the project will address broad issues that could potentially change the shape of the system interface dramatically, rapid prototyping will be used to produce the experimental artifacts. In rapid prototyping, a partial mock-up of the system is created that will be used for a one-off experiment and then not used for any further development [Preece 1995]. Because each mock-up will only be used for a limited time and may not necessarily resemble the final user-interface, only a small amount of development work should be used to produce each. Instead of programming these initial interfaces in traditional development languages, the use of prototyping tools should be considered which allow the quick production of interactive demonstrations such as Macromedia Flash or the speech-based SUEDE system [Sinha 2000].

Because of the difficulty and expense in transporting experimental set-ups to developing countries and in arranging for local members of the community to participate in experiments, another simplification will be made to the experiments in stage 1 and 2. Instead of illiterate users from developing countries, subjects available locally to the project will be used in experiments. To simulate the written language illiteracy of the target user population, nonsense words could be used on the visual displays of literate subjects so that these users cannot read the screen. The audio interface would still read the actual words that should appear so that literate users will be dependent on the audio screen reader just as illiterate users would be. To simulate degrees of partial literacy, nonsense words can be used to replace subsets of the words on the visual display.

At the end of this stage, an initial draft of a design specification for the interface should be created which takes into account the findings of these experiments. While many of the recommendations made in the previous chapter already suggest the shape of this specification, a formal statement of the design should be produced at this time. This document will be needed to guide the development of the initial user-interface prototype for the stage 2 of the project.

Stage 2: Building, Laboratory Testing, Re-Design of Prototype

In stage 2, a prototype version of the system's interface will be produced, tested in a laboratory setting, and re-designed in an iterative design process. At the beginning of the stage, a mock-up of the interface will be created according to the design specification draft written at the end of stage 1. This initial artifact will be a low-fidelity, vertical prototype -- that is, a version of the interface which initially bears little resemblance to the final design and only implements a subset of its features [Preece 1995]. As user trials are conducted and the interface specification is refined according to the results of these trials, the mock-up will slowly evolve into a high fidelity, full prototype of the illiterately accessible interface.

While the experimental mock-ups used in stage 1 did not need to resemble the actual interface in order to test HUDS-inspired design questions, the mock-up of the interface in stage 2 will be a prototype of the actual system and should always follow the current draft of the design specification. While it should be similar in appearance and interactivity to the actual design, this mock-up does not need to be a fully functional piece of software. For instance, users should be able to submit information, but the demo applications in this mock-up could merely return an arbitrary piece of output. Only the interface is being tested, not the performance of the applications; so, it could still be possible to use prototyping tools like Macromedia Flash or SUEDE to produce this mock-up. The only requirement for this prototype is that it be able to simulate the working interactive dialogue of the system and allow users to manipulate elements of the interface, not just view them [Hammond 1984].

During laboratory trials, subjects would interact with the prototype, and the designer would take note of any difficulties they have using the system to perform certain tasks. The prototype software could assist the designer by electronically logging certain user behaviors indicative of errors: requesting help screens, invalid inputs entered, lengthy time needed to complete a task, or instances where the user backtracks when getting lost in the interface [Hammond 1984]. The experimenter could also ask users about their impressions of the system by engaging them in informal discussions, conducting interviews, facilitating focus groups, or giving a standardized usability questionnaire like the SUMI [Porteous 1993].

Uncovering design problems through prototype experimentation is never an automatic process. Even when automatically logged numerical results indicate the existence a design flaw, they will rarely be able to indicate the nature of the problem. In order to identify significant problems in the system and ways to improve the design, the experimenter will need to notice trends and patterns in the issues discussed by the experimental subjects. The time required to conduct user trials, to collect user's comments, and to interpret the results can make experimental design methods very costly; for this reason, non-experimental design approaches were used during this thesis to advance the design as far as possible before this stage.

As new design issues are identified, the recommendations for developers can be modified and the current draft of the design specification updated. The laboratory prototype should be redesigned to reflect the changes to the specification. This testing and redesign process can be repeated as long as designers feel that the design improvements outweigh the costs of experimentation.

Stage 3: Field-Testing of Prototype System

When designers are satisfied with the recommendations refined through the cycles of laboratory testing, a prototype of the system can be tested on actual target users of the system. During this field-testing, illiterate users could be observed as they operate the prototype in their natural environment; they could be asked to accomplish particular tasks or to speak aloud their thoughts as they use the system [Preece 1995]. Similar observation, interview, and survey approaches as used in the stage 2 laboratory tests could be used in this stage to uncover critical issues with the design. While it could be extremely costly, this stage could also be iterated to make improvements to the design. If the designers had specified a set of usability metrics to determine the success of the final version of the system, these benchmark experiments could also be performed at this time.

Stage 4: Toolkit Production and Distribution

After analyzing the results of the laboratory and field-testing, a final version of the design specification should be written. The specified interface may consist of several standard components that could be produced and stored in a software toolkit/library for application developers. The URSULA research group could use standard software engineering practices to implement these components using a design cycle of programming, testing, and debugging. Testing in this stage would not be usability testing (as in stages 2 and 3) but rather traditional software engineering bug-testing to determine if the software behaves according to its specification.

This toolkit should also include extensive documentation on its use, including principles and guidelines for application developers to advise them how to construct interfaces accessible to illiterate users. Once the recommendations made in the previous chapter have been updated during stages 2 and 3, they can be used as a foundation for this documentation. This toolkit and documentation should be advertised and made available to developers of interfaces for illiterate users, such as those designing for the Simputer platform.

Final Comments

Devices like the Simputer that aim to bridge the digital divide in developing countries have the potential to substantially improve the quality of life of their users.

As information technology proliferates to these areas, the URSULA project has an important contribution to make to this process. By understanding the special needs of illiterate users and producing strategies to help them overcome these challenges, URSULA can ensure that all members of developing communities can take advantage of this new technology.

While the scope of this thesis included only the first part of this design process, the recommendations it has produced already suggest directions for application developers and Simputer system designers to follow. By continuing this line of research to conduct further experimental testing, design specification, and toolkit production, the URSULA project can not only enhance our understanding of illiterately accessible interfaces but also contribute to the Simputer effort more concretely.

Bibliography: Works Cited

[AIGA 1981] American Institute of Graphic Arts. 1981. "Symbols Signs: The System of Passenger/Pedestrian Oriented Symbols Developed for the U.S. Department of Transportation." In *Readings in Human-Computer Interaction: A Multidisciplinary Approach* (R. M. Baecker and W. A. S. Buxton, eds.). Los Altos, California: Morgan Kaufmann Publishers.

[Amory 1994] A. Amory. 1994. University of Natal, pers. comm. In: K. A. Murrell. 1998. "Human Computer Interface Design in a Multi-cultural Multi-lingual Environment." 13th Annual MSc and PhD Conference in Computer Science, University of Stellenbosch, 1 July 1998.

[Andrews 1994] S. J. Andrews. 1994. Some Cultural and Perceptual Implications of Courseware Development and the Use of Technology within a Multi-cultural, Multilingual Society (a cautionary tale). In: *Computer-Assisted Education and Training in Developing Countries.* (P. M. Alexander, ed.) Pages 7 - 13, University of South Africa, Muckleneuk, Pretoria. In: K. A. Murrell. 1998. "Human Computer Interface Design in a multi-cultural multi-lingual environment." 13th Annual MSc and PhD Conference in Computer Science, University of Stellenbosch, 1 July 1998.

[Aral 2001] S. Aral, M Escobari, & R. Nishina. 2001. "Sustainable Access in Rural India (SARI) Project. Pilot Phase Assessment." Kennedy School of Government, Harvard University.

[Avison 1990] D. Avison and T. Wood-Harper. 1990. *Multiview Methodology.* Oxford: Blackwell Scientific Publishers.

[Baecker 1987] R. M. Baecker and W. A. S. Buxton. 1987. *Readings in Human-Computer Interaction: A Multidisciplinary Approach.* Los Altos, California: Morgan Kaufmann Publishers.

[Baldwin 1997] T. Baldwin. 1997. Issues in Designing Natural Language Interfaces. Department of Computer Science. University of Sheffield. http://www.dcs.shef.ac.uk/~stu/HCI/Topic1/

[Barron 1998] A. Barron. 1998. "Designing Web-Based Instruction" ITForum Discussion paper, May 1998. In: K. A. Murrell. 1998. "Human Computer Interface Design in a Multi-cultural Multi-lingual Environment." 13th Annual MSc and PhD Conference in Computer Science, University of Stellenbosch, 1 July 1998.

[Baynham 1995] M. Baynham. 1995. *Literacy Practices: Investigating Literacy in Social Contexts.* London: Longman Group, Ltd.

[Benner 1993] K. Benner, M. Feather, W. Johnson, & L. Zorman. 1993. "Utilizing Scenarios in the Software Development Process." In *Information System Development Process* (N. Prakash, C. Rolland, and B. Pernici, eds.), IFIP Transactions A-20, Elsevier Science Publishers, September.

[Carroll 1995] J. M. Carroll (ed.). 1995. *Scenario-Based Design: Envisioning Work and Technology in System Development.* New York: Wiley.

[Chlebowska 1990] K. Chlebowska. 1990. *Literacy for Rural Women in the Third World.* Paris: United Nations Educational Scientific and Cultural Organization.

[Coats 1987] R. B. Coats and I. Vlaeminke. 1987. *Man-Computer Interfaces.* Oxford: Blackwell Scientific Publications. Chapters 4 and 5. Pages 74 to 144.

[Cole 1995] R. Cole, L. Hirschman, L. Atlas, et al. 1995. "The challenge of spoken language systems: research directions for the nineties," IEEE Transactions on Speech and Audio Processing, Vol. 3, pp. 1-21, 1063-6676, January.

[Colle 2000] R. D. Colle, R. Roman, & F. Yang. 2000. "Access is More than Hardware: Building a Constituency for Telecenters." INET2000, Yokohama, July 18-21.

[Colle 2001a] R.D. Colle, R. Roman, & A. Pfiester. 2001. "Modules for Telecenter Management Training." Interim report to International Telecommunications Union, Geneva, Switzerland.

[Colle 2001b] R.D. Colle and R. Roman. 2001. "Sustainability and Training in the Telecenter Movement." Pacific Telecommunications Conference, Honolulu, Hawaii, January 14-18.

[Davis 1987] J. R. Davis. 1987. "A Voice Interface to a Direction Giving Program." Speech Research Group Technical Memo 2, MIT Media Laboratory, January 1987.

[Deatherage 1972] B.H. Deatherage. 1972. Auditory and other forms of information presentation. In: *Human Engineering Guide to Equipment Design.* (H. P. Van Cott & R. G. Kinkade, eds.) Pages 123-160. Washington D.C.: U.S. Government Printing Office.

[Dey 1997] A. K. Dey, L. D. Catledge, G. D. Abowd, and C. Potts. 1997. Developing voice-only applications in the absence of speech recognition technology. Technical Report GIT-GVU-97-06, GVU Center, Georgia Institute of Technology, March. Submitted to DIS'97. http://citeseer.nj.nec.com/dey97developing.html

[Donovan 1996] R. E. Donovan. 1996. Trainable Speech Synthesis. PhD thesis, Cambridge Univ. Eng. Dept., June 1996. http://citeseer.nj.nec.com/donovan96trainable.html

[Downton 1993] A. Downton. 1993. "Dialogue Styles: Basic techniques and guidelines." In *Engineering the Human-Computer Interface: Student Edition.* (Andy Downton, ed.) London: McGraw Hill Book Company. Pages 31-84.

[Eason 1991] K. D. Eason. 1991. "Ergonomic Perspectives on Advances in Human-Computer Interaction." Ergonomics, 34, 6, 721-41.

[Encore 2002] Encore Software Limited. 2002. "Encore's Simputer." http://www.ncoretech.com/simputer/ (March 13, 2002.)

[Gaver 1986] W. Gaver. 1986. "Auditory Icons: Using sound in computer interfaces." Human Computer Interaction, 2(2), pp167-177.

[Goetze 2001] M. Goetze and T. Strothotte. 2001. "An Approach to Help Functionally Illiterate People with Graphical Reading Aids." Smart Graphics Symposium UK. March 21-23, Hawthorne, NY, USA.

[Gould 1985] J. D. Gould and C. Lewis. 1985. “Designing for usability: Key principles and what designers think.” Communications of the ACM, 28, 3, 300-311.

[Grasso 1996] M. A. Grasso. 1996. "Speech Input in Multimodal Environments: Effects of Perceptual Structure on Speed, Accuracy, and Acceptance." University of Maryland Baltimore County, Doctoral Dissertation Proposal. (TR CS-96-09)

[Grasso 1997] M. Grasso, D. Ebert, and T. Finin. 1997. "Acceptance of a Speech Interface for Biomedical Data Collection." AMIA Annual Fall Symposium (SCAMC). Pages 739-743.

[Greenbaum 1991] J. Greenbaum and M. Kyng, eds. 1991. *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

[Hammond 1984] N. Hammond and P. Barnard. 1984. “Dialogue Design: Characteristics of User Knowledge.” Chapter in *Fundamentals of Human-Computer Interaction* (A. Monk, ed.). London: Academic Press. Pages 127-164.

[Hartson 1989] H. R. Hartson and D. Hix. 1989. “Toward Empirically Derived Methodologies and Tools for HCI Development.” International Journal of Man-Machine Studies, 31, Pages 477-494.

[Hix 1993] D. Hix and H. R. Hartson. 1993. *Developing User Interfaces: Ensuring Usability Through Product and Process.* New York: John Wiley.

[Huenerfauth 2002] M. Huenerfauth. 2002. “Design Approaches for Developing User-Interfaces Accessible to Illiterate Users.” AAAI 2002 Workshop on Intelligent Situation-Aware Media and Presentations, Edmonton, Alberta, Canada, 29 July 2002.

[IDRC 1997] IDRC Study/Acacia Initiative. 1997. "Exploring a Basic Illiterate Web Access System." Prepared by m-powa, Midrand, South Africa. February. http://www.idrc.ca/acacia/outputs/op-audi.htm

[India 2001] Office of the Registrar General, India. 2001. *Provisional Population Totals: India. Census of India 2001, Paper 1 of 2001.*

[Johnson 1992] P. Johnson. 1992. *Human Computer Interaction: Psychology, Task Analysis, and Software Engineering.* Berkshire: McGraw-Hill Europe.

[Law 2000] C. Law and G. Vanderheiden. 2000. "The Development of a Simple, Low Cost Set of Universal Access Features for Electronic Devices." CUU2000: ACM Conference on Universal Usability. Arlington, VA, November 16-17.

[Levow 1997] G.-A. Levow. 1997. "Making Sense of Silence in Speech User Interfaces." CHI '97 Workshop: Speech User Interface Design Challenges. Atlanta, GA. 22-27 March.

[Maberly 1966] N. C. Maberly. 1966. Mastering Speed Reading. New American Library, Inc. New York. In: Grasso M.A (1996). "Speech Input in Multimodal Environments: Effects of Perceptual Structure on Speed, Accuracy, and Acceptance." University of Maryland Baltimore County, Doctoral Dissertation Proposal. (TR CS-96-09)

[Marshall 1987] C. Marshall, C. Nelson, and M. M. Gardiner. 1987. "Design Guidelines." Chapter in *Applying Cognitive Psychology to User-interface Design.* (M. M. Gardiner and B. Christie, eds.) Chichester: John Wiley & Sons. 1987. Pages 221-278.

[Murrell 1998] K. A. Murrell. 1998. "Human Computer Interface Design in a multi-cultural multi-lingual environment." 13th Annual MSc and PhD Conference in Computer Science, University of Stellenbosch, 1 July 1998.

[Mynatt 1992] E. Mynatt and K. Edwards. 1992. "Mapping GUIs to Auditory Interfaces." UIST92: The Fifth Annual Symposium on User Interface Software and Technology Conference Proceedings, November.

[Mynatt 1994] E. Mynatt and G. Weber. 1994. "Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches," CHI94 Conference Proceedings, Boston, MA, April.

[Nardi 1992] B. A. Nardi. 1992. "The Use of Scenarios in Design." SIGCHI Bulletin, October.

[Norman 1983] D. A. Norman. 1983. "Design Principles for Human-Computer Interfaces." In *Readings in Human-Computer Interaction: A Multidisciplinary Approach* (R. M. Baecker and W. A. S. Buxton, eds.). Los Altos, California: Morgan Kaufmann Publishers.

[Olinsky 2002] C. Olinsky. 2002. "Dynamic Adaptation for Language and Dialect in a Speech Synthesis System." Project Proposal. Media Lab Europe and Department of Computer Science, University College Dublin. February 2002. http://www-2.cs.cmu.edu/~colinsky/main.htm

[Porteous 1993] M. A. Porteous, J. Kirakowski, and M. Corbett. 1993. *Software Usability Measurement Inventory Handbook.* Human Factors Research Group, University College, Cork, Ireland. In: J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. 1995. *Human-Computer Interaction.* Addison-Wesley Pub. Co.

[Preece 1995] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey. 1995. *Human-Computer Interaction.* Addison-Wesley Pub. Co.

[Redmond-Pyle 1995] D. Redmond-Pyle and A. Moore. 1995. *Graphical User Interface Design and Evaluation (GUIDE): A Practical Process.* London: Prentice Hall.

[Ribeiro 2002] J. Ribeiro. 2002. "Net Telephony Now Legal in India." IDG News Service, Bangalore Bureau. ITworld.com. 1 April 2002. http://www.itworld.com/Net/3303/020401india/

[Roselli 1998] D. Roselli. 1998. Lecture Slides for CS260: Research Topics in Human Computer Interaction, University of Calafornia, Berkeley. 27 February. http://bmrc.berkeley.edu/courseware/cs260/spring98/lectures/speech/

[Rudnicky 1996] A. I. Rudnicky. 1996. Speech Interface Guidelines. School of Computer Science, Carnegie Mellon University. http://www.speech.cs.cmu.edu/rspeech-1/air/papers/SpInGuidelines/SpInGuidelines/html

[Shriver 2000] S. Shriver, A. Black, and R. Rosenfeld. 2000. "Audio Signals in Speech Interfaces," ICSLP 2000. http://citeseer.nj.nec.com/shriver00audio.html

[Singh 1993] R. Singh. 1993. "Communication Technology for Rural Development." New Delhi: B.R. Publishing.

[Sinha 2000] A. K. Sinha, S. R. Klemmer, J. Chen, J. A. Landay, N. Aboobaker, and A. Wang. 2000. "SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces." In CHI Letters, The 13th Annual ACM Symposium on User Interface Softward and Technology: USIT 2000. 2(2): Pages 1-10.

[Stifelman 1993] L. J. Stifelman, B. Arons, C. Schmandt, and E. A. Hulteen. 1993. "VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker." INTERCHI, Amsterdam, April 24-29.

[Street 1984] B. Street. 1984. Literacy in Theory and Practice. Cambridge: Cambridge University Press.

[Simputer 2001] Simputer Trust. 2001. "Simputer Homepage." http://www.simputer.org/ (March 13, 2002.)

[Yankelovich 1995] N. Yankelovich, G.-A. Levow, & M. Marx. 1995. "Designing SpeechActs: Issues in Speech User Interfaces," SIGCHI95, Human Factors in Computing Systems, Denver, CO, May 7-11.

[Youll 2000] J. Youll. 2000. "World Dialog: Video/Voice Messaging for Developing Regions." Preliminary Research Proposal, MIT Media Lab. 4 February 2000. http://web.media.mit.edu/~jim/projects/world/world.pdf

Bibliography: Works Consulted

B. Arons. 1991. “Hyperspeech: Navigating in speech-only hypermedia.” Hypertext '91, pages 133--146. ACM, 1991.

B. Arons, and E. Mynatt. 1994. “The future of speech and audio in the interface.” SIGCHI Bulletin, 26, 4, 44-48.

S. Bayer, R. Kozierok, and J. Kurtz. 1995. “Multimodal Interfaces on the Web.” Third Python Workshop.

S. A. Brewster, P. C. Wright, and A. D. N. Edwards. 1993. “An evaluation of earcons for use in auditory human-computer interfaces.” INTERCHI'93, pages 222--227. ACM Press, Addison-Wesley.

S. A. Brewster. 1994. “Providing a structured method for integrating non-speech audio into human-computer interfaces.” DPhil Thesis, University of York.

W. Buxton, S. A. Bly, S. P. Frysinger, D. Lunney, D. L. Mansur, J. J. Mezrich, and R. C. Morrison. 1985. “Communicating with Sound.” In *Readings in Human-Computer Interaction: A Multidisciplinary Approach (R.M. Baecker and W.A.S. Buxton, eds.).* Los Altos, California: Morgan Kaufmann Publishers.

P. R. Cohen. 1992. “The Role of Natural Language in a Multimodal Interface.” ACM Symposium on User Interface Software and Technology, Monterey California, ACM Press, November 15-18.

P. Constantinides, S. Hansma, C. Tchou and A. Rudnicky. 1998. “A schema based approach to dialog control.” International Conference on Spoken Language Processing, pages 409 - 412, Sidney, Australia.

M. Cooper. 2000. “Disconnected, Disadvantaged, and Disenfranchised: Explorations in the Digital Divide.” Consumer Federation of America. Washington. 11 October. http://www.consumersuinion.org/resources/publications/htm

T. Dutoit. 1997. *Introduction to Text-to-Speech Synthesis.* Dordrecht, The Netherlands: Kluwer.

M. Eskenazi, A. Rudnicky, K. Gregory, P. Constantinides, R. Brennan, K. Bennett, J. Allen. 1999. "Data Collection and Processing in the Carnegie Mellon Communicator," Eurospeech-99, Budapest, Hungary.

S. Everett, K. Wauchope, and D. Perzanowski. 1992. "Talking to a Natural Language Interface: Lessons Learned," Navy Center for Applied Research in Artificial Intelligence Naval Research Laboratory. http:// www.aic.nrl.navy.mil/papers/1992/AIC-92-016.ps

E. Francik. 1996. "Computer- & screen-based interfaces: Universal design filter." Technical Report. Human Factors Engineering, Pacific Bell. Version 2. June 6.

J. C. Fromer. 1998. “Learning optimal discourse strategies in a spoken dialogue system.” Technical Report Forthcoming, MIT AI Lab M.S. Thesis

M. A. Grasso. 1997. “Speech Input in Multimodal Environments: Effects of Perceptual Structure on Speed, Accuracy, and Acceptance.” University of Maryland Baltimore County, Doctoral Dissertation.

M. A. Grasso, D.S. Ebert, T.W. Finin. 1998. “The Integrality of Speech in Multimodal Interfaces.” ACM Transactions on Computer-Human Interaction, 5(4):303-325.

D. Litman, S. Pan, and M. Walker. 1998. “Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent.” 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conf. on Computational Linguistics, 780-786.

A. Marcus. 1983. “Graphic Design for Computer Graphics.” In *Readings in Human-Computer Interaction: A Multidisciplinary Approach (R.M. Baecker and W.A.S. Buxton, eds.).* Los Altos, California: Morgan Kaufmann Publishers.

P. Martin and A. Kehler. 1994. "SpeechActs: A Testbed for Continuous Speech Applications," Submitted to ACL `94, Las Cruces, NM, June.

P. Martin, F. Crabbe, S. Adams, E. Baatz, and N. Yankelovich. 1996. “Speechacts: A spoken language framework.” IEEE Computer, 29(7):33-40.

J. Preece, D. Benyon, G. Davies, L. Keller and Y. Rogers. 1993. *A Guide to Usability: Human Factors in Computing.* Addison Wesley, Wokingham, U.K.

J. Ribeiro. 2000. "Simputer Aims at the Developing World." PCWorld. IDG News Service, Bangalore Bureau. Friday, 23 June 2000.

R. Rosenfeld, X. Zhu, A. Toth, S. Shriver, K. Lenzo, and A. Black. 2000. "Towards a Universal Speech Interface," ICSLP 2000, Beijing, China.

A. Rudnicky and W. Xu, 1999. "An agenda-based dialog management architecture for spoken language systems," Workshop on Automatic Speech Recognition and Understanding, Keystone, Colorado, Dec. 1999.

A. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, A. Oh. 1999. “Creating Natural Dialogs in the Carnegie Mellon Communicator System.” Eurospeech 1999, 4, 1531-1534.

A. Rudnicky, C. Bennett, A. Black, A. Chotimongkol, K. Lenzo, A. Oh, and R. Singh. 2000. “Task and Domain Specific Modeling in the Carnegie Mellon Communicator System.” ICSLP 2000, Beijing, China.

A. Shepard. 1989. “Analysis and Training in Information Tasks.” In *Task Analysis for Human-Computer Interaction* (Diaper D., ed.). Chichester: Ellis Horwood.

S. Shriver, A. Black, and R. Rosenfeld. 2000. "Audio Signals in Speech Interfaces," ICSLP 2000, Beijing, China.

C. A. Simpson, M. E. McCauley, E. F. Roland, J. C. Ruth, B. H. Williges. 1985. “System Design for Speech Recognition and Generation.” In *Readings in Human-Computer Interaction: A Multidisciplinary Approach (R.M. Baecker and W.A.S. Buxton, eds.).* Los Altos, California: Morgan Kaufmann Publishers.

L. J. Stifelman. 1995. "A Discourse Analysis Approach to Structured Speech." Presented at the AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation. March 27-29, Stanford University.

C. A. Thompson, R. J. Mooney, and L. R. Tang. 1997. “Learning to parse natural language database queries into logical form.” Workshop on Automata Induction, Grammatical Inference and Language Acquisition.

W. Verplank and S. Kim. 1986. “Graphic Invention for User Interfaces: An Experimental Course in User Interface Design.” *SIGCHI Bulletin,* 18(3), 50-67.

W. Xu and A. Rudnicky. 2000. “Language modeling for dialog system.” ICSLP 2000, Beijing, China.

W. Xu and A. Rudnicky. 2000. “Task-based dialog management using an agenda.” ANLP/NAACL Workshop on Conversational Systems, May, pp. 42-47.

N. Yankelovich and E. Baatz. 1994. "SpeechActs: A Framework for Building Speech Applications," AVIOS `94 Conference Proceedings, San Jose, CA, September 20-23.

N. Yankelovich. 1996. "How Do Users Know What to Say?" ACM Interactions, Volume 3, Number 6, November/December 1996. Sun Microsystems, Inc.

N. Yankelovich and C.D. Mclain. 1996. “Office Monitor.” CHI '96 Conference on Human Factors in Computing Systems, Vancouver, British Columbia, Canada, April 14-18.