

VOICE AUTHENTICATION USING VOICEXML

AZRUL BIN ABDUL WAHAB

**INFORMATION TECHNOLOGY
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2004**

VOICE AUTHENTICATION USING VOICEXML

by

AZRUL BIN ABDUL WAHAB

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Technology)

JUNE 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

t
QA
76.76
.A997
2004
1. VoicexML (Document
markup language)
2. Automatic speech
recognition
3. IT/IS -- Thesis

CERTIFICATION OF APPROVAL

**Voice Authentication through Speech Recognition
Using VoiceXML**

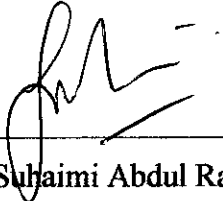
by

Azrul Bin Abdul Wahab

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

JUNE 2004

Approved by,



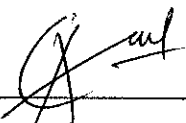
(Mr. Suhaimi Abdul Rahman)

**UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK**

June 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources of persons.



AZRUL BIN ABDUL WAHAB

ABBREVIATIONS

1. ASP = Active Server Pages
2. ASR = Automatic Speech Recognition
3. DTMF = Dual-Tone Multiple Frequency
4. HTML = Hypertext Markup Language
5. HTTP = Hypertext Transfer Protocol
6. IP = Internet Protocol
7. IVR = Interactive Voice Response
8. JSP = Java Server Pages
9. PSTN = Public Switched Telephone Network
10. SSL = Secure Sockets Layer
11. STT = Speech-to-Text
12. TTS = Text-to-Speech
13. VoIP = Voice over IP
14. W3C = World Wide Web Consortium
15. URI = Uniform Resource Identifier
16. XML = eXtensible Markup Language

ABSTRACT

User Authentication through voice is one of the methods to ensure the protection of the sensitive data over the Internet. In this research, author wants to explore the technology and acquire clear understanding of VoiceXML, its concept and its architecture. Provided with understanding of VoiceXML concept and architecture, this research will examines 3 levels of security using VoiceXML capabilities as a solution for validating users. The identified solutions will lead to development of VoiceXML prototype using available VoiceXML application development tool. This project has two main objectives. The first objective is to understand VoiceXML technology architecture and learn to develop and design a VoiceXML application. The second objective of the project is to observe three levels of security as a solution for validating users. This project involved two approaches which are performing research on VoiceXML technology and developing a prototype that conclude the findings on the research performed. In the development of the prototype, Voice Application Life Cycle methodology is used which include 4 phases; Planning, Prototyping and Iteration, Development, and Launch. As the result, the prototype is expected to take full advantages of current VoiceXML technology. The prototype can be used as a template for future use by developers in order to make their voice application achieve the goal of user authentication which is to make the right information reliably and securely available to the right people. The final product for the project is a prototype that has been called Voice Authentication through Speech Recognition that focuses on different level of security using voice authentication with VoiceXML.

ACKNOWLEDGEMENT

Completing a project on this title is a very intensive process and it takes the support and dedication of many people to make it possible. Above all, I would like to express my gratitude to the Lord Almighty for giving me the strength, wisdom and patient to complete this project on time.

My deepest gratitude goes to Mr. Suhaimi Abdul Rahman, Final Year Project (Information Technology) Supervisor for all the guidance, advises, incentives, and of course his patience which have given me the motivation and strength towards finishing this project.

Special credits go to Mr. Pavel Cenek (OptimTalk Principal), an independent developer from the Czech Republic, whom had provided me with the useful open source tool known as OptimTalk and also the advices through the e-mail exchanges during the development process. Your collaboration, support and precious times are very much appreciated.

I must give thanks to my beloved parents, En. Haji Abdul Wahab bin Haji Hasan and Pn. Hajjah Samseah bt Wan Ahmad Seraji, whom had given me the inspiration to carry the project through to the end. Thank you both for your love and encouragement.

Lastly, this project would not be complete without acknowledging the contributions of my friends, for being helpful and committed throughout the project development. I could not mention anyone's name without slighting a dozen others, so I would just like to thank each and every one of my friend.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	i
CERTIFICATION OF ORIGINALITY	ii
ABBREVIATIONS	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi

CHAPTER 1: INTRODUCTION

1.1	Background of Study	2
1.2	Problem Statement	4
	1.2.1 Problem Identification	4
	1.2.2 Significant of the Project	4
1.3	Objectives and Scope of Study	5
	1.3.1 The Relevancy of the Project	5
	1.3.2 Objectives	5
	1.3.3 Scope of Study	6
	1.3.4 Feasibility of the Project	6

CHAPTER 2: LITERATURE REVIEW

2.1	Introduction	7
2.2	VoiceXML	9
2.3	VoiceXML Architecture	11
2.4	VoiceXML Development Tools	12
2.5	Speech Recognition	14
2.6	User Authentication	15
	2.6.1 Security Issues	15
	2.6.2 Voice Authentication	15
	2.6.3 VoiceXML is the Answer	16

CHAPTER 3:	METHODOLOGY/ PROJECT WORK	
3.1	Project Work Procedure . . .	18
3.2	Research Process . . .	18
3.3	Voice Application Life Cycles . . .	18
3.3.1	Planning . . .	19
3.3.2	Prototyping and Iteration . . .	19
3.3.3	Development . . .	20
3.3.4	Launch and Maintenance . . .	20
3.4	Tools/equipment required . . .	20
3.4.1	Software Requirements . . .	21
3.4.2	Hardware Requirements . . .	21
CHAPTER 4:	RESULT AND DISCUSSION	
4.1	Results and Findings . . .	22
4.1.1	VoiceXML Concept and Architecture	22
4.1.2	User Authentication . . .	25
4.1.3	Three Levels of Authentication using VoiceXML . . .	25
4.2	Product Results and Discussion . . .	28
CHAPTER 5:	CONCLUSIONS	
5.1	Relevancy to the Objectives . . .	33
5.2	Recommendation . . .	34
REFERENCES	35

APPENDICES

APPENDIX A:	PROJECT TIMELINE / GANTT CHART
APPENDIX B:	RESEARCH PROCESS DIAGRAM
APPENDIX C:	VOICEXML ARCHITECTURE DIAGRAM
APPENDIX D:	PROGRAM FLOW CHART

LIST OF FIGURES

Figure 1.1:	Voice Application Life Cycles
Figure 2.1:	VoiceXML basic process flow
Figure 3.1:	Username Prompting
Figure 3.2:	Password Prompting
Figure 3.3:	Authorized Login
Figure 3.4:	Unauthorized Login
Figure 4.1:	VoiceXML Simulator 1.0
Figure 4.2:	VoiceXML Simulator 1.0.1

LIST OF TABLES

Table 1.1:	Voice Authentication Solutions for Various Markets
Table 2.1:	VoiceXML Development Tools (Web-Based tools)
Table 2.2:	SDKs (PC-based Development Tools)
Table 2.3:	Open Source SDKs (PC-based Development Tools)
Table 3.1:	Audio Files
Table 4.1:	VoiceXML Session Variables

CHAPTER 1

INTRODUCTION

The global impact and universal penetration of the Web have predominantly been driven by the simplicity of the open Hypertext Markup Language (HTML) standard. The Web development paradigm brought vendor and network independence to distributed applications, and drastically reduced the cost and skill required to quickly deliver powerful solution. VoiceXML is an emerging open standard that brings the web development paradigm to the interactive voice response (IVR) market, which means that existing Hypertext Transfer Protocol (HTTP) gateways to enterprise services and data built using technologies such as Secure Sockets Layer (SSL) and cookies can be seamlessly extended to the phone. VoiceXML has rapidly received almost universal adoption and support from all corners of the voice technology industry.

Speech applications are not very new to us. We are all very familiar with the interactive voice response (IVR) applications that we have encountered over the years. In the past several years, there has been a tremendous amount of activity in the area of voice access of Net information. VoiceXML is an XML-based markup language for distributed voice applications, much as HTML is a language for distributed visual applications. VoiceXML is designed for creating audio dialogues that feature synthesized speech, digitized audio, recognition of spoken and dual-tone multifrequency (DTMF) key input, recording of spoken input, telephony, and mixed interactive conversations. The goal is to provide voice access and interactive voice response to Net-based content and applications. VoiceXML brings the power of Web development and content delivery to voice response applications and frees authors and designers of such applications from low-level programming and resources management. It enables integration of voice services with data services using familiar Internet-centric paradigm, and it gives users the power to seamlessly transition between applications. Document servers provide the dialogues, which can be external to the browser implementation platform.

1.1 Background of Study

VoiceXML is a standard based on XML that allows Web applications and content to be accessed by a phone. Voice application developer can develop speech-based telephony applications using VoiceXML. The standard was developed by the VoiceXML Forum, which was founded by AT&T, IBM, Lucent, and Motorola. The goal of the VoiceXML Forum is to create a language that gives Web developers the ability to deliver content from their Web site over the telephone by using their existing coding skills. The intention is for the VoiceXML language to act as an interface to the low-level functions of programming and system resource management. The developer learns the tag set of the language, builds an interface, and lets the interface handle the access to the low-level functions.

VoiceXML unites the power of the Internet with the ubiquity of the telephone, make it possible for business to replace legacy, proprietary IVR platform with a unified architecture for delivering automated self-service from any device. Without exception, every major voice and call center technology company has embraced VoiceXML, and tens of thousands of developers have already begun building and deploying VoiceXML applications.

User Authentication is an essential to all private networks that are using the resources of the Internet, allowing access to privileged information such as customer records, propriety company data, and personalized account information. User Authentication using Speech Recognition is one of the methods to ensure the protection of the sensitive data over the Internet. The crucial characteristic of this project is to develop a speaker authentication prototype working with high precision to be used for different security purposes. As the name implies, voice authentication is a binary decision system that decides whether a spoken utterance belongs to the claimed person or an impostor tries to log in. Obviously, acceptance of an impostor to the system is completely intolerable if a security system is the case.

As enterprises and organizations increasingly turn to biometrics to ensure their customers safety, the market opportunities for voice authentication are infinite. That is why this Voice Authentication topic is selected for author’s Final Year Project. Table 1.1 below shows the voice authentication solutions for various markets.

Table 1.1 Voice Authentication Solutions for Various Markets

Market	Application	Driver
Financial Services	Access to Banking, Brokerage	Reduce Financial Risk, Reduce Fraud
Telecom	Call Center Applications Unified Messaging Auto Attendant	Reduce Fraud, Protect Personal Information, Competitive Advantage
Retail	Order Entry Personalized Service	Reduce Fraud, Increase Revenue
Enterprise and IT	Access to Intranet, Extranet and Corporate Applications	Increase Security, Reduce Cost
Travel	Frequent Customer Services	Convenience Personalization
Internet	Authenticate Users for Internet Banking and e-Commerce	Reduce Financial Risk

1.2 Problem Statement

Web sites need ways to block unauthorized users from gaining access to private information. One way to do this is by using user authentication which prompt user to enter their username and password. The goal of user authentication is to make the right information reliably and securely available to the right people. This standard authentication addresses several security issues such as hacking and password cracking which will cause unauthorized access.

1.2.1 Problem Identification

By using user authentication, Web developers created login systems for their Web sites, which include requesting usernames and passwords before allowing users to gain access to secure or paid materials. From the research that has been done, this standard authentication address several security issues such as hacking and password cracking which will cause unauthorized access.

1.2.2 Significant of the Project

This project examines three levels of security for validating users through voice authentication procedure. VoiceXML technology is capable of using existing resources (username and password) plus adding voice authentication as another way of verifying the identity of the user. This project will create a user authentication system, using multiple ways to confirm the user's identity.

1.3 Objectives and Scope of Study

1.3.1 The Relevancy of the Project

There are two different approaches involved in each stage of the development process.

- *Research on the VoiceXML technology.*

Research will be carried out on how VoiceXML is capable of using existing authentication resources (username and password) and adding it with voice authentication as another way of verifying the identity of the user. Here, studies regarding its concept, architecture, security and programming will be taken in focus.

- *Development of the prototype.*

Developing the VoiceXML prototype (Voice Authentication Using VoiceXML), which demonstrates three levels of security as a solution for validating users.

1.3.2 Objectives

1. To understand VoiceXML technology architecture and learn to develop and design a VoiceXML application.
2. To examines three level of security using VoiceXML capabilities as a solution for validating users.

1.3.3 Scope of study

The scope of this project is to do research on how VoiceXML is capable of using existing authentication resources (username and password) and adding it with voice authentication as another way of verifying the identity of the user. Here, studies regarding of its concept, architecture, security and programming will be taken in focus. The prototype that will be design and develop in this project is “Voice Authentication Using VoiceXML” that demonstrate Voice Authentication Using VoiceXML to conclude the findings of three levels of user validation using VoiceXML.

1.3.4 Feasibility of the Project

The product of this project is a Voice Authentication Using VoiceXML prototype focus on three levels of security as a solution for validating users. The prototype applies technical requirement that is obtain from the research. Feasibility of the project is depending on the tools and time available. The scope of project seems to be feasible for author to complete on time. However, there are some voice authentication procedures that require very advanced tools/platform to be implemented like voice verification (Voice Print). The scheduled tasks for this semester of the final year project are summarized in the Gantt chart in the APPENDIX A.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Even though we are familiar with the interactive voice response (IVR), and other voice application, VoiceXML is considered as new standard for voice technology. It helped energize the consumers, and hence the industry, to varying degrees.

A *session* begins when the user starts to interact with a VoiceXML interpreter and continues as VoiceXML documents are loaded and unloaded. The session ends when requested by the user, VoiceXML document or interpreter context. The platform defines the default session behavior, although this can be overridden in part by VoiceXML.

VoiceXML documents define applications as a set of named *dialog states*. The user is always in one dialog state at any time. Each dialog specifies the next dialog to transition to using a URL.

VoiceXML dialogs include: *forms* and *menus*. A menu presents the user with a choice of options and the transitions to another dialog state based upon the user's selection. A form defines an interaction that collects values for each of the *fields* in the form. Each field may specify a prompt, the expected input, and evaluation rules. The form can be submitted to a server in much the same way as for HTML.

An *application* is a set of VoiceXML documents that share the same application root document. The root document is automatically loaded whenever one of the application documents is loaded, and remains loaded until there is a transition to a different application, or when the call is disconnected. The root document information is available to all documents in the same application.

Each dialog state has one or more *grammars* associated with it, that are used to describe the expected user input, either spoken input or touch-tone (DTMF) key presses. In the simplest case, only the dialog's grammars are active in that dialog. In more complex cases, other grammars can be active.

- grammars defined within the dialog itself
- external grammars referenced by links
- grammars defined at the document level and marked as being globally active
- grammars defined in the root application document and active throughout the application

A *subdialog* is like a function call: it allows you to call out to a new dialog and then returns to the original dialog, retaining the local state information for that dialog. Sub dialogs can be used to handle confirmations and to create a library of re-usable dialogs for common tasks.

VoiceXML allows you to define named *variables* for holding data. These can be defined at any level and their scope follows an inheritance model. You can test the values of variables to determine what dialog state to transition to next. Variable expressions can also be used for conditional prompts and grammars etc.

Events are thrown when the user fails to respond to a prompt, or when the input can't be understood. VoiceXML allows you to write handlers for catching events. These follow an inheritance model, and events can be caught at a higher level if there is no corresponding handler at the dialog level.

VoiceXML allows you to use *scripting* (ECMAScript) when you need additional control over the application. VoiceXML employs a form filling metaphor. You can define a complex grammar for collecting the values of several fields in a single response. Any unfilled fields can be handled by special subdialogs defined inline within each dialog.

2.2 VoiceXML

VoiceXML uses the syntax and structure of XML to create a language to Voice-Enable the existing Web Sites. VoiceXML provides a high-level programming interface to speech and telephony resources for application developers, service providers and equipment manufacturers. As such, the language follows all of the syntactic rules of XML with semantics that support the creation of interactive speech applications. Standardization of VoiceXML will simplify creation and delivery of Web-based, personalized interactive voice-response services; enable phone and voice access to integrated call center databases, information and services on Web sites, and company intranets; and help enable newvoice-capable devices and appliances. VoiceXML is expected to expand access to the Internet through telephones and other devices using both speech and ordinary touch-tone user interfaces.

According to Chetan Sharma and Jeff Kunins (2002)

Every now and then, there comes an agreed-upon, widely adopted standard or enhancement in a technology that starts a tidal wave in the industry by enhancing performance, by reducing costs at least a factor of 10, or by allowing for application or services that were not easily attainable before. XML, VoiceXML, and speech recognition are such new standards and technologies for the voice industry. (p.6)

VoiceXML is an XML-based markup language for distributed voice applications, much as HTML is a language for distributed visual applications. The establishment of a VoiceXML forum standardize a voice markup language is probably the single largest reason for the growth in the interest and market potential of voice-based applications and services. (p.7)

2.4 VoiceXML Development Tools

The development tools can be divided into two categories: Web-based tools and Software Development Toolkits (SDKs). Web-based tools such as BeVocal, Tellme, and VoiceGenie allow us to build, test, log, and run applications online for no cost. These vendors have also provided numerous tutorials and examples, which will help developers in getting a jump start. SDKs, on the other hand, offer self-contained offline development environments, which allow us to use our personal computer or desktop to work on our applications. Table 2.1 and 2.2 below show both development tools for Web-based tools (Table 2.1) and SDKs (Table 2.2) available on the market. Table 2.3 in the other hand, will shows all Open Source development tools available for education and research.

Table 2.1: VoiceXML Development Tools (Web-Based tools)

Vendor	Website	Description
BeVocal	cafe.bevocal.com	Online development tools, components, debugger, and documentation
HeyAnita	freespeech.heyanita.com	Online development tools, components, debugger, and documentation
Tellme Networks	studio.tellme.com	Online development tools, components, debugger, and documentation
VoiceGenie	developer.voicegenie.com	Online development tools, components, PC-based debugger, documentation
Voxeo	community.voxeo.com	Online development tools, components, debugger, documentation

2.3 VoiceXML Architecture

A typical VoiceXML implementation consists of a VoiceXML interpreter and a VoiceXML interpreter context. The interpreter is responsible for executing VoiceXML code, servicing the real-time control of multiple VoiceXML applications that may be running simultaneously. The interpreter context is responsible for handling support activities such as loading the initial document when a call is received and invoking the interpreter once a voice command is received. Both the interpreter and the interpreter context work with a speech recognition engine(s), text-to-speech (TTS) engine(s), and media server(s). This infrastructure connects to the public switched telephone network (PSTN) using telephony switching software and hardware.

According to Kenneth G. Rehor (2001)

A VoiceXML application consists of several components:

- *Application Server*: Typically a Web server, which runs the application logic, and may contain a database or interfaces to an external database or transaction server.
- *VoiceXML Telephony Server*: A platform that runs a VoiceXML *interpreter* that acts as a client to the application server. The interpreter understands VoiceXML dialogs and controls speech and telephony resources. These resource include ASR, TTS, audio play and record functions, as well as a telephone network interface.
- *Internet-style network*: A TCP/IP-based packet network that connects the application server and telephony server via HTTP.
- *Telephone Network*: Typically the *Public Switched Telephone Network (PSTN)*, but could be a private telephone network (e.g. PBX), or VoIP packet network.
Caller: Any telephone that can connect to the telephone network.

APPENDIX C shows the architecture diagram of VoiceXML.

Les Hamilton (2001) is very definite: "VoiceXML is a new flavor of XML that defines structures for playing prerecorded voice prompts as well as text-to-speech generation for presentation to the user over the telephone. The integrated response from the user is handled by either DTMF (touch tone) or speech recognition. The World Wide Web Consortium's (W3C) working draft on "Voice Browser" activity defines the standards for VoiceXML. W3C is diligently working to expand access to the Web by allowing people to interact with Web sites via spoken commands. This technology allows any telephone to access Web-based services and is especially helpful to people with disabilities. It will also improve interaction with display-based Web content in cases where the mouse and keyboard may be missing or inconvenient".

Carla King (2001) points out that the VoiceXML "is an HTML-like language for specifying voice dialogs. It brings together speech and telephony technologies such as automatic speech recognition (ASR) and text-to-speech (TTS) in a markup language so your software can take direction from users' spoken words or their telephone keypad tones, and respond to them via synthesized speech or audio files. VoiceXML provides this dialog management capability for the application using conventional Web and application servers".

Table 2.2: SDKs (PC-based Development Tools)

Vendor	Development Tool	Description
Nuance Communications	V-Builder™	PC-based VoiceXML tool
Audium Corporation	Audium Central	Java-based VoiceXML tools and components
IBM	WebSphere Voice Toolkit	PC-based VoiceXML tools and components
Cambridge VoiceTech	Voice Studio	PC-based VoiceXML tools and components
Telera	DeVXchange, AppBuilder™	PC-based VoiceXML tool and online development tools, components, debugger, documentation

Table 2.3: Open Source SDKs (PC-based Development Tools)

Organization	Development Tool	Description
Faculty of Informatics, Masaryk University, Brno, Czech Republic.	OptimTalk	VoiceXML interpreter
Public Voice Lab, Vienna, Austria	publicVoiceXML	VoiceXML platform
Columbia University Department of Computer Science, NYC, USA	sipvxml	SIP-based VoiceXML interpreter

2.5 Speech Recognition

Speech Recognition involves the computer taking the user's speech and interpreting what has been said. This allows the user to control the computer (or certain aspects of it) by voice, rather than having to use the mouse and keyboard, or alternatively just dictating the contents of a document.

One of the features that VoiceXML supported is this Speech Recognition feature. With this prevailing feature, voice application can understand user's spoken input rather than DTMF input only like traditional IVR system.

Chetan Sharma and Jeff Kunins (2002) point out that "Automatic Speech Recognition (ASR) is a technology that allows a machine to understand human speech. Over the years, human speech interactions have become more sophisticated. Over the past 30 years, through much research and development, speech recognition accuracy has increased tremendously; processor costs have gone down dramatically; and, with the advent of the Internet and VoiceXML, there has been a general enthusiasm for voice-based solutions among the business community and consumers alike." (p.15)

2.6 User Authentication

User Authentication is a method used by Web developers to protect and secure sensitive information over the Internet. Conventional way of User Authentication is by prompting user a username and unique password.

2.6.1 Security Issues

Combination of username and password is simply to ensure the right information reliably and securely available to the right people. However, this standard authentication method of providing a login and password often poses several security issues like unauthorized access, unknown users, and hacking activities.

2.6.2 Voice Authentication

Voiceprints are as unique as fingerprints. No two are alike. And voiceprints and voice authentication technology are being increasingly used by companies around the world to protect access to information, secure transactions, and replace passwords and Pins. Companies are turning to voice authentication for a number of reasons. Voice authentication is more effective. In fact, a University of Edinburgh study determined that voice authentication was "100 times more secure than traditional PIN-based methods." Voice authentication is also easier to implement than other biometrics. Unlike fingerprinting, iris scanning and facial recognition no expensive high-resolution cameras or other specialized devices are needed. It's more convenient. Enrollment is just by answering a few questions to record voiceprint. It's more flexible and can be combined with speech recognition and knowledge verification to further enhance security. And it's less expensive. Since it works from any telephone, the cost to deploy is extremely low.

A voice identification system, like other biometric technologies, requires that a "voice reference template" be constructed so that it can be compared against subsequent voice identifications. To construct the "reference template" an individual must speak a set phrase several times as the system builds the template.

A major concern for voice identification systems is how to account for the variations in one's voice each time a voice identification occurs. The rate and pitch at which an individual speaks at one moment is not always the same as the next moment in time. To help eliminate these types of variations during voice identification, a process comprising Hidden Markov Modeling is applied.

The basis of this approach is that the system (software) uses language models to determine how many different words are likely to follow a particular word. The realized advantage here is that groups' words (matching word pools) that sound alike, for example "to", "two", and "too", are drastically reduced and actual words are recognized. Error rates that use this type of language modeling are from one to 15 percent (Ruggles, T. 1998).

Customer-driven industries such as financial services, government and enterprise call centers will continue to strive for higher security, but also for higher convenience, providing anytime, anywhere access. Whereas, telephones have traditionally been the ubiquitous mobile device, the growing trend of multi-modal devices (PDA's, automobiles and other electronic devices) are also looking to incorporate security measures to protect sensitive information contained on the device. Here, too, speaker verification stands to provide an ideal way to authenticate a user in a secure and convenient manner, even without having to state your 'mother's maiden name.'

2.6.3 VoiceXML is the Answer

VoiceXML provides an alternative to these expensive and complicated biometric systems. Designed to leverage the existing Web infrastructure, it evokes a markup language that is analogous to HTML, a standard for creating Web sites. Since it is similar to HTML, Web developers can easily learn VoiceXML, leading to a large pool of developers who can quickly create voice applications.

There are three main benefits of Voice Authentication using VoiceXML:

1. **Improve Security:** This is the key objective of using voice biometrics; improving the security of sensitive information and reducing fraud. A password is what people know. But the voiceprint is what they have.
2. **Reduce Costs:** Using an automated authentication process reduces salary expenses, toll-costs and call hold times and allows live agents to focus on revenue-generating calls.
3. **Simple:** Since VoiceXML is similar to HTML, Web developers can easily learn VoiceXML, leading to a large pool of developers who can quickly create voice applications. VoiceXML allow developers to build telephone voice applications with nothing more than an HTTP server compare to traditional IVR which is very hardware-oriented and low level. From this, great innovation shall spring.

According to Mark Miller (2002)

Web developers have created login systems for their Web sites, which include requesting username and passwords before allowing users to gain access to secure or paid materials. VoiceXML is capable of using those existing resources plus adding voice authentication as another way of verifying the identity of the user.

Further detail about ‘Voice Authentication with VoiceXML’ will be discussed in Chapter 4.

CHAPTER 3

METHODOLOGY/ PROJECT WORK

3.1 Project Work Procedure

There are two main activities that the author must complete in order to fulfill this Final Year Project. First one is the research on VoiceXML technology architecture and the other one is to develop VoiceXML prototype which will demonstrate Voice Authentication through speech recognition.

3.2 Research Process

The research is done based on a model illustrated in the book title "*The Research Process for Applied and Basic Research*" by Uma Sekaran. APPENDIX B shows the basic research process that has been run through by the user in doing the research.

3.3 Voice Application Life Cycles

The prototype development is being implemented by applying "The Voice Application Life Cycle" (Chetan Sharma and Jeff Kunins, 2002, p.329). Nevertheless some slight changes and alteration has been added and deducted to suit the project establishment due to the project's time constraint. Author has decided to use only 4 stages out of 6 to suit with the complexity of the project. The Voice Application Life Cycle is illustrated below in Figure 1.1

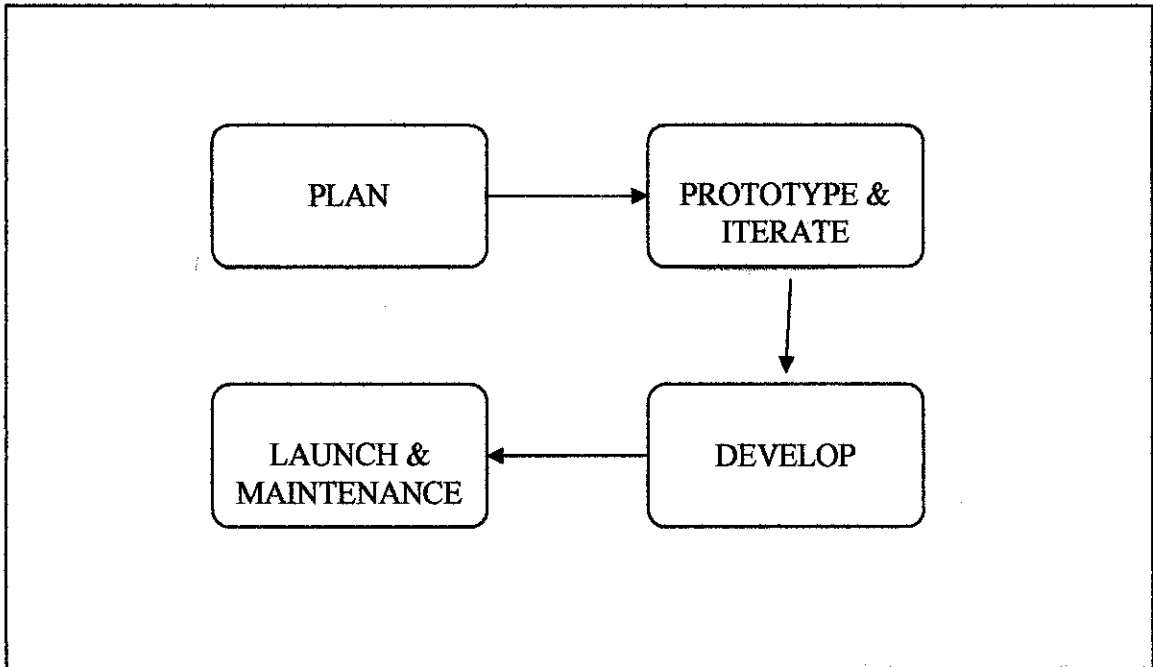


Figure 1.1 Voice Application Life Cycles

3.3.1 Planning

As with all software development projects, the initial planning stage is ultimately the most critical for determining whether a project will succeed. At this stage, Author discuss with the project's Supervisor regarding the project's requirement and the budget availability. Feasibility Analysis also been conducted by observing at the availability of the VoiceXML tools. At this stage also, Author was being briefed by project's supervisors about application scope and feature details.

3.3.2. Prototyping and Iteration

In this second stage the key processes is where all the prototype development took place. In this stage, author developed initial prototyping by learning the syntax of VoiceXML language. During the initial prototyping, usability testing being conducted with the guidance from project's Supervisor to test the reliability of the prototype.

3.3.3 Development

Once a very clear plan for the application has been established between the Author and the project's Supervisor, after an iterative series of usability tests and design revisions, then all is clear for the Author to actually develop the complete application. During this phase, the author prepared all the audio files needed for the prototype. Table 3.1 below shows the list of audio files prepared for the prototype. At the end of this development, once again usability testing being performed to test the reliability of the prototype before it is launch.

Table 3.1: Audio Files

File Name	Description
username.wav	Prompting user to speak their Username.
password.wav	Prompting user to speak their Password.
authorized-login.wav	Alert user about an authorized login.
invalid-login.wav	Alert user about an unauthorized login.

3.3.4. Launch and Maintenance

In this final stage, the prototype developed is launched and ready to enter maintenance mode.

3.4 Tools/equipment required

For this prototype development, author will be using both Web-Based development tool and SDK (PC-based development tool). For SDK, author will be using Open Source SDKs which freely available on the internet. However, those Open Source SDKs will not be able to support some VoiceXML features needed for this prototype development.

3.4.1 Software Requirements

- **VoiceXML Application Development Tools (Web-Based Tools)**
 - Bevocal Café (developers.bevocal.com)
 - Bevocal Developer Account
 - Internet Browser (Microsoft Internet Explorer)

- **VoiceXML Application Development Tools (Open Source SDKs)**
 - OptimTalk Runtime Application
 - publicVoiceXML platform

- **Speech Application Programming Interface (SAPI)**
 - Microsoft SAPI 5.1
 - Microsoft Speech SDK 5.1

- **User Interface Development Tool**
 - Microsoft Visual Basic 6.0

3.4.2 Hardware Requirements

- **Personal Computer / Desktop**
 - Windows 98,NT,2000, or XP
 - Pentium 4 processor with 1.4 GHz or higher
 - 64 MB RAM memory or higher
 - 1 GB hard disk space or higher
 - Sound Card
 - Video Card

- **Voice Tools**
 - Speakers or Headphone
 - Microphone

CHAPTER 4

RESULT AND DISCUSSION

4.1 Results and Findings

This is the critical and the most important part in the project. All research work and product are presented in this section. Both research and prototype must meet the requirements to fulfill the objectives of the project. The expected end result off this project is to study three levels of security as a solution for validating users and as well as the developed application that can authenticate a user through speech recognition. The application is expected to take full advantages of current VoiceXML technology. The application can be used as a template for future use by developers in order to make their voice application achieve the goal of user authentication which is to make the right information reliably and securely available to the right people. At the end of this project, author expected to successfully acquire deep understanding of VoiceXML technology architecture and learn to develop and design a VoiceXML application.

4.1.1 VoiceXML Concept and Architecture

VoiceXML is a language for creating voice-user interfaces, particularly for the telephone. It uses speech recognition and touchtone (DTMF keypad) for input, and pre-recorded audio and text-to-speech synthesis (TTS) for output. It is based on the Worldwide Web Consortium's (W3C's) Extensible Markup Language (XML), and leverages the web paradigm for application development and deployment. By having a common language, application developers, platform vendors, and tool providers all can benefit from code portability and reuse.

With VoiceXML, speech recognition application development is greatly simplified by using familiar web infrastructure, including tools and Web servers. Instead of using a PC with a Web browser, any telephone can access VoiceXML applications via a VoiceXML "interpreter" (also known as a "browser") running on a telephony server. Whereas HTML is commonly used for creating graphical Web applications, VoiceXML can be used for voice-enabled Web applications.

A VoiceXML implementation can take voice commands over the phone, translating those spoken commands to text. The Speech-to-Text (STT) engine handles the conversation of spoken language to text. Once the speech has been transformed to text, it is sent to a processing mechanism, which can be any that is available on Web sites. PHP, Perl, JSP, ASP, or any other sever-side script can be used to process the request. This gives access to all the resources on the receiving computer, including databases, flat files, and any other types of content that are needed to process the request.

The result of the processing is sent back to the VoiceXML application. The application then translates the results by using a Text-to-Speech (TTS) translator, outputting the response verbally to the caller. In addition, the VoiceXML application can have prerecorded audio file responses using human voices that are applicable as a response to the request. The application may then ask the user for more information and continue the session or terminate the call. Figure 2.1 below illustrated the basic processing flow of VoiceXML.

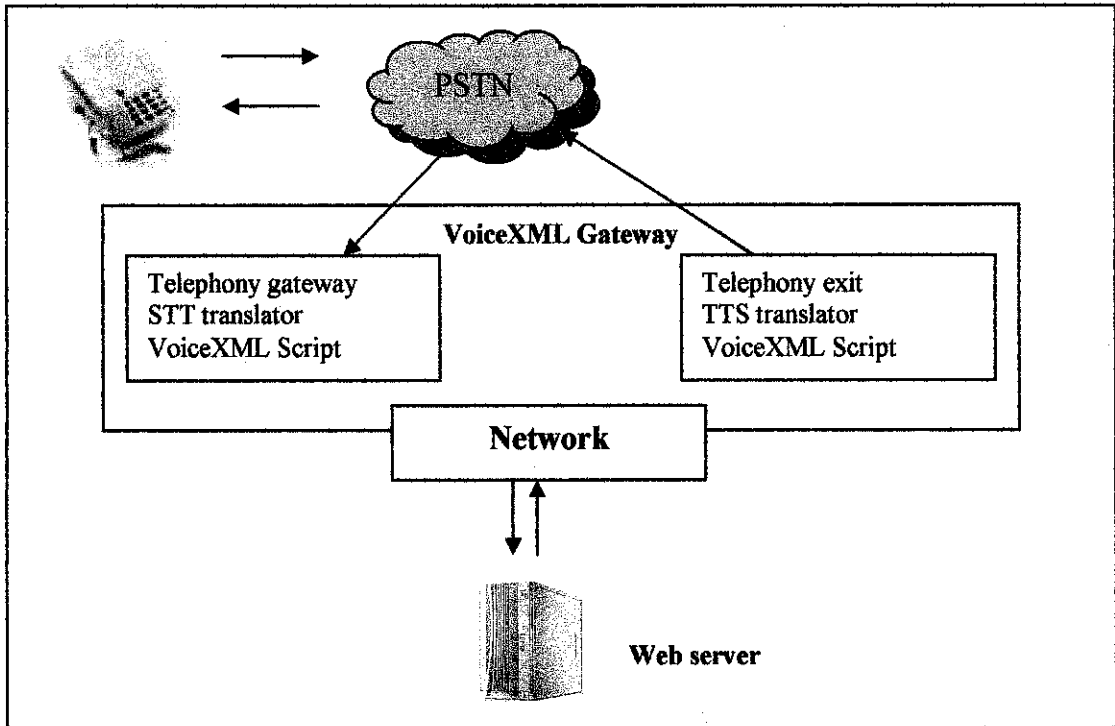


Figure 2.1 VoiceXML basic process flow.

One of the most important concepts of the VoiceXML language is that it isolates the user interaction from the processing of a request. As shown in Figure 2.1 above, the VoiceXML interpreter handles the interface between the telephone and the processing layer. It does not handle the processing of the request. The VoiceXML front end acts as a filter to the service layer that is independent of any specific processing mechanism, the VoiceXML application is portable across processing platforms.

The user interface consists of a series of dialogs that guide the caller, collecting information for the processing implementation. The interface may be single document with a set of menu or a series of documents that are linked through calls to different Uniform Resource Identifiers (URIs). APPENDIX C shows the architecture diagram of VoiceXML.

4.1.2 User Authentication

There are two ways of restricting access to documents: either by the hostname of the browser being used, or by asking for a username and password. The former can be used to, for example, restrict documents to use within a company. However if the people who are allowed to access the documents are widely dispersed, or the server administrator needs to be able to control access on an individual basis, it is possible to require a username and password before being allowed access to a document. This is called user authentication.

Almost every enterprise application will be equipped by a user authentication. For web developers, User Authentication is used to protect and secure sensitive information over the Internet. Conventional way of User Authentication is by prompting user a username and unique password. While authentication does allow resources to be restricted to particular users, there are still potential security issues related like unauthorized access, unknown users, and hacking.

4.1.3 Three Levels of Authentication using VoiceXML

Unique characteristics of VoiceXML that support telephony, speech recognition and voice features provide different level of authentication to be applied on the application. VoiceXML is capable of using existing resources (username and password) plus adding voice authentication as another way of verifying the identity of the user. This will provide the application with three levels of security using the caller's phone number, a username/password combination, and a voice print recognition system.

4.1.3.1 Verify a Caller's Phone Number

Since VoiceXML application is being accessed through phone call, a simple way to begin a secure login is to test the caller's phone number through standard session variables. Session variables within VoiceXML application make information available to the application such as the caller's phone number, the number the caller dialed, and type of machine

the call was originated from (pay phone, cell phone, and so on). But however, not all VoiceXML interpreters can support this session variable. Since the author is using the Open Source SDK, this feature cannot be simulate or test because the SDK is PC-Based which not involved telephony network. Nonetheless, these feature, provide one level of security on the VoiceXML application.

A session variables has a value of *undefined* if the service is not supported. Session variables can be accessed within VoiceXML application and used for making decisions on how to process a call. Table 4.1 below is a list of session variables and the type of information they contain.

Table 4.1: Session Variables

Session Variable	Definition
session.telephone.ani	Automatic number identification
session.telephone.dnis	Dialed number identification service
session.telephone.iidigits	Information indicator digits
session.telephone.uui	User-to-user information

4.1.3.2 Spoken Username and Password

In this second level of security of VoiceXML application, users are prompted to provide the username and password just like the conventional User Authentication method but instead of typing the username and password, users are required to provide them verbally.

This kind of authentication method can be implemented easily within VoiceXML application is because of the Speech Recognition feature supported by VoiceXML.

Speech Recognition involves the computer taking the user's speech and interpreting what has been said. This allows the user to control the computer (or certain aspects of it) by voice, rather than having to use the mouse and keyboard, or alternatively just dictating the contents of a document. With this prevailing feature, voice application can understand user's spoken input rather than text input or DTMF input.

4.1.3.3 Voice Authentication (Voice Print)

In addition to requesting a user password, when the user speaks, the voice can be checked against stored voice patterns known as voice print. Voice prints are as unique as fingerprints. No two are alike. This security feature can be implemented in VoiceXML application as third level security. Most VoiceXML platform can store's a user's voice print and make it available for use during authentication. The implementation, however, will be platform specific. Since the author is using Open Source tool, this feature will not be implemented in the prototype application as the tool have no ability whatsoever to support this powerful feature.

Even though the prototype will not implement this voice print feature, but it is worth learning it since it is a third level of security offered by VoiceXML to give the VoiceXML application the triple-strength security.

4.2 Product Results and Discussion

The final product for the project is a prototype that has been called Voice Authentication Using VoiceXML that focuses on different level of security using voice authentication with VoiceXML.

For the prototype development, the author is using an Open Source tool known as OptimTalk. OptimTalk is a VoiceXML platform that can be used for building, running, tuning and evaluating dialog applications that use natural language for communication with users. OptimTalk is being developed by Pavel Cenek, an independent developer from the Czech Republic and research institute Norut IT, Tromsø, Norway in cooperation with the Laboratory of Speech and Dialogue at Faculty of Informatics, Masaryk University Brno, Czech Republic.

As discussed before, through research and finding, author has identified three levels of security that can be implemented using VoiceXML in the voice application. The three levels are verifying caller's phone number, verbal username and password, and lastly voice print recognition. These three levels of security are very powerful to provide triple-strength security to the voice application which can solve the security issued pose by conventional method of user authentication.

The major concern in developing the prototype is the availability of the development tool that can support the three level of security discussed above. Since VoiceXML is the new technology on voice application area, it is hard for the author to find suitable tool for the implementation of the project. Only few open source tools have been developed for the purpose of educational and most of the tool cannot support certain VoiceXML tags. Nonetheless, research is still performed by the author in all those three level of security. Figure 3.1, 3.2, 3.3 and 3.4 below illustrate the prototype runtime environment. Figure 4.1 and 4.2 shows the VoiceXML Simulators developed to simulate the Voice Authentication. APPENDIX D shows the program flow chart for the prototype.

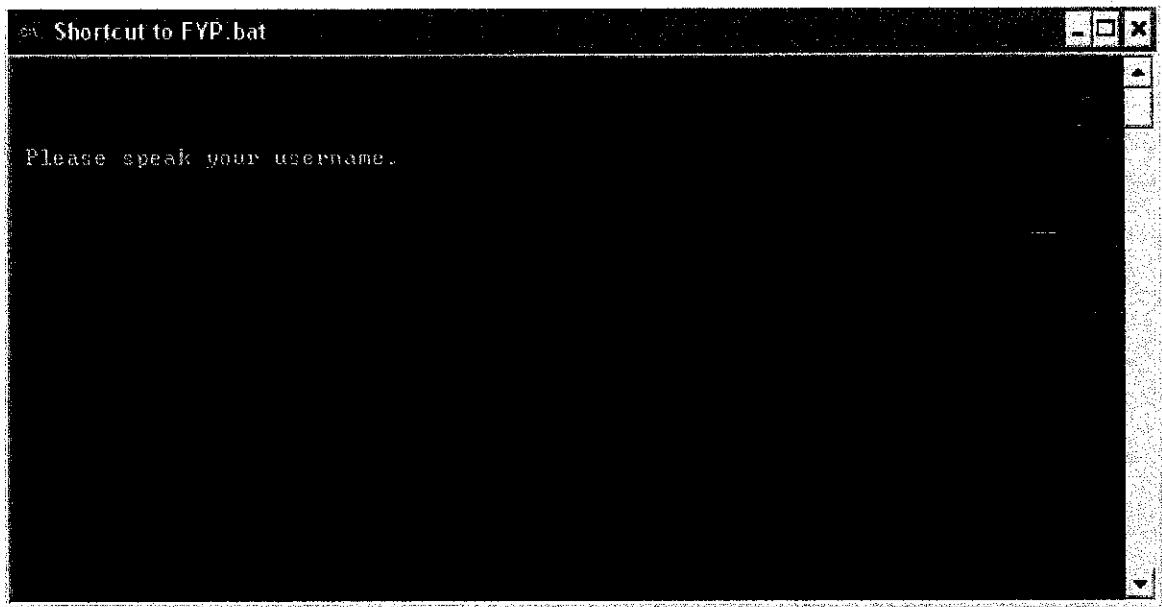


Figure 3.1 Username Prompting

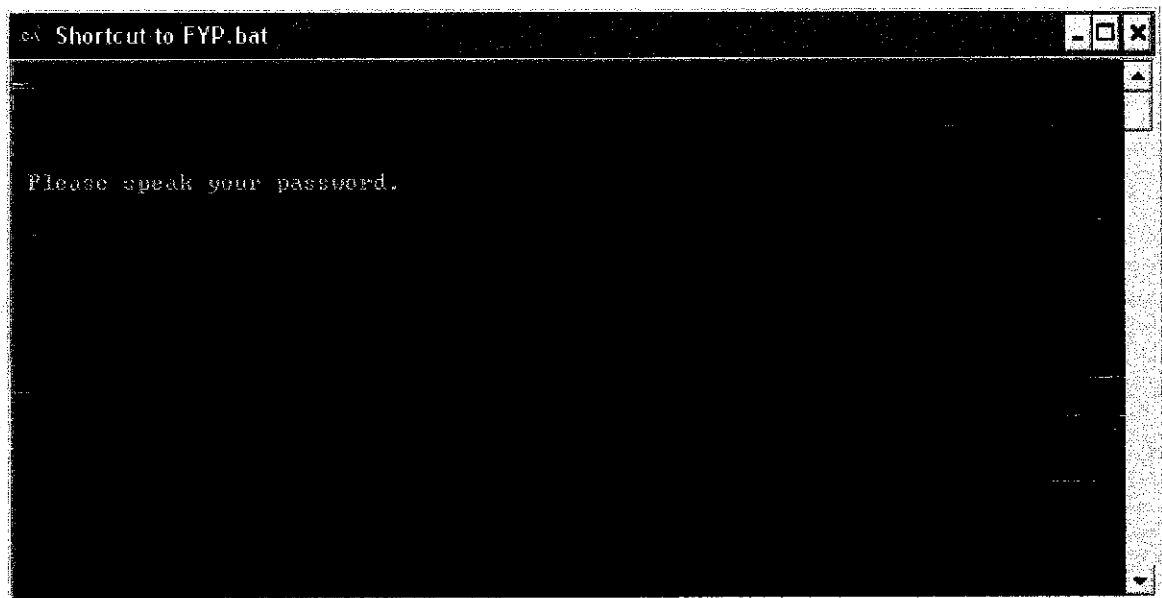


Figure 3.2 Password Prompting

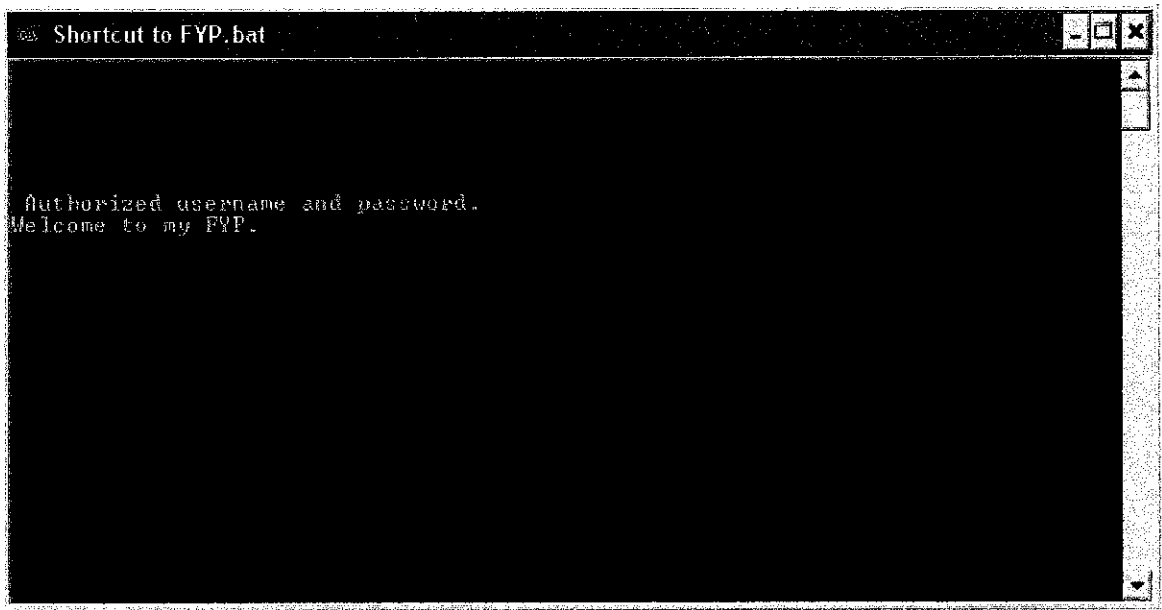


Figure 3.3 Authorized Login

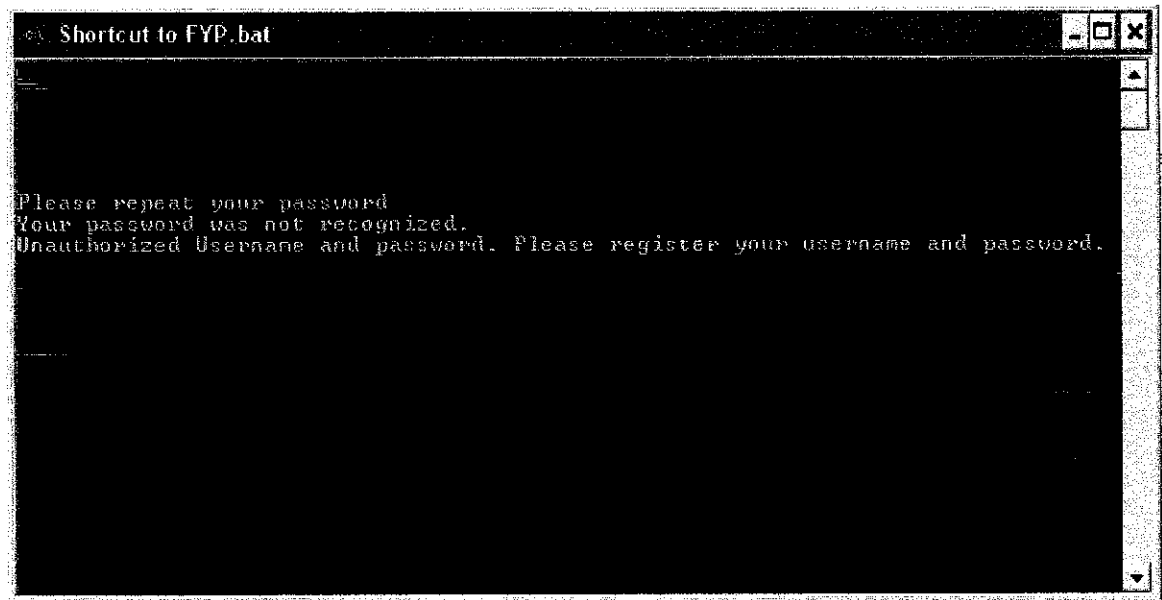


Figure 3.4 Unauthorized Login

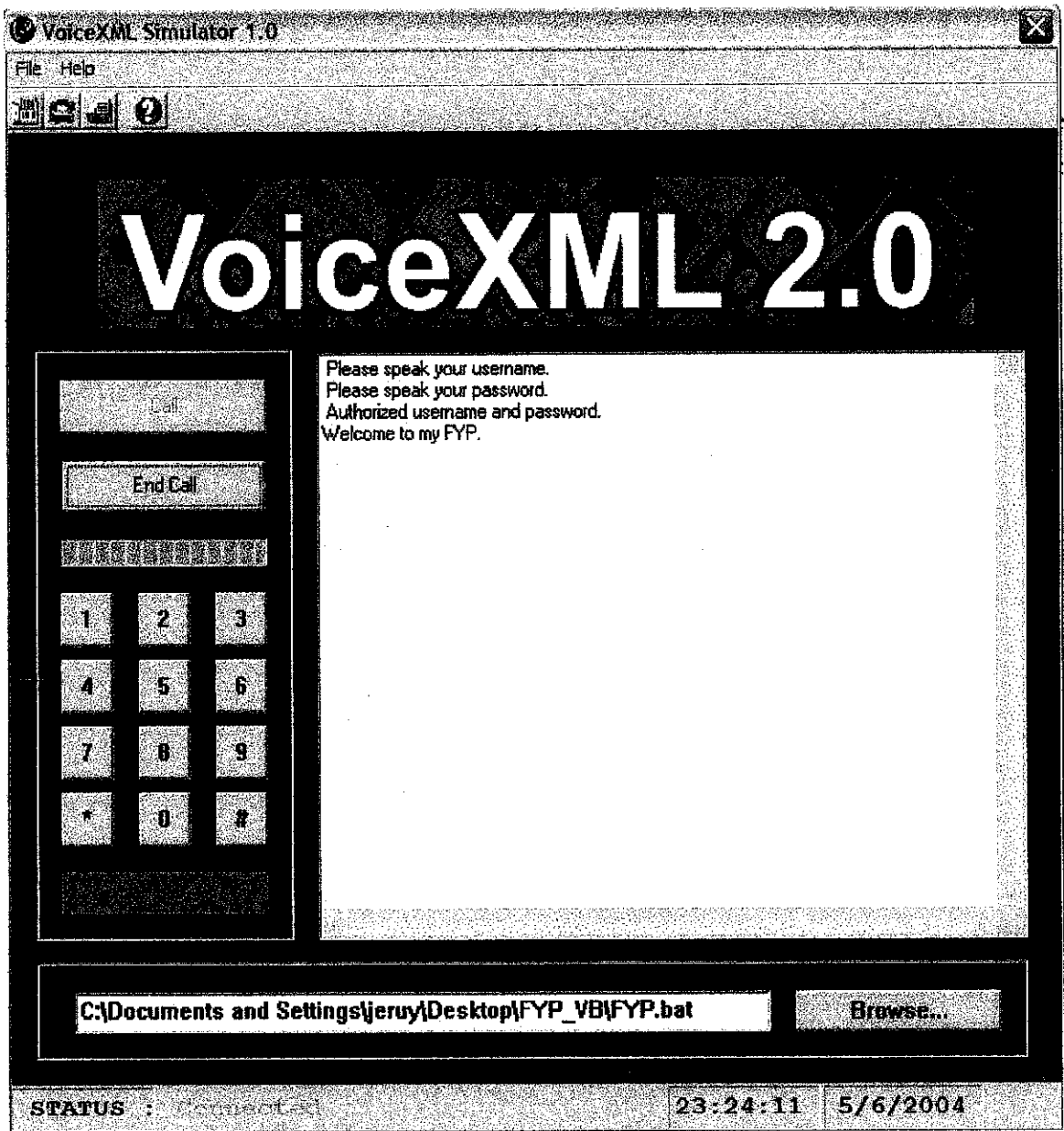


Figure 4.1 VoiceXML Simulator 1.0

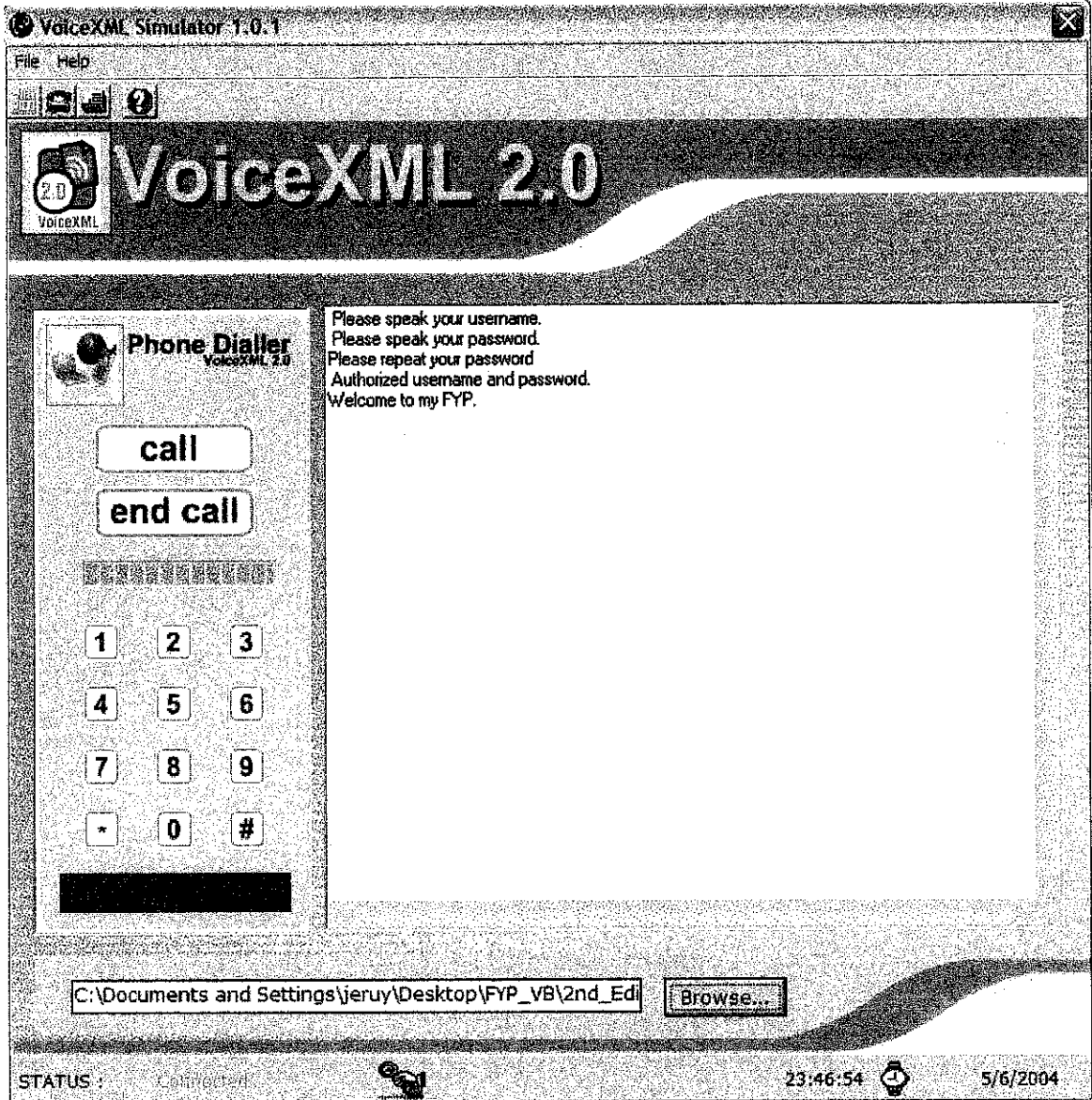


Figure 4.2 VoiceXML Simulator 1.0.1

CHAPTER 5

CONCLUSIONS

5.1 Relevancy to the Objectives

As stated in chapter 1, this project has two main objectives. The first objective is to understand VoiceXML technology architecture and learn to develop and design a VoiceXML application. Author had completed the research on the VoiceXML architecture and documented the findings in the Literature Review chapter of this report. The result was excellent and author gained a lot from it. With the understanding of VoiceXML architecture and concepts of voice application motivated the author to study the syntax of VoiceXML. Thus, with the deep understanding of the syntax of VoiceXML, author is able to develop and design a VoiceXML application. This is proved through the prototype that had been developed.

The second objective of the project is to observe three levels of security as a solution for validating users. Begin with understanding of User Authentication; author had identified three solutions for validating users through voice authentication procedures using VoiceXML. The results and discussions about User Authentication and three level of security of VoiceXML had been documented on Result and Discussion section (Chapter 4). Even though the prototype not be able to implement all three level of security discussed earlier in the prototype, but still it considered as achieving the last objective because the author managed to design and develop a VoiceXML prototype.

5.2 Recommendation

This project takes advantage of the current VoiceXML technology to solve problem in insecure user authentication system. The voice authentication routine coupled with a username and password combination is currently one of the most secure methods of using VoiceXML for verifying a user. Voice recognition interfaces solve problems for the user as well as the system administrator. Even though the author had managed to develop the prototype, it will be much better if the prototype can be implemented in End-to-End environment in the telephony network. Provided with required hardware and software, the author recommends the enhancement on the prototype so it can be deployed in the End-to-End environment.

VoiceXML provides an interesting area of study. The only major concern of this research is the availability of the development tools. This is mainly because VoiceXML is a new technology. The implementation of VoiceXML application required expensive hardware and software. Author highly hopes that Universities especially in Malaysia will provide their faculty with the VoiceXML tools to encourage research and development of this technology in this country.

It also will be good if Universities can offer the course or subject on VoiceXML because it is worth learning this new technology.

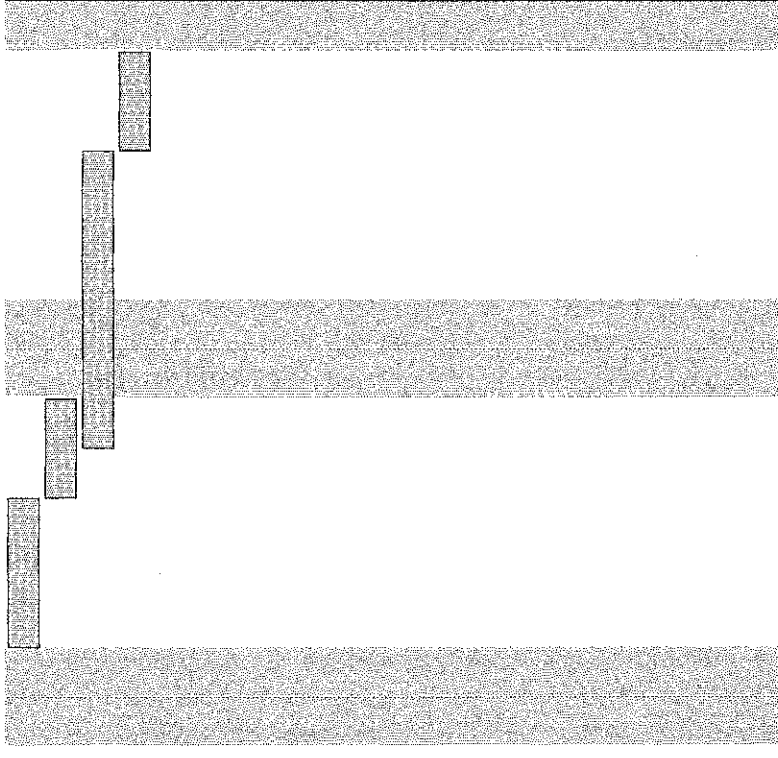
REFERENCES

- 1) Cheetan Sharma and Jeff Kunins. VoiceXML: Strategies and Techniques for Effective Voice Application Development with VoiceXML 2.0. John Wiley & Sons, Inc. 2002.
- 2) Mark Miller. 10 Projects to Voice-Enable Your Website. Wiley Publishing, Inc. 2002.
- 3) Speech Synthesis & Speech Recognition: Overview
(<http://bdn.borland.com/article/0,1410,29580,00.html>)
- 4) Developing a Voice User Interface (VUI) System July 26, 2001
<http://www.carlaking.com/techwriting/techarticles/Sun/vui.jsp>
- 5) Voice Identification
(http://et.wcu.edu/aids/BioWebPages/Biometrics_Voice.html)
- 6) What is VoiceXML, By Kenneth G. Rehor, Volume 1, Issue 1 - January 2001
(www.voicexmlreview.org/Jan2001/features/Jan2001_what_is_voicexml.html)
- 7) www.voicexml.org/resources/devtools.asp
- 8) <http://www.minwar.com/learn/VoiceXML>
- 9) <http://www.voicexmlcentral.com>
- 10) <http://www.voicexmlreview.org>
- 11) www.nwfusion.com/details/793.html?def
- 12) www.telera.com/stageone/files/Telera/collateral/VXML%20Primer.pdf
- 13) www.telera.com/stageone/files/Telera/collateral/AppDev_WP_4-18-02.pdf
- 14) www.nwfusion.com/news/tech/2004/0322techupdate.html
- 15) www.voicexmlreview.org/Jan2001/features/Jan2001_what_is_voicexml.html

APPENDICES

APPENDIX A: PROJECT TIMELINE / GANTT CHART

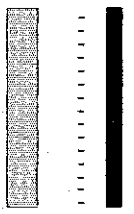
3	☐	User Scenarios and task analyses	3 days	Mon 1/26/04	Wed 1/28/04
4	☐	Feasibility Analysis	2 days	Thu 1/29/04	Fri 1/30/04
5	☐	Application scope and feature details	4 days	Fri 1/30/04	Wed 2/4/04
6	☐	Creative Brief	2 days	Thu 2/5/04	Fri 2/6/04
7	☐	Stage 2 : Prototyping and Iteration	25 days	Mon 2/9/04	Sun 3/14/04
8	☐	Initial prototyping	20 days	Mon 2/9/04	Fri 3/5/04
9	☐	Usability Testing	10 days	Mon 2/23/04	Fri 3/5/04
10	☐	Iterative prototyping	5 days	Mon 3/1/04	Fri 3/5/04
11	☐	Detailed call flows and audio scripts	4 days	Fri 3/5/04	Wed 3/10/04
12	☐	Initial grammar design	2 days	Thu 3/11/04	Sun 3/14/04
13	☐	Stage 3 : Development	17 days	Mon 3/15/04	Tue 4/6/04
14	☐	VoiceXML development and back-end integri	17 days	Mon 3/15/04	Tue 4/6/04
15	☐	QA Planning	3 days	Mon 3/22/04	Wed 3/24/04
16	☐	Recording all audio	2 days	Thu 3/25/04	Sun 3/28/04
17	☐	Initial grammar tuning	3 days	Mon 3/29/04	Wed 3/31/04
18	☐	Usability Testing	4 days	Thu 4/1/04	Tue 4/6/04
19	☐	Stage 4: Launch and Project Sign-Off	42 days?	Thu 4/8/04	Sat 6/5/04
20	☐	Final Report Preparation	9 days?	Thu 4/8/04	Tue 4/20/04
21	☐	Project Presentation Preparation	5 days?	Wed 4/21/04	Tue 4/27/04
22	☐	Project Dissertation Submission	4 days?	Tue 6/1/04	Sat 6/5/04
23	☐	Project Sign-Off	4 days?	Tue 6/1/04	Sat 6/5/04



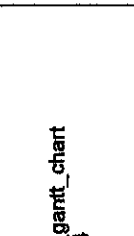
External Tasks
External Milestone
Deadline



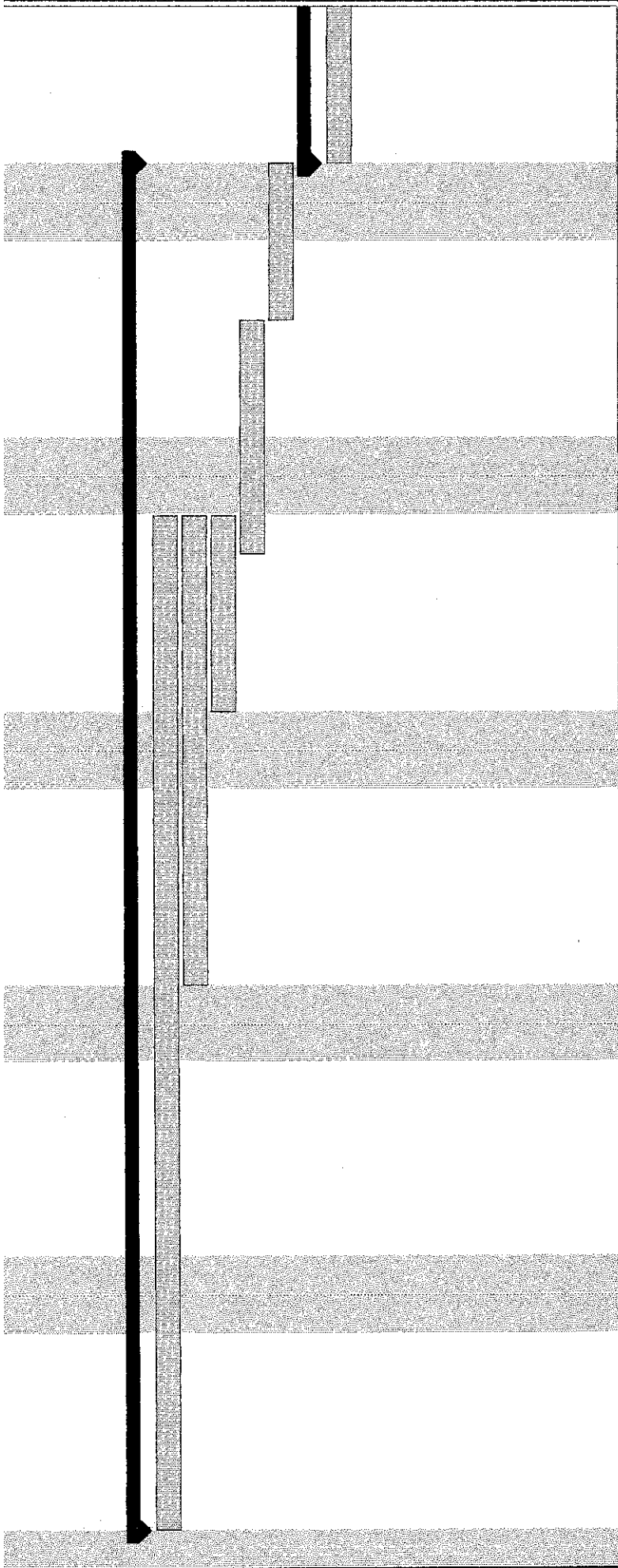
Milestone
Summary
Project Summary



Task
Split
Progress



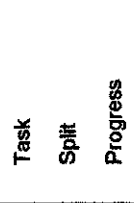
Project: project_gantt_chart
Date: Fri 6/11/04



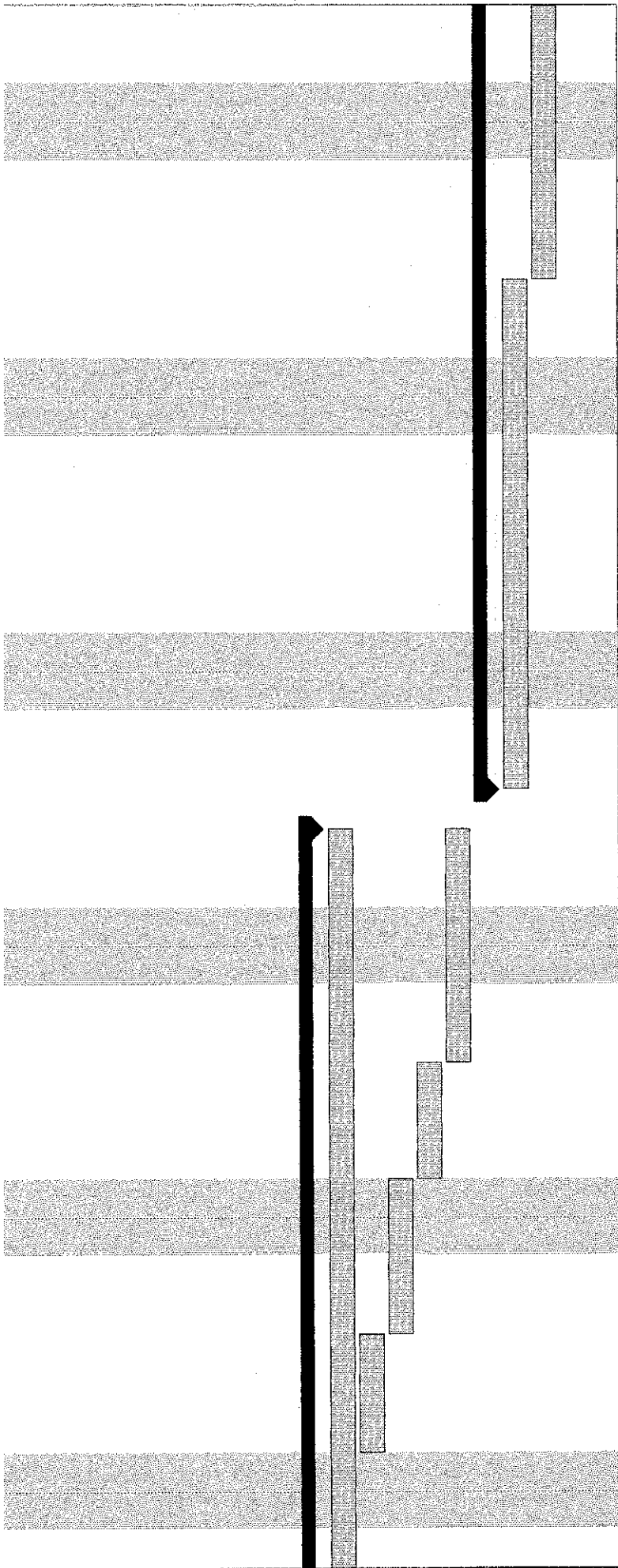
External Tasks
 External Milestone
 Deadline



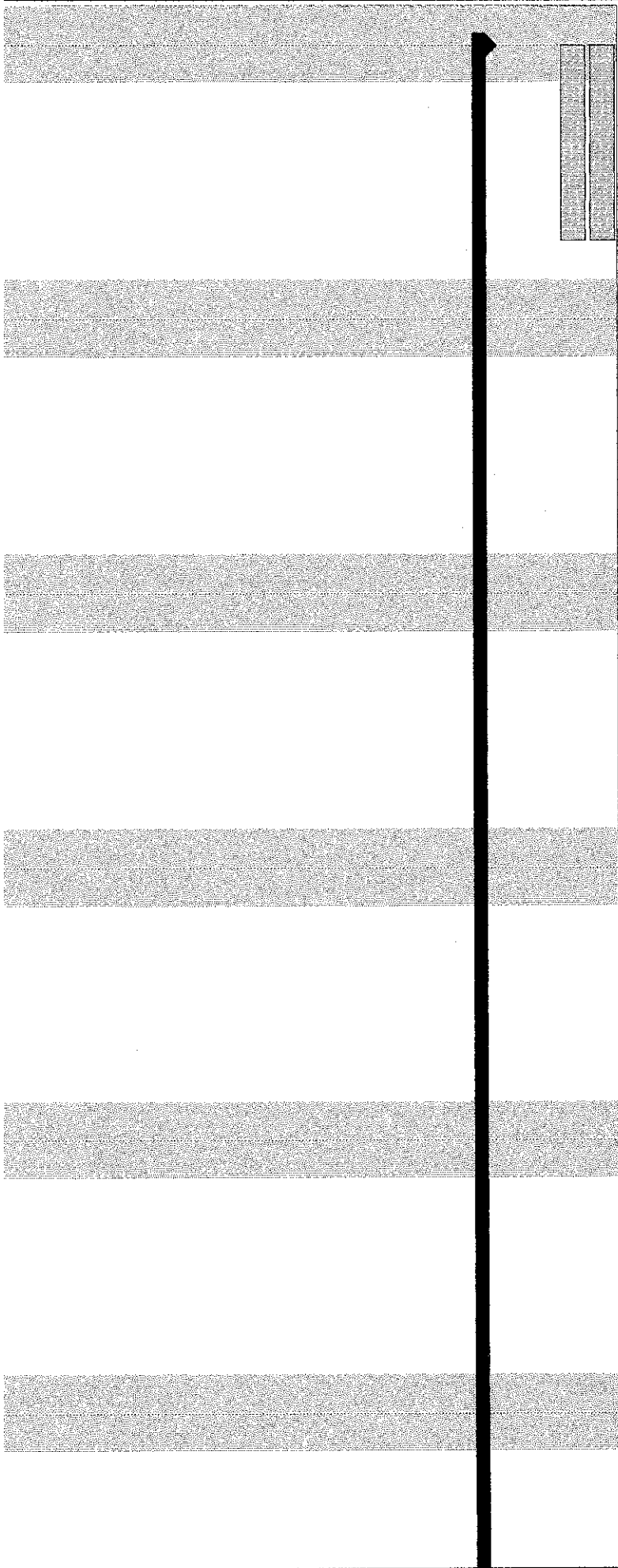
Task
 Split
 Progress



Project: project_gantt_chart
 Date: Fri 6/11/04



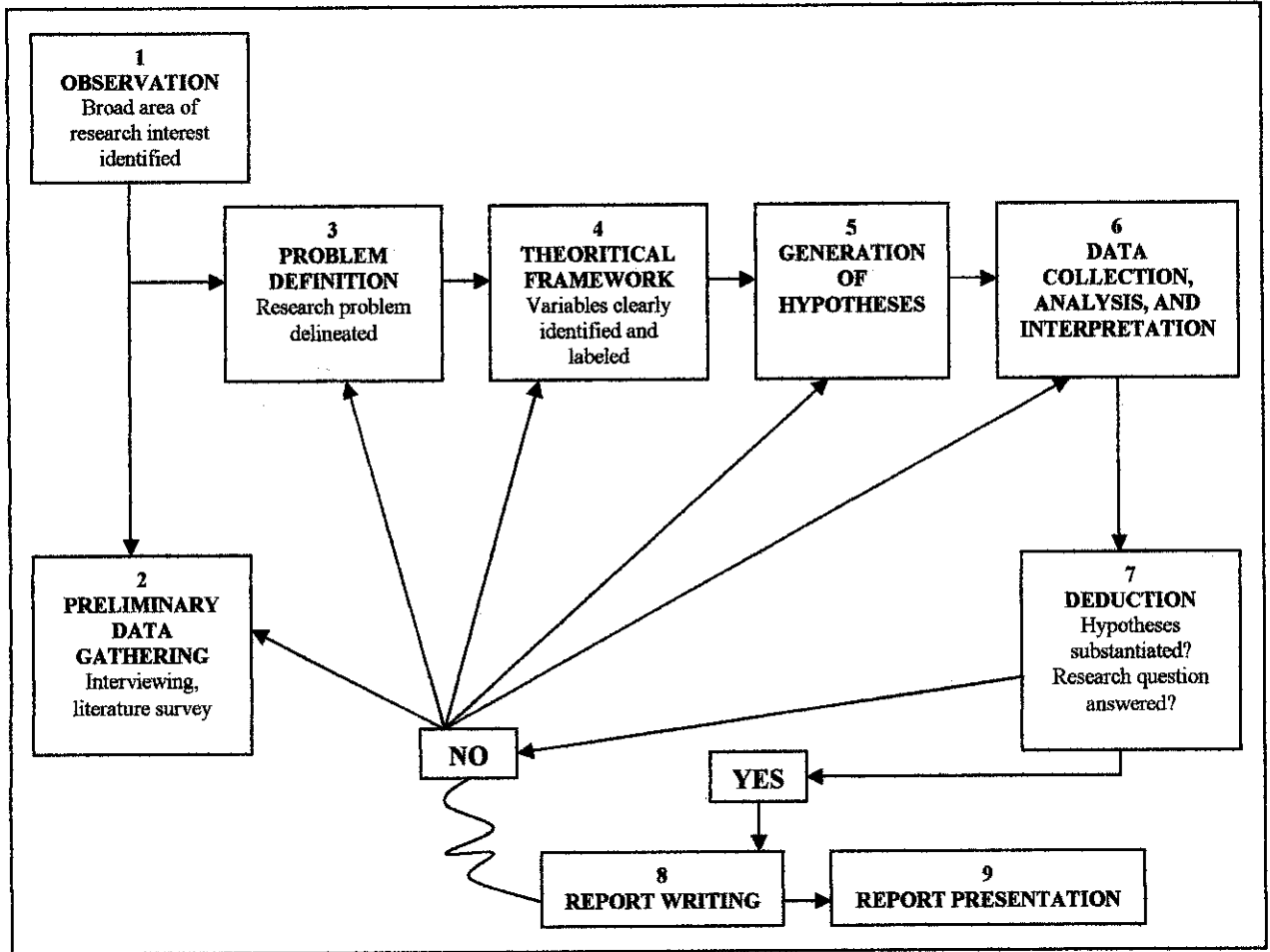
Project: project_gantt_chart
Date: Fri 6/11/04



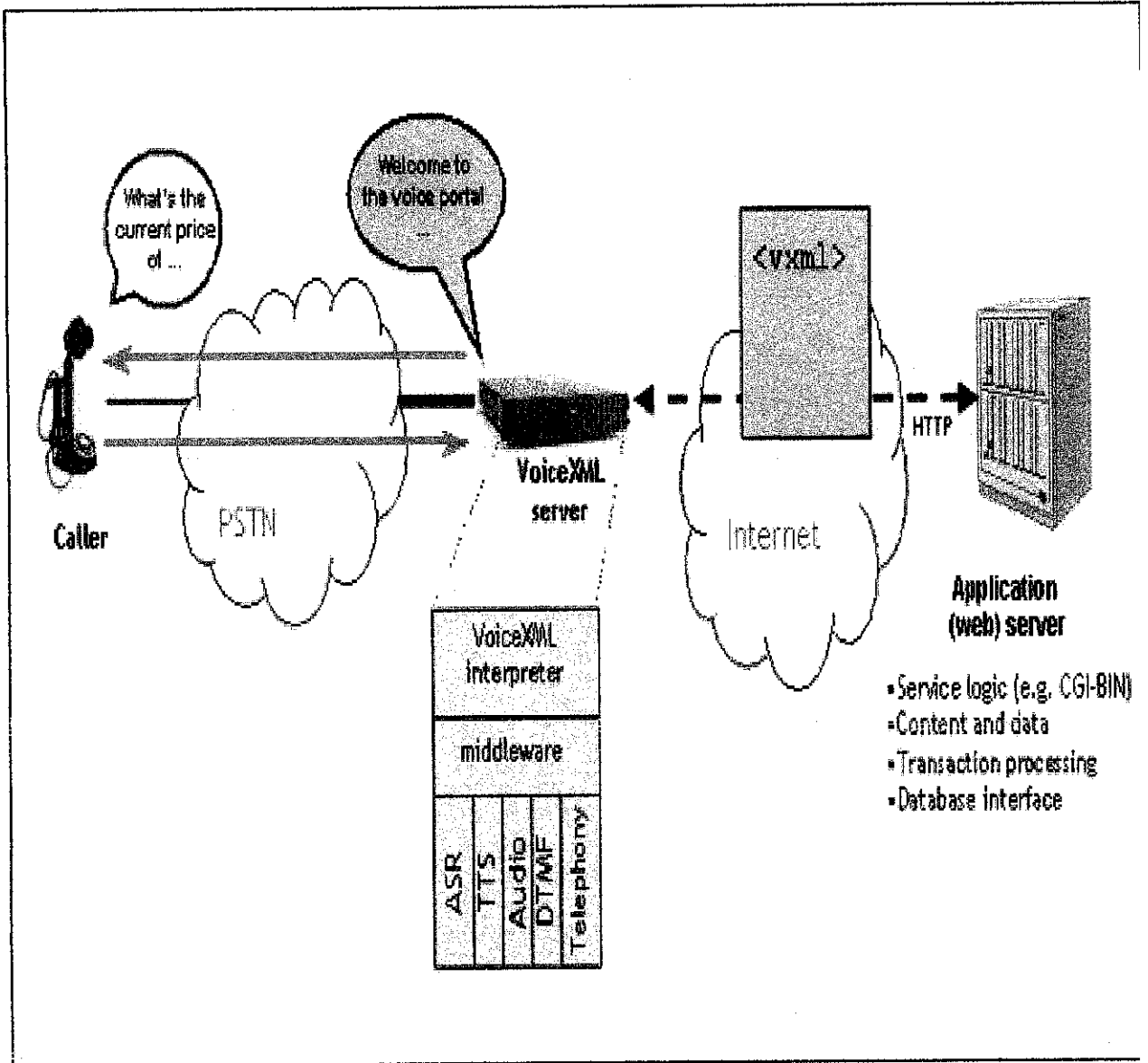
Task
Split
Progress

Project: project_gantt_chart
Date: Fri 6/1/04

APPENDIX B: RESEARCH PROCESS DIAGRAM



APPENDIX C: VOICEXML ARCHITECTURE DIAGRAM



Source: - *What is VoiceXML? Volume 1, Issue 1 - January 2001* By Kenneth G. Rehor

<http://www.voicexmlreview.org/Jan2001/features/Jan2001_what_is_voicexml.html>

Copyright © 2001 VoiceXML Forum. All rights reserved.

APPENDIX D: PROGRAM FLOW CHART

