Fax Engine

for

Superfax

By

Asif Ali

A report submitted to the
Department of Computer Science, Quaid-i-Azam University
as a partial fulfillment of the requirement for the award of the degree of
M. Sc. in Computer Science.

January 2002.

"In The Name Of ALLAH The Most Beneficent The Merciful"

Blessed is the name of our Lord, Mighty, Glorious and Tremendous.

"Our Lord! Condemn us not if we forget, or miss the mark! Our Lord! Lay not on us such a burden as thou didst lay on those before us! Our Lord! Impose not on us that which we have not the strength to bear! Pardon us, absolve us and have mercy on us, Thou art our Protector, so give us victory over the disbelieving folk."

(Al-Baqra)

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: April 3, 2002

FINAL APPROVAL

This is to certify that we have read the project report submitted by Mr.Asif Ali and it is our judgement that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of Master of Science in Computer Science.

COMMITTEE:

1. EXTERNAL EXAMINER
   Dr. Muhammad Mehboob Yasin
   Department of Computer Science
   Ghulam Ishaq Khan Institute
   TOPI

2. SUPERVISOR/INCHARGE
   Ms. Memoona Afsheen
   Lecturer
   Deptt. Of Computer Science
   Quaid-I-Azam University
   Islamabad.

3 INCHARGE
   Mr. Khalid Saleem
   Incharge
   Deptt. Of Computer Science
   Quaid-I-Azam University
   Islamabad.

To...

My Parents, Nani Jan and Amman

ACKNOWLEDGEMENT

First, I owe gratitude to Almighty Allah, the most merciful and compassionate, most Gracious and beneficial Whose favor made it possible for me to accomplish this task assigned to me.

I solute my loving parents for their prayers, support and giving me all the love in the world. I cannot forget my beloved sisters and brothers whose love gives me the courage to achieve more and more.

It is with genuine pleasure that I acknowledge my debt to my teachers and internal supervisor Miss. Memoona Afsheen who provided me personal stimulation support and criticism. Her experience, searching questions and arguments have contribution in designing and developing system and writing this report.

I am also thankful to Dr. Afzal Bhatti, Miss Onaiza Maqbool, Mr. Khalid Saleem, and all other teachers for providing me useful advices during my study.

I am also thankful to my uncles Irshad Ahmed and Mohammad Umer for their kind guidance and advices.

And finally my words cannot say gratitude to my external supervisor Mr. Ashfaq Ahmed (Director operations, Worldtel, Islamabad) for providing me essential and useful guideline in analyzing, designing, and developing the system. I would also like to thank my friends Tahir, Imran Rafiq, M Irfan, M.Imran Bukhari, the little Jin Abdul Qadir (QBDS), Khair Mohammad soomro, Abdul Jabbar memon, Shah nawaz Jiskani and other class fellows for their companionship and moral support during my stay at Q.A.U. I would also like to name some other people Abdur Rehman Cheema, Aman bhai, Sohail bhai and Shahid Latif bhai for their nice company and cooperation during my stay at Q.A.U.

In the end I would like to say that I had never forgot the love for the little Atif. I always missed him during my stay at Q.A.U. Islamabad.

Asif Ali

January 2002.

Project Brief

Project Title:	Fax Engine
Organization:	Supertel
Undertaken by:	Asif Ali
Supervised by:	Madam Memoona Afsheen *Lecturer,* *Computer Science Department,* *Quaid-i-AzamUniversity, Islamabad.*
External Supervisor:	Mr. Ashfaq Ahmed (Worldtel)
Starting Date:	August 28, 2001
Completion Date:	January 31, 2002
Software Used:	Visual C++ 6.0, Dialogic SDK
Operating System:	Windows NT, Windows 2000 Server
System Used: -	Pentium III, 500 MHz

Preface:

Chapter 1: It gives a brief introduction of "Fax Engine". It also contains Objectives and Scope of the system.

Chapter 2: Describes the Requirements of the system. It also contains use case description.

Chapter 3: It gives the Design of "Fax Engine" and description of the classes.

Chapter 4: This chapter named "Implementation" gives the description of platform, language and tools used for the development of the system.

Chapter 5: This briefly describes the testing strategies used for testing. It also contains Test Cases applied and their Results.

Chapter 6: This briefly describes the evaluation of the system and possible future enhancements in the system

Chapter 7: This chapter gives details of the technologies used to develop the system.

Appendix A: It gives description of some data types used in some modules.

Appendix B: It contains the System and Deployment diagrams of the system.

Appendix C: It gives the Class Diagram of the system.

Appendix D: It contains the Use Case Diagram of the system.

Appendix E: It contains Sequence Diagrams of some use cases.

Appendix F: It gives the resources required to develop and test the system.

Table OF Contents

5 System Testing

6 System Evaluation and Future Enhancements

7 Technology

Chapter # 1

Introduction

1.1 INTRODUCTION TO ORGANIZATION

Supertel is a well-established telephone company providing services in 22 cities of Pakistan. Basically Supertel provides telephone service to their customers. For this purpose they issue their customers telephone unit cards and customer can use it anywhere at supertel phone booths. Now Supertel decided to start a new service for their customers i.e. *Store and Forward Fax* (through Internet). This service will be cheaper for the customers because Internet will be used to send fax. This software will be developed under the license of PTCL and will be legal software. The schedule for all the system development process is described in the report and it will take only 16 weeks for the whole development of system.

1.2 SYSTEM OVERVIEW

Fax has become an indispensable part of our everyday life. We use fax for business and for personal communications. In our times when this world has shrunk to become a global village and businesses are communicating daily with their associates around the world, expenses incurred on faxes have become exorbitantly high but unavoidable. In addition, numerous hours and valuable resources are wasted due to undeliverable or incorrectly delivered faxes. According to a recent survey, expenses incurred using fax by businesses are exceeded only by voice calls. This store and forward fax system offers immediate savings on fax expenses as well as a platform for guaranteed delivery.

"Store and forward fax" is the system to provide fax services using Internet. The main purpose of this system is cost reduction. Since telephone lines used for transferring fax cost higher, Internet is used for the same purpose, with low cost. System will maintain accounts for the users who will be able to send fax. The user's account information will be maintained in a database. Interactive voice response will be used to make the system more comfortable and user friendly.

A server will be running on the Supertel office in a city. This server will have a telephone line attached with it where it will be listening to calls. The server's phone number will be given to the customer with his customer ID. The customer, when sending a fax, will first connect to this server by calling server's phone number e.g. 2870327. Customer will be guided by voice response for example "press 1 to send the fax", "Enter

your customer ID", "Enter your password", "Enter your destination fax number" etc. Customer will send the fax to the server and the Fax Engine running on the server machine will store it on the hard disk as a TIFF file. Fax Engine will store this TIFF file's address and the destination city's email address in a database. The Mail Engine, also running on the server machine will be checking the database for any new file. It will take that TIFF file, compose a mail with that TIFF file as attachment and send it to the corresponding destination email address. On the receiving end i.e. on the destination city, the Mail Engine will be checking for any new mail in that city's mailbox. When Mail Engine will find a new mail, it will pop the mail and store the attachment to the hard disk. The Mail Engine will inform the Fax Engine about this fax mail by adding a record in a table in the database. This record will contain the TIFF file address and the destination fax number. Fax Engine, that will be checking the database for new file and will take this file and send it to the destination fax machine. Whether this fax will be sent to the destination fax machine successfully or not, Fax Engine will tell this to the Mail Engine by updating the database. Now Mail Engine will compose an acknowledgement mail and send it back to the fax sender station. From there acknowledgement will be sent to the source fax machine about the status of the fax i.e. success or failure. Customers' information as well as fax information will be stored in the database. The process sequence of the system is described in appendix B (fig. 1.2.) and the physical architecture is described in appendix B (fig. 1.1).

1.3 FAX ENGINE

This is a group project and three students are working on it. I am assigned the task of developing a fax engine that will control the send and receive of fax.

1.3.1 PROBLEM DEFINITION

Fax Engine is an important module of **Super Fax**, the store and forward fax system. It fulfils one of the major functionality of the system of sending and receiving faxes to and from the fax machine.

When a fax machine sends a fax document to the server, Fax Engine will receive it and converts it into a TIFF file. It will then update the database to inform the Email Engine of the arrival of a new fax. This Email Engine, developed by one of my project mates, will be examining the database continuously for any incoming fax. Email Engine will get the TIFF file form the specified location and send it to the destination end server. Now, when email will arrive at the other side, the Email Engine will update the database to indicate the Fax Engine of the arrival of a pending fax. Fax Engine will be examining the database there continuously for any incoming email. It will get the attached TIFF file from the specified location and will send the fax to the destination machine.

1.4 OBJECTIVES AND SCOPE

The idea behind Fax Engine is to develop an application based on Dialogic voice and fax board that will be used to handle the sending and receiving of faxes in Superfax, the store and forward fax system. This engine will be synchronized with IVR (Interactive voice response) and Email Engine.

1.4.1 OBJECTIVES

The objectives of the above-described modules are:

- ✓ To receive faxes from the fax machine
- ✓ To store the incoming fax into TIFF format so that it can be sent over the Internet
- ✓ To send the fax to the destination fax machine
- ✓ To get the destination fax machine address so that the fax can be sent to it
- ✓ To provide Email Engine, the email address corresponding to the destination city code so that it can send email to the destination Email Server

1.4.2 SCOPE

The Scope of the Fax Engine is as follows:

- ✓ Fax Engine will establish connection between fax machine and the Application to send and receive fax
- ✓ Fax Engine will serve the Email Engine by providing it the file that it will email to the destination side Email server

- ✓ Fax Engine will store the incoming fax into TIFF format so that it can be sent as an email attachment over the Internet
- ✓ Fax Engine will fax the TIFF file to the destination fax machine

Chapter # 2

Requirement Analysis

2.1 INTRODUCTION

Analysis involves understating the problem, establishing the services the system should provide, and the constraints under which it must operate. Analysis is one of the very vital activities in the development of a project. All the later stages depend on analysis. Infect analysis works as a base for all the later stages of the project. A good analysis leads to a good final product.

2.2 REQUIREMENT DEFINITION

A software requirement definition is an abstract description of the services, which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system. It should not be concerned with system design characteristics.

System requirements may be,

- Functional
- Non functional

a) FUNCTIONAL REQUIREMENT

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

b) NON-FUNCTIONAL REQUIREMENT

These are the constraints on services or functions offered by the system. They include timing constrains, constraints on the development process, standards and son on.

2.2.1 FUNCTIONAL REQUIREMENT OF THE SYSTEM

Following are the functional requirements of the system:

- ✓ Communication between Fax machine (sender) and the Fax server should be established.
- ✓ The system should differentiate among different types of tones e.g.
  - International call tone

- Fax machine tone
- Busy tone
- Local phone call tone etc.

✓ The system should receive the incoming fax at the server

✓ The system should receive single page faxes as well as multiple page faxes

✓ The system should convert the incoming fax to TIF format

✓ The system should be able to fax TIFF file to the destination fax machine

✓ The system should release the connection at the end of the communication

2.2.2 NON FUNCTIONAL REQUIREMENT OF THE SYSTEM

Following are the non-functional requirements of the system,

✓ Object oriented design methodology should be followed for the development for system design.

✓ The system should be developed using Dialogic SDK and VC++ 6.0

✓ Windows NT Server 4.0 should be used for the system development.

✓ According to prefixed schedule of the project a complete operational system with complete documentation is expected in the end of Jan 2002.

✓ Full documentation and the complete code will be given to the organization

2.3 USE CASE MODELING

A use case is a modeling technique used to describe what a new system should do or what an existing system already does. The primary components of use cases are:

- Use cases
- Actors
- The system model

a) Use Case

The functionality of the system is represented by a number of use cases and each use case specifies a complete function. A use case must always deliver some value to an actor, the value being whatever the actor wants from the system.

b) Actor

The actor is an external entity that has an interest in interacting with the system. Often it is the human user of the system, but it can also be another system or some kind of hardware device that needs to interact with the system.

An actor is a role that a user plays with respect to the system.

2.3.1 USE CASES OF THE SYSTEM

Following are the use cases of the system:

- Establish Connection
- Detect Tone
- Send Fax
- Receive Fax
- Convert Fax to TIF File
- Store Fax
- Get Fax Number
- Get Email Address

2.3.2 ACTORS OF THE SYSTEM

Following are the actors in the system:

- Fax machine
- Fax card
- Customer
- Database

Use Case Diagram:

Use Case diagram is given in Appendix D (fig. 2.1) .

2.4 USE CASE DESCRIPTION

Here is the brief description of the above mentioned use cases:

1. **Establish Connection:**

   Objective:

   The objective of this use case is to establish connection between Fax machine (sender) and the application

   Each customer is given an account number, she/he dials the account number and the fax number of the destination fax machine. The system validates the customer account information and destination fax machine address, if o.k., the connection establishes.

2. **Detect Tone:**

   Objective:

   Objective of this use case is to separately identify the different types of tones.

   When the customer dials his account number and the destination fax machine 's address, the sender fax machine generates a tone that is detected at the server. If the tone detected is not a Fax machine's tone then the system does not respond it.

3. **Send Fax:**

   Objective:

   Objective of this use case is to send fax from the server (computer) to the fax machine

   After receiving the fax at the server from the fax machine, the destination fax machine's address is found and the fax is then sent to the destination fax machine

4. **Convert Fax to TIF File:**

   Objective:

   Objective of this use case is to store the incoming fax as an image file so that it can be sent as an email attachment to an email sever over the Internet.

   When the customer sends the fax, it is first received at the server where it is converted to TIF format.

5. **Get Destination Fax Address:**

   Objective:

   Objective of this use case is to get the destination fax machine 's address.

When customer dials the destination machine address and his account number, all these digits are received as a single long string at the server then the destination fax address is extracted from this string

6. **Retrieve Email Address:**

   Objective:

   Objective of this use case is to retrieve email address corresponding to city code of the destination fax machine's address.

   Email engine sends an email with an attached TIF file to the email server corresponding to the destination Fax machine's city. The email is retrieved from the database for the Email Engine.

7. **Store Fax:**

   Objective:

   Objective of this use case is to store the incoming fax in a folder on the hard disk

   After receiving the fax, Fax Engine stores it in a folder on the hard disk and updates the address of this folder in the database.

8. **Receive Fax:**

   Objective:

   Objective of this use case is to receive the incoming fax.

   Fax Engine receives every incoming fax at the server and updates its address to the database.

2.5 CLASS IDENTIFICATION

By analyzing the use cases, following classes are identified:

CFax	CConnect
CChannel	CPhaseHandler
CFaxSender	CFaxReceiver
CCallAnalyst	CDatabase
CFile	CFaxEngineDialog

Chapter # 3

System Design

3.1 Introduction

Design is the first step in development phase for any engineered product or system. It may be defined as:

"*The process of defining a device, a process or a system in sufficient detail to permit its physical realization*"

It is on the highest priority among the phases that software engineer mostly emphasizing. The importance of design can be stated with a single word "Quality". Any fault or flaw left during this phase not only become problematic in the end but also reduces the efficiency and reliability of resulting software largely. Software design serves as foundation for the development and maintenance steps. Without design we risk building an unstable system that will fail when small changes are made.

There are two approaches to design a system

- ✓ Structured approach.
- ✓ Object oriented approach.

The **Fax Engine** has been designed following object oriented approach and unified modeling language.

Why Object Oriented approach is used?

The methodology chosen for the development of my project is object oriented. The first reason of choosing this methodology is of course the scientific nature of the project. The most appropriate methodology that can be thought of is object oriented. The second reason is quite obvious as well, when you talk about Visual C++ the first thing that comes to your mind is its object-oriented nature. It makes the understanding of the problem more realistic and the developer is always very close to the implementation while designing the development of system.

There are some other reasons, which compel us to use this approach.

- ✓ It become more problem oriented instead of system oriented
- ✓ It breaks up the system into sets of classes each having simple operations
- ✓ The dynamic aspect of real word is also captured.
- ✓ Object oriented design simplifies the system and helps in system maintenance

The first step in Object oriented design is the construction of Class diagram. Initial classes are identified during object-oriented analysis and these classes are refined and attributes and member functions for each class are identified in this phase. Moreover relationship among various classes is also identified in this phase.

3.2 CLASS DIAGRAM

Class diagram is given in Appendix C.

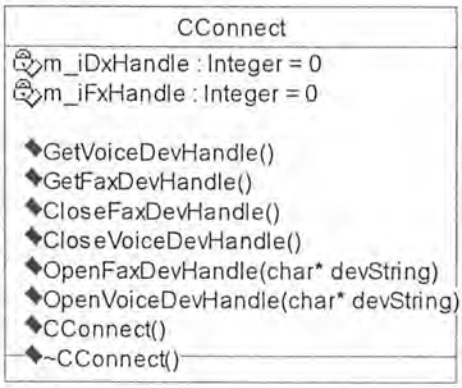
3.3 DESCRIPTION OF CLASSES

Description of classes used in the project is given below:

CConnect Class

Description:

This is the most important class of Fax Engine. Fax Engine uses Dialogic Fax card to send and receive faxes. This class is used to open handle to the Fax Card resource to make use of it.



Operations:

Dialogic VFX40/ESC is a voice and fax board.The VFX/40ESC board consists of a D/41ESC baseboard and a FAX/40E daughterboard that provides 4-channels of enhanced voice and fax services in a single slot.

OpenVoiceDevHandle(char* devString)

This method is to open handle to voice board resource i.e. D/41ESC. Parameter passed to the method is the device name. There can be multiple boards on a single system and each board can have maximum four telephone lines connected to it. So the device name is represented as: dxxxBiCj where ' I ' represents the board number and ' j ' represents the channel number.

OpenFaxDevHandle(char* devString)

This method is used to open handle to fax board resource i.e. FAX/40E. Device name is passed as a parameter to this method.

CloseVoiceDevHandle()

This method is used to close the Dialogic voice channel device previously opened with OpenVoiceDevHandle() method. It releases the handle and breaks any link the calling process has with the voice device channel through this handle

CloseFaxDevHandle()

This method is used to close the Dialogic voice channel device previously opened with OpenFaxDevHandle(). It releases the handle and breaks any link the calling process has with the FAX device channel through this handle

GetVoiceDevHandle()

This method is used to get the voice device handle previously opened with OpenVoiceDevHandle().

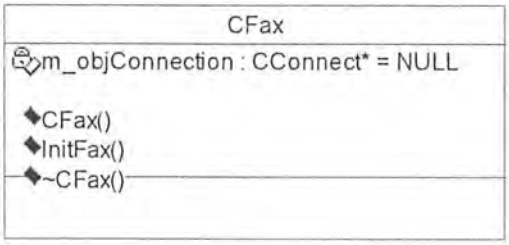
GetFaxDevHandle()

This method is used to get the fax device handle previously opened with OpenFaxDevHandle().

CFax Class

Description:

This is the base class for both CFaxSender and CfaxReceiver and it contains CConnect class type object so that both (Fax sender and Receiver) can make use of Fax card. Basic initializations for sending and receiving fax are performed in this class.



Operations:

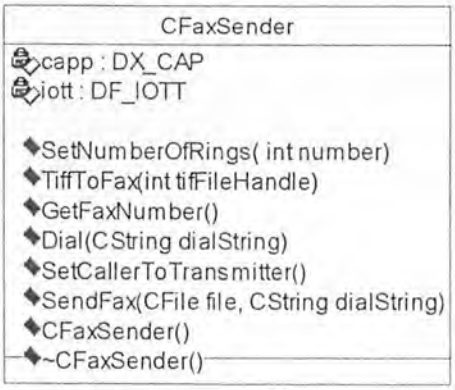
InitFax()

When an object of CFax class is created this function is called in its constructor to perform basic initializations.

CFaxSender Class

Description:

This class is responsible for faxing the TIF file from the server (computer) to the fax machine.



Operations:

SetNumberOfRings(int number)

When a fax is sent from the server (Computer) to a fax machine, Fax Engine dials the fax number of the destination fax machine to indicate the recipient of the incoming fax. This method is used to set the number of rings before the fax is printed out of the dastination fax machine.

TiffToFax(int tifFileHandle)

This method is used to initialize DF_IOTT data structure so that the tif file can be sent as a fax to the fax machine.

GetFaxNumber()

This method is used to get the destination fax number from the database to send the fax to it.

Dial(Cstring dialString)

This function is used to dial the destination fax machine number.

SetCallerToTransmitter()

This method sets the initial fax state. It is used to set the initial fax state of the specified fax channel to a CALLER. This method is issued prior to issuing the first send function of a fax session to maintain the correct fax state of the application.

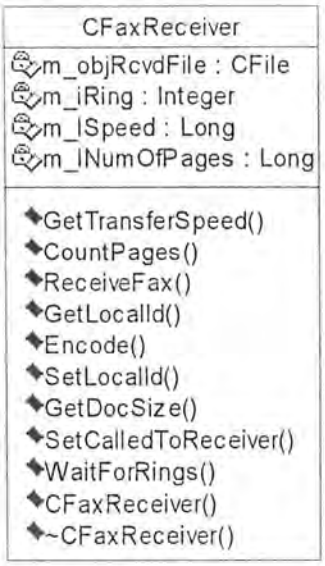
SendFax(CFile file, CString dialString)

This method is used to actually transmit the file from the server (Computer) to the destination fax machine. This method handles the transmission of a single as well as multiple pages of fax from the server (computer) to the fax machine.

CFaxReceiver Class

Description:

This class is responsible for receiving fax from the fax machine and of storing the received fax on the hard disk in TIFF-F format.



Operations:

GetTransferSpeed()

This method is used to get the fax transfer speed (in baud) of the last received page. This information is available after Phase B is completed.

CountPages()

This method returns the number of transferred fax pages during the current fax call.

GetDocSize()

This method returns the size of the fax received. This function is issued after successfully receiving the fax at the server.

Encode(unsigned long scheme, void *valuep)

This method is used to encode the incoming fax data. The incoming fax data can be encoded in any of the following schemes:

- MH – Modified Huffman encoding
- MMR – Modified Modified Read encoding

SetCallerToReceiver()

This method sets the initial fax state. It is used to set the initial fax state of the specified fax channel to be a CALLED. This method is issued prior to issuing the first receive function of a fax session to maintain the correct fax state of the application

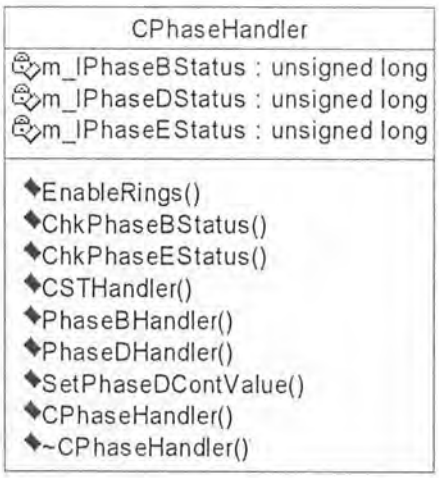
WaitForRings(int numOfRings, int hookStat, int timeout)

This function waits for a specified number of rings and changes the hook state after rings are detected.

CPhaseHandler Class

Description:

This class is used to handle various events generated during processing of different phases of sending and receiving fax. Reception and transfer of fax takes place in five phases.



Operations:

ChkPhaseBStatus(int fxHandle)

In Phase B negotiation between the two stations, to determine their RECEIVER and TRANSMITTER state as well as other fax parameters, take place. This method is used to check how the call is progressing during phase B.

ChkPhaseEStatus(int fxHandle)

The Phase E status values indicate errors during the course of a fax transmission /reception. This method returns Phase E information, describing errors that occurred during the fax transmission/reception.

SetPhaseDContValue(int fxHandle)

This method is used to set the inter-page Phase D continuation value for transmitting TIFF files.

PhaseDHandler(unsigned long x)

The Phase D status values indicate the status of a fax transmission/reception. This method is used to handle different events generated during phase D and to check the phase D status.

PhaseBHandler(unsigned long x)

This method is used tohandle different events generated during phase B.

EnableRings(int state, int dxHandle)

This method enables or disables the detection of ring events.

CFile Class

Description:

Fax Engine receives each incoming fax on the server and stores it in TIF format on hard disk. This class is used to store information about the received fax in TIF file.

Operations:

SetId()

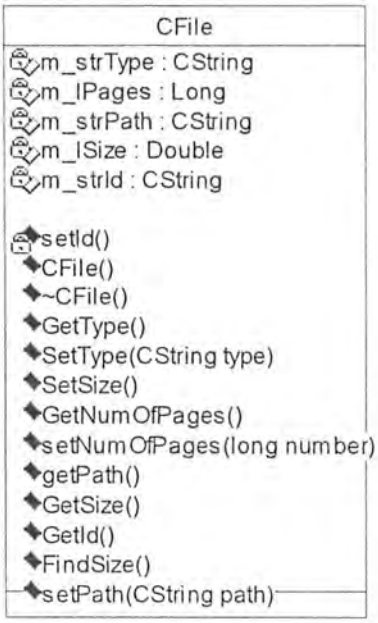
Each incoming fax is stored on the hard disk with a unique id. This method is used to assign a unique id to each incoming fax.

GetId()

This method is used to get the id of the file from a CFile object.

SetType(CString type)

This method is used to set the type of the file in a CFile object.



GetType()

This method is used to get the type of the file from the CFile object.

FindSize()

This method is used to find the size of the received file.

SetSize()

This method is used to set the size of the file in a CFile object.

GetNumOfPages()

This method is used to get the number of pages of a received fax from the CFile object.

SetNumOfPages()

This method is used to set the number of pages of a received fax in a CFile object.

SetPath (CString path)

Each incoming fax is stored on the hard disk. This method is used to set the path of this file.

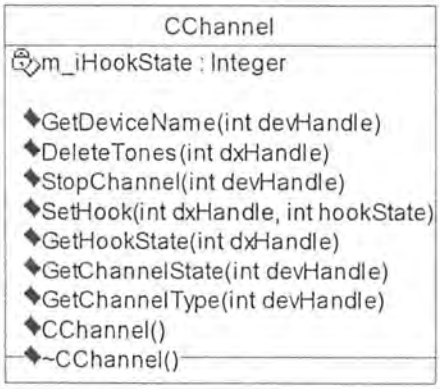
GetPath()

This method is used to get the path of the stored file.

CChannel Class

Description:

This class is used to find the current status of the channel and to set the hook state ON and OFF to send and receive fax respectively.



Operations:

GetDeviceName(int devHandle)

This method is used to get the board name.

DeleteTones(int dxHandle)

This method removes all user-defined tones previously added to the channel.

StopChannel(int devHandle)

This method forces termination of currently active I/O functions on a channel. It forces a channel in the busy state to become idle.

SetHook(int dxHandle, int hookState)

This function provides control of the hookswitch status of the specified channel. A hookswitch state may be either *on-hook* or *off-hook*.

GetHookState(int dxHandle)

This method is used to get the current hook state of the channel.

GetChannelType(int devHandle)

This method is used to get the type of channel.

CCallAnalyst Class

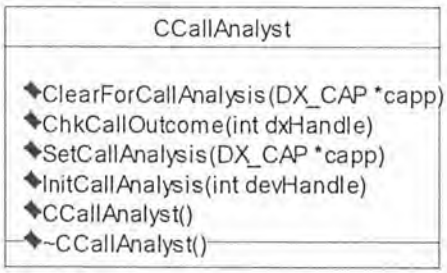
Description:

Call Analysis is used to determine the progress of a call after dialing into the Public Switched Telephone Network (PSTN), where a wide variety of signal possibilities can occur. There are two types of call analysis:

1- Basic Call Analysis

2- Perfect Call Analysis

We can't detect fax machine's tones in Basic call analysis; this class is used to enable perfect call analysis.



Operations:

InitCallAnalysis(int devHandle)

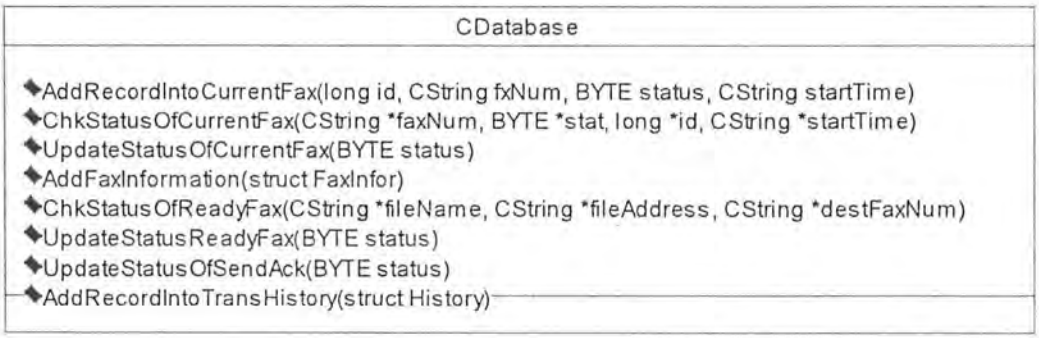
This method is used to initiate perfact call analysis. This method initializes and activates PerfectCall Call Analysis on the channel.

ChkCallOutCome(int dxHandle)

This method is used to determine the call outcome after dialing the fax number.

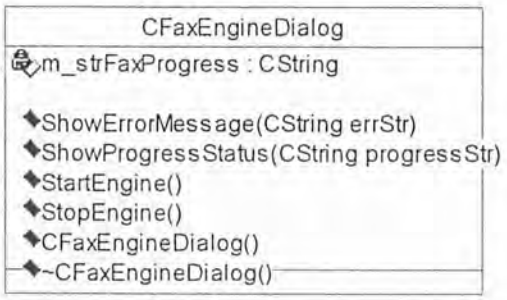
CDatabase Class:

Fax Engine perform various tasks based on certain values of database tables. This class is used to interact with database and to check some tables continuously.



CFaxEngineDialog Class:

This is an interface class and is used to define a dialog that will show the progress of fax in PSTN during send and receive operations. It is also used to start and stop Fax Engine.



Sequence Diagram

Sequence diagrams are given in appendix E.

Chapter # 4

System Implementation

4.1 INTRODUCTION

The development phase of the project starts after the design. During this phase a design in the form of shapes and text is converted into working software. The software is developed in such a way so that it can meet the requirements and specification of the users. The implementation phase of any system is concerned with the tools used in the development work and the components used to implement the system. This chapter explains all the steps taken for the development of the software.

4.2 TOOL SELECTION

When it comes to implementation, the first thing is to choose appropriate tools for the development of the software. The decision of tool selection is important and crucial. The decision should be wise enough to avoid further difficulties and problems in the later stages of the development life cycle. Therefore, there are two things to be catered:

1. Operating System Selection
2. Programming Language Selection

4.2.1 OPERATING SYSTEM SELECTION

An operating system should be chosen that fulfils the requirements of the software and is available to the users easily. Operating System selection was not a difficult decision at all. As we are going to develop the system using Dialogic SDK that is written for Windows NT so the best operating system for the development of this software can be Microsoft Windows NT. Other advantages are:

- ✓ Windows provide considerable advantages to both users and programmers over the conventional MS-DOS environment
- ✓ Windows provide a large library of functions that make the programming task easy.
- ✓ Windows provide consistent graphical user interface in which graphics provide a utilization of screen.
- ✓ Windows support multitasking and multithreading and is independent of underlying hardware.

4.2.2 PROGRAMMING LANGUAGE SELECTION

The selection of an appropriate language is of vital importance. A language should be selected keeping in view both the requirements and the nature of the project. Whatever language is selected it should support the desired programming tasks. I have selected VC++ to develop the system.

Why Visual C++?

- ✓ It is a pure Object Oriented language.
- ✓ It provides programming productivity and lets the programmer to focus and emphasis on the business language.
- ✓ It supports the creation of attractive and advanced Graphical user Interface.
- ✓ As we have discussed earlier that we have to use Dialogic SDK, for the development of this software, that consists of certain APIs which are written in Visual C++ so it is easy to use these APIs in VC++ than in any other language.
- ✓ Programs developed in VC++ execute efficiently.
- ✓ Scientific nature of our project demands certain properties in the developing language and VC++ meets all those demands.
- ✓ It is a non-functional requirement from the organization that the system should be developed in VC++.

Chapter # 5

System Testing

5.1 INTRODUCTION

Testing is a process of executing a program with the intent of finding an error.

A good test case is one that has a high probability of finding an as-yet undiscovered error

A successful test is one that uncovers an as-yet undiscovered error.

5.2 TESTING STRATEGIES

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests. Different testing strategies may be adopted depending on the type of system to be tested and the development process used.

Top Down Testing: Where testing starts with the most abstract component and works downwards.

Bottom-up Testing: Where testing starts with the fundamental components and works upwards.

5.3 OBJECT ORIENTED TESTING STRATEGIES

To adequately test object oriented system three things must be done

- ✓ The definition of testing must be broadened to include error discovery techniques applied to OOA ad OOD model.
- ✓ The strategy of unit and integration testing must change significantly.
- ✓ The design of test cases must account for the unique characteristics of object oriented software.

5.3.1 UNIT TESTING

In unit testing different modules of developed system were tested independently. The purpose of this testing was to determine that each individual module is functioning properly and to locate logical and coding errors. Initial tests focus on each module individually, assuring that they function properly as units. Hence the name unit testing.

The advantage of this type of testing is that errors if any, an be dealt with at this level, rather than when the system is finally tested. Thus resulting in much easier error detection.

5.3.2 INTEGRATION TESTING

After unit testing, integration testing was conducted., Several errors were reported and corrected. The main objective is to determine the irregularities in the developed system. Integration testing addresses the issues associated with the dual problem of verification and program construction.

5.3.3 BLACK BOX TESTING

In this strategy we test whether the developed application conforms to its specification or not . Black box testing attempts to find out errors in the following categories:

- ✓ Incorrect or missing functions
- ✓ Interface errors
- ✓ Errors in data structures or external database access
- ✓ Performance errors
- ✓ Initializations and termination errors

5.4 BLACK BOX TESTING OF THE FAX ENGINE

Fax Engine is divided into two parts:

- ✓ Fax Receiver
- ✓ Fax Sender

Unit testing as well as integrate testing have been applied to both modules and found them working properly. Following are the test cases and their results under which the system have been tested.

5.5 TEST CASES

Test case # 1

Purpose	To test whether connection between the fax machine and the fax engine established or not.
a)	
Value	2870329 (Fax machine dials telephone line's number to which the Fax Card is connected)
Result	Connection established

b)

Value 2870329

Result Connection error report

Test Case # 2

Purpose To test whether the fax machine rings or not when its number is dialed from the computer.

When the fax machine is not busy

a)

Value 2870328 (Fax number)

Result fax machine rings

b)

Value 287038

Result No ring back received i.e. the fax machine does not rings due to some problem with telephone line

When the fax machine is busy

c)

Value 2870327 (Fax number)

Result Busy tone detected

Test Case # 3

Purpose To check whether a single page of fax is received on the computer or not.

Value A single page of fax was sent from the fax machine to the computer (Server).

a) Result The connection established and the page is received on the computer properly

b) Result The connection established but the page is not received properly due to some noise in the line.

Test Case # 4

Purpose	To check whether multiple pages sent from the fax machine are received on the computer properly or not .
Value	Five pages were sent from the fax machine to the computer (Server)
a) Result	The connection established and all the pages were received on the computer properly.
b) Result	The connection established but the pages were not received properly due to some noise in the line.

Test Case # 5

Purpose	To check whether a single page TIF file can be sent from the computer to the fax machine or not.
Value	A single paged TIF file was sent from the computer to the fax machine.
a) Result	The file was successfully received at the fax machine as a fax.
b) Result	The file was not received successfully at the fax machine because it could not be opened successfully.
c) Result	The file was not received successfully at the fax machine because of noise in telephone line.

Test Case # 6

Purpose	To check whether a multi-paged TIF file can be sent from the computer to the fax machine or not.
Value	A multi-paged TIF file was sent from the computer to the fax machine.
a) Result	The file was successfully received at the fax machine as a fax.
b) Result	The file was not received successfully at the fax machine because it could not be opened successfully.
c) Result	The file was not received successfully at the fax machine because of noise in telephone line.

Test Case # 7

Purpose	To check whether the database is updated or not, when a fax is received or send.
Value	A fax is received on the computer from the fax machine.
Result	As soon as the fax was received the, Fax receiver updated the database and Fax sender sent the fax to the destination machine.

Chapter # 6

Evaluation & Future Enhancements

6.1 System Evaluation

On the basis of above test cases it is evaluated that:

- To send or receive fax, the connection between the fax machine and the server should be established. When a fax machine dials the number of the fax card at the server, it generates signals and CNG tone is detected. If the connection is not established then this tone can not be detected on the server. It has been tested this tone is received properly on the server. This shows that the communication between fax machine and server (computer) establish properly.

- During the send or receive fax process the called fax machine may be busy, in such a case the calling fax machine receives a busy tone. It has been tested that the Fax Engine receives this busy tone properly. It shows that it is working properly

- To receive a fax, from a fax machine, on the computer is a fundamental task performed be the fax Engine. It has been tested that the Fax Engine receives single as well as multiple pages of fax properly.

- This Fax Engine is designed to be used in a store and forward fax system. So every incoming fax on the server is to send forward to another fax machine. To meet this requirement all incoming faxes are stored on the hard disk in TIFF files. It has been tested that the Fax Engine stores the incoming fax in a TIFF file on the hard disk accurately.

- It has been tested that the Fax Engine sends the TIFF file as a fax to the destination fax machine accurately.

- It has also been tested that at the end of the receive or send fax operation, the connection is released always.

This evaluation shows that the system (Fax Engine) meets all requirements and is working properly.

6.2 FUTURE ENHANCEMENTS

This system is working properly according to the organization's requirements. However, in future, following enhancements can be made to the system

- ✓ In future, system can be enhanced to send the fax from desktop to the fax machine.
- ✓ At this time, only Tiff files can be sent as a fax to the fax machine, in future the system can be enhanced to send the other files e.g. document files, .jpg files etc. as faxes to the fax machine.
- ✓ The system can be enhanced to provide Email to fax service to the users.
- ✓ Tiff files will be stored in encrypted form to ensure the privacy of fax documents.
- ✓ If there is some noise in telephone line, fax is not received properly. The system can be enhanced to receive faxes properly even if the line is noisy.

Chapter # 7

Technology

7.1 INTRODUCTION

As with many messaging technologies, faxes can be handled real-time or store-and-forward.

Real-time Fax

Real-time faxing connects the sender directly to the end user's receiving machine or system, and faxes are deposited in the end user's real or electronic inbox.

Store and Forward Fax

Store and forward faxing uses a server - local or remote - to receive the faxes, for later pickup or forwarding to the user's inbox on demand. Most of the Internet fax technologies use store and forward at some point.

7.2 STANDARDS

The ITU (International telecommunication union) and IETF (Internet Engineering Task force) are working together to continue to evolve both the real-time Fax over IP (FoIP) network standard (T.38) as well as the store and forward Fax over IP network standard (T.37). Both T.37 and T.38 were approved by the ITU in June 1998.

7.2.1 REAL TIME FOIP STANDARD (T.38)

T.38 defines the protocol for real-time delivery of fax over IP (FoIP). Any real-time protocol over IP that bridges the call path must meet requirements of the T.30 protocol (the protocol for standard fax calls over PSTN). To solve this problem, ITU-T Study Group 8 developed the T.38 protocol. Transporting a fax using T.38 takes only a half-duplex channel and 14.4 kbps plus packet overhead. Transporting a fax using the G.711 channel takes a full duplex 64 kbps plus packet overhead.

T.38 is modeled as a smart T.30 interpreter. It executes extensive training, signaling, and data exchange with T.30 to determine the line quality on a PSTN network. This is meaningless with a packet network, since the IP packets can take any available route. Gateways at each end execute full T.30 for communication with fax machines. However, all the data is not transferred over IP. While the fax machine sends the entire

CNG/CED-type tones for signaling, the gateways using T.38 only exchange octets that indicate whether they've succeeded or failed at detecting tones.

7.2.2 STORE AND FORWARD FOIP STANDARD (T.37)

T.37 merely defines the format in which fax is to be delivered as an e-mail attachment. With T.37, the fax is sent over IP as e-mail attachment and delivered to the destination over the public switched telephony network (PSTN) by the gateway closest to the destination. While T.37 allows for cost savings through toll arbitrage, from a user's perspective it is a store and forward model and is not real time. You can't immediately receive confirmation that the fax was successfully delivered. Although a sender might claim a fax "has been sent," the T.37 model doesn't let you automatically assume a confirmation of delivery has been sent. If an en route e-mail server happened to be down (perhaps due to some worm or virus attack); the confirmation won't go through.

7.3 PSTN FAX CALL PROCEDURE

Fax machines in common use today implement the ITU recommendations T.30 and T.4 protocols. The T.30 protocol describes the formatting of non-page data, such as messages that are used for capabilities negotiation. The T.4 protocol describes formatting of page Tiff data. T.30 and T.4 have evolved substantially over time and are now quite complex because they attempt to describe the behavior of an evolving set of fax machines. The timing related to the message interaction and phases of the call is critical and is one of the major causes of problems in the transmission of Fax over Packet networks.

ITU (International Telecommunication Union) T.30 and T.4 fax protocols specify five phases for the flow of a fax session. These five phases are:

- Phase A – Fax call establishment (Begin fax session)
- Phase B – Control and capabilities exchange
- Phase C – Message transmission
- Phase D – End of page and Multi-Page signaling
- Phase E – Fax call release – Disconnect (End of fax session)

Phase A

The Fax call is established, either through a manual process, where someone dials a call and puts the machine into fax mode, or by automatic procedures, where no human interaction is required. In both cases, the answering fax machine returns an answer tone, called a CED, which is the high pitched tone that you would hear when you call a fax machine. If the call is automatically dialed, the calling station will also indicate the fax call with a calling tone (CNG), which is a short periodic tone that begins immediately after the number is dialed. These tones are generated to allow a human participant to realize that a machine is present on the other end call. These tones are sometimes used to recognize the presence of a fax call, although they are not a very reliable indication.

Once the line connection with the CALLED party is established, the following takes place during Phase A:

- Fax tone detection
- Digital handshake detection

In preparation for Phase B, the CALLER station is initially given fax TRANSMITTER status, and the CALLED station is initially given fax RECEIVER status.

Phase B

The Control and Capabilities Exchange phase of the fax call is used to identify the capabilities of the fax machine at the other end of the call. It also negotiates the acceptable conditions for the call. The exchange of control messages throughout the fax call is sent using the low speed (300 bps) modulation mode. Every control message is preceded by one-second preamble, which allows the communication channel to be conditioned for reliable transmission.

The called fax machine begins the procedure by sending a Digital Identification Signal (DIS) message, which contains the capabilities of the fax machine. An example of a capability that could be identified in this message is that the V.17 (14000 bps) Data Signaling Rate is supported. At the same time, the Called Subscriber Information (CSI) and Non-Standard Facilities (NSF) messages are optionally sent. Non-Standard Facilities are capabilities that a particular fax manufacturer has built into a fax machine to

distinguish their product from others. These are not required to be supported for interoperability.

Once the calling fax machine receives the DIS message it determines the conditions for the call by examining its own capabilities table. The calling machine responds with the Digital Command Signal (DCS), which defines the conditions of the call. At this stage, high-speed modem training begins. The high-speed modem will be used in the next phase of the fax call to transfer page data. The calling fax machine sends a Training Check Field (TCF) through the modulation system to verify the training and ensure that the channel is suitable for transmission at the accepted data rate. The called fax machine responds with a Confirmation to Receive (CFR) which indicates that all capabilities and the modulation speed have been confirmed and the fax page may be sent

Phase C

In this phase fax document pages are transmitted depending on the parameters negotiated between the CALLER and CALLED applications during Phase B. The high-speed modem is used to transmit the page data that has been scanned in and compressed. It uses the ITU T.4 protocol standard to format the page data for transmission over the channel.

Phase D

After the page has been successfully transmitted, the calling fax machine sends an End of Procedures (EOP) message if the fax call is complete and all of the pages have been transmitted. If only one page has been sent and there are additional ones to follow, it sends a Multi-Page Signal (MPS). The called machine would respond with Message Confirmation (MCF) to indicate the message has been successfully received and it is ready to receive more pages.

Phase E:

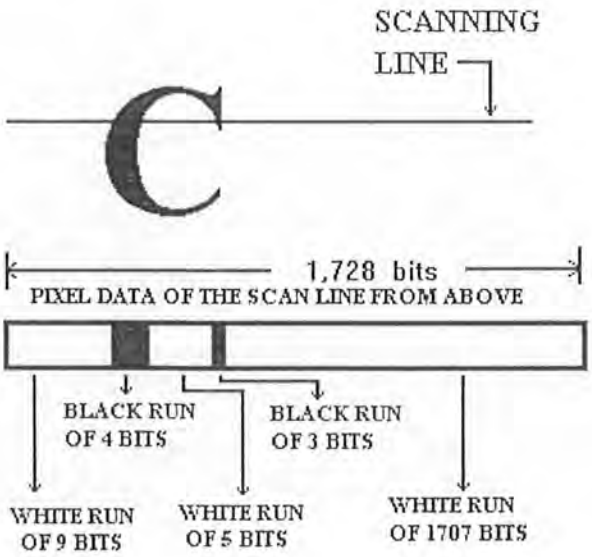
The release phase is the final phase of the call where the calling machine sends a Disconnect Message (DCN). While the DCN message is a positive indication that the fax call is over, it is not a reliable indication since the fax machine can disconnect prematurely without ever sending the DCN message.

7.4 ENCODING SCHEMES

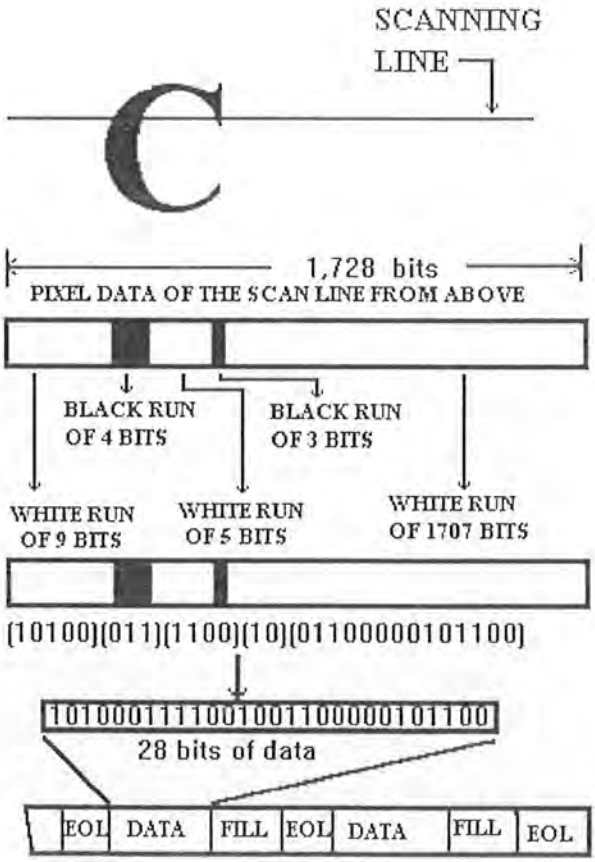
The method that G3 fax machines use to reduce data is called coding. Typically with any line of data read by a fax machine there are white areas and black areas that make up the document line. If we look at the pixel information that makes up a line of picture data you would find groups of black pixels and groups of white pixels. These groups are referred to as black run lengths and white run lengths. Based on these characteristics of run lengths of black or white data, codes can be assigned to the different size run lengths and only these codes need to be transmitted over the phone lines. There are many coding schemes but the most efficient one, which is used by all G3 fax machines, is the Modified Huffman Scheme or MH. Also a variation of this is the Modified Read Scheme (MR) and the Modified Modified Read Scheme (MMR). A newer coding scheme is the Trellis Coding Scheme. Each one of these coding methods reduce the amount of data needed to be sent over the phone lines and therefore each improvement in coding yields an improvement in data transmission speed.

7.4.1 MODIFIED HUFFMAN SCHEME (MH)

The MH coding scheme is referred to as a one-dimensional coding scheme. One line of data is scanned and coded for transmission. Let us consider an example to understand MH coding scheme.



As shown above, the document that is being scanned is a large letter "C". One line is indicated which is 1,728 bits long. This line is broken down into run lengths of black and white. As shown the first run is a white run of 9 bits, the second run is of black and is 4 bits long this breaking down of the scan line to run lengths continues to the end of the line and then a new line will be scanned and broken down into run lengths. After the line is broken down into run lengths, the next step is to code it. MH coding is made up of two factors, the run length (bits) and the run color. Once the run lengths are created they are referenced to the MH coding table to determine what the binary code is that represents that particular run.



Shown above is the scan line converted to MH code, which is then inserted into a standard data line. As you can see the scan line started out as 1728 bits of information but

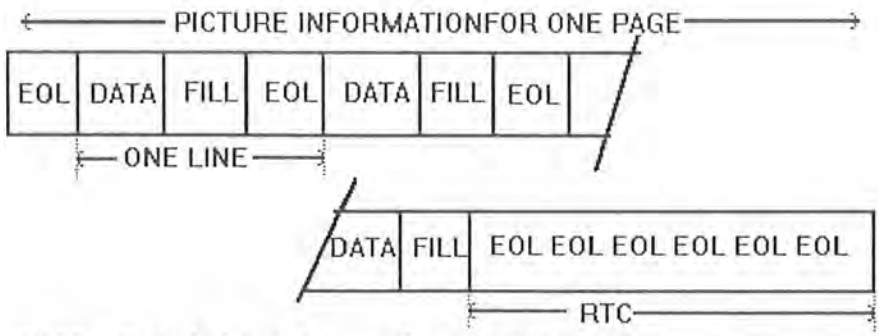
with the MH coding performed on this line it is reduced to only 28 bits of data. Therefore the transmission time for this scan line has been greatly reduced.

7.4.2 MODIFIED READ SCHEME

The Modified Read coding scheme is very similar to the MH coding scheme except that it will scan a line of data and then use this line for reference when it scans the next line. After comparing the two lines only the changes will be transmitted therefore reducing the data data, there is a limit to how many lines of data will be compared to a reference line. If the reference line has an error then that same error will be reproduced on any line that was compared to it. For standard transmission quality the reference is set to 2 lines and on even farther. Because this coding scheme is more likely to create errors in the picture fine quality it is set to 4 lines.

7.4.3 STRUCTURE OF PICTURE SIGNALS

Shown below are the components of a picture signal.



DATA = Coded picture information of each line
FILL = In case the Data & EOL time is less then MTT. This is used to etend the transmission time.
EOL = (end of line) End of picture signal for one line of data.
RTC = Indicates end of picture signa for one page. (6 consecutive EOL's) (return to control)

Appendix A

Data Types

Data Types

Dialogic SDK uses very complex data types to provide certain functionality. Below is the description of two important data types that are used:

DF_IOTT

The DF_IOTT structure provides a description of the fax data to be transmitted.

```
typedef struct
{
   unsigned long io_offset;          // Starting page number or offset
   unsigned long io_length;          // Number of pages or length of data
   char *io_bufferp;                 // Memory transfer start buffer location
   DF_IOTT *io_prevp;                // (Optional) Pointer to previous DF_IOTT
   DF_IOTT *io_nextp;                // Pointer to next DF_IOTT entry (for linked list)
   void *io_datap;                   // Pointer to additional data associated
                                     // with io_datatype
   int io_fhandle;                   // File descriptor
   unsigned short io_type;           // Entry type (file, memory; linked, contiguous,
                                     // last structure; select user-defined I/O
                                     // functions for transmit)
   unsigned short io_datatype;       // Type of data to transmit
   unsigned short io_phdcont;        // Phase D continuation value to send
   unsigned short io_width;          // Width of image  (raw and ASCII)
   unsigned char io_resln;           // Vertical resolution of image (raw and ASCII)
   unsigned char io_coding;          // Encoding of stored data (raw)
   unsigned char rfu[2];             // Reserved for future use
} DF_IOTT;
```

DX_CAP

The DX_CAP structure modifies parameters that control Frequency Detection, Cadence Detection, Loop Current, and Positive Voice Detection.

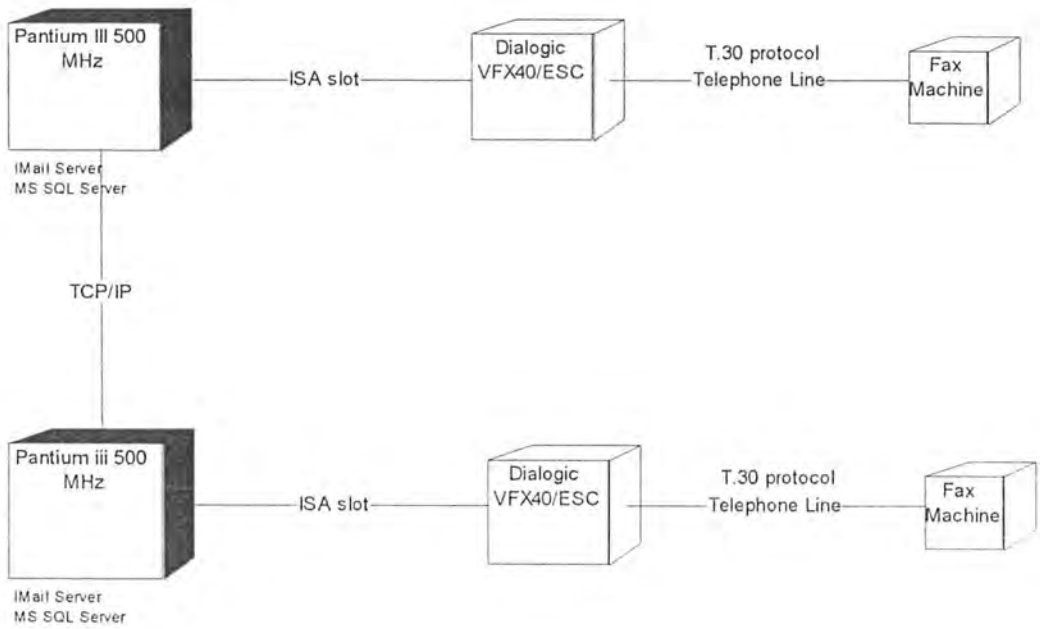
```
typedef struct DX_CAP
{
  unsigned short ca_nbrdna;          // # of rings before no answer.
  unsigned short ca_stdely;          //Delay after dialing before analysis.
  unsigned short ca_cnosig;          //Duration of no signal time out delay.
  unsigned short ca_lcdly;           //Delay after dial before lc drop connect
  unsigned short ca_lcdly1;          // Delay after lc drop con. before msg.
  unsignedshortca_hedge;             //Edge of answer to sendconnect message.
  unsigned short ca_cnosil;          // Initial continuous noise timeout delay.
  unsigned short ca_lo1tola;         // % acceptable pos. dev of short low sig.
  unsigned short ca_lo1tolb;         // % acceptable neg. dev of short low sig.
  unsigned short ca_lo2tola;         // % acceptable pos. dev of long low sig.
  unsigned short ca_lo2tolb;         // % acceptable neg. dev of long low sig.
  unsigned short ca_hi1tola;         // % acceptable pos. dev of high signal.
  unsigned short ca_hi1tolb;         // % acceptable neg. dev of high signal.
  unsigned short ca_lo1bmax;         // Maximum interval for shrt low for busy.
  unsigned short ca_lo2bmax;         // Maximum interval for long low for busy.
  unsigned short ca_hi1bmax;         // Maximum interval for 1st high for busy
  unsigned short ca_nsbusy;          // Num. of highs after nbrdna busy check.
  unsigned short ca_logltch;         // Silence deglitch duration.
  unsigned short ca_higltch;         // Non-silence deglitch duration.
  unsigned short ca_lo1rmax;         // Max. short low dur. of double ring.
  unsigned short ca_lo2rmin;         // Min. long low dur. of double ring.
  unsigned short ca_intflg;          // Operator intercept mode.
  unsigned short ca_intfltr;         // Minimum signal to qualify freq. detect.
  unsigned short rfu1;               // reserved for future use
  unsigned short rfu2;               // reserved for future use
```

```
unsigned short rfu3;                // reserved for future use
unsigned short rfu4;                // reserved for future use
unsigned short ca_hisiz;            // Used to determine which lowmax to use
unsigned short ca_alowmax;          // Max. low before con. if high>hisize.
unsigned short ca_blowmax;          // Max. low before con. if high<hisize.
unsigned short ca_nbrbeg;           // Number of rings before analysis begins.
unsigned short ca_hi1ceil;          // Maximum 2nd high dur. for a retrain.
unsigned short ca_lo1ceil;          // Maximum 1st low dur. for a retrain.
unsigned short ca_lowerfrq;         // Lower allowable frequency in hz.
unsigned short ca_upperfrq;         // Upper allowable frequency in hz.
unsigned short ca_timefrq;          // Total duration of good signal required.
unsigned short ca_rejctfrq;         // Allowable % of bad signal.
unsigned short ca_maxansr;          // Maximum duration of answer.
unsigned short ca_ansrdgl;          // Silence deglitching value for answer.
unsigned short ca_mxtimefrq;        // max time for 1st freq to remain in bounds
unsigned short ca_lower2frq;        // lower bound for second frequency
unsigned short ca_upper2frq;        // upper bound for second frequency
unsigned short ca_time2frq;         // min time for 2nd freq to remains in
                                       bounds
unsigned short ca_mxtime2frq;       // max time for 2nd freq to remain in bounds
unsigned short ca_lower3frq;        // lower bound for third frequency
unsigned short ca_upper3frq;        // upper bound for third frequency
unsigned short ca_time3frq;         // min time for 3rd freq to remains in bounds
unsigned short ca_mxtime3frq;       // max time for 3rd freq to remain in bounds
unsigned short ca_dtn_pres;         // Length of a valid dial tone (def=1sec)
unsigned short ca_dtn_npres;        // Max time to wait for dial tone(def=3sec)
unsigned short ca_dtn_deboff;       // The dialtone off debouncer(def=100ms)
unsigned shortca_pamd_failtime;     //WaitforAMD/PVDaftercadence
                                       break(default=4sec)
unsigned short ca_pamd_minring;     // min allowable ring duration(def=1.9sec)
byte ca_pamd_spdval;                // Set to 2 selects quick decision(def=1)
```

```
    byte ca_pamd_qtemp;                 // The Qualification template to use for
                                           PAMD
    unsigned short ca_noanswer;         // time before no answer after first
                                           ring (default=30sec)
    unsigned short ca_maxintering;      // Max inter ring delay before connect
                                           (8 sec)
};
```

Appendix B

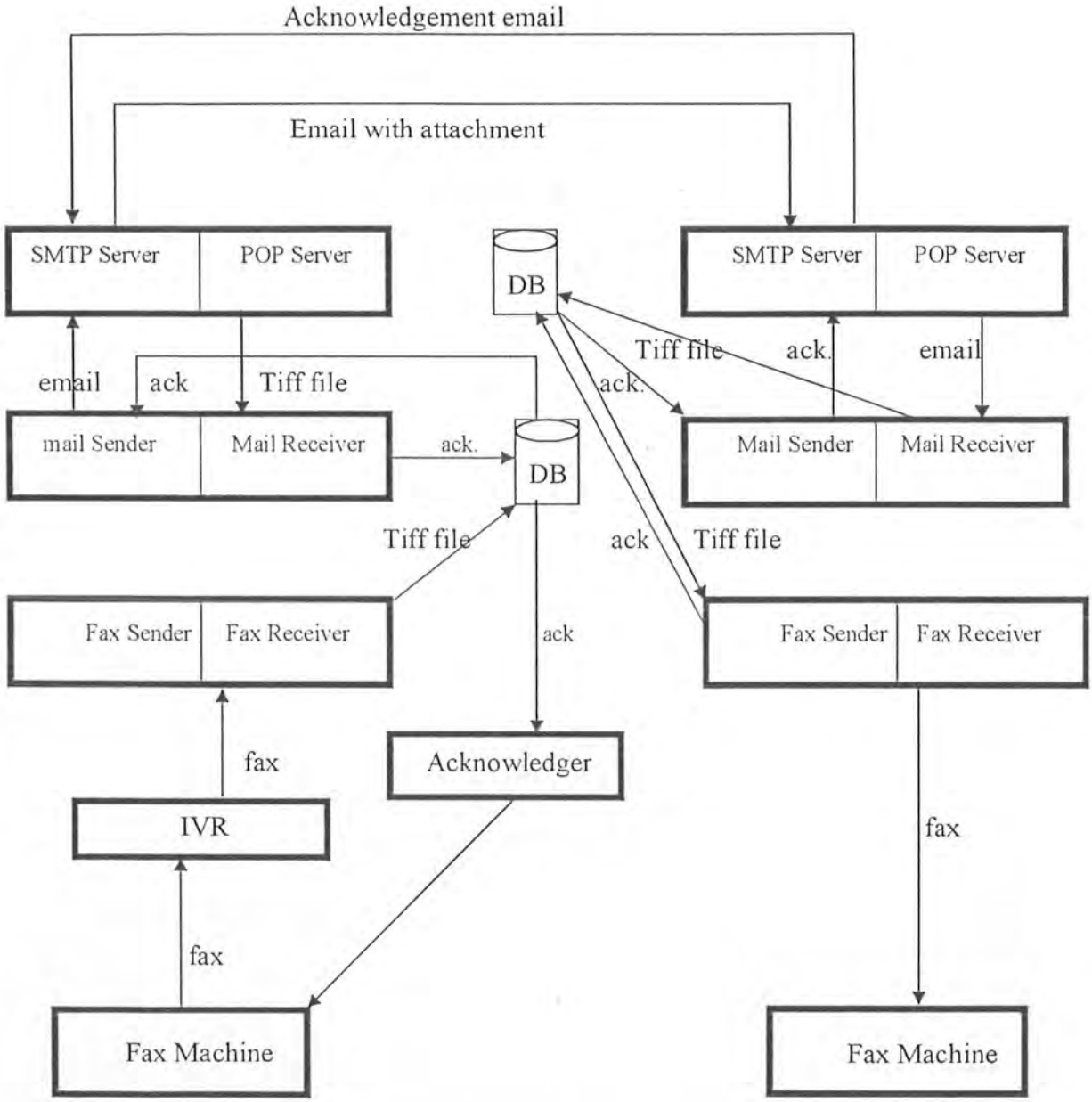
Deployment Diagram and System Diagram



Deployment Diagram

(Fig.1.1)

System Diagram

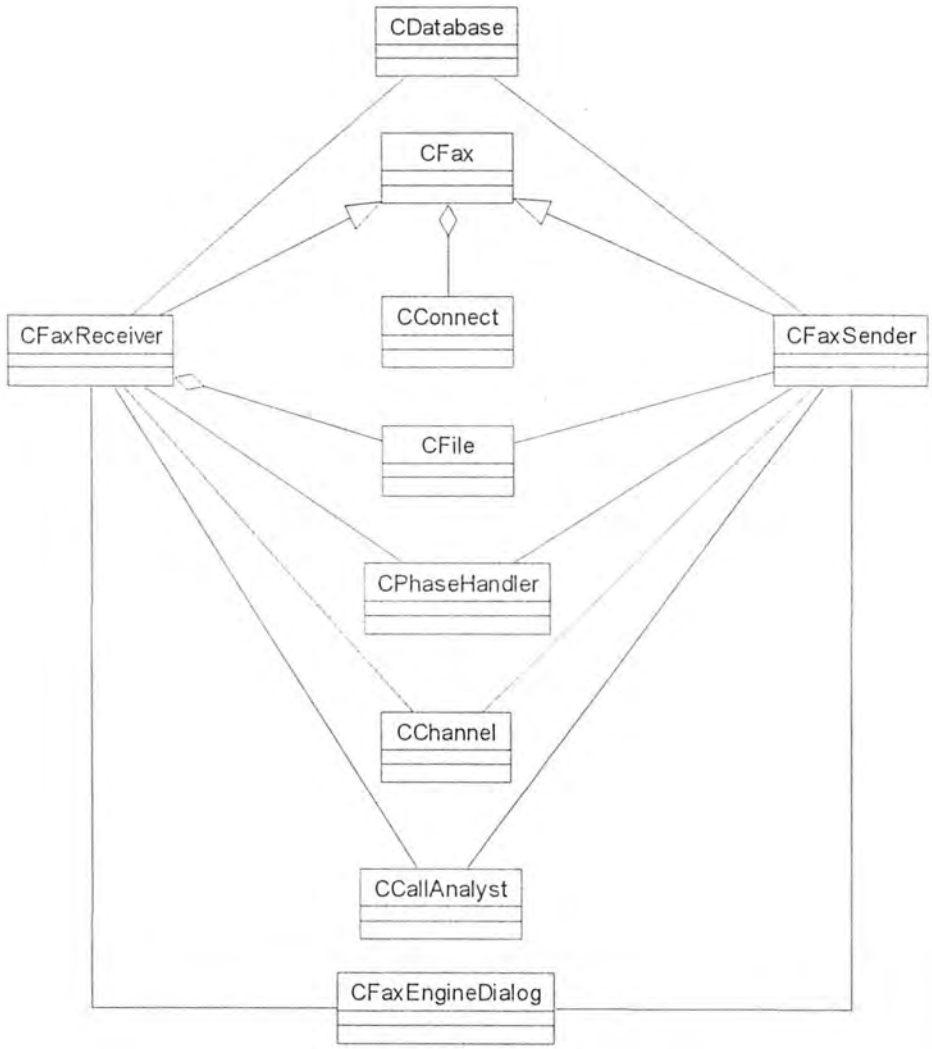


Overall Working of Superfax

(Fig. 1.2)

Appendix C

Class Diagram

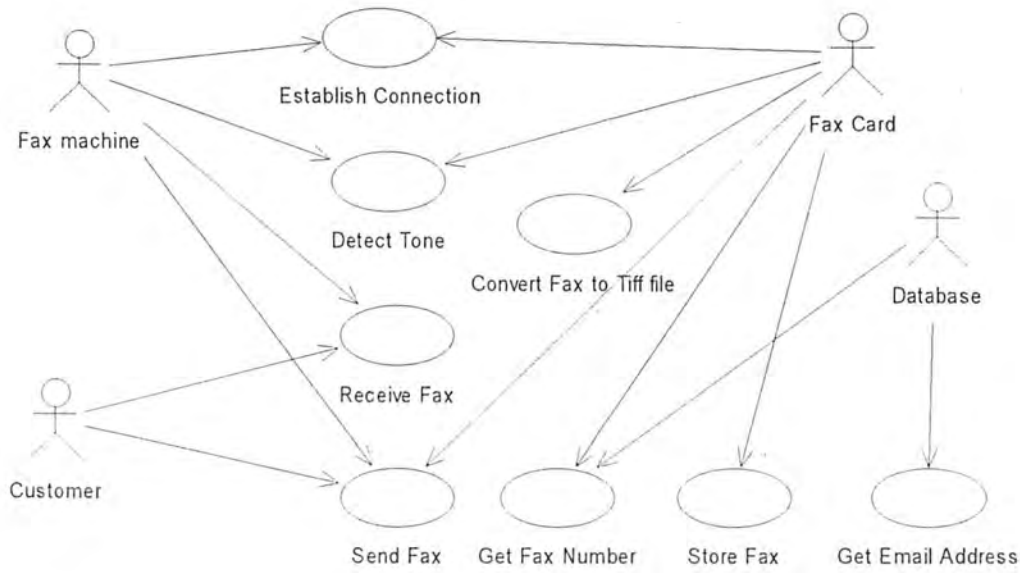


Class diagram

(Fig. 3.1)

Appendix D

Use Case Diagram



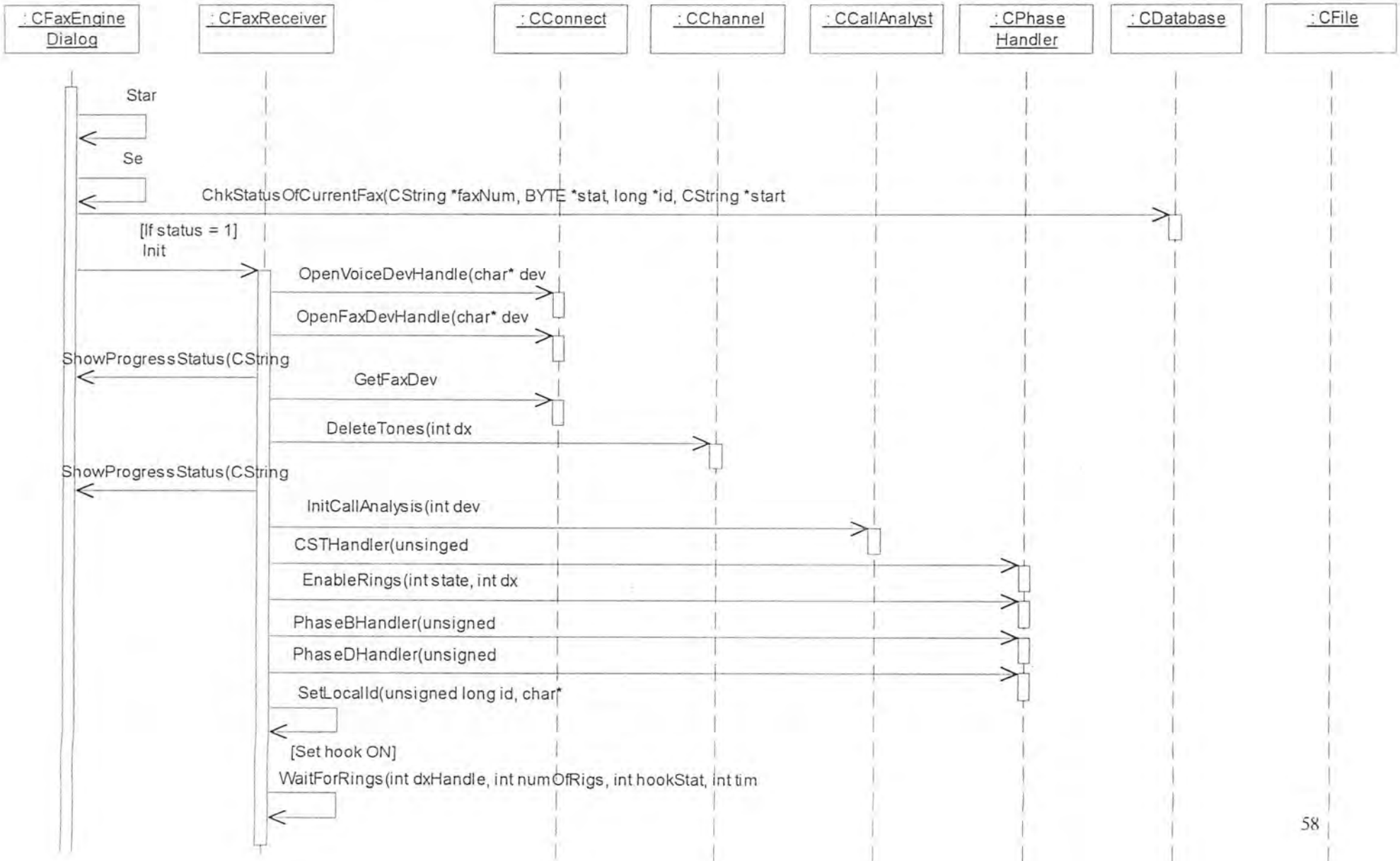
Use Case Diagram

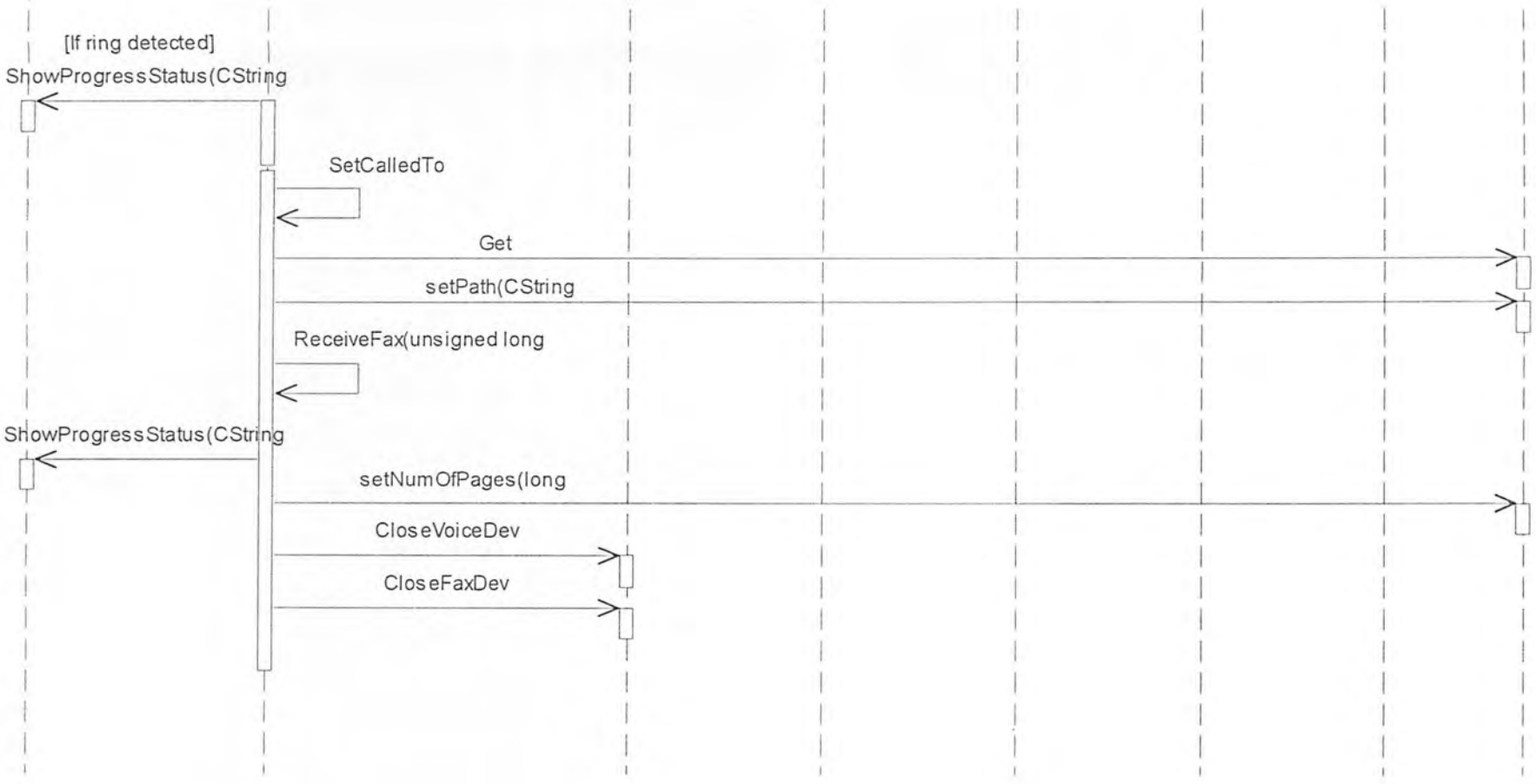
(Fig. 2.1)

Appendix E

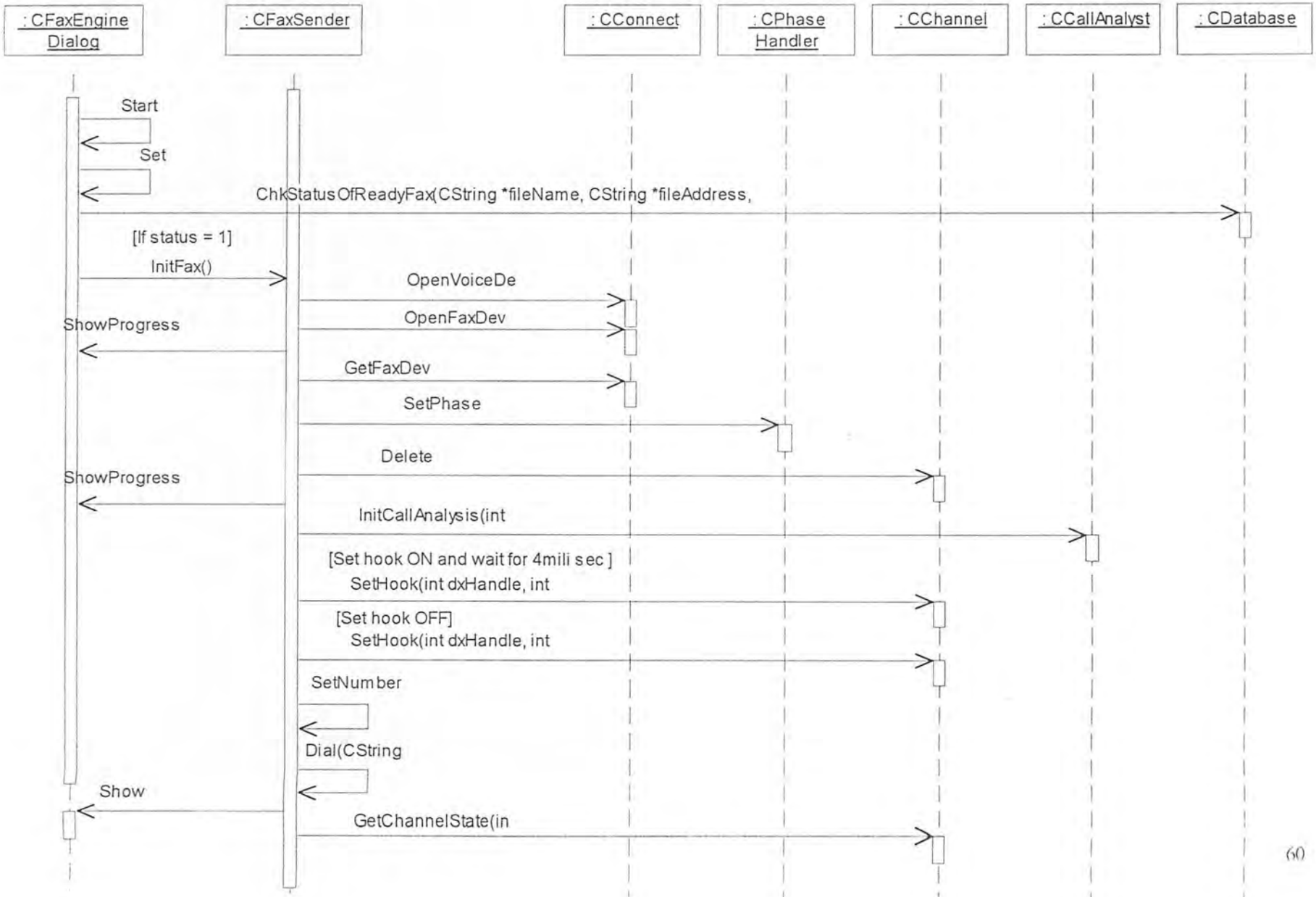
Sequence Diagrams

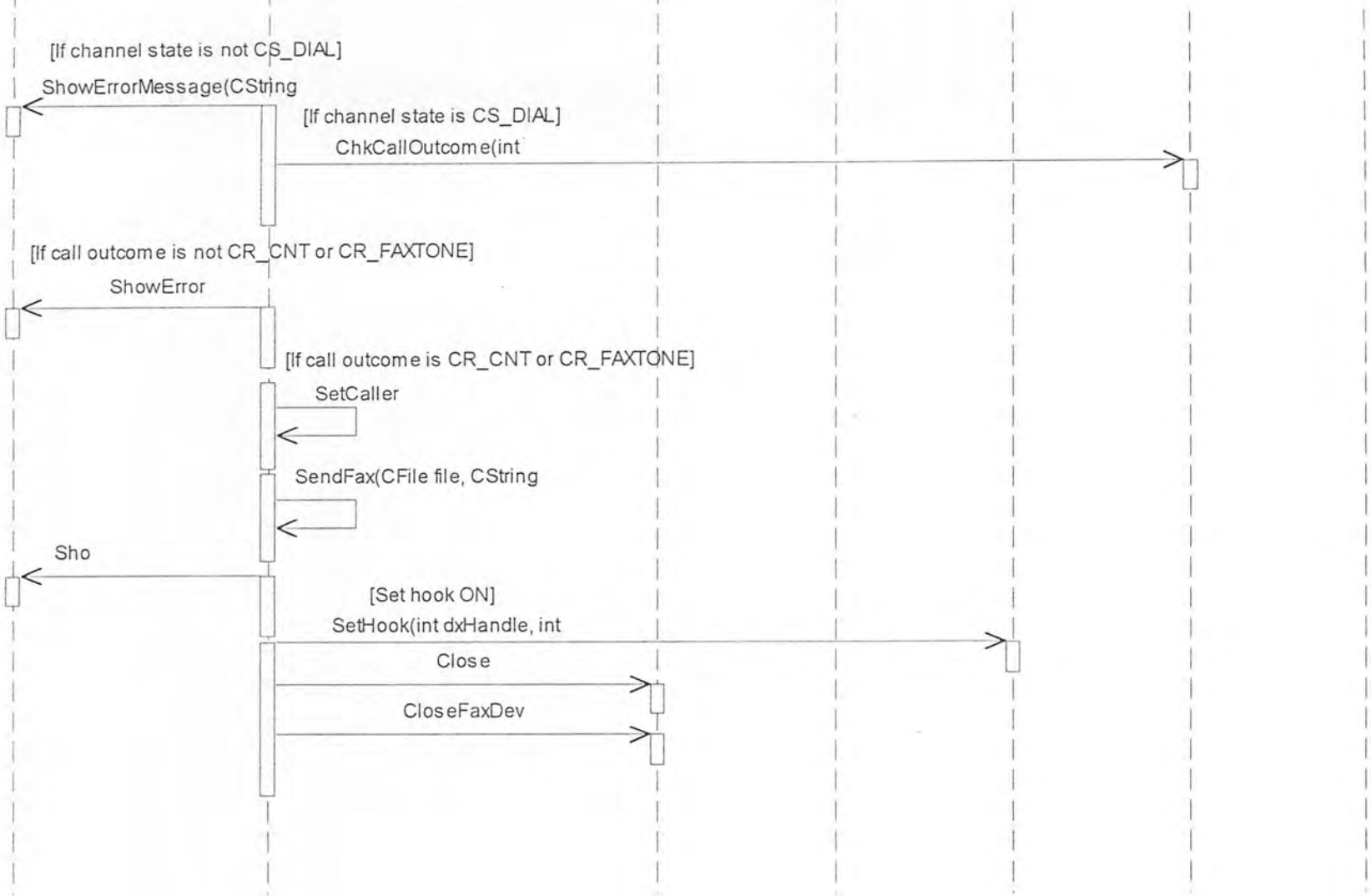
RECEIVE FAX SCENARIO




[If ring detected]
ShowProgressStatus(CString
SetCalledTo
Get
setPath(CString
ReceiveFax(unsigned long
ShowProgressStatus(CString
setNumOfPages(long
CloseVoiceDev
CloseFaxDev

SEND FAX SCENARIO




[If channel state is not CS_DIAL]
ShowErrorMessage(CString
[If channel state is CS_DIAL]
ChkCallOutcome(int
[If call outcome is not CR_CNT or CR_FAXTONE]
ShowError
[If call outcome is CR_CNT or CR_FAXTONE]
SetCaller
SendFax(CFile file, CString
Sho
[Set hook ON]
SetHook(int dxHandle, int
Close
CloseFaxDev

Appendix F

Resources Required

HARDWARE REQUIREMENTS

- ✓ 200 MHz processor
- ✓ 64 Megabytes RAM
- ✓ 1GB hard disk (Size of the HD can be increased according to the organization's requirements)
- ✓ VFX40/ESC Dialogic Fax and Voice board
- ✓ Space in the computer to hold fax board (Dialogic fax board requires a full length ISA slot)
- ✓ Modem to access Internet
- ✓ Fax machine
- ✓ Various cables to connect the system to the telephone lines

SOFTWARE REQUIREMENT

- ✓ Visual C++ 6.0
- ✓ Dialogic SDK
- ✓ Microsoft Windows NT Server 4.0

Glossary

CALLER An application that places a call

CALLED An application that receives a call

Call Analysis The process used to automatically determine what happened after an outgoing call is dialed. Call Analysis monitors the progress of an outbound call after dialing, which allows you to process the call based on the outcome.

By using Call Analysis you can determine the following:

- If the line is answered and, in many cases, how the line is answered
- If line rings but is not answered
- If line is busy
- If there is a problem in completing the call

device handle Numerical reference to a device, obtained when a device is opened. The device handle is used for all operations on that device.

device channel A Dialogic voice data path that processes one incoming or outgoing call at a time (equivalent to the terminal equipment terminating a phone line). There are 4 device channels on a D/4x board.

event An unsolicited or asynchronous message from a hardware device to an operating system, application, or driver. Events are generally attention-getting messages, allowing a process to know when a task is complete or when an external event occurs.

Fax Session A fax session begins when entering Phase A and ends when exiting Phase E of the T.30 protocol.

Group 3	T.4 standards for facsimile devices developed by CCITT adopted in 1980 and modified in 1984 and 1988. A Group 3 digital fax transmission of an 8.5" by 11" (or A4) page at 9,600 bps is completed in 15 - 30 seconds using PSTN phone lines.
Group 4	T.6 standards for facsimile devices developed by CCITT. Using Public Data Networks or modified PSTN, the data is transmitted using ECM (Error Correction Mode).
hook state	A general term for the current line status of the channel either *on-hook* or *off-hook*. A telephone station is said to be *on-hook* when the conductor loop between the station and the switch is open and no current is flowing. When the loop is closed and current is flowing the station is off-hook. These terms are derived from the position of the old fashioned telephone set receiver in relation to the mounting hook provided for it.
hook switch	The name given to the circuitry which controls on-hook and off-hook state of the Voice device telephone interface.
idle device	A device that has no functions active on it.
MH	Modified Huffman code (MH) is one-dimensional encoding scheme used to compress fax data for transmission used in Group 3 fax devices. For example, a white line with no text, called a run, extending across an 8.5" page equals 1728 bits. Modified Huffman Code compresses the 1728 bits into a 17-bit code word. The lengths for all possible white runs are grouped together into 92 binary codes that will handle any white run length from 0 to 1728.
MMR	Modified Modified Read code (MMR) is a Group 4 facsimile two-dimensional digital encoding scheme with improved transfer speed over Modified Read encoding.

MR	Modified Read code (MR) is a Group 3 facsimile two-dimensional digital encoding scheme with improved transfer speed over Modified Huffman encoding.
off-hook	The state of a telephone station when the conductor loop between the station and the switch is closed and current is flowing. When a telephone handset is lifted from its cradle (or equivalent condition), the telephone line state is said to be off-hook.
on-hook	When a telephone handset is returned to its cradle (or equivalent condition), the telephone line state is said to be on-hook.
RECEIVER	An application that receives or is capable of receiving a fax document
TIFF/F	Tagged Image File Format Class F. TIFF is a tag based general purpose raster format used to exchange image data between application programs. Class F indicates specific format information for fax applications.
time out	In telephone networks, an event which occurs at the end of a predetermined interval of time.
TRANSMITTER	An application that transmits or is capable of transmitting a fax document

References:

[HAN&MAN 98]	Hans-Erik Eriksson & Mangus Panker ,*"UML toolkit"*, John Wiley & Sons.inc.
[MAR&KEN]	Martin Fowler & Kendall Scott, *"UML Distilled"*, 2nd edition , Addison-Wesley
[IAN96]	Ian Sommerville, *"Software Engineering"*, 5th edition, Addison-Wesley
[ROG98]	Roger S. Pressman, *"Software Engineering"*, 4th edition, McGraw Hill