Visual Language for Computer Telephony

(Module 2)

By

Muhammad Siddique Bhatti

Computer Science Department

Quaid-i-Azam University, Islamabad

January 2005

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: March 12, 2005

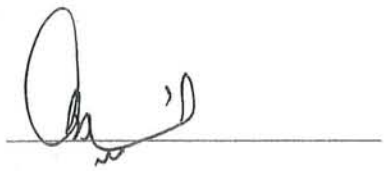
FINAL APPROVAL

This is to certify that we have read the project report submitted by Mr. Muhammad Siddique Bhatti and it is our judgment 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:

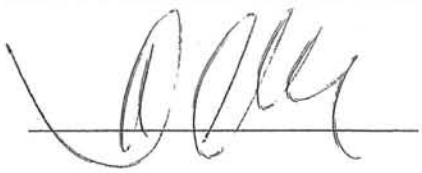
1. External Examiner

   Dr. Arshad Ali Shahid
   National University of Computer Science
   and Engineering Science
   FAST House, Rohtas Road, G-9/4
   Islamabad.



2. Supervisor

   Mr. Asif Qumer
   Lecturer
   Department of Computer Science
   Quaid-i-Azam University
   Islamabad



3. Chairman

   Dr. M. Afzal Bhatti
   Department of Computer Science
   Quaid-i-Azam University
   Islamabad

Dedicated to ...

My loving and caring **parents** whose prayers and affection has enabled me to accomplish such a difficult task, and to my brothers and sisters who always guide and encourage me, and make me feel comfortable even at toughest times.

PREAMBLE

In The Name of Allah, the Most Gracious, the Most Merciful

Praise Be To Allah, Lord of Words
The Beneficent, the Merciful
Master of the Day of Judgment
Alone We Worship, Alone We Ask For Need
Show Us the Straight Path
The Path of Those Whom Thou Haste Favored
Neither the Path of Those
Who Earn Thins Anger of Those
Who Go Astray

Al-Fatiha **(Al-Quran)**

Project Brief

Project Title:	Visual Language for Computer Telephony (Module 2)
Undertaken By:	Muhammad Siddique Bhatti
Supervised By:	Mr. Asif Qumer
	Lecturer, Department of Computer Science
	Quaid-i-Azam University, Islamabad
Starting Date:	September 2004
Completion Date:	January 2005.
Tool Used:	Microsoft Visual C++ 6.0
Operating System:	Microsoft Windows 2000 Professional.
System Used:	Intel® Pentium® 3 CPU 650 MHz.

Abstract

Visual Computer Telephony Script Editor is Windows software that helps you quickly implement a professional voice response system into your business. Developing IVR solutions is easy with Visual IVR Script Designer .Allows fast and easy design of an Interactive Voice Response system. Visual IVR Script Designer is easy to learn, and allows a complete GUI for the whole IVR development process. A list of simple IVR blocks is available on the toolbox. These blocks can be dragged and dropped on the workspace. Each block has Properties and Event Handlers. It provides operator with a client area calling "Canvas" where he would be able to drop the selected components and arrange them. Edit Event Handlers to link the modules to one another. Correct translation of the drawn objects on the Canvas in the way they are connected using XML script.

Preface

This report represents description of Analysis, Design, Development, Implementation and Testing of "Visual Language for Computer Telephony". The entire work is presented in chapters and some Appendixes.

Chapter-1	This chapter contains the material about the problem domain, scope objective and feasibility of problem to be solved.
Chapter-2	This chapter concerns about requirement engineering and analysis of system.
Chapter-3	This chapter explains about the System analysis phase. It includes usecase diagram, System Sequence Diagram.
Chapter-4	This chapter introduces Object-Oriented design of the system. It includes class diagram, sequence diagram and their description.
Chapter-5	This chapter introduces the implementation of the system. It discusses the technology used for the development of this product. Also given the detailed information about project classes.
Chapter-6	This chapter covers the testing phase of the system
References	Reference of books used in this documentation.
Bibliography	Bibliography of books used in project development process.
Webliography	URL's of web sites, those were useful in development process.
Appendices	Appendices are given at the end, which include process model, feasibility study, usecase description, sequence diagrams, resource estimates and Gantt chart.
Glossary	Provides description of some terms used in the report.
Visual Glossary	Some graphical notations used in the report and their description.

Contents

List of Appendices

Table of Figures

Acknowledgement

First of all I would like to extend my sincere and humble gratitude to Almighty Allah whose blessings and guidance has been a real source of all my achievements in my life.

Furthermore sincere thanks are due to my supervisor **Mr. Asif Qamar** for his guidance and great help during all the phases of this project and his guidance during my studies. He always remained a source of learning for me.

I will be failing in my duties if I don't express my warmest thanks to my external supervisor **Mr. Imdad Hussain** (System Analyst in Visual Soft) for his help, guidance and valuable suggestions in algorithm design, reading materials and other difficulties of the project. I must also thank **Mr. Muhammad Jamil** (Director in PTCL) for his kind help at every stage of development.

It is my bounden duty to pay tributes to honorable teachers of Computer Science Department, Quaid-i-Azam University Islamabad. Greatest thanks go to my favorite teachers, **Dr. Farhana Shah, Miss Ifra Farrukh Khan** and **Mr. Ghazanfar Farooq** for their kind help and guidance.

I feel greatest pleasure in expressing my deepest appreciation and heartiest gratitude to ***Dr. Afzal Bhatti***, Chairman Department of Computer Science, for his determined efforts for the betterment of the department. He has always been very kind and affectionate. He is a great source of inspiration for me.

I am indebted to my parents and other family members for their prayers and moral & financial support during my studies.

I can never forget to mention my batch mates the **"Pointers"** for their sincere help, encouragement and enjoyable company during my stay at Islamabad. Their enjoyable company was really a great source of pleasure for me at university campus and I always feel the presence of their well wishes for me.

Muhammad Siddique Bhatti
January 1, 2005.

Chapter 1

Overview

1.1 Introduction to Organization

Visual Soft (Pvt.) Ltd. is a privately held company headquartered in Rawalpindi, Pakistan.

Visual Soft offers comprehensive Software/Hardware design, development, integration and testing Services. Visual Soft's vision is to become a leading Product Development Company. Visual Soft build on the product concept to help companies realize a commercial product with a focus on their valued proposition consisting of low cost, high quality and on time delivery.

1.2 Existing System

Since current working system is based on MFC (Microsoft Foundation Classes).

The major problem in Existing System is:

Bad Design

The GUI (Graphical User Interface) is not properly designed It is not user friendly because the Icons are not property managed.

Data Lose

Since the data base is not property managed in existing system, hence the call data may be lose.

Slow Execution

The existing system is based on MFC (Microsoft Foundation Classes) so, it's very heavy and slow to execute.

Searching Problem

Since the data base in not property managed and system is very slow, hence system does not search properly.

No Flow Graph Information

There is no flow graph information in existing system. It causes the call understanding problem of out going calls.

XML Generation and Parsing

There is no concept of XML generation and parsing in existing system.

Compatibility Problem

The existing system is not properly compatible with related hardware (Dialogic Card).

The major requirement of the organization is to build a system in Visual C, ATL (Active Template Libraries) COM (Component Object Model).

1.3 Problem Definition

It is required to build the flow graph in which all of the components/controls will be added. This environment will be capable of designing a flow graph using all integrating components/controls. After designing this flow graph XML files will be generated. An XML file of specific pattern or flow graph will contain design time information and properties of those components that are used in that flow graph. Database/Data Structure of specific pattern is also required to generate design time information and properties of those components. These XML file and database will contain the logic of flow graph, describing how these components are interacting and what application (IVR) this flow graph is representing.

A Visual Interface is required to be developed in which all of the components/controls will be integrated. These components will be used here to build a flow graph for specific telephony applications. This environment will be capable of Drag Drop facility of controls and properties of these components can be set at design time.

On execution of flow graph an XML file of specific pattern will be generated and those patterns will be used in telephonic cards (Dialogic) further. These XML file will contain design time information of components and also the execution logic of flow graph.

Database will also be used to store the design time information of components and also the execution logic of flow graph.

1.3 Scope

There are three major scenarios to discuss the scope of Visual Language for Computer Telephony. Those are discussed below.

1.3.1 Context

Visual Language for Computer Telephony will be used in an environment in which the dominating and frequently used operating system is Microsoft Windows. Visual Language for Computer Telephony must be built fit into a larger system or business context. So the constraint imposed as a result of context on is that, Visual Script Editor for Component based Computer Telephony must be installed and run in an environments which is compatible with Microsoft Windows.

1.3.2 Information Objectives

In information objective the data objects produced as outputs of system and data objects, which are required as inputs of system are discussed.

1.3.2.1 Input

- Components added by user in Client Area.
- Properties of Components set by user when integrated into "Computer Telephony Visual Script Editor"
- User mouse actions against components.

1.3.2.2 Output

Each component has certain set of properties which are set at design time these properties are used in XML file generated by script editor.

1.3.3 Functions and Performance

Functions are those tasks, which will be performed by our system in order to transform the input data to output. General functions that can be performed with our Components are following.

- Providing support for drag and drop.
- Generating XML file.
- Provide facility of Cut, Copy and Past.
- Provide facility of Moving and Removing controls.

- Providing save facility.
- Open Previous Session.

1.4. Objectives

Since there is no existing system in Visual Soft, hence Visual Language for Computer Telephony will work in visual soft first time.

- To develop a system this provides operator a strong GUI (Graphical User Interface) as main application.
- System will prove to be a cost effective, minimizing the distances at affordable rates.
- System user interface during implementation will be given special focus to make the system more user-friendly, the complexities will be removed.
- Provides a property box showing properties of objects where user can edit the properties.
- Correct translation of the drawn objects on the drawing board in the way they are connected.
- Provides operator with a drawing area calling drawing board where he would be able to draw the selected components and arrange them.
- Provides operator with a tool box facilitating him selecting the tool for drawing.

1.5. Resource Identification

There is need of resources for doing any kind of job. Following resources are required to accomplish the development of this system.

1.5.1. Human Resources

The teams working on Telephony in Visual Soft (Pvt.) Ltd. has maintained some software development metrics. These metrics indicates size, complexity and feasibility of the project. According to these metrics, complexity and size of the project indicates that

effort required for this project is 8 person months, since two people are working on this project hence this project should complete within four months.

1.5.2. Hardware Resources

Hardware resources required for the development are as follow.

Resource Name	Minimal	Recommended	Description
Processor	233 MHz	850 MHz	To run software required for development
RAM*	64 MB	256 MB	To run software required for development
Hard Drive	5GB	10GB	To run software required for development and store data.
Printer	Dot Matrix	HP DeskJet 600	To get hard copy of documentation for end user.
Keyboard	Standard	Standard	To give input to the computer
Mouse	Standard	Standard	To give input to the computer
Color Monitor	With resolution 800x600	With resolution 1024x768	To view out put.

1.5.3. Software Resources

Software resources required for the development are as follow.

Resource Name	Minimal	Recommended	Description
Operating System	MS Windows 98	MS Windows 2000 professional	To run the hardware.
Development Tool	Visual C++ 6.0 with ATL COM	Visual C++ 6.0 with ATL COM	To develop the system.
Documentation Tool	MS Office 97	MS Office XP or Star Office TM 5.2	To prepare the documentation.
Management Tool	MS Project 98	MS Project 2000	For analysis & Design
CASE Tool	Rational Rose 2.0	Rational Rose 5.0	For analysis and Design
Help Tool	MSDN library	MSDN library	For development help

1.6 Feasibility Study

The feasibility study is project management activity that is reviewed by project management to assess the reliability and at upper management level to assess the project status. The feasibility study finally results in "go/no-go" decision

Please see Appendix-C for more details.

1.7 Process Model

The process model selected for the software development cycle is the "Spiral Model". For detailed description on process model please refer the Appendix-A.

Conclusion

The problem definition is very important before moving forward during the software development. Without knowing what the problem is?, one can't solve it. The preliminary investigation about the problem, its scope and various constraints one can face in solving the problem and to set what objectives are behind finding the solution of the problem provides the fruitful supplement to the next phases of solution.

Chapter 2

Requirement Engineering

2.1 Introduction

The requirement engineering phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a software requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the ``what" phase. [SOM00]

The requirement document tries to capture the requirements from the customer's perspective by defining goals and interactions at a level removed from the implementation details.

The requirement document does not specify the architectural or implementation details, but specifies information at the higher level of description. The problem statement, the customer's expectations, and the criteria for success are examples of high-level descriptions. There is a fuzzy line between high-level descriptions and low-level details

The requirements specified in this document provide a base for the design and development of the software. This document has the following three main phases:

- Requirement elicitation.
- Requirement definition.
- Requirement specification.

2.2 Requirement elicitation

At this phase a communication between the customer and the developer is arranged. If the customer knows his requirements then he initiates the communication. If the customer is unable to express his requirements then the communication takes the shape of question answer session. At first the developer asks some generic questions about the motivation behind the need to develop the system. This leads to a basic understanding of the problem. The next session enables the analyst to gain a better understanding of the problem.

The requirements elicitation phase was carried out by questioning the Visual Soft's Project manager because he is a person who has defined this project.

2.3 Requirement definition

Requirement definition gives an abstract view of the system services and the constraints under which the system will operate. Following is the requirement definition of my system:

2.3.1. Functional requirements

Drag and Drop Facility

The user will be able to add the controls in the editing window (canvas) using drag and drop facility.

Select Control

The user will be able to select the controls by clicking the mouse on the relevant control.

Property Sheet

Every control has some basic properties it's also a function requirement that when any control is selected than a property sheet will be open in the container.

Generate XML

When a flow graph of controls will be generated in the editing window (canvas) than XML file will be generated.

Copying of Controls

The system should give the facility of copying of controls in the specified location of the canvas.

Removing of Controls

The system should give the facility of copying of controls.

Pasting of Controls

The system should give the facility of copying of controls.

Moving of Controls

The system should give the facility of copying of controls.

Printing

When a flow graph is maintained in the editing window and user wants to take a print of required flow graph there the system give the facility of flow graph.

Save

If user wants to save any flow graph, there must be the facility of saving in the system.

Open Previous Session

If user require any previous file that was saved. Than there must be a facility of opening the previous file, when user give the accurate name and extension.

2.3.2 Non-functional requirements

Non-functional requirements define system properties and constraints. The system properties are reliability, response time and memory requirements. The constraints are the capabilities of the I/O devices attached to the system and the data representations used by other systems connected to the required system. The following are three main categories of the non-functional requirements. [SOM00]

2.3.2.1 Product requirements

The system to be developed should have the user friendly GUI (Graphical User Interface). Error or informative messages should be illustrative.

2.3.2.2. Organizational requirements

- The Operating system to be used during the development phase is Microsoft Windows 2000 Professional.
- The development tool is MS Visual C++. The product and its documentation are to be delivered by February 6, 2004.

2.3.2.3. External requirements

The system to be developed is a common property of the **Visual Softs (Pvt.) Ltd.** Department of Computer Science (Quaid-i-Azam University Islamabad) and the student who is developing the system (me). The system is acceptable to the Visual Soft and all those people who have seen its working prototype.

Conclusion

The requirement engineering emphasis on intense communication between the customer and system engineer. This is carried out through various activities like requirement elicitation, analysis and negotiation and definition. This phase culminates with the *System Specification* document.

Chapter 3

System Analysis

3.1 Introduction to Object Oriented Analysis

The object oriented analysis model encompasses a description of static and dynamic characteristics of classes that describe the system or product. The objective of object oriented analysis is to develop a model that describes the software as it works to satisfy the customer requirements. [CRL2001]

For Analysis and Design purposes the Unified Modeling Language (UML) is used. In UML, a system is represented using five different "views" that describe the system from distinctly different perspective. Each view is defined by a set of diagrams. The following views are present in UML.

3.1.1. User Model View

The use-case is the modeling approach of choice for the user model view.

3.1.2. Structural model view

Data and functionality are viewed from inside the system. That is, static structure (classes, objects and relationship) is modeled.

3.1.3. Behavioral Model

This model represents the dynamic or behavioral aspects of the system. It also depicts the interactions or collaborations between various structural elements described in the user model and structural models view.

3.1.4. Implementation View Model

The structural and behavioral aspects of the system are represented as they are to be built.

3.1.5. Environmental Model View

The structural and behavioral aspects of the environment in which the system is to be implemented are represented.

However the UML analysis modeling focuses on *User Model* and *Structural Model* views of system.

3.1.1. User Model View

3.1.1.1. Identification of Actors

Actors of this system are those users who will be using this project (Visual Language for Computer Telephony).

3.1.1.2. Use Case Identification

Use cases are represented graphically in a use case diagram to allow the analyst to visualize each use case in the context of other use cases in the system or subsystem to show its relationship with actors and other use cases. Use case names are text strings that contain letters, numbers and most punctuation marks except for colon, which is used to separate use case names from the name of packages, and it is good idea to keep them short. Use case names are normally made up of an active verb and a noun or noun phrase that concisely describe the behavior of the system that you are modeling. There is only one actor of the system, which is the user of Visual Language for Computer Telephony.

The relationship of usecase with functional requirement is shown in the table.

Functional Requirements	**Usecases**
Addition of Controls	Control Add
Selection of Controls	Control Select
Display the Property Sheet	Property Sheet Display
XML Generation	Generate XML
Cut Control	Control Cut
Copy Control	Control Copy
Paste Control	Control Paste
Move of Control	Control Move
Removing of Control	Control Remove
Drag and Drop Facility	Drag and Drop
Save Flow Graph	Flow Graph Save

Open Previous Session	Previous Session Open

The use cases description is provided in the Appendix-D.

3.1.1.3. Usecase Diagram

In usecase diagram use case are drawn as an ellipse, the name of the use case usually written inside the ellipse.

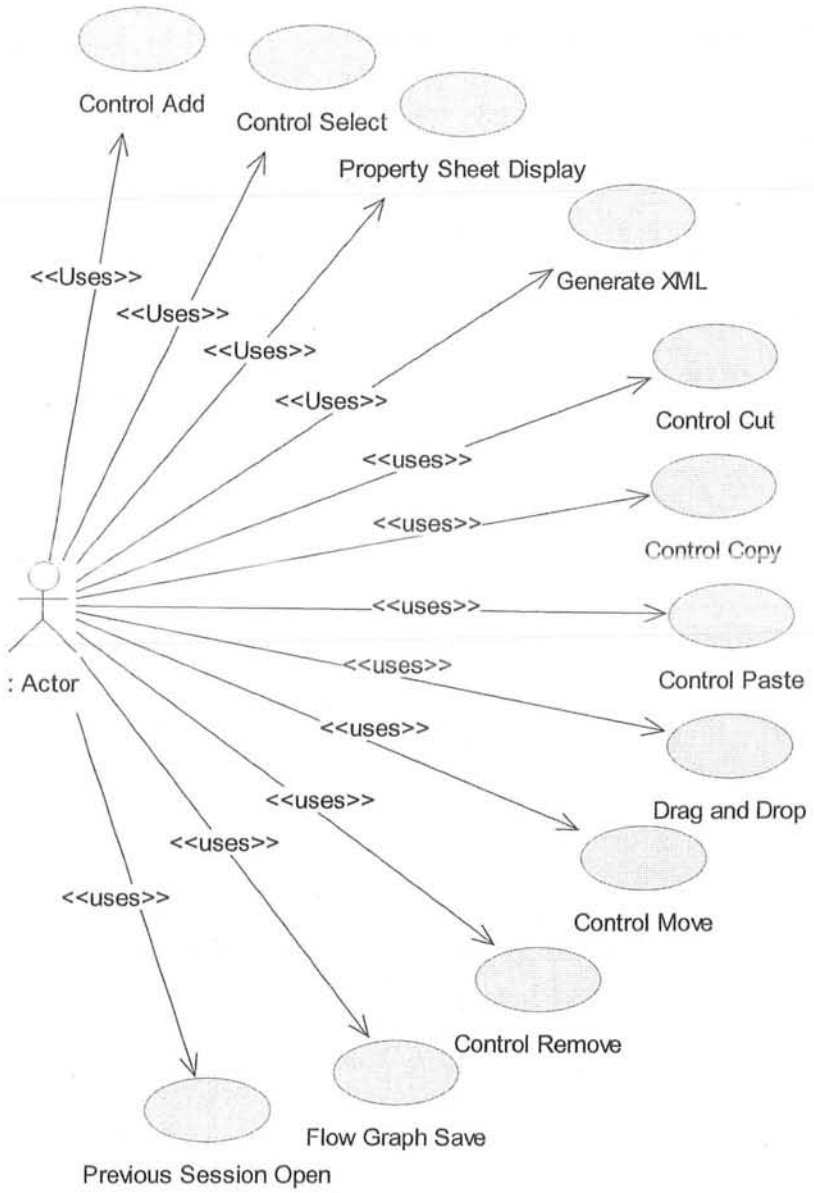


Fig.3.1 Usecase Diagram

3.1.1.3.1. System Sequence Diagrams and Operations Contracts

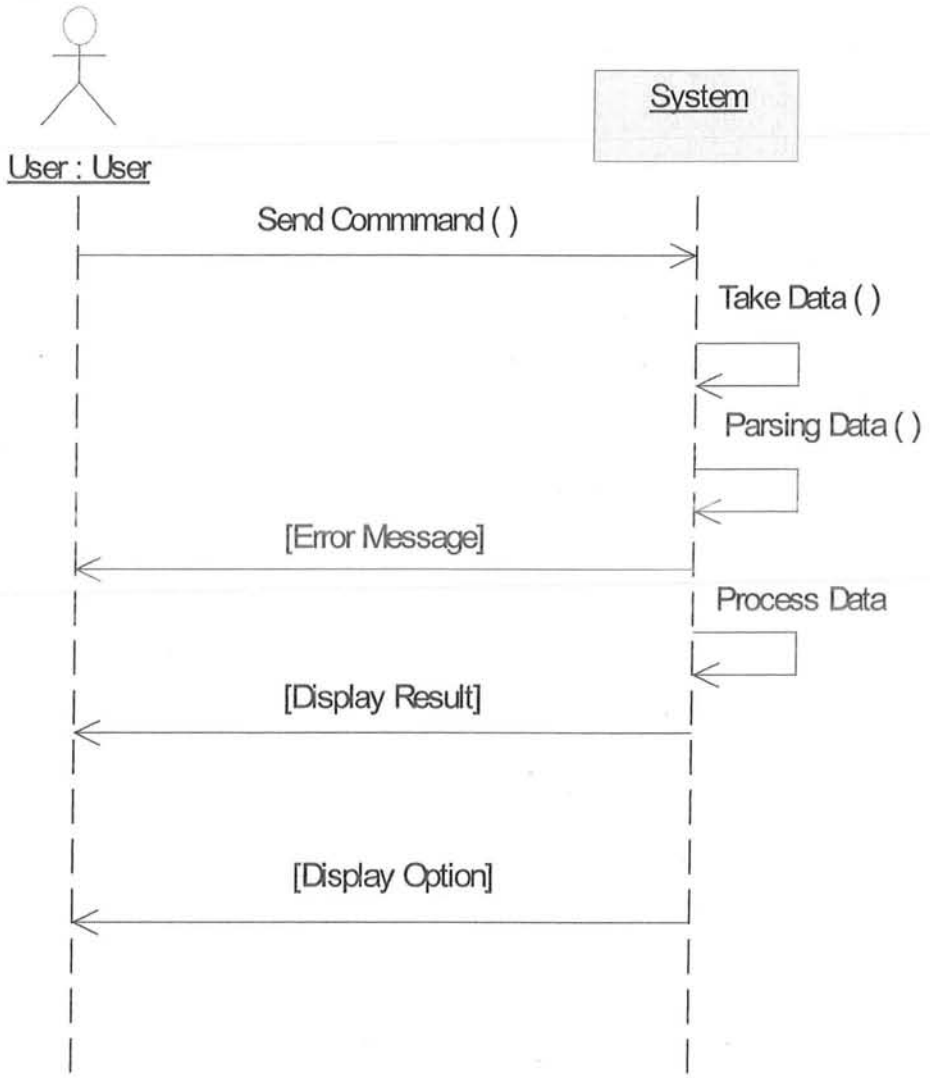


Fig 3.2 Generate XML System Sequence Diagram

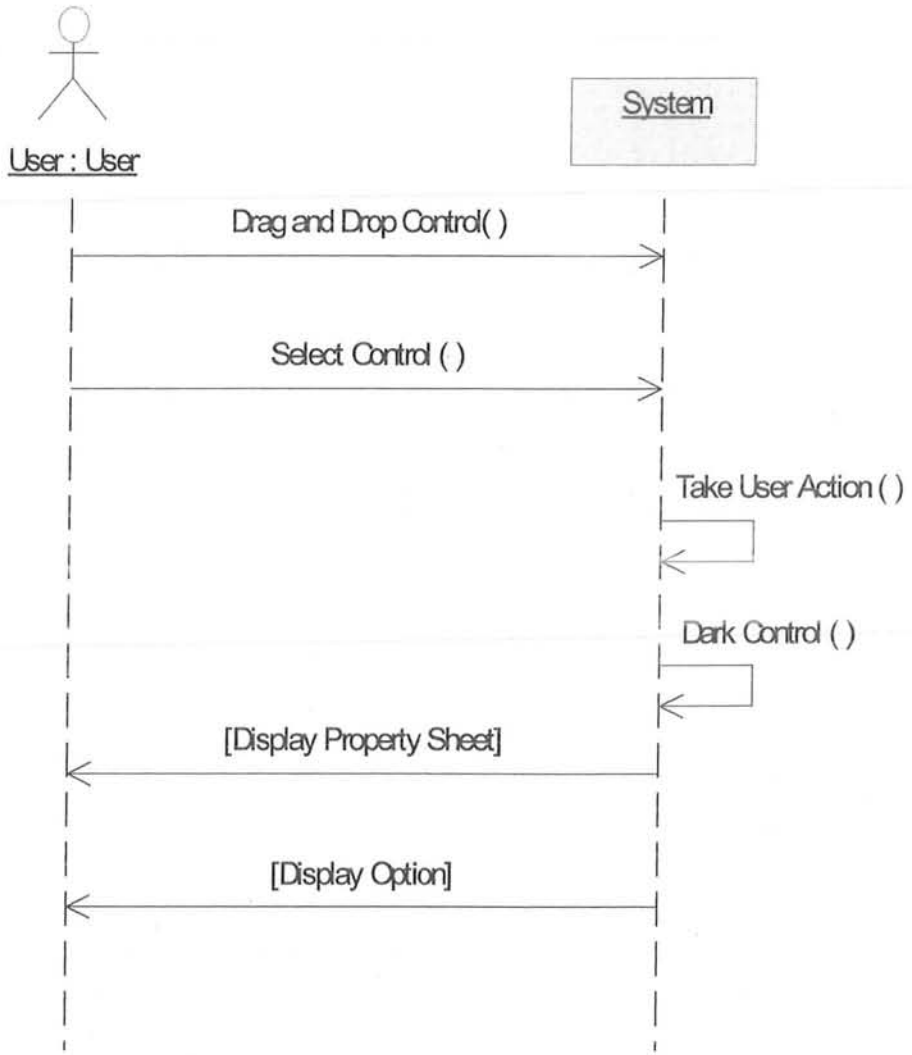


Fig.3.3 Display Property Sheet System Sequence Diagram

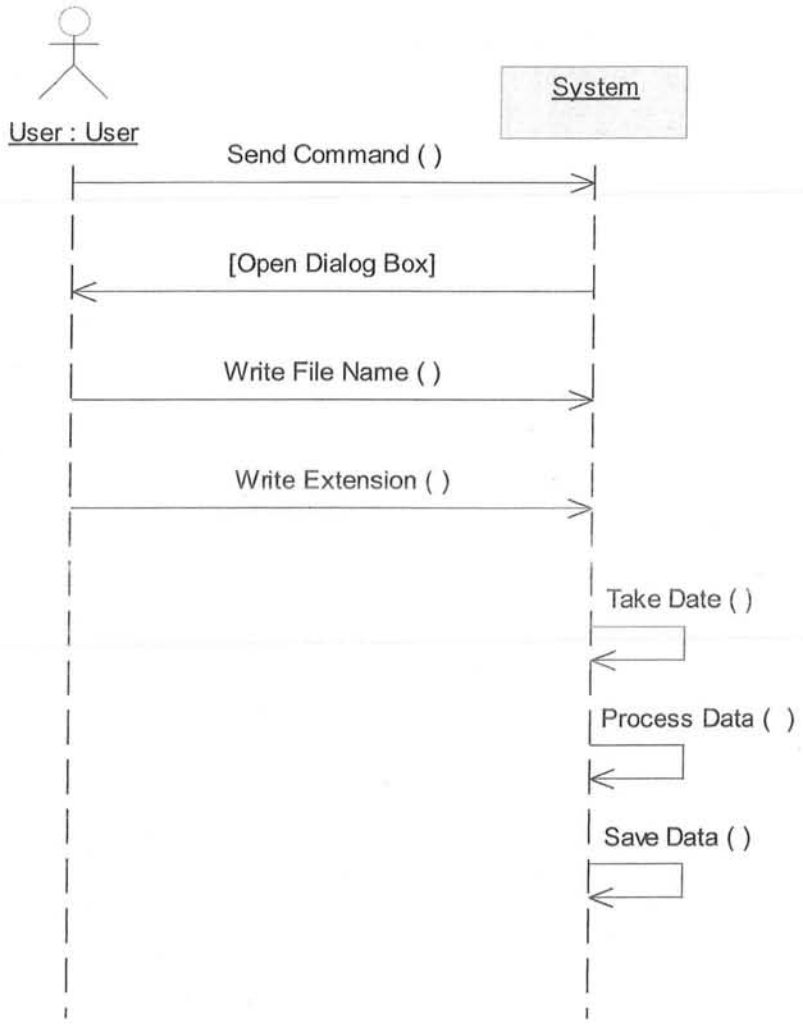


Fig.3.4 Save Flow Graph System Sequence Diagram

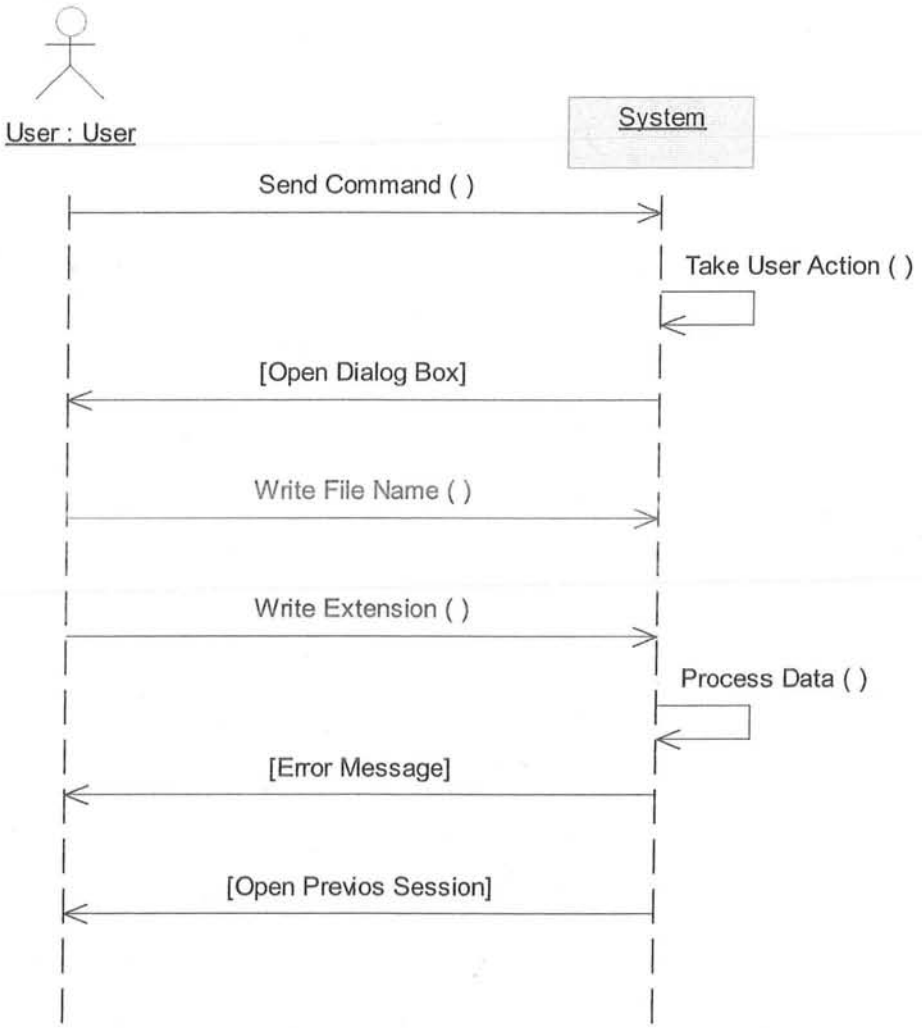


Fig.3.5. Open Previous Session System Sequence Diagram

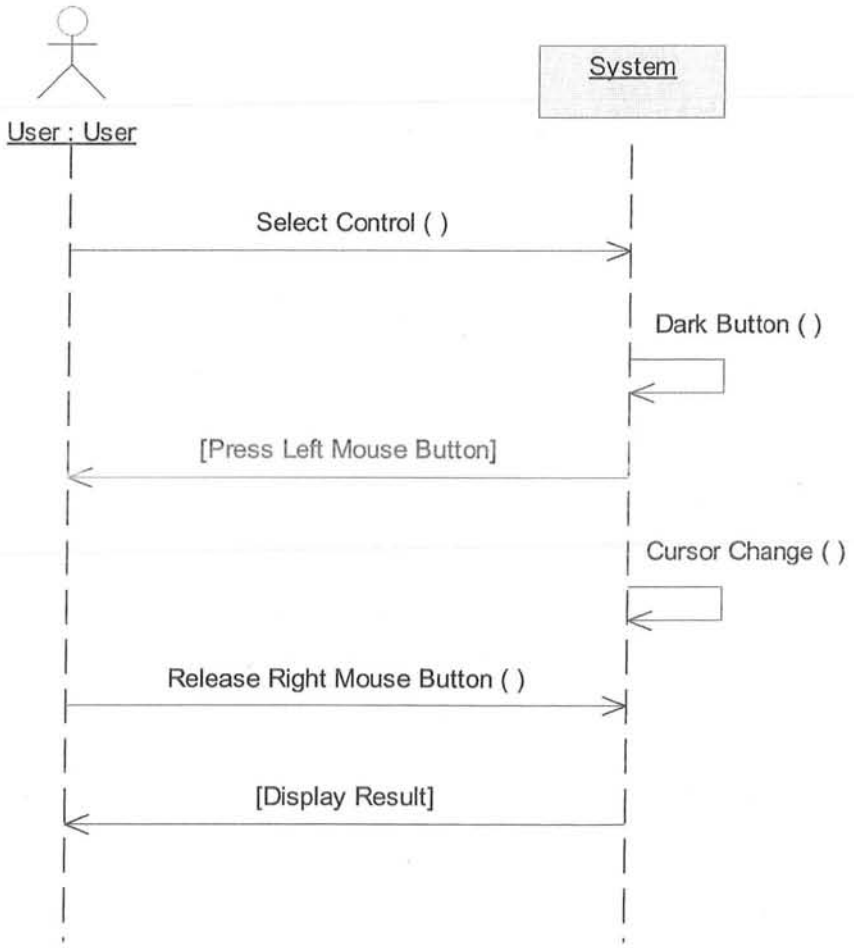


Fig.3.6. Drag and Drop System Sequence Diagram

ContratCo1: Generate XML

Operation:	GenerateXML(CVSTelpCtrlBase *pCtrl)
Cross-Reference:	Use case: GenerateXML
Pre-Condition:	-Controls must be added in the canvas -Flow graph must be created
Post-Condition:	- XML file will be generating.

Contract CO2: Flow Graph Save

Operation:	SaveFlowGraph (String SessionName)
Cross-Reference:	Use Case: Flow Graph Save
Pre-Condition:	- Save option in the file menu in acting state - Save option will be available in the toolbar.
Post-Condition:	- Dialog box will be open - Write the file name - Write the extension name

Contract CO3: Drag and Drop

Operation:	Drag and Drop (uint BtnIDPressed)
Cross-Reference:	Use Case: Drag and Drop
Pre-Condition:	- Mouse in acting state
Post-Condition:	- Specified control will be in the dropped location.

Contract CO4: Previous Session Open

Operation:	PreviousSessionOpen (file name to be browsed)
Cross-Reference:	Use Case: Previous Session Open
Pre-Condition:	- Mouse in acting state
Post-Condition:	- Specified control will be in the dropped location.

3.2. Structural Model

Object oriented methodology provides information hiding, abstraction and modularity, by viewing a software system as a set of interconnecting data objects. These data objects have their own private status represented by a set of attributes (data members) and a set of member functions to change their state. The data objects communicate with one another through messages and by calling each other's member functions directly [PRE01].

3.2.1 Identification of Classes

Following classes are used to create ActiveX controls

CVSDialCtrl	CVSSwitchCtrl
CVSStopCtrl	CVSStartCtrl
CVSIFCtrl	CVSLoopCtrl
CVSWaitCtrl	CVSGetDigitCtrl
CVSRecordCtrl	CVSDisconnectCtrl
CVSGetHangupCtrl	CVSTelpCtrlBase

Following classes are used to design the system.

CVSPropertyView	CMainFrame
CPicture	CPropList
CMyDnDToolBarDoc	CMyDnDToolBarView

4.3. Domain Model or Conceptual Model

A domain model is visual representation of conceptual classes or a real world object in the domain of interest [FLO96].This is also called a Conceptual Model.

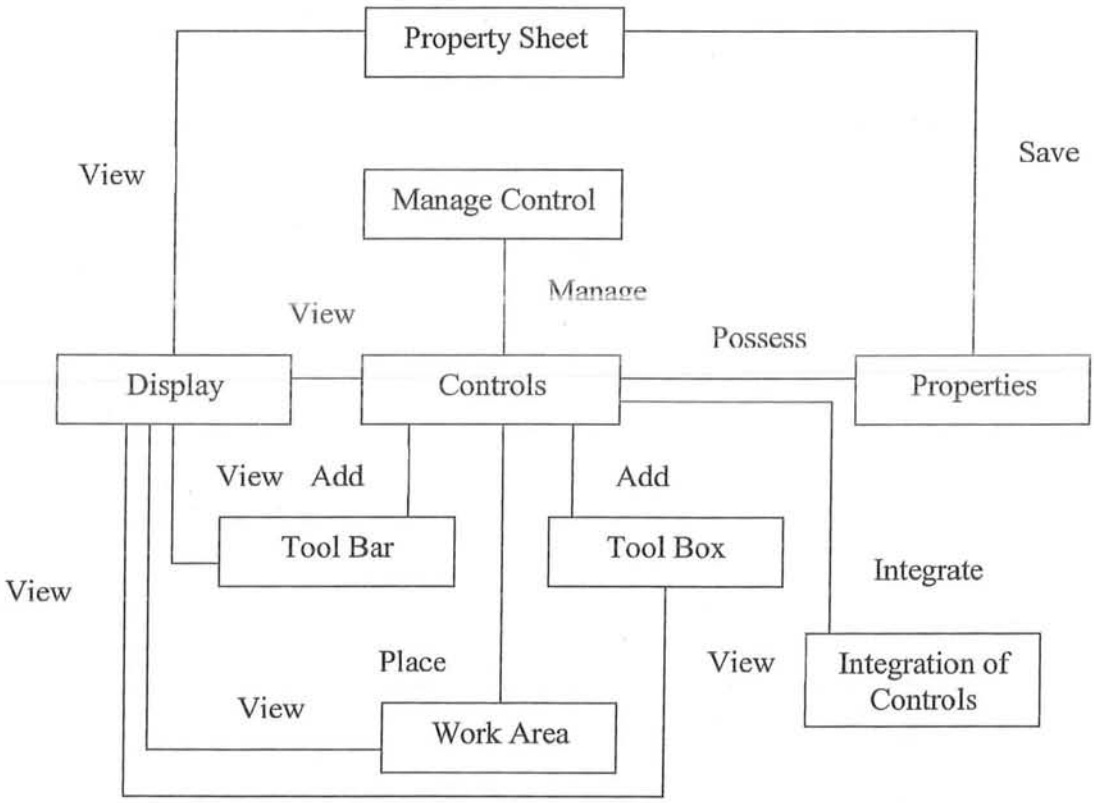


Fig3.4 Domain Model

3.3 Requirement specification

Requirement specification is detailed and precise description of the system requirements. The requirement specification is mainly divided into two categories. These are functional requirements specification and non-functional requirements specification. There are many techniques to document the functional requirements specification of the system to be developed. These include:

- Structured Natural Language specification.
- Design Descriptive language specification (PDL).
- Requirement Specification Language.
- Graphical Notation Specification.
- Formal Specification (Z-VDM).[SOM00]

For VLCT (Visual Language for Computer Telephony), I found the Structured Natural Language Specification the most appropriate. It is a restricted form of natural language for requirement specification. Its proper structure maintains the expressiveness and understandability of the natural language. Thus it is acceptable at both the managerial and technical levels.

A type of structured natural language specification is the Form Based Specification. In this technique standard forms or templates are used to express the detailed requirements. Form may be designed around the objects manipulated by the system, the functions performed by the system or the events processed by the system. Functionally oriented specifications are easily understandable to the customer. As a matter of fact most of the customers are interested in the functionality of the system. They want to see the functionality of the system in each product during the development process.

Standard form for the functionally oriented form based specification is as follows:

Requirement name: Add Controls

Function:	Container should add the controls
Description:	The user will be able to add the controls in the editing window using drag and drop facility
Inputs:	Mouse select the control in the toolbox.
Source:	The controls will be added by user.
Outputs:	Relevant control must be in the canvas and its relevant properties are also shown in the container.
Destination:	Editing window in the container.
Requires:	Mouse in acting stage.
Pre-conditions:	There must have some controls in the toolbox.
Post-conditions:	NIL

Requirement name: Select Control

Function:	Container should select the control
Description:	If user want to select the control for adding or for any purpose than he/she will be able to select the control using the key board or using the mouse.
Inputs:	NIL
Source:	The control will be selected by user.
Outputs:	If the control is selected than its button must be bold.
Destination:	Tool bar in the container.
Requires:	Mouse or keyboard in acting state.
Pre-conditions:	There must have some controls in the toolbox.
Post-conditions:	NIL

Requirement name: Property Sheet Display

Function:	Container should have Property sheet of the relevant control.
Description:	When user added any control in the canvas of the container than its relevant properties must be shown in the container.
Inputs:	Control must be selected when added in editing window of the container.
Source:	The control will be selected by user.
Outputs:	Property sheet of relevant control should be displayed in the container.
Destination:	Tool bar in the container.
Requires:	The control must be selected.
Pre-conditions:	There must have some controls in the toolbox.
Post-conditions:	NIL

Requirement name: Generate XML

Function:	Container should generate XML
Description:	When any flow graph should be generated in the canvas of the container than its XML file must be generated.
Inputs:	There must be any flow graph in the canvas.
Source:	The canvas of the container.
Outputs:	The file will be added in the specified location.
Destination:	System database
Requires:	NIL
Pre-conditions:	There must have flow graph in the toolbars.
Post-conditions:	NIL

Requirement name: Copying of controls

Function:	Container should have facility of copying of controls
Description:	If user added any control in the canvas and wants to copy the control in any location of canvas than user must have the facility of copying of controls.
Inputs:	The control must be selected in the canvas.
Source:	The canvas of the container.
Outputs:	The control will be added in the specified location.
Destination:	The canvas of the container.
Requires:	NIL
Pre-conditions:	At least one control in the canvas.
Post-conditions:	NIL

Requirement name: Removing of controls

Function:	Container should have facility of removing of controls
Description:	If user added any control in the canvas and than wants to remove any control from the canvas than there must be facility of removing of specified control.
Inputs:	The control must be selected from the canvas.
Source:	The canvas of the container.
Outputs:	The control will be removed from the specified location.
Destination:	The canvas of the container.
Requires:	NIL
Pre-conditions:	At least one control in the canvas.
Post-conditions:	NIL

Requirement name: Pasting of controls

Function:	Container should have facility of Pasting of controls
Description:	If user copy any control placed in the canvas and after that user wants to past in any where of the canvas than container must give the facility of pasting of control.
Inputs:	Selected control must be copied.
Source:	The canvas of the container.
Outputs:	Paste the control at the user specified position.
Destination:	The canvas of the container.
Requires:	Some controls must exist in the canvas.
Pre-conditions:	User must copy any control in the canvas.
Post-conditions:	NIL

Requirement name: Moving of controls

Function:	Container should have facility of moving of controls
Description:	If user wants to move the control from any where in the canvas. Than container give the facility of moving of controls in the canvas.
Inputs:	The control must be selected in the canvas.
Source:	The editing window of the container.
Outputs:	The control will be removed at the specified location in the canvas.
Destination:	Canvas of container.
Requires:	Mouse in acting state.
Pre-conditions:	User must select the control.
Post-conditions:	NIL

Requirement name: Printing of controls

Function:	Container should have facility of printing of controls.
Description:	If user draws any flow graph in the canvas and wants to take print for this flow graph than container give the facility of printing.
Inputs:	User must select the print option.
Source:	The print button in the toolbar and printer.
Outputs:	The required flow graph will be printed.
Destination:	Printer
Requires:	Printer in acting state.
Pre-conditions:	Selection of print option.
Post-conditions:	NIL

Requirement name: Save Control

Function:	Container should Save Control
Description:	If user draw any flow graph in the canvas and want to save this flow graph than container give the facility of printing.
Inputs:	User must select the save option
Source:	The save button in the toolbar.
Outputs:	The required flow graph will be saved.
Destination:	The system Database.
Requires:	NIL
Pre-conditions:	Selection of save option
Post-conditions:	NIL

Requirement name: Open Previous Session

Function:	Container should Open Previous Session
Description:	If user want to open the session that user was saved previously than container give the facility of open. After opening the precious session we can see pattern is same as the pattern user can save.
Inputs:	Select the Dialog Box of Open in the File menu. Write the correct file name and extension.
Source:	The canvas of the container.
Outputs:	The relevant file will be open.
Destination:	Canvas of the container.
Requires:	NIL
Pre-conditions:	User must select the open option in the container.
Post-conditions:	NIL

Conclusion

This chapter describes the system requirements analysis phase, which transforms the requirements to UML representations that describe the characteristics of the system. The system analysis patterns are further transformed to various design patterns.

Chapter 4

System Desgin

4.1 Introduction

The software design is aimed at finding the conceptual solution that fulfils the requirements, rather than its implementation. For example description of database schema and software objects.

The object oriented design emphasis on defining software objects and how they collaborate to fulfill the requirements

The software design is a process of finding

- How the classes described in the object oriented analysis will work together in the software.
- How links and associations should be implemented
- How purchased or otherwise acquired components will be can help
- Improving our estimates of cost and time to market.
- Assessing the prerequisite in terms of labor and infrastructure

A successful design is one that can translate the requirements into finished product. A good design means that the software complexities are reduced .The design must contain following characteristics [PRE97].

- The design ensures the accurate translation of customer's requirements.
- It should be readable and understandable.
- It should provide the complete picture of software.
- Design should form the base for programming and maintenance.

UML specification of An Object oriented design should include following models.

- Static Model
- Interactive Model
- Behavioral Model

4.2 Static Modeling

The static modeling is aimed at building the class model, which involves identifying the classes that should exit in our system. This is major part in designing an object-oriented system. The system we build consists of software objects that interact

with each other to fulfill the requirements. The secret of object-oriented design is to end up with the class model which doest not misrepresent the conceptual reality of system domain.

The static model includes

- **Identifying Classes**

There are four types of classes

Conceptual Class

The conceptual or essential perspective

CMainFrame CPicture CPropList

Software Class

A class representing a specification or implementation, perspective of a software component.

CVSDialCtrl
CVSStopCtrl
CVSIFCtrl
CVSWaitCtrl
CVSRecordCtrl
CVSGetHangupCtrl
CVSSwitchCtrl
CVSStartCtrl
CVSLoopCtrl
CVSGetDigitCtrl
CVSDisconnectCtrl
CVSTelpCtrlBase

Design Class

It is a synonym for software class.

CVSPropertyView
CPicture
CMyDnDToolBarDoc
CMainFrame
CPropList
CMyDnDToolBarView

Implementation Class

A class implemented in an object oriented language.

CVSTelpCtrlBase

- **Finding Associations and Relationships**

An association is a relationship between types (or more specially, instances of those types) that indicates some meaningful and interesting connection, these associations are shown in domain model in chapter 3, figure 3.4.

- **Creating Class Diagram**

A Class Diagram illustrates the specifications for software classes and interfaces (for example Java interfaces) in an application.

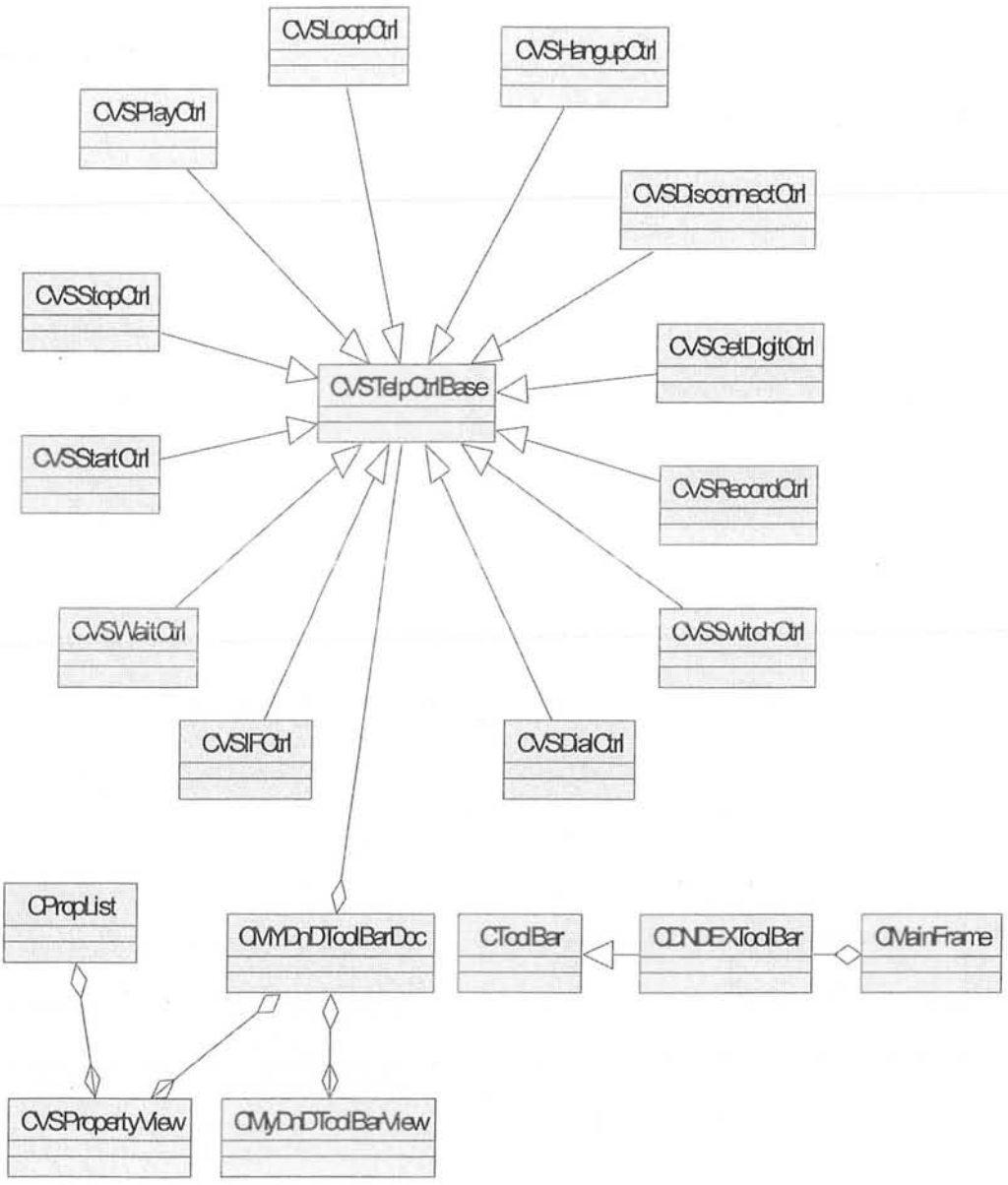


Fig4.1 Class Diagram

4.2.1 Class Relationships

The objects of classes communicate with each other to perform any function. The type of relationship at class level defines this communication. The interaction relationships are defined in the class relationship diagram. Class diagram is UML standard for describing the class infractions. There are different UML standards notations to represent classes. There are normally three types of relations between the classes

- Association
- Generalization (Inheritance)
- Aggregation (containment)

Different constructs used in the class diagram are defined in glossary; their visual representations are mention in Visual Glossary.

4.3 Interactive Modeling

Interactive modeling emphasis on the illustration of software objects interaction via messages.

The Interaction diagram/Model is generalization of two UML diagrams types; both can be used to express similar messages interaction. These are

- Collaboration Diagram
- Sequence Diagram

4.3.1 Collaboration Diagram

This diagram represents the interaction of objects in a graph or network format in which object can be formed or placed any where in the diagram.

4.3.2 Sequence Diagram

This diagram illustrates the interaction in kind of fence format in which each new object is added to the right.

However there is flexibility of choice in using any of them.

4.3.3 Selection of Interactive Model

I have preferred the sequence diagram because of following reasons.

1. The sequence Diagram clearly shows the sequence or time ordering of messages
2. Simple Notation

And my reason of rejecting the collaboration was motivated by following facts about the collaboration diagram.

1. It is difficult to see the sequence of messages.
2. It uses the more complex notation

Sequence Diagrams

The sequence diagrams are given in Appendix-F.

4.4 System Behavior

The system behavior is the description of what a system does with out explaining how it does it.

One part of that description is the sequence diagram. Other parts include the Use-Cases, system contracts and State Diagram.

4.4.1 State Diagram

The state diagram represents the events and state of state of things-transaction, use cases, people and so forth.

4.4.5 Selection of Behavioral Model

I have chosen the state dependent-class state diagram for my system due to following reason.

In general, the business information systems have minority of state dependent class .By the contrast the process control and Telecommunication domains have many state-dependent objects.

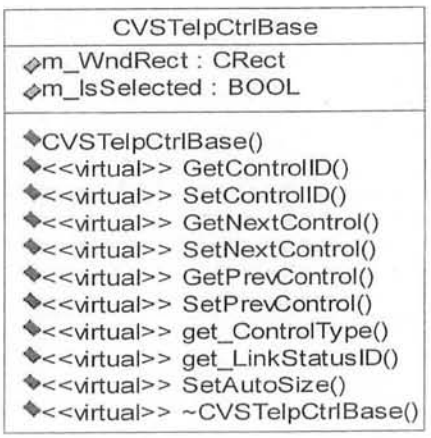
As I have system from telecom-related to some sort so I will be creating the state diagrams for the state-dependent classes.

4.5. State Diagrams

The state Diagrams are given in the Appendix-G.

4.1. Design of Attributes and Operations.

We can mention the design the attributes and operations of different classes as under.

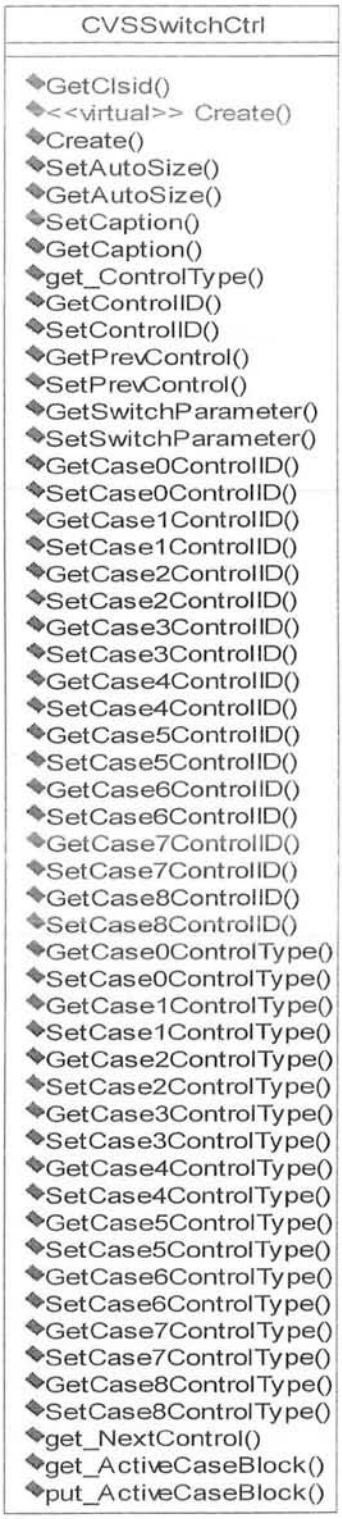
CVSIFCtrl This class is used to construct the if control CVSSIFCtrl is wrapping the functionality switch ActiveX control. We can manage all the checks related to if statement in this class.	
CVSTelpCtrlBase It is a base class All the classes related to controls are inherited to this class.	

CVSSwitchCtrl

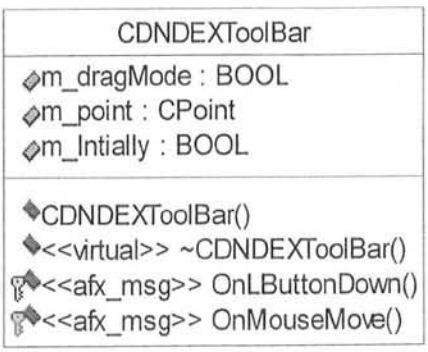
This class is used to construct the switch control

CVSSwitchCtrl is wrapping the functionality switch ActiveX control.

We can manage all the checks related to the switch statement in this class.

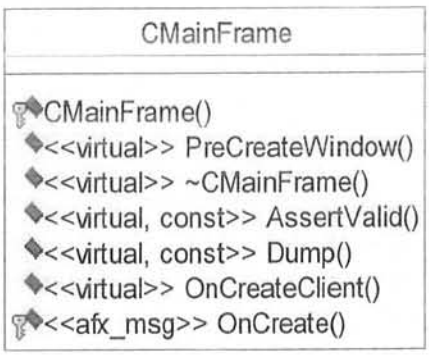


CVSLoopCtrl This class is used to construct the loop control CVSloopCtrl is wrapping the functionality switch ActiveX control. We can manage all the checks related to the conditional loop.	
CVSHangupCtrl This class is used to construct the HangUp control CVSHangupCtrl is wrapping the functionality switch ActiveX control. We can discuss the hanging options of the computer telephony in this class.	

CVSGetDigitCtrl This class is used to construct the GetDigit control CVSGetDigitCtrl is wrapping the functionality switch ActiveX control. In this class we can manage how to get the digit.	
CVSDisconnectCtrl This class is used to construct the Disconnect control CVSDisconnectCtrl is wrapping the functionality switch ActiveX control.	
CDMDEXToolBar Using this class we can achieve drag and drop functionality.	

CVSDialCtrl This class is used to construct the Dial control CVSDialCtrl is wrapping the functionality switch ActiveX control.	
CVSRecordCtrl This class is used to construct the Record control CVSRecordCtrl is wrapping the functionality switch ActiveX control.	
CVSStartCtrl This class is used to construct the start control CVSStartCtrl is wrapping the functionality switch ActiveX control.	

CVSStopCtrl This class is used to construct the Stop control CVSStopCtrl is wrapping the functionality switch ActiveX control.	
CVSWaitCtrl This class is used to construct the wait control CVSWaitCtrl is wrapping the functionality switch ActiveX control.	
CPropList This class provide interface to the property sheet. Set the properties of any control. Get the properties of any control. Show the properties of control. If property of any control change than it can modify the container.	

CVSPlayCtrl This class is used to construct the Play control CVSPlayCtrol is wrapping the functionality switch ActiveX control.	
CMainFrame This class is used to split window into two portions.	

CMyDnToolBarView

This class is used to represents the all the documents.

This class creates control.

Objects are also created in CMyDnToolBarView class.

It can also handle all the windows messages.

CVSPropertyView This class is used to provide the the view of the property and canvas.	
CMyDnDToolBarDoc *It is a document class* *It is used to store all the data related to this system.* *In this class we can manage the database of the system.* *Or in this class we can manage the data structure for this system.*	

User Interface

About User Interface of the system, we show main page of the system and all parts of Visual Language for Computer Telephony in **Appendix – H**

Conclusion

In this chapter we are discussing the design of attributes and operations of all the classes of Visual Language for Computer Telephony and about User Interface.

Chapter 5

System Implementation

5.1 Platform, Programming language and Tool Selection

Platform, programming language and related tools are selected according to nature and requirement of this project.

5.1.1 Platform Selection

Platform selection in the software implementation phase is very critical. When choosing an operating System you must ensure that all those features that your software requires are supported by operating system. The selection of Platform should ensure that it will not create any Copy Rights problems when software is deployed.

The Platform I have selected for development is Microsoft® Windows XP.The reason for choosing this operating system is that it provides the feature and capabilities that are required during the software development phase. Also the SDK documentation I am working with, suggests that Windows® XP is good choice as for the Operating system selection is concerned. Although I will be using Windows® 2000 and Windows®98 for software testing purposes.

The built software will be able to run on the Microsoft Windows based operating system i.e. Windows® 98, Windows® 2000 and Windows XP.

5.1.2 Tool Selection

The Tool selected for development of this project is Visual C++ and Microsoft XML 4.0.

5.1.3 Code Documentation Standard

Code of project has been documented well with coding standard as told by external project in charge. These standards are described below

- Each Class name begins with capital letter C and first letter of class name is also capitalized.
- Fully object oriented design is followed in implementation of this system.
- A general Design structure is used for the system, this design generalization will enable future enhancements.

- Each Class is well documented in form of class documentation. Documentation of these classes is provided in the appendices.

5.1.3.1 Visual C++ Features

Visual C++ contains several unique language features in high demand among advanced programmers. These language features contribute to making C++ the most powerful of all the Microsoft-provided Visual Studio languages:

- **Templates.** Templates comprise several compile-time language features which are largely unique to C++, enabling many code reuse and performance enhancing capabilities.
- **Pointers.** Pointers give C++ developers direct access to machine level memory locations, enabling the highest performing applications.
- **Multiple inheritances.** C++ offers developers a full gamut of object-oriented programming (OOP) features for implementing the widest variety of OOP programming patterns.
- **Compile-time attributes.** C++ attributes provide a shorthand means of writing highly tuned repetitive boilerplate code using a simple and robust syntax.

5.1.3.2 Unique Environment Features

Visual C++ also contains a variety of environment features that assist programmers in creating flexible and powerful applications, including:

- **Optimizing Compiler.** The Visual C++ Optimizing Compilers tune applications for a variety of scenarios, including target machine environments, floating point calculations.

- **Runtime Code Security Checks**. Programmers can write more-secure native Windows-based applications using advanced compiler features that help protect applications from malicious attack.

- **32- and 64-bit support**. Visual C++ compilers are available for multiple hardware platforms including 32- and 64-bit Intel and AMD microprocessors, and a variety of embedded device microprocessors, enabling truly scalable applications.

- **Advanced Error Reporting.** Applications are always susceptible to programmer error. Visual C++ enables developers to more easily identify and correct bugs, even in deployed code, with Mini dump technology.

- **Advanced debugging.** The Visual Studio Debugger and Visual C++ provide seamless support for simultaneously debugging both native and managed code.

5.2 Component Diagram

A component represents a physical piece of program code, either as source code or DLL. Following is a component diagram which shows the interrelations between components.

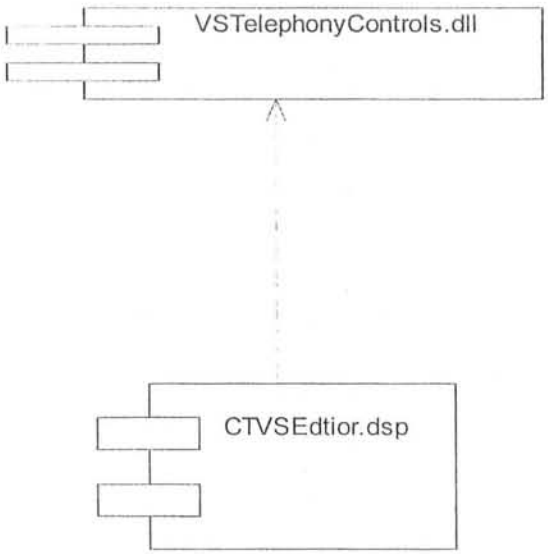


Fig 5.1 Component Diagram

Conclusion

This provides the brief overview of tools and technologies used in the development of the system. The coding convention used in the software source code is also mentioned in this chapter.

Chapter 6

System Testing

6.1. Software Testing Introduction

Simply stated, quality is very important. It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality. A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version. Quality is a distinguishing attribute of a system indicating the degree of excellence. The importance of software testing can not be overemphasized. Once the source code has been generated, software must be tested to allow errors to be identified and removed before the delivery of software. While it is not possible to remove the every error in large software package, our goal is to remove as many as possible in the early software development cycle.

6.2. Software Testing Objectives

Testing of the application is done to achieve following objectives.

- Execution of the program is done with intent to find the errors in the program.
- Test Cases are designed so that these have high probability of finding an as-yet-undiscovered error.

6.3 Boundary Value Analysis

Boundary value analysis is a test case design technique that complements equivalence partitioning. Rather than selecting any element of an equivalence class, BVA leads to the selection of test cases at the "edge" of the class.

The reason to use this technique is that a greater number of errors tend to occur at the boundaries of the input domain rather than in the "center". The test cases are designed so that these test cases satisfy the requirements of BVA also.

6.4 Test Cases

Test cases are designed so that these maps with following functional requirements of the system.

R.1. Drag and Drop

R.2. Select controls

R.3. Property Sheet

R.4. Generate XML

R.5 Copy controls

R.6. Paste Controls

R.7. Cut Controls

R.8. Remove Controls

R.9. Move Controls

R.10. Save Flow Graph

R.11. Open Previous Session

Req. No.	**R.1**
Date of Test	8/12/2004
Program	VLCT.dsp
Description	The user will be able to add the controls in the canvas using drag and drop facility
Input	Selection of any control using mouse
Expected Output	The specified control should be drag and drop.
Actual Output	The specified control is added in the canvas using drag and drop.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that there were no errors in "Drag and Drop"

Req. No.	**R.2**
Date of Test	15/12/2004
Program	VLCT.dsp
Description	If user want to select the control for adding or for any purpose than he/she will be able to select the control using the key board or mouse.
Input	Go to the toolbox using mouse
Expected Output	Specified control should be selected.

Actual Output	Same as expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is achieved.

Req. No.	**R.3**
Date of Test	18/12/2004
Program	VLCT.dsp
Description	When user added any control in the editing window of the container than its relevant properties must be shown in the container.
Input	One control must be selected.
Expected Output	Property related to the specified control should be shown in the canvas.
Actual Output	Same as expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained.

Req. No.	**R.4**
Date of Test	25/12/2004
Program	VLCT.dsp
Description	When any flow graph should be generated in the editing window of the container than it XML file must be generated.
Input	To design flow graph in the canvas.
Expected Output	XML file should be generated.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	**R.5**
Date of Test	1/1/2005

Program	VLCT.dsp
Description	If user added any control in the canvas and wants to copy the control in any location of canvas than user must have the facility of copy.
Input	User chooses the copy button in the toolbar.
Expected Output	If we use the past option from the past button at the toolbar than specified control should be pasted.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	R.6
Date of Test	1/1/2005
Program	VLCT.dsp
Description	If user copy any control placed in the canvas and wants to past in any where of the canvas than container must give the facility of pasting of control.
Input	User chooses the copy button in the toolbar.
Expected Output	If we use the past option from the past button at the toolbar than specified control should be pasted.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	R.7
Date of Test	1/1/2005
Program	VLCT.dsp
Description	If user cut any control from any location of the canvas than the specified control must be cut from any location of the canvas
Input	User chooses the copy button in the toolbar.

Expected Output	If we use the past option from the past button at the toolbar than specified control should be pasted.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	**R.8**
Date of Test	18/1/2005
Program	VLCT.dsp
Description	If user added any control in the canvas and after that if user want to remove any control from the canvas than there must be facility of removing of control.
Input	The control must be selected from the canvas.
Expected Output	The specified control should be remove from the canvas
Actual Output	Same as expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained.

Req. No.	**R.9**
Date of Test	25/12/2004
Program	VLCT.dsp
Description	If user wants to move the control from any where in the canvas. Than container give the facility of moving of controls in the canvas.
Input	The control must be selected in the canvas.
Expected Output	The control should be moved from the current location to the specified location.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	R.10
Date of Test	1/1/2005
Program	VLCT.dsp
Description	If user draw any flow graph in the canvas and want to save this flow graph than container give the facility of saving.
Input	User must selected the save option
Expected Output	The specified file should be saved.
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Req. No.	R.11
Date of Test	1/1/2005
Program	VLCT.dsp
Description	If user want to open the session that user was saved previously than container give the facility of open. After opening the precious session we can see the pattern is same as the pattern user can save.
Input	Select the Dialog Box of Open in the File menu.
Expected Output	The required file should be open
Actual Output	The actual out obtained turned out to same as was expected.
Test Conductor	Muhammad Siddique Bhatti

Result of Test: Test showed that desired result is obtained

Conclusion

This chapter provides the introduction to the various testing strategies applied to the Visual Language for Computer Telephony.

References

EFRENCES:

[SOM00] Somerville, I., Software Engineering, *Publisher*, 6th Edition 2000

[PRE01] Pressman, R.S., Software Engineering A Practitioner's Approach, *McGraw Hills*, 2001

[ALF91] *Alford, M.W.* A Requirement Engineering Methodology for Real Time System, IEEE *Tran on Software Engineering* SE-3 (1) 60-90[123], 1991

[DAV93] *Davis*, A.M., Software Requirements: Analysis and Specification. Englewood Cliff NJ: *Prentice Hall*

[EAS93] *Easter Brook, S.* Domain Modeling with Hierarchies of alternative Viewpoint complete

[GIL98] Gil Held, Voice and Data Internetworking, McGraw-Hill

[HOO98] Hoofer, G., System Analysis and Designing, *Prentice Hall*

Bibliography

Bibliography

[SOM00] Sommerville Ian, "Software Engineering", Addison Wesley, 1998.

[BER99] Oestereich Bernd, "Developing Software with UML", Addison Wesley, 1999.

[CHA98] Chapman Davis, "Teach Yourself Visual C++ 6 in 21 Days", SAMS, 1998.

[TOT98] Toth Viktor, "Programming Windows 98/NT Unleashed", SAMS, 1998.

[STR99] Eric Stroo, "Desktop Applications with Microsoft Visual C++ 6.0", Microsoft Press, 1999.

[PRE97] Pressman Roger S., "Software Engineering A practitioner's Approach", IEEE Software, 1997.

[LAR01] Craig Larman, "Applying UML and Patterns", Second Edition.

Webliography

WEBLIOGRAPHY:

http://www.google.com

http://www.microsoft.com

http://www.whatis.com

http://www.ultavesta.com

http://www.openh323.com

http://www.iconnecthere.com

http://www.cisco.com

http://www.vovida.com

Process Model

The process model used for the development of the Visual Language for Computer Telephony is the "Spiral Model". The spiral model has the following prominent features:

- **Requirement analysis**

  The Spiral model uses an incremental approach toward the software development. The Visual Language has the following major components to be developed.

  - Graphical User Interface for Computer Telephony.
  - A user manual to help the user use the system and relavent theoritical concepts.

  The Visual Language for Computer Telephony needs some theoritical concepts from Telecommunications, which are quite new for me as a software developer. So this model provides me the opportunity to do requirements analysis in a cyclic fashion.

- **Generic model**

  In its essence, the spiral model is a generic model. It incorporates all other process models likes "Waterfall" model and the "Incremental" model.

- **Flexible**

  Each loop in the spiral model represents a phase of the software process. The inner most loop might be concerned with system feasiblity, the next loop with system requirements definition, the next loop with system design. Then other loops may be a combination of all these loops to achieve a potential increment.

  In Visual Language for Computer Telephony, I focused on the requirements difinition and specification supplemented by the "Throw away prototypes" of the software.

- **Switching to other models**

The Spiral model provides the option of using a different model at each phase of the spiral. For example the phase involving development of the GUI can use "Evolutionary Prototyping Model" and some safety critical phase may use the "Formal Transformation" technique.

- **Risk consideration**

  The risk analysis is done at each phase to decide whether to continue work on the current activity or switch to next increment.

  I have four months to complete the project. I also have to study two subjects in the department, both the subjects are optional. The time clash resolution from the "Department's Time Table Incharge" results in frequent rescheduling of the time slots. This suffers my project scheduling and deadlines. The Spiral model provides the opportunity to adjust my project scheduling accordingly.

Risk Management

Risk#	Risk	Category	Imapact
1	Inability to meet the dead line	Project risk	critical
2	System Crash	Product Risk	critical
3	Lack of training on new tools to be used	The Development Envoirment Risk	marginal

Risk Mitigation, Monitoring and Management Plan

Risk#1

Mitigation

Schedule of each task to be done in the project will prepared and followed.

Monitoring

In following the "Spiral Model" more importance will be given to schedule risks.

Management

- If inspite of all these efforts I failed to meet the deadline then I will prepare throwaway prototypes made for modules not yet completed and made some enhancement to them to integrate them with developed system.
- I will ignore the non-functional requirements.

Risk#2

Mitigation

I will make sure before starting the project that the system I am going to use is virus free and its parts are not out of order.

Monitoring

- System scan will be performed at the start and end of the day to ensure no virus are present in system
- Taking regular backup of project.

Management

The backup taken during the monitoring plan will be used in case of system crash.

Risk#3

Mitigation

- Online help will be use to master tool and technology
- Manual ,books or documentation of tool helps a lot

Monitoring

I will purchase good books search the relevant material on the net.

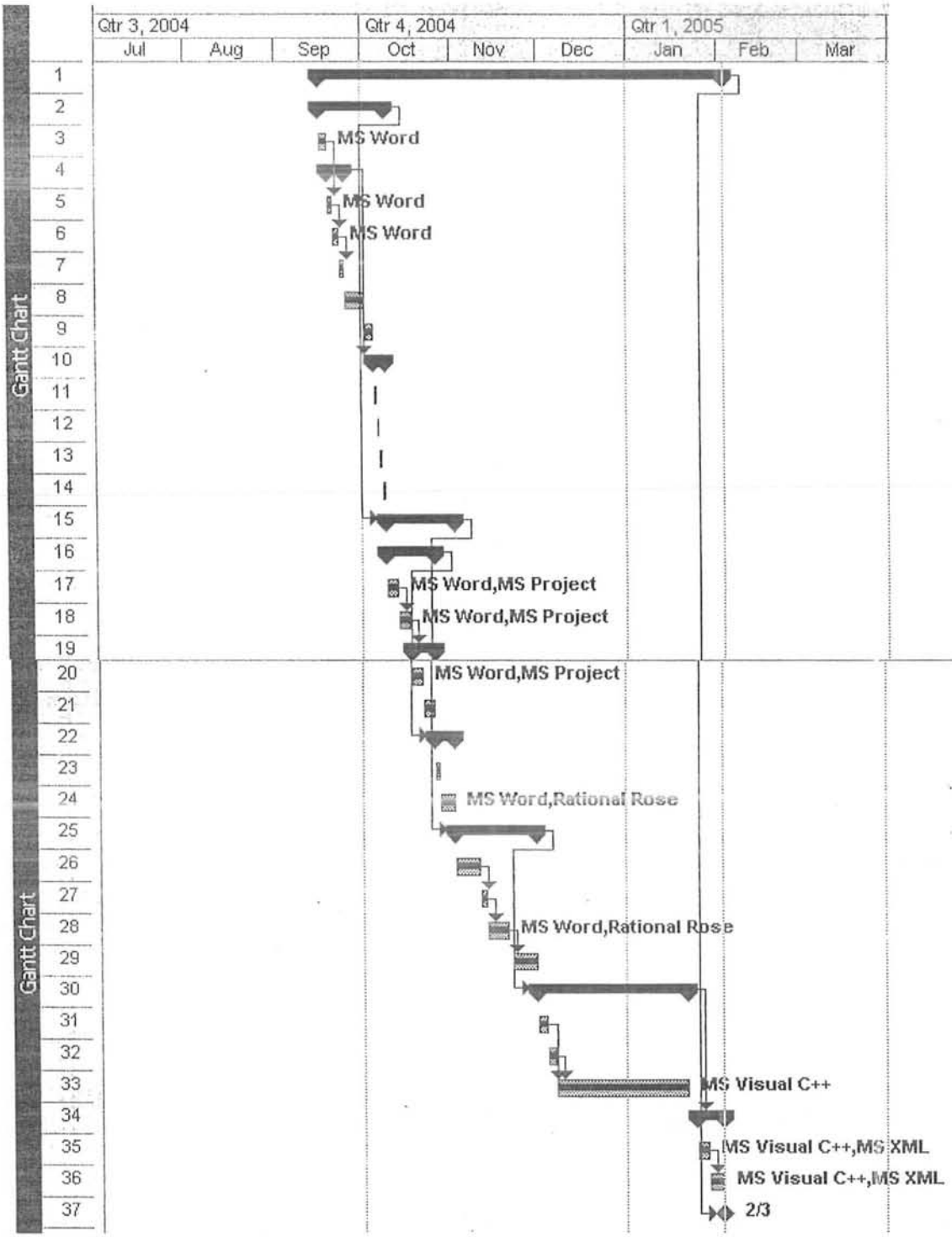
Management

If the things go wrong I will be consulting with the some language and tool professional.

APPENDIX- B

Gantt Chart

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Visual Language for Computer Telephony**	##########	**Thu 9/16/04**	**Wed 2/2/05**		
2	**Domain Study**	**20 days?**	**Thu 9/16/04**	**Fri 10/8/04**		
3	Problem Definition	3 days?	Thu 9/16/04	Sat 9/18/04		MS Word
4	**System Study**	**5 days?**	**Sun 9/19/04**	**Fri 9/24/04**		
5	System Context	1 day?	Sun 9/19/04	Mon 9/20/04	3	MS Word
6	Information Objectives	2 days?	Tue 9/21/04	Wed 9/22/04	5	MS Word
7	Function and performance	2 days?	Thu 9/23/04	Fri 9/24/04	6	
8	Wap Applications Study	6 days?	Sat 9/25/04	Fri 10/1/04		
9	Selection Of Process Model	2 days?	Sat 10/2/04	Mon 10/4/04		
10	**Feasibility Study**	**4 days?**	**Tue 10/5/04**	**Fri 10/8/04**	4	
11	Technical Feasibility	1 day?	Tue 10/5/04	Tue 10/5/04		
12	Operational Feasibility	1 day?	Wed 10/6/04	Wed 10/6/04		
13	Economical Feasibility	1 day?	Thu 10/7/04	Thu 10/7/04		
14	Legal Feasibility	1 day?	Fri 10/8/04	Fri 10/8/04		
15	**System Analysis**	**20 days?**	**Sat 10/9/04**	**Mon 11/1/04**	2	
16	**Requirement Engineering**	**14 days?**	**Sat 10/9/04**	**Mon 10/25/04**		
17	Requirement Identification	3 days?	Sat 10/9/04	Tue 10/12/04		MS Word,MS Project
18	Requirement Definition	4 days?	Wed 10/13/04	Sat 10/16/04	17	MS Word,MS Project
19	**Requirement Specification**	**7 days?**	**Mon 10/18/04**	**Mon 10/25/04**	18	
20	Functional Requirements	4 days?	Mon 10/18/04	Thu 10/21/04		MS Word,MS Project


Gantt Chart
Qtr 3, 2004
Qtr 4, 2004
Qtr 1, 2005
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
MS Word
MS Word
MS Word
MS Word,MS Project
MS Word,MS Project
MS Word,MS Project
MS Word,Rational Rose
MS Word,Rational Rose
MS Visual C++
MS Visual C++,MS XML
MS Visual C++,MS XML
2/3

Feasibility Study

The feasibility study is project management activity that is reviewed by project management to assess the reliability and at upper management level to assess the project status. The feasibility study finally results in "go/no-go" decision

Feasibility study is conducted by keeping following four activities in mind

1. Technical feasibility
2. Economical feasibility
3. Operational feasibility
4. Legal feasibility

1.7.1 Technical Feasibility

The Visual Language for Computer Telephony will be technically feasible because of following

1. Using VC ++ 6.0, which is efficient and reliable tool, we will develop the system.
2. By proposed solution it is to enhance it in future.

1.7.2 Economical Feasibility

Economic Feasibility is directly related to cost of software. Cost of software depends on resources that are used. The system will be economically feasible because of following

1. As the system is developed for academic purposes in order to complete partial fulfillment of M.Sc degree so the evaluation versions of software can be used and there is no need to spend extra money to purchase software resources.

1.7.3 Operational Feasibility

The system will be operationally feasible because of following

1. As we have studied the system very well and by using the technique of brainstorming so the end product will meet all the requirements of the user such as easy editing and a reusable component with satisfactory reliability.

1.7.4 Legal Feasibility

The system will be legally feasible because Legal feasibility is related with copyrights laws of soft wares that are used in development. The development is involved with evaluation versions of software for academic purposes not for commercial purposes.

1.1.1 Control Add

Use case:	Control Add
Actors	User of application
Trigger	User invokes "Toolbox" option from menu
Flow of events	
1	User go to the toolbox through mouse
2	Select the any required control
3	The required control will be dark
4	After selection of control push the left button of mouse
5	Move the mouse to the required location
6	Release the left button
7	The control should be added in the editing window through drag and drop.

1.1.2 Control Select

Use case:	
Actors	User of application
Trigger	User invokes "Toolbox" option from menu
Flow of events	
1	User will go to the toolbox through mouse
2	Take the mouse cursor at the required control
3	Push the left button of the mouse
4	Required control must be selected

1.1.3 Property Sheet Display

Use case:	Property Sheet Display
Actors	User of application
Trigger	User selects any control.
Flow of events	
1	User go to the toolbox through mouse

2	Add the control from toolbox in the editing window (canvas)
3	Select any control in the editing window (canvas)
4	Relevant properties of the control must be displayed in the container.

1.1.4 XML Generate

Use case:	XML Generate
Actors	User of application
Trigger	User selects the "XML button" from toolbar.
Flow of events	
1	User selects any control from the canvas.
2	User go to the toolbar
3	Select the XML button from the toolbar.
4	XML file must be generated.

1.1.5 Control Cut

Use case:	Control Cut
Actors	User of application
Trigger	User select the "cut button" option from toolbar.
Flow of events	
5	User selects any control from the canvas.
6	User go to the toolbar
7	Select the cut button from the toolbar.
8	Specified control must be cut.

1.1.6 Control Copy

Use case:	Control Copy
Actors	User of application
Trigger	User select the "copy button" option from toolbar.
Flow of events	

1	User selects any control from the canvas.
2	User go to the toolbar through mouse
3	Select the copy button from the toolbar.
4	Specified control must be copy.

1.1.7 Control Past

Use case:	Control Past
Actors	User of application
Trigger	User selects the "past button" option from toolbar.
Flow of events	
1	User selects any control from the canvas.
2	User go to the toolbar
3	User cut or copy the required control through mouse
4	Select the required location where user wants to past the control.
5	Control must be pasted.

1.1.8 Control Move

Use case:	Control Move
Actors	User of application
Trigger	User selects any control
Flow of events	
1	User selects any control from the canvas.
2	Drag and drop through mouse
3	Control should be replaced from one place to another place.

1.1.9 Control Print

Use case:	Control Print
Actors	User of application
Trigger	User selects the "Print Button" or "Print Option" from menu.
Flow of events	

1	User selects the controls from the toolbox.
2	User create the flow graph
3	Go to the print button from the toolbar or print option from menu.
4	Flow graph must be printed.

1.1.10 Flow Chart Save

Use case:	Flow Chart Save
Actors	User of application
Trigger	User selects the "Save Button" or "Save Option" from menu.
Flow of events	
1	User selects the controls from the toolbox.
2	User create the flow graph
3	Go to the save button from the toolbar or save option from menu.
4	Write the name of file
5	Write the extension of file
6	Flow graph must be saved.

1.1.11 Previous Session Open

Use case:	Previous Session Open
Actors	User of application
Trigger	User select the "Open Button" from toolbar or "Open" Option from menu.
Flow of events	
1	User selects the Open option from the file menu
2	A dialog Box will be open
3	Write the exact name of required file
4	Write the exact extension of required file
5	The required file will be open.

APPENDIX- F

SEQUENCE DIAGRAMS

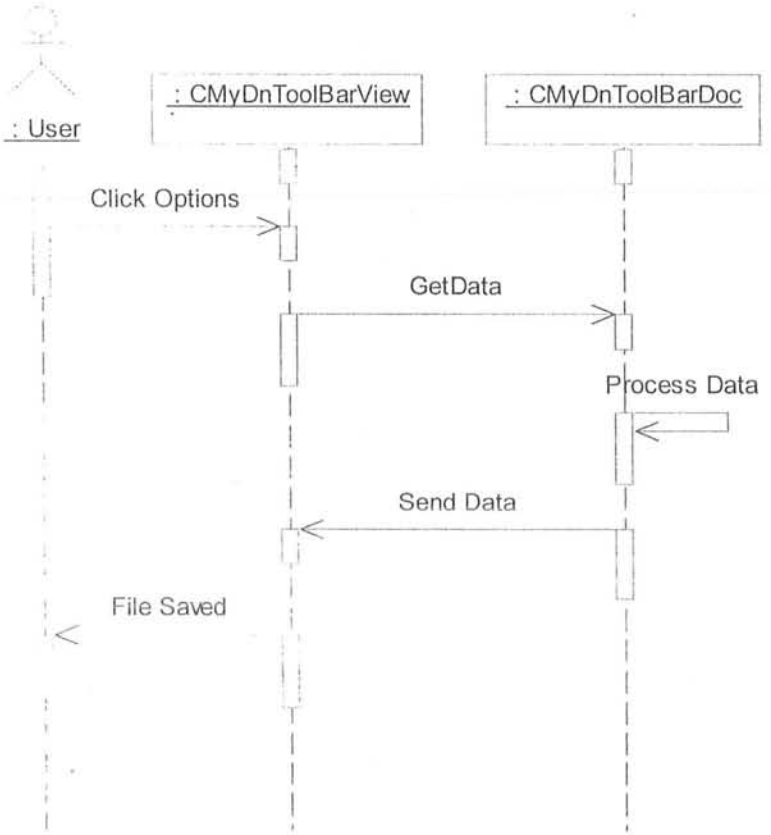


Fig f1. Sequence Diagram of Save File

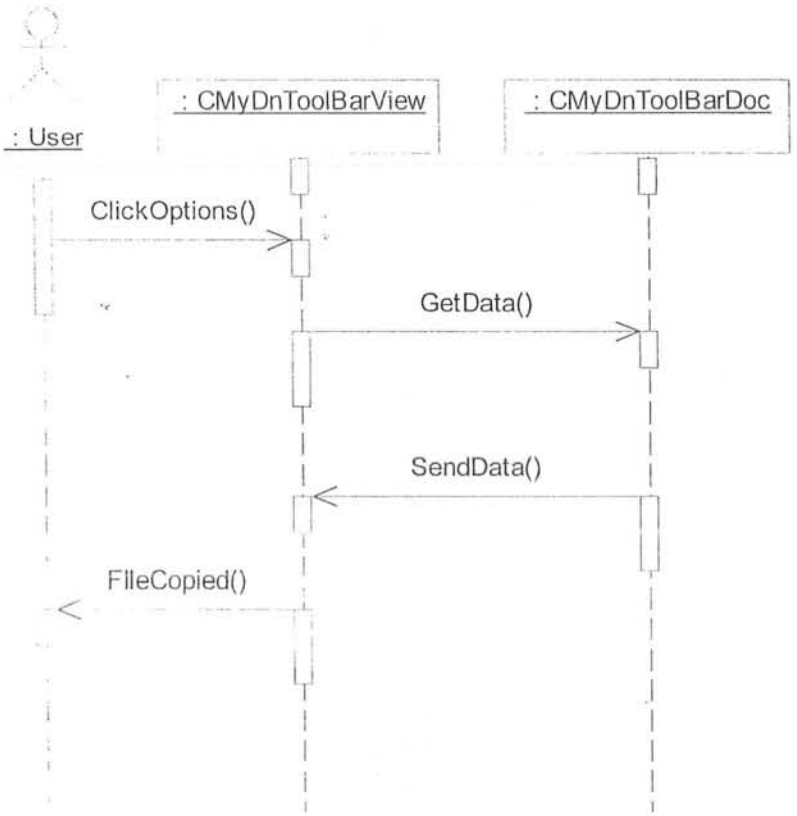


Fig f2. Sequence Diagram of Copy Control

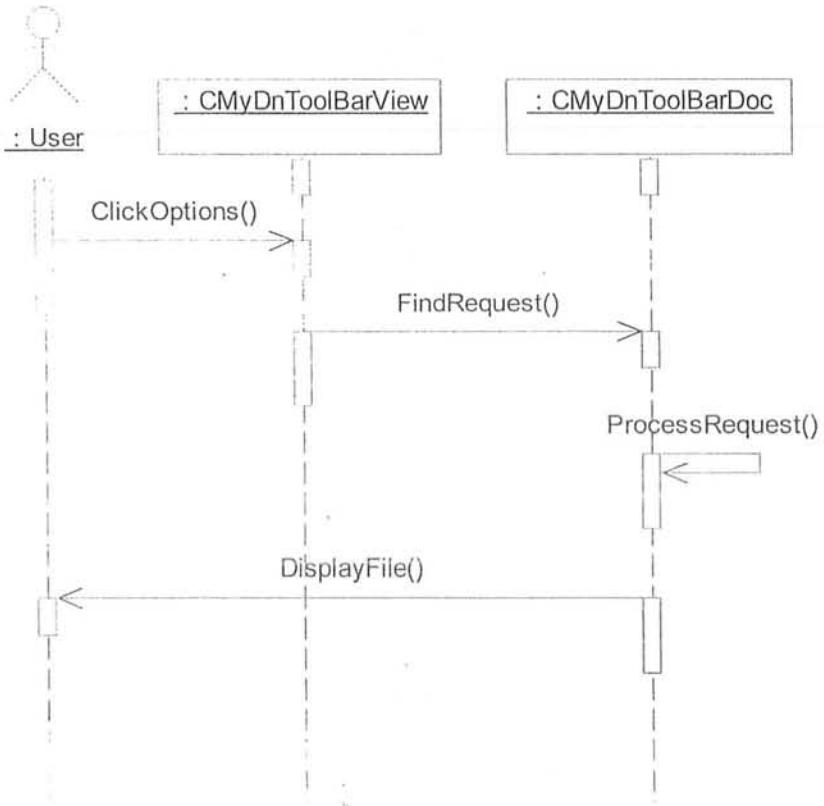
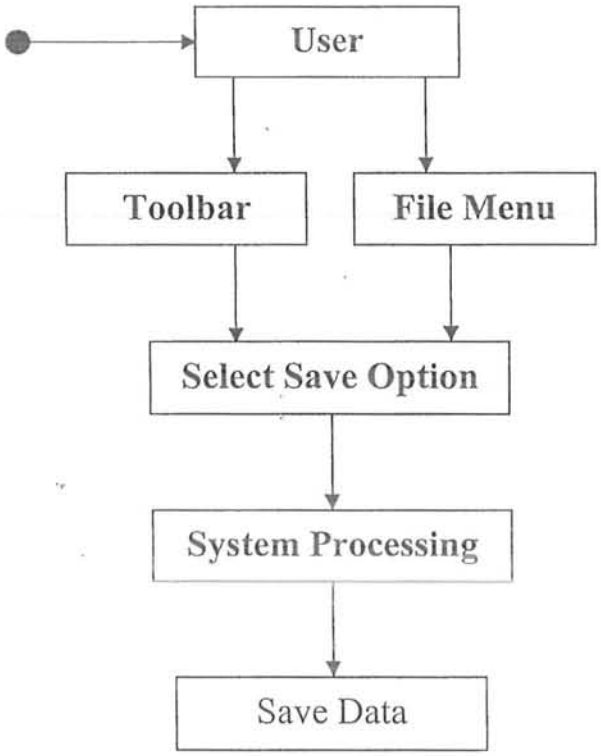


Fig f3. Sequence Diagram of Open File

APPENDIX- G

STATE DIAGRAMS



State Diagram of Save Data in Visual Language for Computer Telephony

APPENDIX- H

USER INTERFACE

User Interface

User Interface is user friendly. We can mention the main page of visual language for computer Telephony, It includes

Status Bar

In Status bar the name of file is shown, for example if the file name is Visual Language than visual language will be shown in status bar.

Menu Bar

In menu bar all the menus is shown and if user can click any menu then its relevant options will open.

Toolbar

In toolbar all the shortcut buttons are shown. All options including cut, copy, paste, generate XML etc are included in this toolbar.

Toolbox

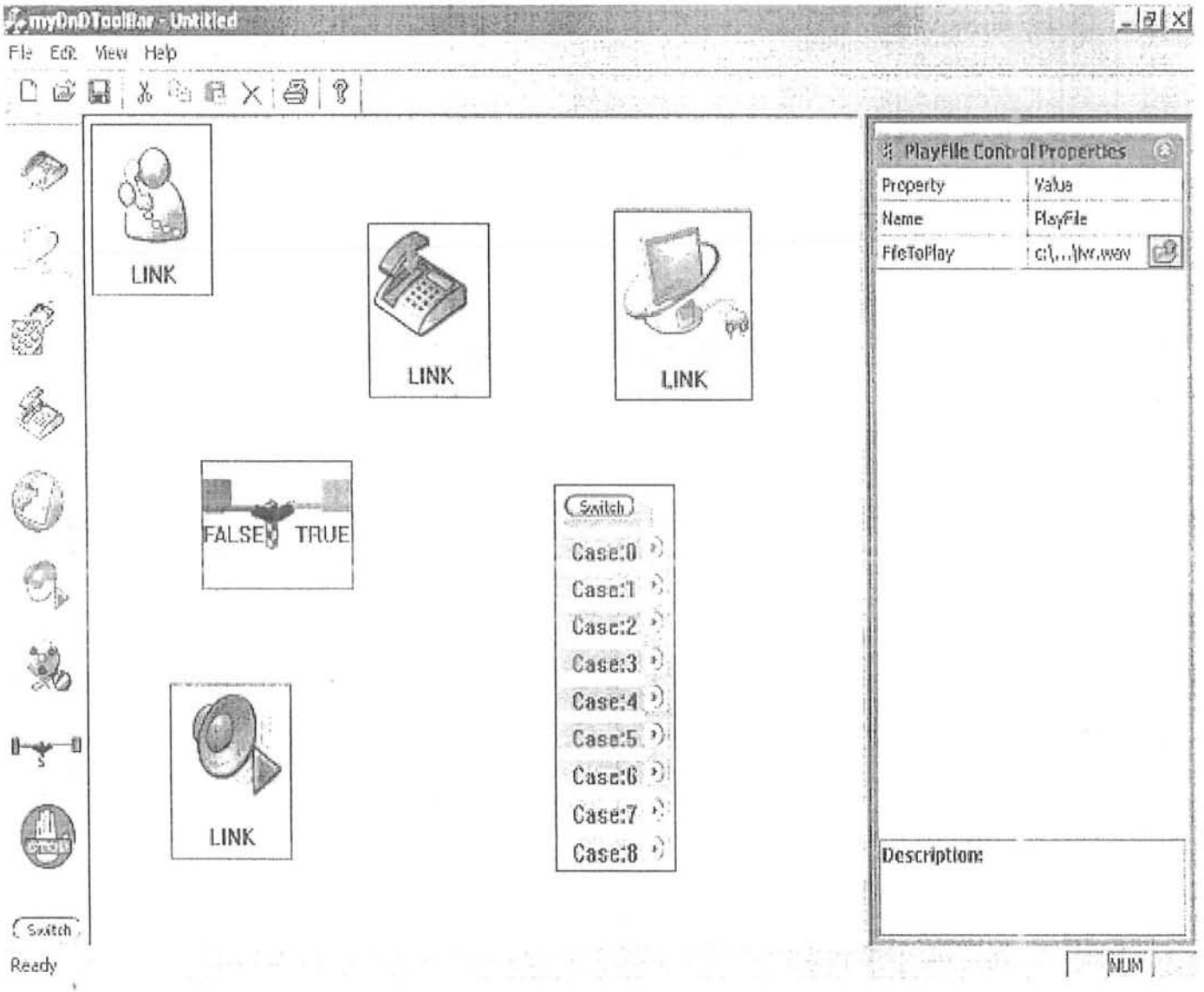
In toolbox all the control or components are adjusted, these controls/components are able to drag in editing window.

Property Window

If any control in editing window is selected than its relevant properties will be displayed.

Editing Window

In editing window we can add all the controls and make a flow graph, it can perform all the operations like cut, copy, past, move, remove etc.



Main Page of Visual Language for Computer Telephony

Glossary

A

API
Application Programming Interface

ATL
Active Template Library

C

COM
Component Object Model.

D

DLL
Dynamic Linked Library, used to wrap some functionality of code, which is dynamically linked during execution of software.

Dial up
A type of communication that is established by a switched-circuit connection using the telephone network.

DSP
digital signal processors: segment the voice signal into frames and store them in voice packets

E

Encode
To convert input data, for example, analog signals, characters, and commands, into a digital form recognizable by a computer1

I

Internet telephony
Internet telephony refers to communications services - voice and voice messaging applications - that are transported via the Internet, rather than the Public Switched Telephone Network (PSTN).

Internet telephony service
Internet telephony service is defined as the provision of real-time, interactive, multimedia telecommunications services between human users, using the public Internet.

P

Preprocessing

There is need of some tasks before recognition like Noise removal, Segmentation, Re-sampling and Skeltonization etc; and these are collectively called pre-processing

PSTN

Public Switched Telephone Network: The traditional telephone network as used in homes around the world.

S

SDK

Acronym for Software Development Kit

Signaling

The process of sending messages over the network for purposes of communication.

T

Test Cases

Test Cases are a set of inputs, which are used to check the system for its reliability and verification of code and validation of requirements, is done with the help of these.

TCP

Transport Control Protocol

U

User

An Alias for System operator

Use Case

A scenario based methodology for election of requirements

UML

Unified Modeling Language, These are notation used to describe system models of OOA & OOD etc.

Visual Glossary

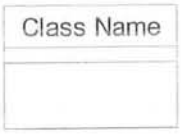
Actor:

Syntax:

Association (Unidirectional):
Syntax:

Aggregation:
Syntax:

Class:
Syntax:



End State:
Syntax:

Interface:
Syntax:

Message:
Syntax:

Return Message:
Syntax:

Self Message:
Syntax:

Stereotype:
Syntax:

<< Type>>

Start State:
Syntax:

Use Case:
Syntax: