

OKLAHOMA STATE UNIVERSITY

REMOTE CANCER DIAGNOSIS

By

SREE RAMA KRISHNA RAO MALLINA

Bachelor of Engineering

University of Madras

Madras, India

1995

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1998

OKLAHOMA STATE UNIVERSITY

REMOTE CANCER DIAGNOSIS

Thesis Approved:

W. Nick Stunt

Thesis Adviser

J. Chandler

M. E. Hedlund

Wayne B. Powell

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my sincere appreciation and gratitude to my Guru, Dr. William Nick Street for his guidance, supervision, encouragement and help for the completion of my thesis work. His patience and constructive ideas helped me make this thesis work an enjoyable and memorable experience. I consider it a privilege to have worked under his supervision. I would also like to express my deep sense of gratitude and appreciation to Dr. John P. Chandler and Dr. G. E. Hedrick for serving on my graduate committee and for providing valuable suggestions and ideas.

My greatest appreciation, thanks and love to my parents Dr. Seshagiri Rao Mallina and Mrs. Vijaya Lakshmi Mallina, and my brothers for all the love, support and inspiration that they have given me. Also, I thank all my friends, for their support and much needed help.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
II. LITERATURE REVIEW.....	2
2.1. Breast Cancer and its impact on the society.....	2
2.2. Role of Early Detection.....	3
2.3. Diagnosis of Breast Cancer.....	4
2.3.1. Mammography.....	5
2.3.2. Ultrasound.....	5
2.3.3. Biopsy.....	7
2.3.4. Fine needle aspiration.....	7
2.4. Xcyt system.....	8
2.5. Client/Server model.....	9
2.6. Internet Technologies.....	11
2.6.1. Internet.....	11
2.6.2. World Wide Web.....	12
2.6.3. Web browser.....	12
2.6.4. HTML.....	14
2.6.5. CGI.....	14
2.6.6. JavaScript.....	15
2.6.7. Java.....	16
III. THE SYSTEM.....	17
3.1. Platform requirements.....	17
3.2. Functionality of the system.....	18
3.3. System files.....	19
IV. DESIGN AND IMPLEMENTATION.....	21
4.1. Initial screen.....	21
4.1.1. Upload Files.....	24
4.1.2. Run.....	24
4.1.3. Save Result.....	27
4.1.4. Save Result on Server and Exit.....	27
4.1.5. Error dialog.....	32

Chapter	Page
4.2. User Interface.....	32
4.2.1. File.....	34
4.2.2. Edit.....	34
4.2.3. Colorscale.....	35
4.3. Xcyt system.....	35
4.3.1. Xcyt class.....	38
4.3.2. Snake class.....	40
4.3.3. Node class.....	40
4.3.4. Snakelist class.....	40
4.3.5. Nucleus class.....	41
4.3.6. Features class.....	41
4.3.7. Diagnosis class.....	41
4.3.8. Grayscale class.....	41
4.3.9. PlugInFilter class.....	41
4.3.10. EmptyListException class.....	42
4.3.11. format class.....	42
4.3.12. menu class.....	42
4.3.13. runThread class.....	42
V. CONCLUSION AND FUTURE WORK.....	43
5.1. Conclusion.....	43
5.2. Future work.....	44
REFERENCES.....	45

LIST OF TABLES

TABLE	Page
1. Annual incidence rates of breast cancer per 100,000 females.....	3
2. List of source files.....	19
3. List of files needed for running the application.....	20
4. List of classes used for the Xcyt system.....	38

LIST OF FIGURES

FIGURE	Page
1. Diagnostic methods for breast cancer.....	6
2. Client/Server model.....	10
3. Model of the system.....	22
4. Initial screen.....	23
5. Screen to upload files.....	25
6. Flowchart of file uploading.....	26
7. Flowchart of saving data file on server.....	28
8. Screen for saving data on the local machine.....	29
9. Error dialog box.....	29
10. Screen displaying the stored data file on the server.....	30
11. Final screen.....	31
12. User interface.....	33
13. Features window.....	36
14. Diagnosis window.....	36
15. Model of the Xcyt system.....	37
16. Flow diagram of the Xcyt system.....	39

CHAPTER I

INTRODUCTION

Breast cancer is the most common form of cancer among women. It accounts for many new cases every year and is the cause of many deaths. However early diagnosis of breast cancer by screening and physical examination improves the chances of survival. Many research studies have shown that fine needle aspiration biopsy of breast along with physical examination, mammography, and/or ultrasound could be used as a more cost-effective means of diagnosis of patients with palpable breast masses. But, visual examination of fine needle aspiration has the drawback of not being able to reliably classify some cases as either benign or malignant. As a result, many women with benign breast masses must undergo surgery for diagnosis. At the University of Wisconsin a system called Xcyt was developed to overcome this drawback of visual fine needle aspiration.

The thesis work implements the existing Xcyt system using the latest Internet technologies, and makes it available on the Internet. Chapter II of this thesis discusses the impact of breast cancer on the society, diagnostic methods for breast cancer, the Xcyt system, and the latest Internet technologies. Chapter III provides the system details. Chapter IV describes the design and implementation issues involved in developing the system. Finally, Chapter V concludes the thesis work and provides some suggestions for future work.

CHAPTER II

LITERATURE REVIEW

2.1. Breast Cancer and its impact on the society

Breast cancer, by far the most common form of cancer in women, comprises 23% of cancer in women and a total of 13% of all cancers [8]. It causes a devastating impact on the patient and her family. Breast cancer is the number one killer of women in the United States [8]. The number of new cases diagnosed as breast cancer each year are more than double the number of new cases of colorectal cancer, the second major type in women [1]. In United States there are roughly 70,000 new cases each year and 31,000 deaths each year [8]. One out of every fifteen newborn girls will develop breast cancer later in life. In every seventeen minute period, three new cases of breast cancer will be diagnosed and in those same seventeen minutes, one woman will die of breast cancer [8].

In UK the breast cancer constitutes 20% of all cancer deaths and 4.5% of all female deaths from all causes [12]. It is believed that one out of every twelve women develop breast cancer in their lifetime. Fifteen thousand deaths per year are attributed to the disease in UK. The following table illustrates the annual incidence rates of breast cancer per 100,000 females.

<i>High</i>	<i>Rate/100,000</i>
<i>U. S. A. (white)</i>	62.3 - 62.4
<i>U. K.</i>	42.5 - 51.1
<i>Scandinavia</i>	41.0 - 48.6
 <i>Intermediate</i>	
<i>Cali, Colombia</i>	27.3
<i>Yugoslavia</i>	22.8
<i>Bombay, India</i>	20.4
 <i>Low</i>	
<i>South Africa</i>	11.9 - 13.6
<i>Japan</i>	11.0 - 12.4

Table 1. Annual incidence rates for breast cancer per 100,000 females [8].

The breast cancer mortality rate has remained very nearly 25 per 100,000 over the past forty years. It is responsible for 25% of deaths in women aged 35 to 54 years. The breast cancer incidence is directly proportional to the age. The older the patient, the greater the chance of developing breast cancer.

2.2. Role of Early Detection

Early diagnosis and treatment of breast cancer provides one of the highest chances of survival among cancer types in women. Early detection means finding the incidence of the disease as early as possible, possibly even before there are any symptoms. It is the only method with proven potential for lowering the death rate from the disease [18].

Detection of cancer when it is small and confined to the breast, improves survival and

reduces mortality. Screening and physical examination have made major contributions to the earlier detection of breast cancer. Mammography is the only reliable means of detecting non-palpable cancers and can detect many minimal breast cancers when they appear to be curable [3]. It is the only method available for detecting ductal carcinoma in situ (an early form of breast cancer).

2.3. Diagnosis of Breast Cancer

The diagnosis of breast cancer begins with the clinical examination followed by a radiographic or an echographic examination, and if necessary, a cytological analysis. Various diagnostic methods have been developed to evaluate palpable and non-palpable breast lesions. Mammography has proven to be the only reliable means of detecting non-palpable cancers and can detect many minimal breast cancers and provide a curative approach for this disease. Mass screening of asymptomatic women by mammography is one proven way of reducing the rate of mortality [12]. Physical examination, mammography and/or ultrasound, core needle biopsy, thermography, fine needle aspiration (FNA) and open excisional biopsy are used predominantly in the diagnosis of patients with palpable breast masses. The diagnostic methods are as follows:

- Mammography
- Ultrasound
- Biopsy

2.3.1. Mammography

A mammogram (X - ray) of the breast is taken and examined by a radiologist or other doctor to detect the presence of breast cancer, either in the form of micro-calcifications or the presence of soft tissue masses of the order of 1 cm in diameter. For women over 50 mammography is the best way of finding early tumors. Mammography is useful for spotting early changes in the breast, when it is difficult to feel a lump. About 25% of breast cancers diagnosed in mass screening programs are detected by mammography [4]. The sensitivity and specificity of X-ray mammography was determined to be 89 and 72 respectively, in a recent study conducted by Bone [2].

2.3.2. Ultrasound

Ultrasound uses sound waves to get a picture of the inside of the body (breast). It is usually used for women under 35 whose breasts are too dense or solid to give a clear picture with mammography. Sonographic differentiation of benign from malignant solid masses leads to a large number of false-positive diagnosis [11,3]. Ultrasound fails to detect many cancers smaller than 1 cm in size, and almost never detects cancer in the form of clustered calcifications as on mammograms [11,17]. For these reasons, ultrasound is used as an adjunct to mammography and physical examination.

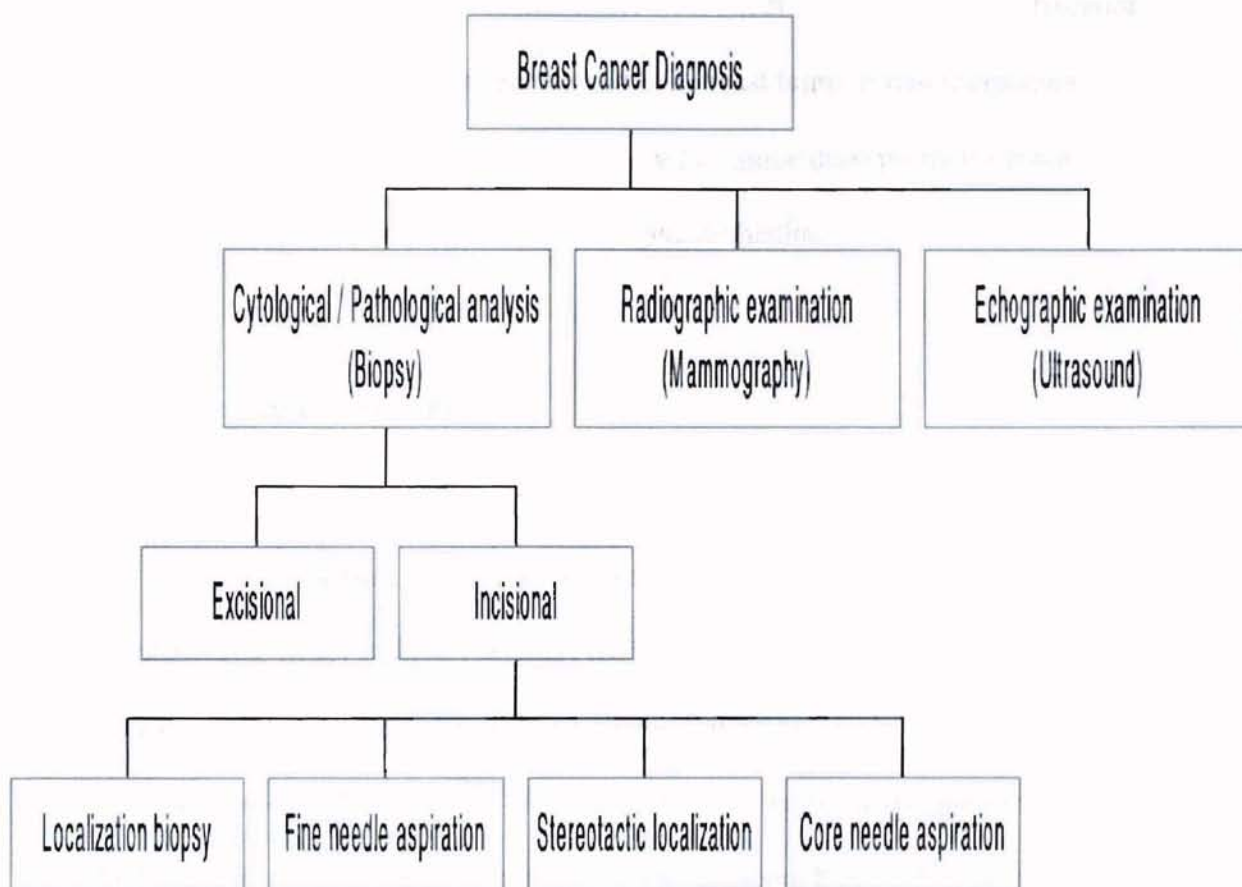


Figure 1. Diagnostic methods for breast cancer.

2.3.3. Biopsy

Tissue is removed from a breast lump by a surgeon and examined by a pathologist under a microscope to identify tissue changes that are characteristic of breast cancer. Tissue samples for biopsy are obtained either with needles or with surgery. Excisional biopsy is the only certain way to learn whether a breast lump or mammographic abnormality is cancerous. The choice of biopsy technique depends on the nature and location of the lump as well as the women's general health.

2.3.4. Fine needle aspiration

FNA uses a fine-gauge needle and syringe, to remove clusters of cells from a solid mass. These cells are then examined microscopically by the attending physician or by a cytopathologist. Many research studies [16] have concluded that fine needle aspiration biopsy of breast along with physical examination, mammography, and/or ultrasound can be used as a preoperative and as a more cost-effective means of diagnosis of patients with palpable breast masses, and as an alternative to excisional biopsy. FNA biopsy is being used increasingly because more than 80% of the open excisional biopsies performed each year in the United States are found to be benign [15]. Fine needle aspiration classifies the cases into either benign (negative for cancer), malignant (positive for cancer), or indeterminate (suspicious for cancer). The doctor may confirm the diagnosis with an excisional biopsy before proceeding with any treatment.

FNA is a minimally invasive procedure that can be performed on an outpatient basis. Since most breast masses are benign, the procedure has significant potential for

avoiding unnecessary surgery. However, adoption of the technique has been slowed by the difficulty of obtaining definitive diagnoses from cytological information alone.

2.4. The Xcyt system

The Xcyt system [20,21,22] was developed at the University of Wisconsin, to overcome the drawback of visual fine needle aspiration. The system was able to further classify the inconclusive or indeterminate cases into either benign or malignant with the corresponding probability of malignancy for each case. It also revealed the fact that computer analysis of a preoperative FNA could be used for prognosis of breast cancer in place of the traditional pathologic examination of axillary lymph nodes, thus avoiding the routine axillary lymph node dissection [23].

The Xcyt system takes a digital image of the fine needle aspirate in the form of a GIF image, as the input. The graphical user interface of the Xcyt system allows the user to input the approximate location of sufficient nuclei with the help of a mouse button. A deformable contour program is run on all the cells chosen to determine the exact boundary of each cell nuclei. Once the exact boundaries of all the nuclei are determined, the system computes ten nuclear features for each cell. The mean value, worst value, and the standard error of each feature are computed resulting a total of 30 features. The mean texture, the worst area, and the worst smoothness are used as input to a classification system that determines whether the sample is benign or malignant, as well as computing a probability of malignancy. The system has been shown to be highly accurate, correctly classifying 98.9% of 273 samples since its introduction into clinical practice in 1994.

2.5. Client/Server model

The concept of client/server model was introduced with the advent of distributed computing. Client/server technology is one of the most important applications of computer networks [19]. Figure 2 illustrates the client/server model.

The client/server model for networked computer systems consists of three main components:

- The server
- The client
- The network

A server is the software process that runs on the information provider's computer. The client is a software process that runs on the end user's computer. The user initiates a request for information through the client software. This request travels over the network to the server. The server interprets the request and takes the desired action by itself or spawns slave processes to serve the client's request. The action may be a database access or request for some hypertext document. The results of the requested transaction are sent back to the client. All client/server communications follow a set of rules, or protocols (predefined set of sequences in which information is exchanged between the client and server software) which are defined for the client/server systems [5]. A client can access many servers using the protocols that both server and client understand.

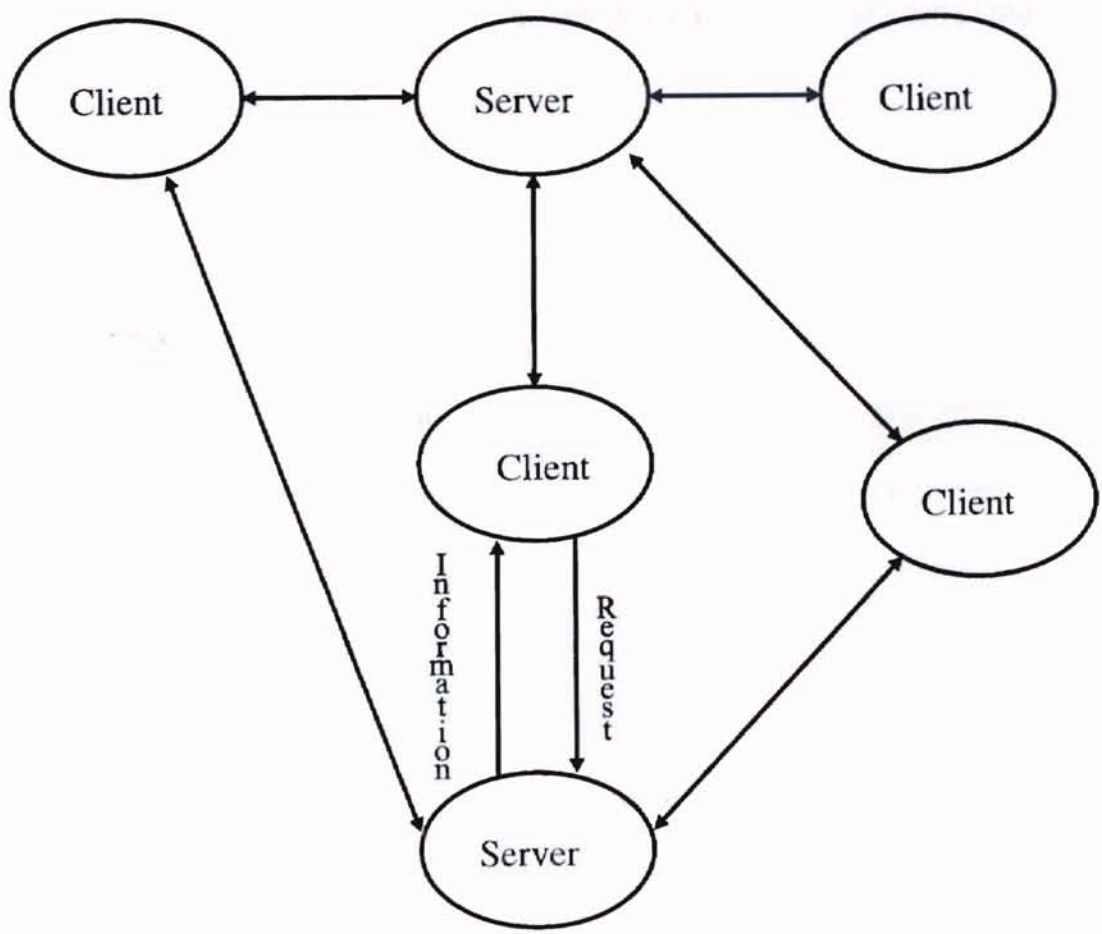


Figure 2. Client/Server model.

Client software can be customized to the user's hardware system, allowing the server to be independent of the client software. Because of this feature the client software like a Web browser particular to a user's system can access any Web server. Separate versions of the information need not be developed for any particular hardware platform, because necessary customizations are written into the client software for each platform.

2.6. Internet Technologies

2.6.1. Internet

The Internet is a globally interconnected network of networks. The Internet consists of several components, including network infrastructure, servers, documents, browsers, and applications. "It is the cooperatively run, globally distributed collection of computer networks that exchange information via the TCP/ IP protocol suite" [13]. The central protocol of the Internet is TCP/IP (Transmission control protocol/Internet protocol), which is a collection of transport and application protocols. The Internet is a client/server network. It serves as a vehicle for data communication for many information dissemination protocols. Though gateways, many other networks in cyberspace can exchange data with the Internet. Because of this and also because of the large amount of information available on it, the Internet serves as a common ground for the networked region of cyberspace.

2.6.2. World Wide Web

The World Wide Web has revolutionized the computer industry and has dramatically changed the online world. The Web is a flexible system for communication that can be used in many ways, ranging from individual communication on home pages through group communication and mass communication. It is an interlinked collection of servers on the Internet that provide information. The information can be in the form of text, graphics, or executable programs (hypertext). The Web is an application system employing a set of software programs like FTP, HTTP, Graphics, Video, HTML, Text, Gopher, etc. [6]. These are deployed and used on many networks. The Web gives the information providers the ability to distribute and gather information globally and instantly. The Web allows the end users to dynamically access information provided by millions of people and organizations. The Web is technically defined as "A hypertext information and communication system popularly used on the Internet computer network with data communications operating according to a client/server model. Web clients (browsers) can access multi-protocol and hypermedia information using an addressing scheme" [13].

2.6.3. Web browser

A browser is basically a Web client. A browser can be viewed as a client process in the client/server model. The functions of the browser are as follows [6]:

- To locate Web servers and communicate with them,

- To read and display the HTML documents requested by the user in the appropriate form, and
- To jump to the hyperlinks that are embedded in the documents when the user clicks on them.

The browsers are programmed for handling the fundamental Internet protocols and media formats like HTTP, Gopher, FTP, HTML, HTTPS, GIF etc. At times the browser makes use of helper applications to display specialized media formats such as movies or sound [13]. The non-Java-enabled browser has a limitation that it can handle only protocols for which it is programmed or media formats for which it has helper applications. But, the new generation browsers called the Java-enabled browsers are capable of handling protocols or media formats they are not aware of. The Java-enabled browser is able to interpret any protocol or media format with the aid of a protocol handler and content handler respectively.

A Java-enabled browser is capable of displaying Java applets. The Java-enabled browser is embedded with a Java interpreter. When a document with a Java applet is requested, the applet's bytecode is transferred to the user's computer. The bytecode is pre-compiled by the Java compiler using the Java source code file for that applet. The Java-enabled browser on the user's computer interprets the bytecode and provides the display. The user may have further interaction with the applet, but with no further downloading from the providing Web server. The bytecode contains all the information necessary to run the applet.

2.6.4. HTML

HTML is the foundation on which the Web is built. The actual content of the Web is stored in documents. These documents are created using HTML (Hyper Text Markup Language). The HTML documents consist of text, tags which control formatting, and hyperlinks to other documents. Documents also contain references to other resources using URLs (Uniform Resource Locators). Tags in the HTML document are enclosed within the "<" and ">" bracket pairs. Some HTML tags have attributes, which qualify the tag's meaning [7]. For example, a Java applet is included in an HTML document using the tag `<applet code = "test.class" height = 100 width=100>`, where height and width specify the height and width of the applet. HTML is used for developing Web sites, home pages, and electronic documents for display over the Internet.

2.6.5. CGI

HTML started as a very static way of displaying information on the Web. To make the Web medium dynamic and interactive, server side programs (gateway programs) were introduced. The Common Gateway Interface (CGI) is the programming interface that is built into the WWW communication protocols [14]. CGI is a set of standard protocols that allow Web servers to communicate with external programs. CGI is the widely used specification for creating executable programs that run on Web servers. CGI scripts are written in languages like C, Perl, UNIX shell, AppleScript, Java etc. The scripts take data input from the users through HTML forms, process it, and return the appropriate results.

The CGI provides two methods for gathering data from a form: the GET method and the POST method, which are built into the original specifications for HTTP 1.0 [14]. The GET method is intended to move entities from the server to the client. The GET method specifies a URL which contains a path describing the entity to be retrieved. The entity could be either a static HTML document or a program. When the entity is a program, HTTP specifies that the program will be run, and its output is sent in response to the request. The POST method is intended to move entities from the client back to the server. The CGI environment variables like `CONTENT_LENGTH`, `REMOTE_HOST`, and `QUERY_STRING` are sent as headers to the server by the client, and others like `SERVER_NAME` are set by the server [9]. The server determines the length of the content by examining the environment variable `CONTENT_LENGTH` and then sends those bytes to the standard input (STDIN) of the program named in the URL. Then the CGI scripts process the data from the STDIN and take the intended actions.

2.6.6. JavaScript

Though CGI and the server side programs make the Web interactive, still most of the transaction processing is done on the server side. The scripting languages like JavaScript and VBScript can be embedded into the HTML documents, reducing network traffic by keeping simple tasks local, thus improving the response times for the users. The main purpose of JavaScript is to link HTML with the Java programming language. JavaScript is easy-to-use, open, cross-platform scripting language to link objects and resources from both HTML and Java.

2.6.7. Java

Java is an object-oriented programming language for creating distributed and executable applications. Since Java is a programming language it is capable of doing everything that the scripting languages like JavaScript, Jscript, or VBScript can do and even more. Java applications that run on the Web are called Java applets. Java applets make the Web more interactive and dynamic. The Java compiler generates an architecture-neutral object file format called a bytecode. These bytecodes can be translated at run-time (on the fly) into machine code for the particular processor on which the application is running. Java is portable in the sense that it does not use different formats for the primitive data types. With the help of Java applets the processing is offloaded to the user's system. In applications involving heavy computations, the user need not worry about the speed of transmission from the host machine because the data is computed locally. Though Java is intended to be used in both networked and distributed environments it is highly secure. It does not allow any manual memory allocation and deallocation. It does not support any pointer arithmetic and prohibits local file input and output for untrusted applets and as well for signed classes [10]. It also restricts the applet's ability to establish arbitrary network connections. Finally, Java has good exception handling features and an automatic garbage collecting system.

CHAPTER III

THE SYSTEM

The thesis work implements the Xcyt system using object-oriented technology, platform independent technology like Java, server side (gateway programs) programs, and JavaScript, to make the system available on the WWW (Internet) and provide wide access to the system for many users distributed geographically.

The Xcyt system is implemented as a Java applet to make it platform independent and remotely executable. All the class files needed by the Xcyt system are packed into a single Java Archive called *xcytjarfile*, to reduce the overhead involved in establishing network connections to the Web server, for fetching each class file. As explained in Chapter II the Xcyt system takes a GIF image as input. Since Java applets loaded over the Internet are restricted to access local files, client/server programming and JavaScript are used for handling local file I/O.

3.1. Platform requirements

1. Server:

- HTTP (1.0 and above) Web server,
- Perl compiler,
- C compiler,

- JDK 1.1.3 and above, and
- Support CGI (POST method)

2. Client:

- Any browser that supports JDK 1.1.3, and JavaScript

3.2. Functionality of the system

The system provides the user with the facility to load input image files from the server, or from the client machine. The GUI with event handling capabilities allows the user to input the approximate location of a sufficient number of cell nuclei. The nuclei are internally represented by a multi-linked list. The process of determining the exact boundary of each cell nuclei is visualized with animation. The system allows the user to view the image in grayscale or colorscale enhancing the readability of the image. Multithreading is used to make the system more responsive. The image is converted into a two dimensional array for computing the features. It displays the intermediate computed features in a pop-up window when the user desires to view them. It also displays the benign and malignant distribution curves and indicates the region where the sample lies, classifying the sample as either benign or malignant with the corresponding probability of malignancy in a pop-up window. Finally, it allows the user to save the computed results onto the local machine and/or the server with an appropriate file name.

3.3. System files

This section lists the various files needed to develop and run the application.

Table 2 lists the files needed to develop the system and Table 3 lists the files needed for running the application.

File name	Type
Diagnosis	Java file
Features	Java file
format	Java file
Grayscale	Java file
menu	Java file
Nucleus	Java file
PlugInFilter	Java file
Snake	Java file
Snakelist	Java file
Xcyt	Java file
cgi	Header file ✓
gd	Header file /
util	C file
post	C file ✓
main	C file
Eglobsub	Perl file ✓
Eupload	Perl file /
fwrite	CGI file ✓
* upload	CGI file
* run	CGI file
* test	Java Script file
* upload	HTML file

Table 2. List of source files.

CHAPTER IV

IMPLEMENTATION

File name	Type
extract	Exe file
post	Exe file
xcytjarfile	Jar file
Eglobsub	Perl file
Eupload	Perl file
fwrite	CGI file
upload	CGI file
run	CGI file
test	JavaScript file
upload	HTML file

Table 3. List of files needed for running the application.

CHAPTER IV

DESIGN AND IMPLEMENTATION

This chapter examines the design and implementation issues involved in developing the overall system. The functionality of the system explained in Chapter III is illustrated in Figure 3. The application resides on the server z.cs.okstate.edu (Sun Solaris) at <http://z.cs.okstate.edu:8000/~mallina/iexcyt> location. The application will run on any popular browser that supports JDK1.1.3 and above, and JavaScript.

4.1. Initial screen

The initial screen is as shown in Figure 4. The initial screen provides the following options:

1. Select GIF files from the server
2. Upload Files
- ✓ 3. Run
- ✓ 4. Save Result
- ✓ 5. Save Result on Server and Exit

The user is provided with the option of choosing the input image file from the select list displaying image files from the server or upload an image file from the client to the server. A shell script on the server lists the image files from the appropriate directory.

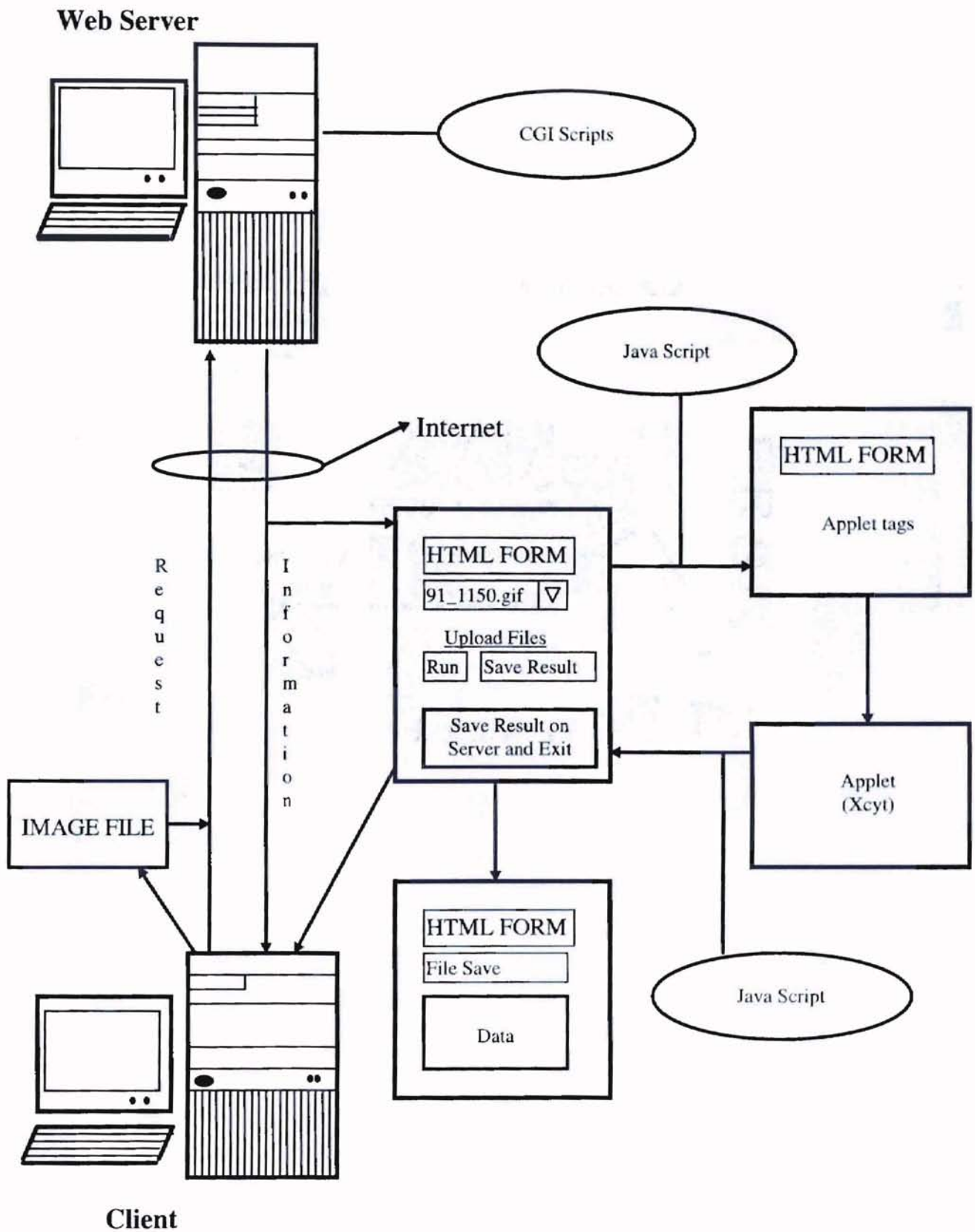


Figure 3. Model of the system.



Figure 4. Initial screen.

4.1.1. Upload Files



When the user clicks on the Upload Files link, a new screen as shown in Figure 5 is loaded. The user can either enter the path of the desired file directly into the box provided or choose to browse the local drive by clicking the *Browse* button. When the *Browse* button is clicked, the system's default file dialog box as shown in Figure 5 pops up allowing the user to browse the local drive. This is accomplished by using file input type in the HTML form. When the *Upload* button is clicked, the selected GIF file is attached to the HTML form (initial screen). A connection is established to the server and the file is sent to the standard input (STDIN) by the POST method. The CGI program, *Eupload* written in perl extracts the uploaded input GIF file from the standard input and stores it in the tmp directory of the server. Later the GIF file is moved to the appropriate directory. Once again the initial screen is loaded. This process is illustrated in Figure 6.

4.1.2. Run

Once the input GIF file is selected, the user clicks on the *Run* button to load the applet. The *Run* button will invoke a JavaScript function *loadApplet()*, to create a new HTML form with the applet embedded in it with the selected file as a parameter to the applet in a new browser window.

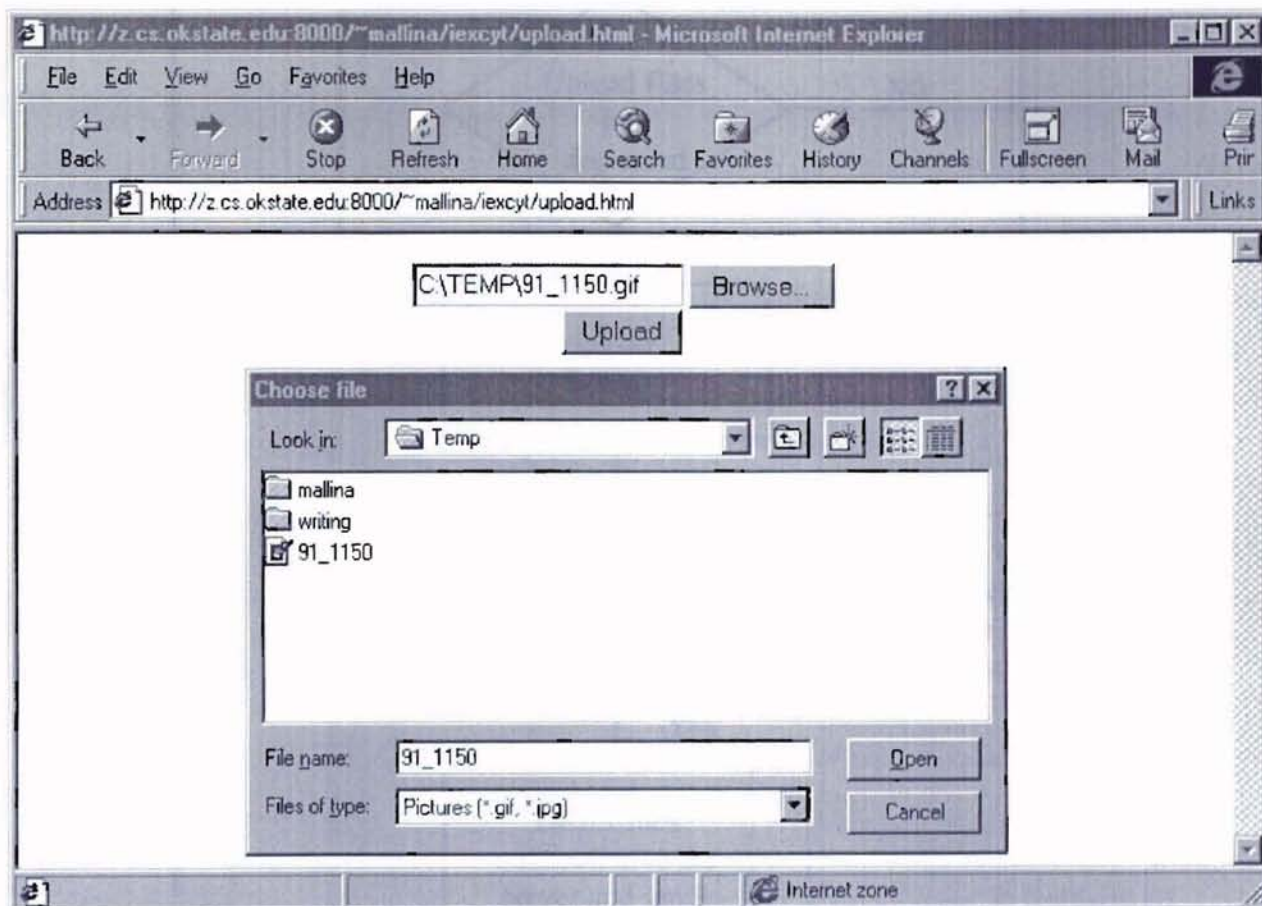


Figure 5. Screen to upload files.

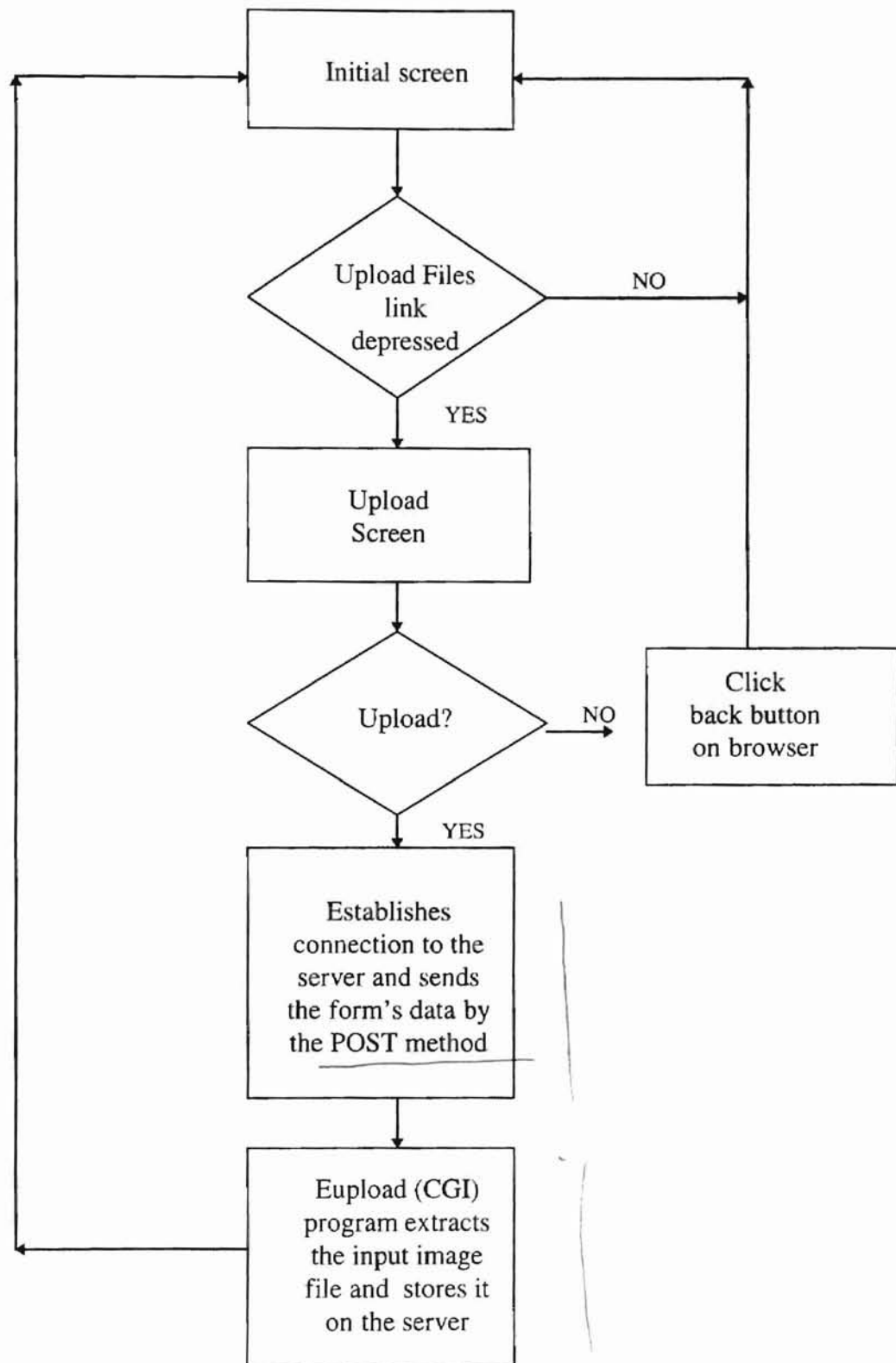


Figure 6. Flowchart of file uploading.

4.1.3. Save Result

When the Save Result button is clicked a JavaScript function returnVal() is invoked, which links the HTML with the Java applet. The JavaScript gets the information to be stored on the local machine from the applet into an HTML form variable, and outputs the information into a new browser window with a file save option as shown in Figure 8.

4.1.4. Save Result on Server and Exit

To save the result onto the server, the user will have to click on the *Save Result on Server and Exit* button. A JavaScript function returnsVal() is invoked, which links the HTML with the Java applet. The JavaScript gets the information to be stored onto the server from the applet into an HTML form variable. A connection is established to the server and the data from the HTML form is passed to the standard input of the server in the QUERY_STRING by the POST method. The CGI program post written in the C programming language extracts the information from the standard input in the form of name value pairs and stores it on the server with the file name tem.\$\$, where \$\$ is the current process id. The CGI program main written in C takes the file tem.\$\$ as a command line argument and formats the data up to 4 decimal places and stores the data file with the file name "imagefilename.data" as shown in Figure 10. The status, whether the data file is successfully written or not, is displayed back to the user by the final screen as shown in Figure 11. This process is illustrated in Figure 7.

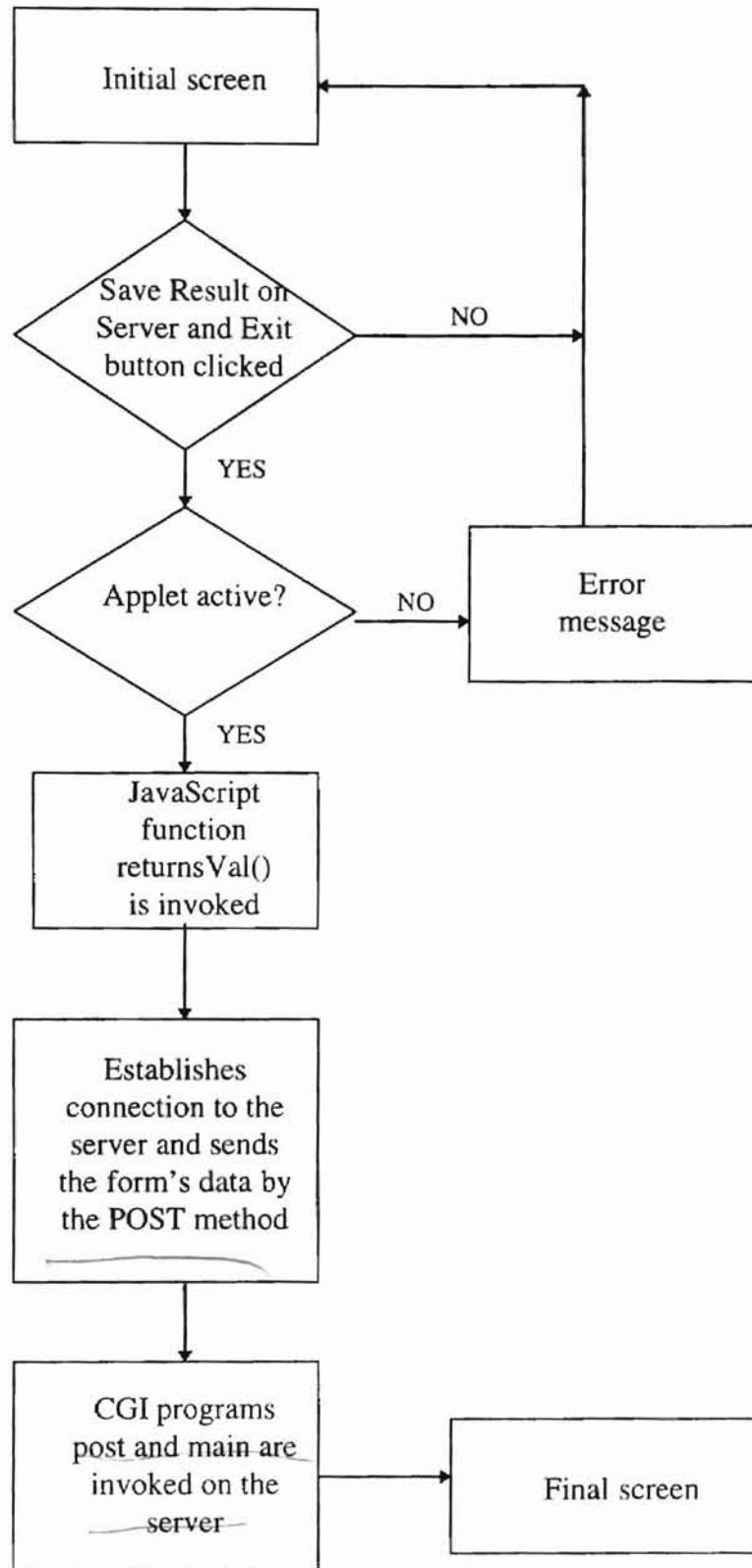


Figure 7. Flowchart of saving data file on server.

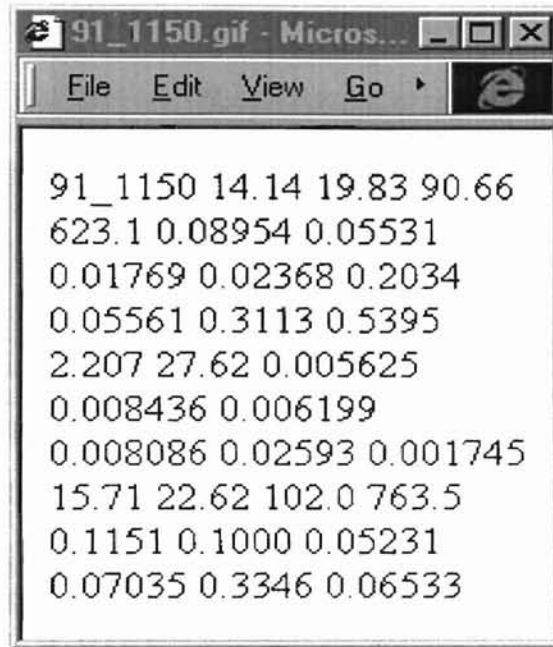


Figure 8. Screen for saving data on the local machine.



Figure 9. Error dialog box.

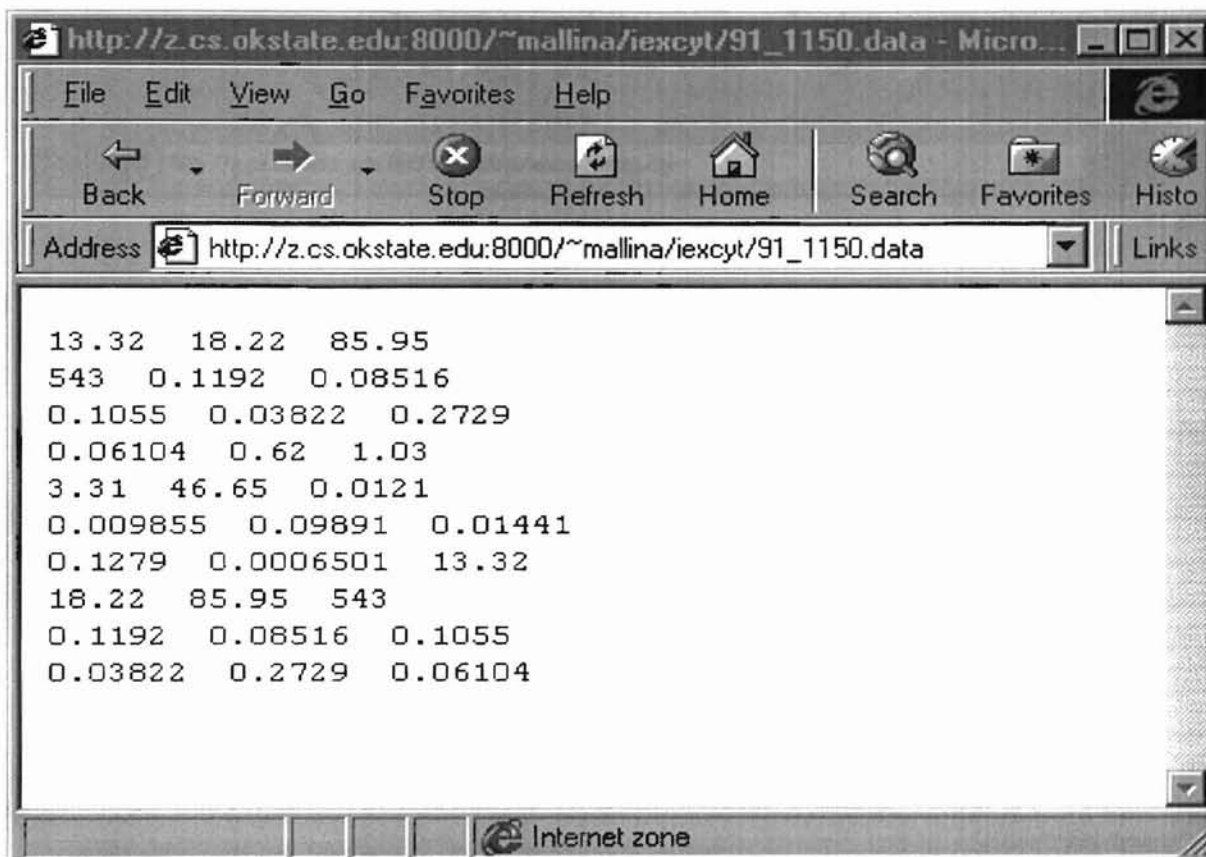


Figure 10. Screen displaying the stored data file on the sever.

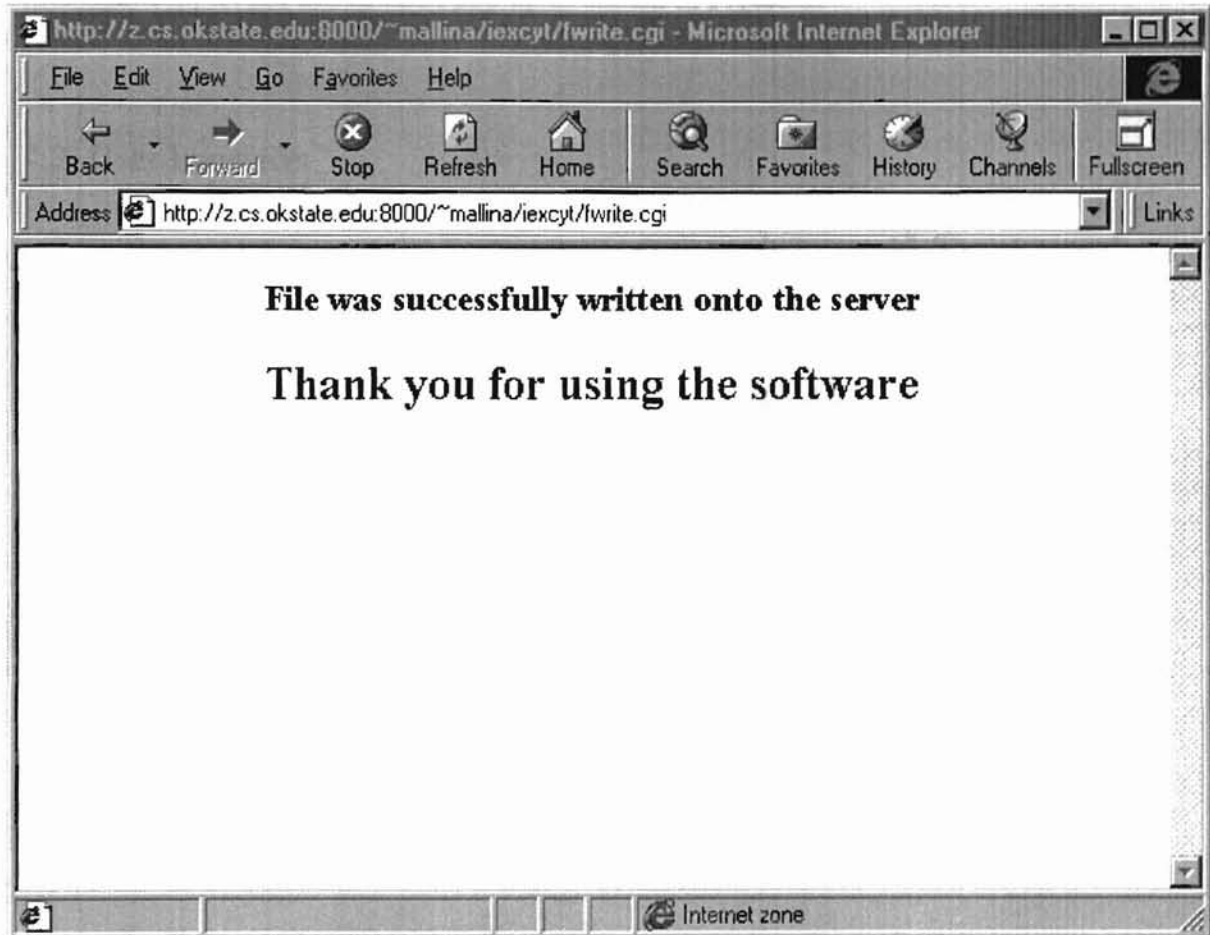


Figure 11. Final screen.

4.1.5. Error dialog

If the user clicks either the *Save Result* button or the *Save Result on Server and Exit* button without the applet being loaded, a JavaScript alert dialog box with the message “Applet not active” will pop up as shown in Figure 9.

4.2. User Interface

The user interface of the Xcyt system is as shown in Figure 12. The user interface is spawned in a frame instead of being embedded in the browser window because an applet inside a Web page cannot have menus [10]. The GUI with event handling capabilities allows the user to input the approximate location of the nuclei and run the application. A mouse is used to initialize the boundaries of the nuclei. The menu of the graphical user interface provides the following options:

1. File
 - Initialize
 - Exit
2. Edit
 - DeleteSnake
 - RunSnakes
 - ComputeFeatures
 - Diagnosis

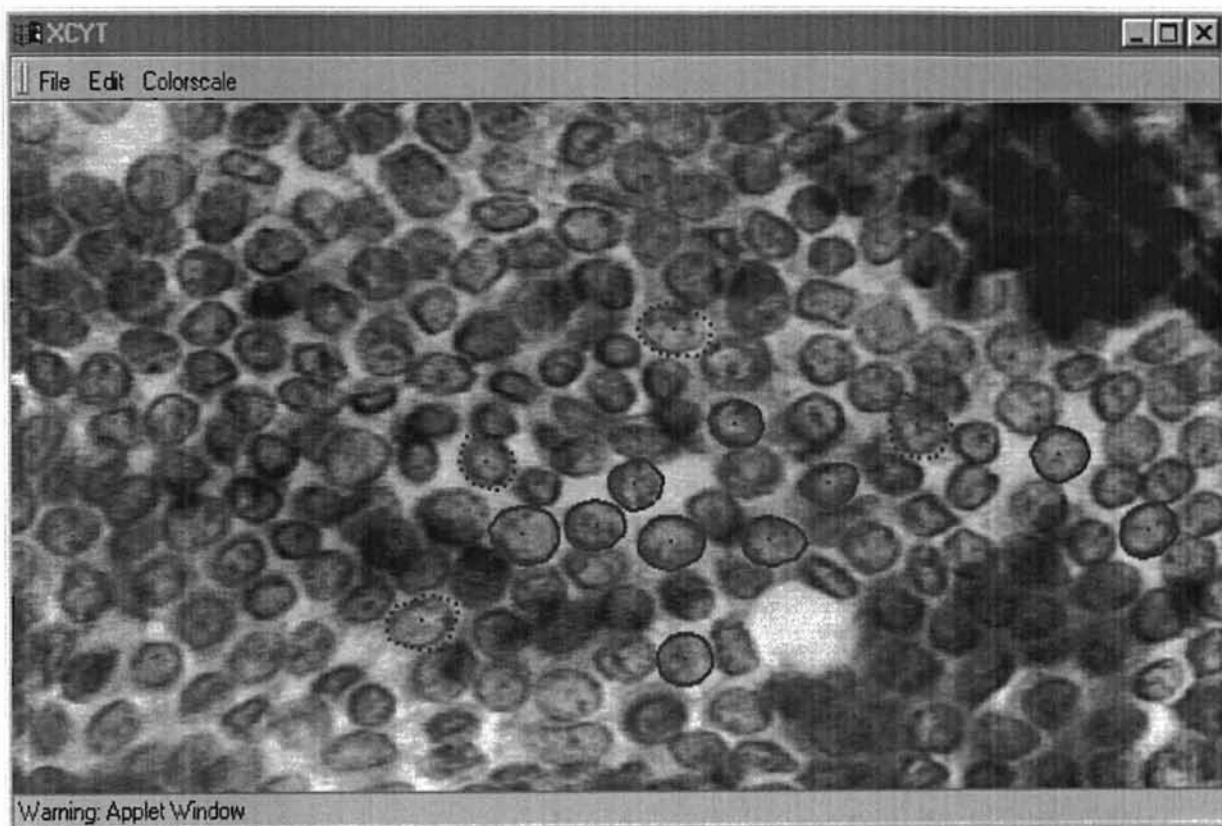


Figure 12. User interface.

3. Colorscale

- Grayscale
- Colorscale

The user interface is implemented in the *Xcyt* class and the menu in the *menu* class.

4.2.1. File

The *Initialize* option deletes all the nuclei and creates a new instance of the *Snakelist* object by invoking the constructor of the *Snakelist* class. The exit option will destroy the user interface window.

4.2.2. Edit

The *DeleteSnake* option will allow the user to delete the selected nucleus. The user can also delete a nucleus by double clicking on it with the right mouse button. The nucleus whose center is closest to the selected point is deleted from the *Snakelist* object. The *RunSnakes* option is used to compute the exact boundaries of the nuclei. The *RunSnakes* option can also be initiated by pressing the F1 key. The nuclei whose boundaries are not yet determined are displayed with dotted circles and the nuclei whose boundaries are already determined are displayed with completed curves as shown in Figure 12. The process of determining the exact boundary of each nucleus is visualized with animation. Multithreading is used for determining the exact boundaries of the nuclei, to make the system more responsive. The exact boundaries are determined only for the nuclei whose boundaries are not determined.

The *ComputeFeatures* option is used for computing 10 nuclear features for each nucleus. The mean value, worst value, and the standard error of each feature are computed resulting a total of 30 features. The computed features are displayed in a window as shown in Figure 13.

The *Diagnosis* option is used for classifying the sample into either benign or malignant. The probability of malignancy is calculated by using the benign and malignant distribution data. The distribution curves, machine diagnosis and the probability of malignancy are displayed in a window as shown in Figure 14.

4.2.3. Colorscale

The *Colorscale* option is used for viewing the image in colorscale or grayscale. It is implemented in the *Xcyt* class.

4.3. Xcyt system

The model of the Xcyt system is illustrated in Figure 15. The Xcyt system takes the following as inputs:

1. Input image file (GIF file1),
2. Image of the benign and malignant distribution curves (GIF file2),
3. User inputs, and
4. Parameters to the applet like the GIF file names

Features	Mean	StdError	Worst
Radius	14.14	0.31	15.71
Texture	19.83	0.53	22.62
Perimeter	90.66	2.2	102.07
Area	623.18	27.62	763.52
Smoothness	0.08	0	0.11
Compactness	0.05	0	0.1
Concavity	0.01	0	0.05
Concavepoints	0.02	0	0.07
Symmetry	0.2	0.02	0.33
Fractal_dim	0.05	0	0.06

Dismiss

Warning: Applet Window

Figure 13. Features window.

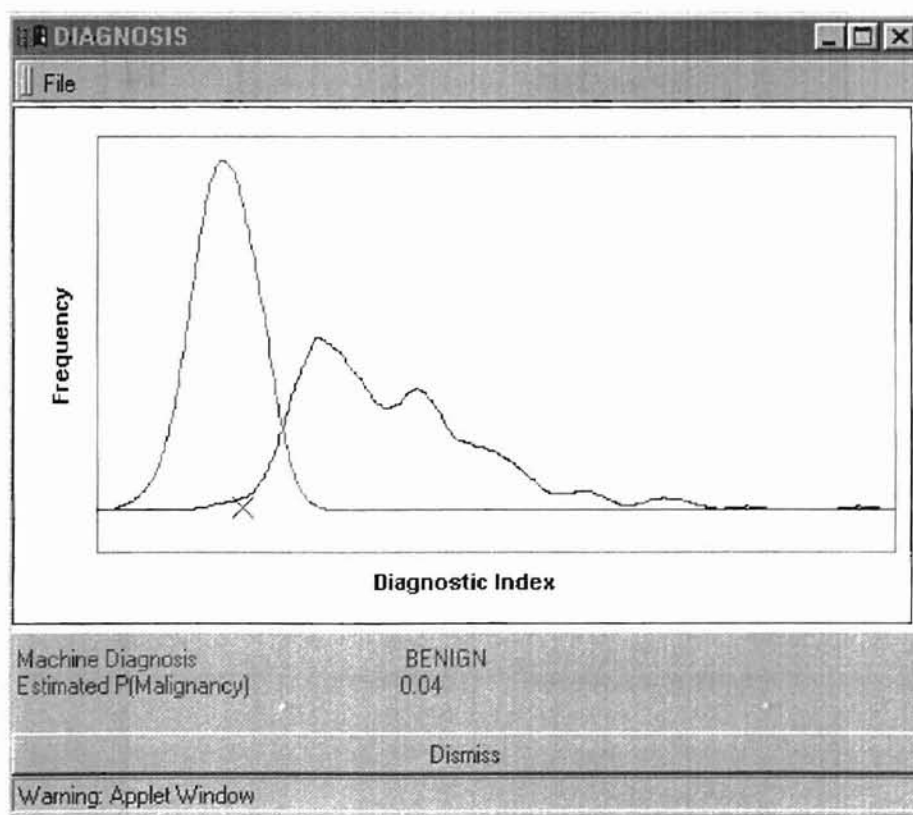


Figure 14. Diagnosis window.

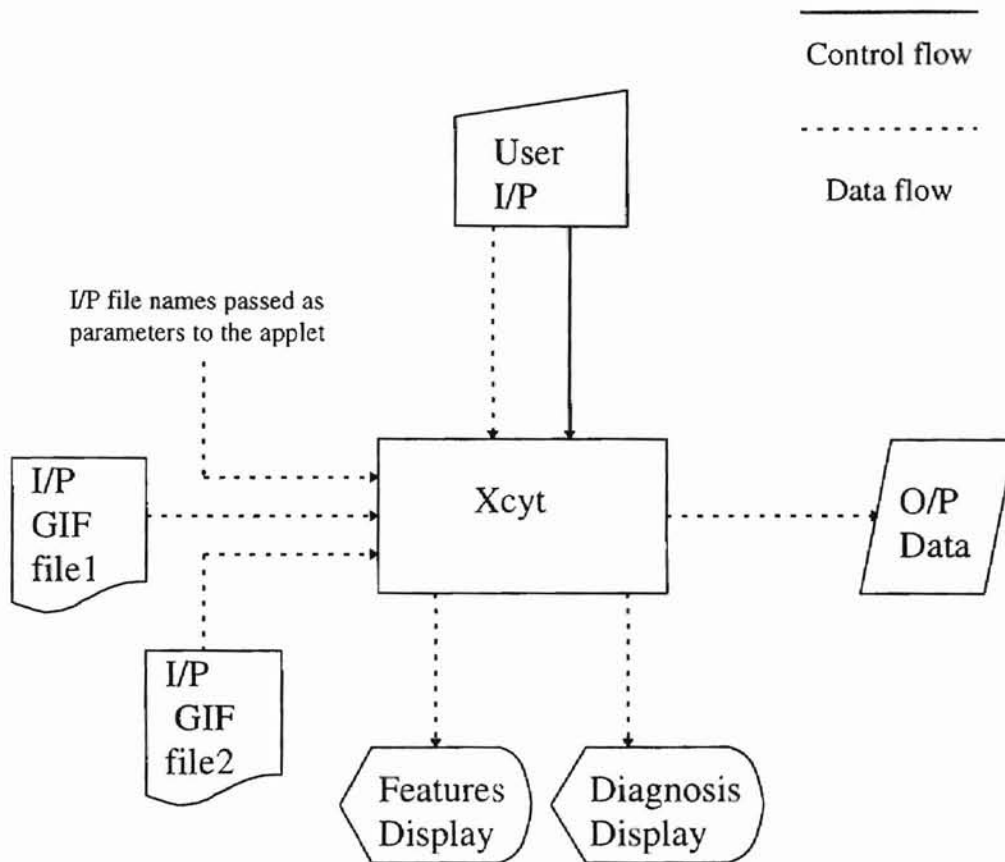


Figure 15. Model of the Xcyt system.

The Xcyt system gives the following outputs:

1. Display of the features,
2. Display of the diagnosis, and
3. Data of the computed features

The flow diagram of the Xcyt system is as shown in Figure 16. The Xcyt system uses 13 classes. These various classes and their implementations are discussed in the following sections. Table 4 lists the various classes of the Xcyt system.

Class name
Diagnosis
EmptyListException
Features
Grayscale
Node
Nucleus
PlugInFilter
Snake
Snakelist
format
menu
runThread
Xcyt

Table 4. List of classes used for the Xcyt system.

4.3.1. Xcyt class

The applet is implemented in the *Xcyt* class which is the main class. This class implements the modules to initialize, start, run and stop the applet. It communicates with the external world for delivering information like computed features and gathering

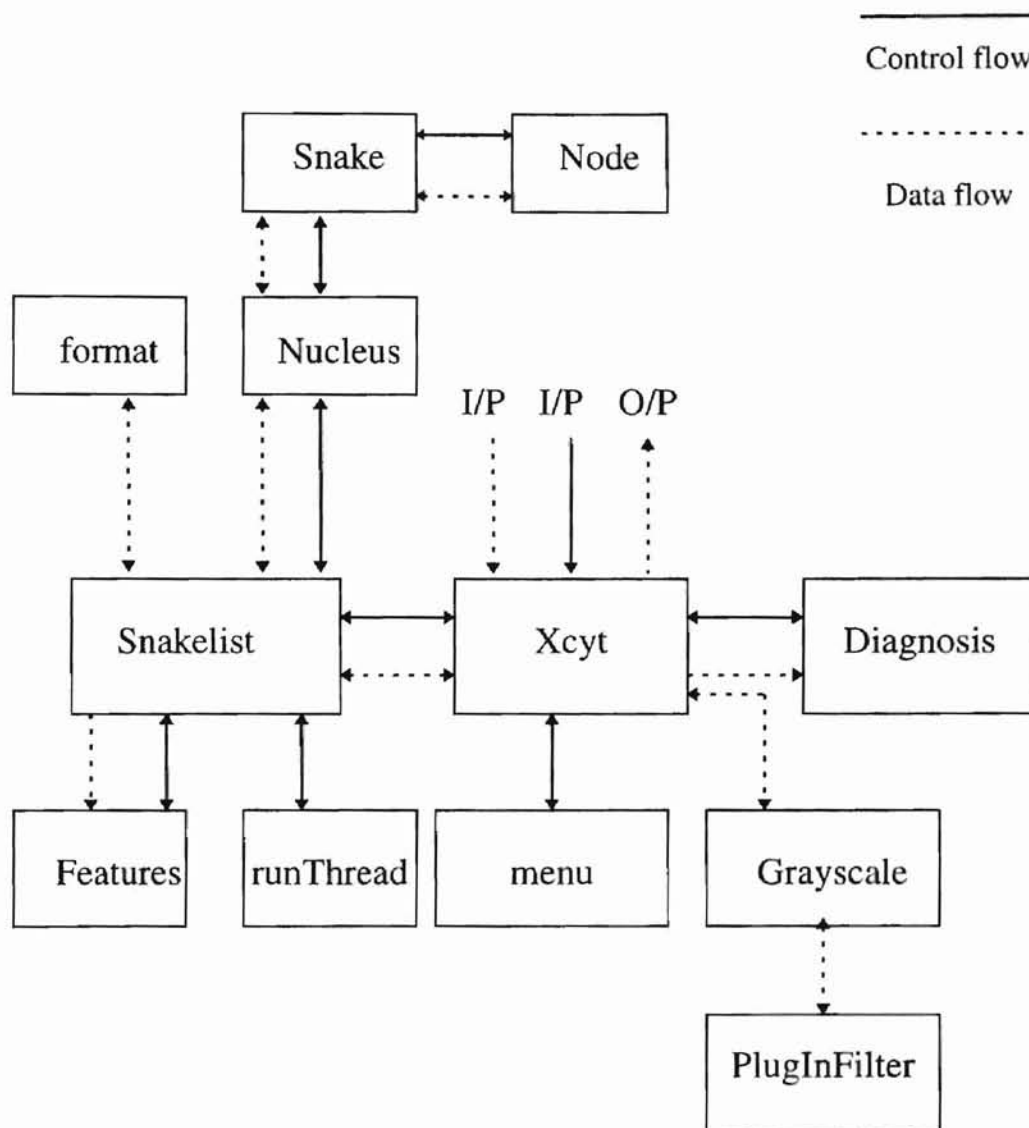


Figure 16. Flow diagram of the Xcyt system.

information like the input file names etc. It is responsible for loading the input image files, creating the user interface, handling graphics capabilities, mouse and key board events. Media tracker is used for loading the input image smoothly and handling loading errors.

4.3.2. Snake class

The *Snake* class is used for storing and manipulation of information pertaining to each nucleus. Each nucleus is internally represented as a linked list of points and each point is an object of type *Node*. This class is responsible for the manipulation of linked list, finding the exact boundary of the nucleus, and computing the nuclear features.

4.3.3. Node class

The *Node* class is used for representing a point (x, y) of the *Snake*.

4.3.4. Snakelist class

The *Snakelist* class is used for storing and manipulation of information pertaining to the nuclei. The list of nuclei are internally represented as a linked list of nuclei. The *Snakelist* class computes and holds the mean, worst and the standard error of the nuclear features of the nuclei resulting in a total of 30 features.

4.3.5. Nucleus class

The *Nucleus* class is used for storing information pertaining to each *Snake*. This class holds the *Snake* itself and the 10 nuclear features of the *Snake*.

4.3.6. Features Class

The *Features* class is used for creating a pop-up window and displaying the 30 nuclear features.

4.3.7. Diagnosis class

The *Diagnosis* class is used for computing the probability of malignancy using the benign and malignant distribution data and classifying the sample as either benign or malignant. It creates a pop-up window and displays the distribution curves, machine diagnosis, and the computed probability of malignancy for the given sample.

4.3.8. Grayscale class

The *Grayscale* class along with the *PlugInFilter* class are used for filtering the input image.

4.3.9. PlugInFilter class

As mentioned in the above section, it is used for filtering the input image.

4.3.10. EmptyListException class

As the name suggests, it is used for handling empty list exception in the *Snake* class.

4.3.11. format class

The *format* class is used for formatting the computed nuclear features up to 4 decimal places.

4.3.12. menu class

The *menu* class is used for creating and handling the menu bar for the user interface.

4.3.13. runThread class

The *runThread* class is used for handling the multithreading in the *Snakelist* class.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1. CONCLUSION

The diagnosis of breast cancer with palpable breast lesions relies on the experience of the examining medical doctor. A reliable computerized system can contribute both speed and accuracy. The Xcyt system has proved to be highly accurate in further classification of the inconclusive or indeterminate breast cancer cases of visual fine needle aspiration into either benign or malignant. It also revealed the fact that computer analysis of a preoperative FNA could be used for prognosis of breast cancer in place of the traditional pathologic examination of axillary lymph nodes, thus avoiding the routine axillary lymph node dissection. As discussed in the Chapters I, II, III, and IV the existing Xcyt system is designed and implemented using object-oriented technology, platform independent technology like Java, server side (gateway programs) programs, and JavaScript and made available on the Internet. As a result, many pathologists and doctors across the world will be able to use the system. The job of pathologists and doctors in diagnosing the patients with palpable breast lesions will become simpler, faster, and more accurate.

5.2. FUTURE WORK

Future versions of the system could include the following modifications.

1. The developed system will run on any machine that supports JDK1.1.3 and above, and JavaScript. But, the server needs to support CGI, Java, and Perl. CGI was used for server side gateway programs because it is robust and is the most widely used specification for creating executable programs that run on Web servers. As the usage of Java Servlets is becoming more popular and bug-free, Java Servlets could be used to substitute the CGI part in the system, thus eliminating the need for CGI and Perl. For a Windows (95/NT) based server, Microsoft ASP (Active Server Pages) and VBSCRIPT could be used to eliminate the CGI part in the system.
2. Currently the system is accessible to any user in the world. This could be restricted by providing a Web password system.
3. A database could be used for storing the data and their file names on the server, instead of storing the data files on the server for later use. Hence, the system would benefit from having built-in interfaces to popular database programs.

REFERENCES

- [1] Bankman, I. N., Christens-Barry, W. A., Weinberg, I. N., Dong, W. K., Ralph, D. S., and Brody, W. R. (1992) An algorithm for early breast cancer detection in mammograms. *Fifth Annual IEEE Symposium on Computer-based Medical Systems*, 362-368.
- [2] Bone, B., Pentek, Z., Perbeck, L., Veress, B. (1997) Diagnostic-accuracy of mammography and contrast-enhanced MR-imaging in 238 histologically verified breast-lesions. *Acta Radiology*, 38: 489-496.
- [3] Brunner, S., Langfeldt, B. (Eds.) (1990) *Advances in Breast Cancer Detection*. Springer-Verlag, Berlin.
- [4] Brunner, S., Langfeldt, B., and Andersen, P. E. (1984) *Early Detection of Breast Cancer*. Springer-Verlag, Berlin.
- [5] Comer, D. E., and Stevens, D. L. (1991) *Internetworking with TCP/IP*. Vol-II, Prentice-Hall, Inc., Englewood Cliffs, NJ.
- [6] Gilster, P. (1997) *The Web Navigator*. John Wiley & Sons, Inc.
- (21) [7] Graham, I. S. (1996) *The HTML Sourcebook: A Complete Guide to HTML 3.0*. John Wiley & Sons, Inc.
- [8] Griem, M. L., Jensen, E. V., Ultmann, J. E., and Wissler, R. W. (1973) *Breast Cancer: A Challenging Problem*. Springer-Verlag, Berlin.
- [9] Harlan, D., Powers, S., Doyle, P., and Foghler, M. O. (1996) *PERL5 for Web Programming*. Que Corp., Indianapolis, IN.
- [10] Horstmann, C. S., and Cornell, G. (1997) *Core Java. Vol 1: Fundamentals*, Sun Microsystems, Inc., CA.
- [11] Kopans, D. B. (1987) Nonmammographic breast imaging techniques: Current status and future developments. *Radiological Clinicology North America*, 25: 961-971.
- [12] Kotre, J. (1993) Image processing in the fight against breast cancer. *Engineering Science and Education Journal*, 41-46.
- [13] Morrison, M., December, J., Colton, P., Fletcher, M., Gulbransen, D., Rawlings, K., Perkins, C. L., Afergan, M., Beveridge, T., Bloch, B., Suresh, K. J., Lesh, R.,

- Leybzon, G., Macinta, T., Park, T., Scott, A., Seguin, C., and Thomsen, D. (1996) *Java Unleashed*, Sams.net Publishing, Indianapolis, IN.
- [14] Niles, R., and Dwight, J. (1996) *CGI by Example*. Que Corp., Indianapolis, IN.
- [15] O'Neil, S., Castelli, M., Gattuso, P., Kluskens, L., Madsen, K., and Aranha, G. (1997) Fine-needle aspiration of 697 palpable breast lesions with histopathologic correlation. *Surgery*, 122: 824-828.
- [16] Rubin, M., Horiuchi, K., Joy, N., Haun, W., Read, R., Ratzner, E., and Fenoglio, M. (1997) Use of fine needle aspiration for solid breast lesions is accurate and cost-effective. *American Journal of Surgery*, 174: 694-698.
- [17] Sickles, E. A. (1983) Breast CT scanning, heavy-ion mammography, NMR imaging, and diaphanography. In: Feig S. A., McLelland R. (eds.) *Breast Carcinoma: Current Diagnosis and Treatment*. Masson, New York, 233-250.
- [18] St-Arneault, G. P. B., and Israel, L. (1976) *Breast Cancer: A Mutidisciplinary Approach*. Springer-Verlag, Berlin.
- [19] Stevens, W. R. (1990) *UNIX Network Programming*. Prentice-Hall, Inc., NJ.
- [20] Teague, M. W., Wolberg, W. H., Street, W. N., Mangasarian, O. L., Suzanne, L., and David, L. P. (1997) Indeterminate fine-needle aspiration of the breast: Image analysis-assisted diagnosis. *Cancer (Cancer Cytopathology)*, 129-135.
- [21] Wolberg, W. H., Street, W. N., and Mangasarian, O. L. (1994) Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters*, 77: 163-171.
- [22] Wolberg, W. H., Street, W. N., and Mangasarian, O. L. (1995) Breast-Cancer Diagnosis and Prognosis via Linear-Programming. *Operations Research*, 43: 570-577.
- [23] Wolberg, W. H., Street, W. N., and Mangasarian, O. L. (1997) Computer-derived nuclear features compared with axillary lymph node status for breast carcinoma prognosis. *Cancer (Cancer Cytopathology)*, 172-179.

VITA

Sree Rama Krishna Rao Mallina

Candidate for the Degree of

Master of Science

Thesis: REMOTE CANCER DIAGNOSIS

Major Field: Computer Science

Biographical:

Personal Data: Born in Vundrajavaram, Andhra Pradesh, India, On January 12, 1974, the son of Dr. Seshagiri Rao and Vijaya Lakshmi Mallina.

Education: Graduated from Gowtham Junior Residential College, Vijayawada, Andhra Pradesh in May 1990; received Bachelor of Engineering degree in Computer Science and Engineering from University of Madras, Madras, India in May 1995; completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in July 1998.

Professional Experience: Graduate Research Assistant, Department of Computer Science, Oklahoma State University, May 1996 to August 1996; Graduate Teaching Assistant, Department of Computer Science, Oklahoma State University, January 1996 to May 1997.