Faculty of Computer Science and Information Technology
University of Malaya
Kuala Lumpur

Automatic
Email Classification System

Prepared by,

PHANG SIEW TING
WEK 000365

Under Supervision of
Dr. LEE SAI PECK

Moderator
Mrs. NAZEAN JOMHARI

being a Dissertation submitted in partial fulfillment
of the requirements for the Degree of
Bachelor of Computer Science

ABSTRACT

The growing problem of unsolicited bulk email and the growth of the volume of email received has generated a reliable need for email tool that assists the users manage the emails in an effective way. For this purpose, several Machine Learning algorithms has been purposed to automate the classification of emails. The algorithms learn to classify emails based on it textual contents, and subsequently assign individual emails into a predefined set of categories or bins in accordance with the preferences of a user.

Automatic Email Classification System is an email reader tool that implements machine learning algorithm in email classification, manipulated by a Graphical User Interface. The interface provided allows building a classifier algorithm, testing it, and applying it to previously unread messages. New messages can be classified and stored into corresponding predefined folder on the fly automatically as they are downloaded from a server.

The automated email classification of the system foreseen a greatest advantage over the existing email clients in the market, such as Microsoft Outlook, Netscape Messenger, etc, which rely mostly on hand-constructed keyword-matching rules.

ACKNOWLEDGEMENT

The development of the Automatic Email Classification System was carried out with the advice, assistance and contributions from many individuals.

First and foremost, I would like to extend my most sincere gratitude to Dr Lee Sai Peck, my project supervisor who has provided me with unlimited support and guidance.

Special thanks to Ms Melissa Seah Ching Yieng, Researcher of Technology Research Group, MIMOS Berhad, for giving me thoughtful advise and informative comments in developing the system.

Special thanks also goes to Mr Teh Ying Wah, lecturer from Department of Information Science, for being giving useful information and advise for my thesis.

Last but not least, my sincere thanks go to my housemates especially Mr Lim Han Shyong, Mr Ee Chee Wan, and my dearest roommate Ms Lee Chia Yee, for valuable help and discussions.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Diagrams

CHAPTER 1

INTRODUCTION

1.1 Emails and E-Communication Evolution

In this fast paced growth of technology, everybody is talking about the electronic communication (e-communication). The use of computer-based e-communication tools has its roots in the 1960s with the advent of email systems running on mainframe computers that allowed text messages to be exchanged among users registered in the operating system. These early e-communication systems gave way to more sophisticated ones with the interconnection of first mainframes, and then desktop computers through networks and the Internet, which have extended the reach of e-mail beyond single organizations.

In recent year, the tremendous growth of the Internet witnesses a new era in e-communication. Email has taken over as one of the most practical and widely used forms of communication. Clearly, email has measurable benefits over previous modes of communication especially in a society filled with computer savvy individuals. Unlike the traditional mail system (e.g. snail mail) or the telephone, email is delivered nearly instantaneously to almost anywhere in the world and costs practically nothing to send.

The explosion in email is dramatically changing the way people interact with one another. Since Email is extremely cheap and easy to send, it has gained enormous popularity not simply as a means for letting friends and colleagues exchange message, but also as a medium for conducting electronic commerce. Internet users by The Gallup Organization found that e-mail remains the No. 1 activity for people [Pastore, 2001]. In business field, email is not only a means of communication, but it is a way of doing business. Studies show that 78 percent of Japanese businesses, 68 percent of US businesses and 62 percent UK businesses have Internet access and use email, while 97 percent of North American workers rely on email as their main communications tool [AmikaNow.com, 2002]. Email is also expected to be the most important application for wireless messaging. There are 500 mobile device users in the world today and this figure is expected to double by 2003 [AmikaNow.com, 2002]. Besides Email is also an effective way to rally members and advise Congress of association views on current legislative issues [Turner, 2001].

1.2 Problem Definition

Email is one the most successful computer applications yet devised. The speed and ease of email explains why email becomes the killer application of Internet. However, email has become the victim of its own success. Several issues have been arising in recent year. In the following subsection, I will discuss two major issues that are caused by the increasingly popular of email-- *Email Overload and Spam.*

1.2.1 Email Overload

Email overload is defined as the overflow of emails in the Internet. The popularity of email has led to high daily volumes of email being transferred and hence, generates the heavy email traffic over the Internet. According to International Data Corp (IDC), everyday 8 billion of emails are exchanged on the Internet. By 2005 the figure will be increased to 35 billion emails a day and the number of email mailboxes is expected to jump to 983 million. IDC holds three factors responsible for the projected e-mail boom: Web services, wireless access and workers without e-mail. And despite a shakeout among free Web e-mail providers, IDC believes the remaining providers will see significant growth in mailboxes tied to Web sites [Pastore, 2001; IDC, 2001].

The excessive high volume of email creates problems for personal information management. Users often have cluttered inboxes containing hundreds of messages, including outstanding tasks, partially read documents and conversational thread. There are millions of email users worldwide who often spend significant proportions of their work time on processing emails. Research shows that a business user will take more than 4 hours per day to deal with an average of 50 work-related messages

[AmikaNow.com, 2002]. Certain individuals also experienced major problems in reading and replying to email in a timely manner, with backlogs of unanswered email, and in finding information in email systems.

United States Congress has become the victim of email overload. Recent news articles have indicated that Congress is so inundated with email from constituents that the messages are largely ignored. The Congress Online Project indicates that as many as 55,000 and 8,000 e-mail messages per month are directed to members of the U.S. Senate and House of Representatives respectively-- roughly 80 million messages a year for the Congress as a whole. The study also figures that the number of messages is increasing by one million each month [Turner, 2001].

1.2.2 Spam

Spam is junk email that is sent to us by someone who has no prior or existing relationship to us. Whether one calls it unsolicited commercial email (UCE), unsolicited bulk e-mail (UBE) or junk mail, spam is defined by the fact that the recipients did not solicit the mail or divulge their email addresses for the purposes of receiving such mail [Boetriger ,2002]

Each day, thousands of spam programs scan web pages, newsgroups, and other online documents to harvest email addresses in bulk. As a result, it is becoming increasingly common for users find their mailboxes cluttered by large quantities of unsolicited bulk email. According to Brightmail's latest interception figures, unsolicited bulk e-mail made up a whopping 36 percent of all e-mail traveling over the Internet, up from 8 percent about a year ago [Lemos, 2002]. Research foresees a *forty-fold* increase in spam between 1999 and 2005, from 40 pieces to 1,600 pieces received per person annually [Boetriger, 2002]. These unsolicited messages arrive in a user's electronic mailbox with headers such as "YOU MUST READ THIS," "MAKE MONEY FAST," or "NEW ADULT WEB SITE WITH HOT LINKS!!!,"and advertise a seemingly limitless variety of products and services, including get-rich-quick schemes, pornography, and even services that allow the user to send his or her own spam.

As a result, many readers of email must now spend a non-trivial portion of their time online wading through such unwanted messages. A European Commission (EC) study indicated that the daily flood of the spam which regularly clogged Internet users' mailboxes is costing Web users 10 billions euros (RM36 billion)[Computimes, 2001]. Besides, the Spam also consumes expensive network resources, such as disk space and CPU utilization. This is especially true for high volume email systems that receive and store thousands of copies of the same spam message.

Businesses also get double impact with the spam. This because they are not only have to pay throughput costs for email that is unrelated to their business mission, but also deal with the costs and loss of productivity associated with staff time that is wasted dealing with spam [Turner, 2001].

1.3 Project Objective

The main objective of the Automatic Email Classification System is to build assists the user to manage the large amount of email message in an efficiency way. It combines e-content interpretation strategies with intelligent personal assistance to address the problem of email overload and Spam. It provides applications with the mechanisms to learn patterns in a set of email and then use this information to automatically file all new email into the defined folders.

- At the business enterprise level the system can improves the corporate management of email, filtering irrelevant messages and consequently make the staff more productive in their job.
- At the personal level the system helps users manage their email, improving response time, and reducing the feeling of overload.

1.4 Project Description

Automatic Email Classification System is a standalone java-based email client: by client, we mean that it interacts directly with the user, allowing mail to be sent, read, filed, printed and otherwise manipulated through a graphical interface.

One of the great advantages of the system over the other email clients in the market is the ability to classify the email ***automatically*** into predefined folders based on the message textual content. It implements the Artificial Intelligence method - Machine Learning algorithm, in which the system will launch a training process to train a classifier. The classifier wil! learn to classify the emails in accordance on the users' preferences, using the email message retrieved from the users' personal account.

After the classifier is generated, each email will be assigned to the predefined folders on the fly when it is downloaded from the Pop3 server.

1.5 Project Scope

Automatic Email Classification System will cover the following areas:

- Implement a password authenticator for authorized access for valid users to their mailboxes.
- Retrieve emails from the server
- Allows the uses to view the full email text, delete, reply or forward the emails, which is manipulated through a Graphical User Interface (GUI)
- Allow the authorized user to send the attachment and view the incoming attached file in various format, includes RTF, HTML, JPEG, BITMAP, ZIP.
- Support standards for POP3, SMTP and MIME
- Provides automated function in building an email classifier, which will learn to classify the email, based on the users' preferences. The learned classifier will be applied to the incoming emails and classify them into the folders.
- As the email content might be change over time, the users can choose to retrain the classifier to produce an up-to-date classifier.
- Sort the mail by sender, subject and date in ascending or descending order in message list.

1.6 Project Limitations

- The system classifies the emails based on the textual content. The attachments will not be taken consideration in the classification process.
- The system assumes that all the emails are written in pure English. The mixture with other languages might affect the classification accuracy

1.7 Project Timeline

ID	Activities	Start	End	Duration	Semester 1				Semester 2			
					Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan
1	Proposal report	3/6	8/6	1 w	■							
2	Introduction	10/6	29/6	3 w	■	■						
3	Literature Review	24/6	3/8	5 w	■	■						
5	System Analysis	5/8	24/8	3 w			■					
6	System Design	26/8	13/9	3 w			■	■				
7	Exam	16/9	4/10	3w				■	■			
8	System Coding	5/10	28/12	11 w					■	■	■	
10	System Testing	16/12	18/1	5 w							■	■
11	System Implementation	12/1	26/1	2 w								■
13	System Documentation	3/6	26/1	28w	■	■	■	■	■	■	■	■

Table 1.1 Project Timeline of Automatic Email Classification System

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to Email

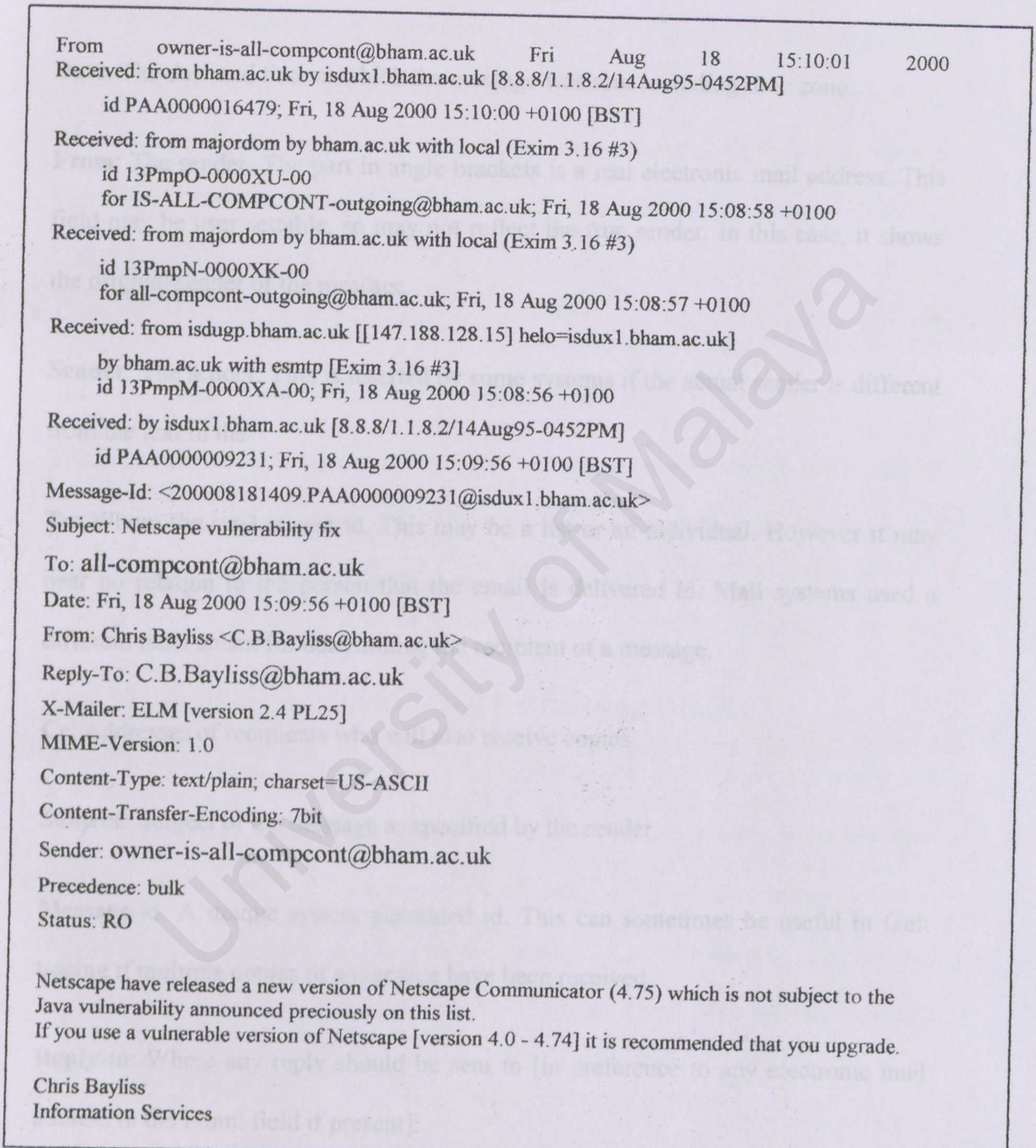
Email is a computer-based method of sending messages from one computer user to another [Email, 2000]. These messages usually consist of individual pieces of text, which you can send to another computer user even if the other user is not logged in (i.e. using the computer) at the time you send your message. The message can then be read at a later time. This procedure is analogous to sending and receiving a letter. Originally, email messages were restricted to simple text, but now many systems can handle more complicated formats, such as graphics and word-processed documents.

When mail is received on a computer system, it is usually stored in an electronic mailbox for the recipient to read later. Electronic mailboxes are usually special files on a computer, which can be accessed using various commands. Each user normally has their individual mailbox.

2.1.1 Structure of an Email Message

Email is a semi-structured document, which consists of a structured header and the unstructured message body. The header includes standard fields such as the sender [the From field], a list of recipients (the To and Cc field), the return path, date and subject.

Figure 2.1 shows an example of typical email message. The header consists of lines beginning with a keyword followed by a colon (:), followed by information on each line.



```
From         owner-is-all-compcont@bham.ac.uk          Fri        Aug        18        15:10:01        2000
Received: from bham.ac.uk by isdux1.bham.ac.uk [8.8.8/1.1.8.2/14Aug95-0452PM]
        id PAA0000016479; Fri, 18 Aug 2000 15:10:00 +0100 [BST]
Received: from majordom by bham.ac.uk with local (Exim 3.16 #3)
        id 13PmpO-0000XU-00
        for IS-ALL-COMPCONT-outgoing@bham.ac.uk; Fri, 18 Aug 2000 15:08:58 +0100
Received: from majordom by bham.ac.uk with local (Exim 3.16 #3)
        id 13PmpN-0000XK-00
        for all-compcont-outgoing@bham.ac.uk; Fri, 18 Aug 2000 15:08:57 +0100
Received: from isdugp.bham.ac.uk [[147.188.128.15] helo=isdux1.bham.ac.uk]
        by bham.ac.uk with esmtp [Exim 3.16 #3]
        id 13PmpM-0000XA-00; Fri, 18 Aug 2000 15:08:56 +0100
Received: by isdux1.bham.ac.uk [8.8.8/1.1.8.2/14Aug95-0452PM]
        id PAA0000009231; Fri, 18 Aug 2000 15:09:56 +0100 [BST]
Message-Id: <200008181409.PAA0000009231@isdux1.bham.ac.uk>
Subject: Netscape vulnerability fix
To: all-compcont@bham.ac.uk
Date: Fri, 18 Aug 2000 15:09:56 +0100 [BST]
From: Chris Bayliss <C.B.Bayliss@bham.ac.uk>
Reply-To: C.B.Bayliss@bham.ac.uk
X-Mailer: ELM [version 2.4 PL25]
MIME-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit
Sender: owner-is-all-compcont@bham.ac.uk
Precedence: bulk
Status: RO

Netscape have released a new version of Netscape Communicator (4.75) which is not subject to the
Java vulnerability announced preciously on this list.
If you use a vulnerable version of Netscape [version 4.0 - 4.74] it is recommended that you upgrade.
Chris Bayliss
Information Services
```

Figure 2.1 An Example of Typical Email lEmail, 2000]

A brief explanation of each field of the header in the following:

Received: These lines indicate the route that the email has taken and which systems have handled it and the times that it was handled.

Date: The date and time at which the message was sent including time zone.

From: The sender. The part in angle brackets is a real electronic mail address. This field may be user settable, so may not reflect the true sender. In this case, it shows the original sender of the message.

Sender: The sender. This is inserted by some systems if the actual sender is different from the text in the

To: Whom the mail is sent to. This may be a list or an individual. However it may bear no relation to the person that the email is delivered to. Mail systems used a different mechanism for determining the recipient of a message.

Cc: Addresses of recipients who will also receive copies.

Subject*:* Subject of the message as specified by the sender.

Message-id: A unique system generated id. This can sometimes be useful in fault tracing if multiple copies of a message have been received.

Reply-to: Where any reply should be sent to [in preference to any electronic mail address in the From: field if present].

X-Mailer*:* Any field beginning with X can be inserted by a mail system for any purpose.

2.1.2 Post Office Protocol (POP3)

POP3 is client/server protocol (currently in version 3) for the management and transfer of stored email between a client and an Internet host acting as server. The POP server is constantly connected to the Internet so it can receive messages as they arrive via SMTP and buffer them for clients. It defines support for a single mailbox for each user. POP3 mail can be retrieved by most popular e-mail products, such as Outlook, Outlook Express, Communicator, Eudora and many more. [123m.com, 2002]

2.1.3 Simple Mail Transfer Protocol (SMTP)

SMTP is an Internet protocol for transferring mail between Internet hosts. SMTP can upload mail directly from the client to an intermediate host but can receive mail only on computers constantly connected to the Internet

2.1.4 Multipurpose Internet Mail Extensions (MIME)

MIME is a standard developed as a set of extensions to SMTP - the standard email protocol in use on Internet [Email, 2000]. Basically, a set of extensions has been defined which state how different types of data are to be encoded as text and included in the message. MIME can use various types of encoding, but the one normally used for binary files is base64. This encodes the file into printable text and offers some advantages over uuencode.

This approach has the advantage of allowing multimedia mail to be carried using existing email services without modification. MIME multimedia mail is constructed and displayed by suitable user interfaces. If such mail is received and read using a non-MIME user interface, any encoded sections of the message are displayed as text and can be extracted and converted using suitable programs. MIME is supported by many packages, for example recent versions of Pegasus, Eudora, the Microsoft exchange client on Windows 95 and some versions of elm and pine on Unix. [Email, 2000]

2.2 Existing System Analysis

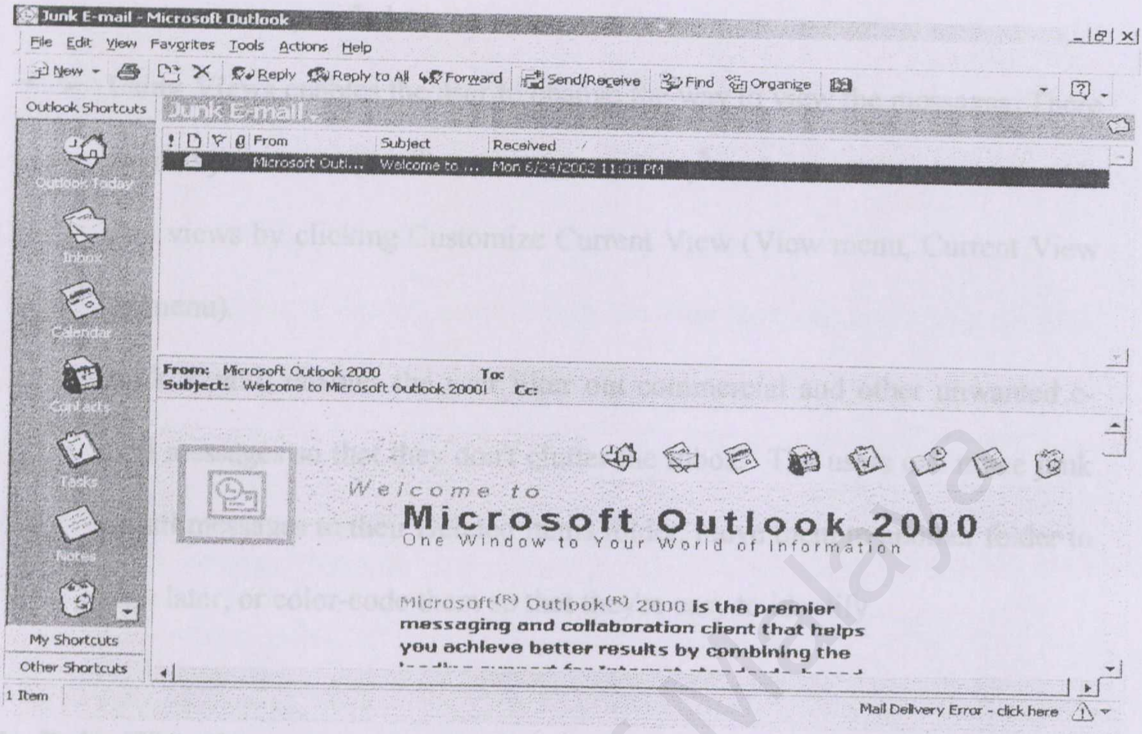
Nowadays, there are many email clients available in the market that help the user email user to manage their email message. There vary in the types of services it provides, the features and also the Graphic User Interface.

Most of the email clients that currently available are designed for both personal use and business use, which manipulates through an interesting and user-friendly Graphic User Interface (GUI), such as Microsoft Outlook, Netscape Messenger, Eudora, Pegasus and so on.

Among the email clients, Microsoft Outlook and Netscape Messenger are the most popular due to its multiple functions and easy-to-use tool in assisting the user to manage their mailbox.

Generally, all the email clients have the same function that provides the service to receive, reply, forward, compose and delete email. The following subsection will give an overview on the services and features of these two email clients, focusing on the way to organize the mailbox as well as filter out the junk email.

2.2.1 Microsoft Outlook 2000



Microsoft Outl... Welcome to ... Mon 6/24/2002 11:01 PM

From: Microsoft Outlook 2000 To:
Subject: Welcome to Microsoft Outlook 2000! Cc:

Welcome to
Microsoft Outlook 2000
One Window to Your World of Information

Microsoft(R) Outlook(R) 2000 **is the premier messaging and collaboration client that helps you achieve better results by combining the**

Figure 2.2 Graphical User Interface of Microsoft Outlook 2000

Microsoft Outlook 2000 is the premier messaging and collaboration client that helps us achieve better results by combining the leading support for Internet standards-based and Microsoft Exchange Server-based e-mail with integrated calendar, contact, and task management features.

a. Organize Tool

Microsoft Outlook 2000 provides a organize tool that assists the user to manage their mailbox. Generally, there are four choices available to organize the inbox by using the organize tool, that's *Using Folders, Using Colors, Using Views and Junk E-mail.*

- ➢ **Using Folders** enables the user to create a new folder to store messages from a specified sender, and then set up a rule to automatically move all future messages from that specified sender into that folder.

- **Using Colors** enables the user to create rules to color-code messages that meet criteria specified.
- **Using Views** enables the user to change the way to view the messages. There are many custom views that can choose from, but the user can also create his own views by clicking Customize Current View (View menu, Current View submenu).
- **Junk E-mail** enables the user filter out commercial and other unwanted e-mail messages so that they don't clutter the Inbox. The users can move junk e-mail messages to their Deleted Items folder, move them to another folder to view later, or color-code them so that they're easy to identify.

b. Rules Wizard

The Rules Wizard that was available as an add-in in Outlook 97 is now a built-in feature that automatically moves, deletes, forwards, or flags incoming and outgoing messages.

The Rules Wizard helps the user manage the e-mail messages by using rules to automatically perform actions on messages. After creating a rule, Microsoft Outlook applies the rule when messages arrive in the Inbox or when the users send a message. The users can add exceptions to their rules for special circumstances, such as when a message is flagged for follow-up action or is marked with high importance. A rule is not applied to a message if any one of the exceptions specified is met.

c. Email Filtering

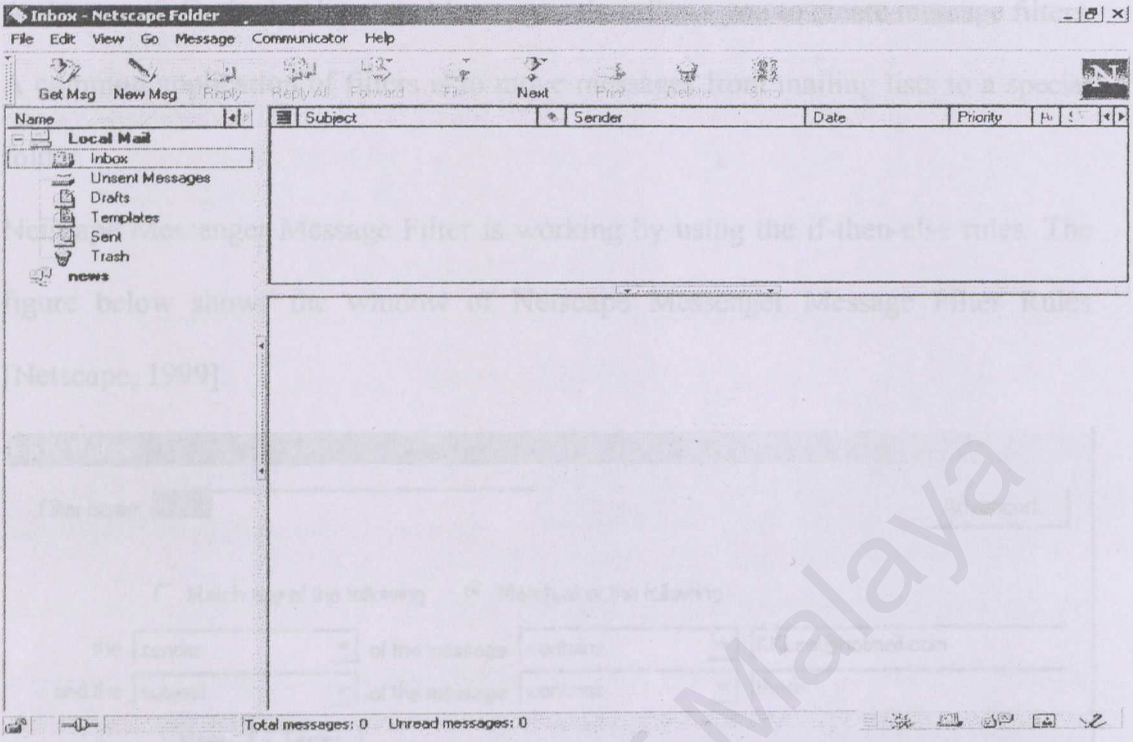
Besides the organize tools and rules wizard, Microsoft Outlook 2000 also provides automatic email filters to move the junk or adult content email messages from the Inbox to a junk e-mail folder, Deleted Items folder, or any other folder specified by the user, automatically. The filters work by looking for keywords. The list below shown a few instances of exactly which words the filter looks for and where the filter looks for them.

Junk email filter:
From is blank
From contains "sales@"
From contains "@public"
To contains friend@
To contains "public@"
Subject contains "!" AND Subject contains "free"
Subject contains "$$"
Subject contains "advertisement"
Body contains ",000" AND Body contains "!!" AND Body contains "$"
Body contains "check or money order"
Body contains "money-back guarantee"
Body contains "money back "

Adult Content Filter:
Body contains "must be 18"
Body contains "adults only"
Body contains "must be 21"
Body contains " xxx "
Subject contains "over 18"
Subject contains "adult s"
Subject contains "be 18"
Subject contains " sex"
Subject contains "free" AND Subject contains "adult"
Subject contains "free" AND Subject contains "sex"

The user can also filter messages based on a list of e-mail addresses of junk and adult content senders. There are third party filters, which are regularly updated, that can be added to Outlook. These filters have the latest lists of commercial and adult content senders.

2.2.2 Netscape Messenger



Total messages: 0 Unread messages: 0

Figure 2.3 Graphical User Interface of Netscape Messenger

Netscape Messenger is one program in the Netscape Communicator suite of programs. It is a powerful, easy-to-use e-mail program that uses a graphical user interface with multiple buttons, windows, and dialog boxes.

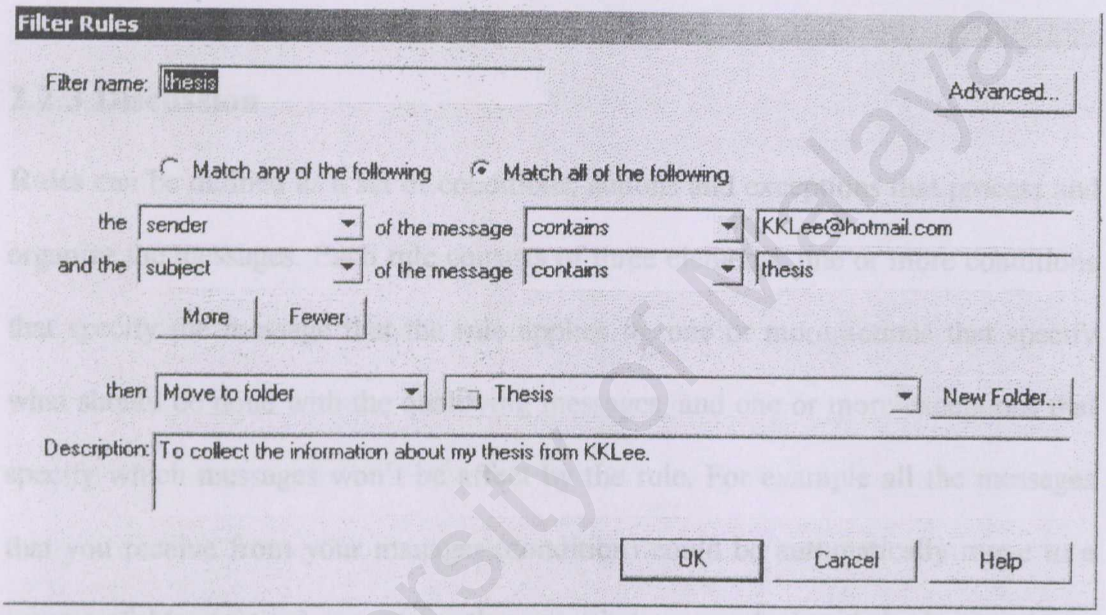
a. Organizing The Inbox

One of the great advantages of Netscape Messenger over Microsoft is the storage method for old messages. Netscape has a folder hierarchy that allows for folders within folders, whereas Microsoft Outlook is a one-layer flat file. It previews all the folders in a tree structure, which give user a clearer view on the folders organization.

b. Email Filtering

As Microsoft Outlook, Netscape Messenger also allows you to create message filters. A common application of filters is to move messages from mailing lists to a special folder.

Netscape Messenger Message Filter is working by using the if-then-else rules. The figure below shows the window of Netscape Messenger Message Filter Rules [Netscape, 1999].



Filter Rules

Filter name: thesis

Advanced...

○ Match any of the following ● Match all of the following

the sender of the message contains KKLee@hotmail.com

and the subject of the message contains thesis

More Fewer

then Move to folder Thesis New Folder...

Description: To collect the information about my thesis from KKLee.

OK Cancel Help

Figure 2.4 Netscape Messenger Message Filter Rules Wizard

Figure 2.4 states a rule named " Thesis", that moves all the emails which the sender is KKLee@hotmail.com AND the subject contains the word "thesis" to the folder "Thesis", with the description "To collect the information about my thesis from KKLee". After you click the OK button, a set of rules will be generated and store in Rules.dat, as shown Figure 2.5.

```
version="6"
        logging="yes"
        name="thesis"
        enabled="yes"
        description=""
        type="1"
        action="Move to folder"
        actionValue="Inbox.sbd/Thesis"
        condition=" OR [from,contains,KKLee@hotmail.com]
                        OR [subject,contains,thesis]"
```

Figure 2.5 Rules.dat

2.2.3 Discussion

Rules can be defined as a set of conditions, actions and exceptions that process and organize the messages. Each rule consists of three elements: one or more conditions that specify the message that the rule applies to, one or more actions that specify what should be done with the qualifying messages, and one or more exceptions that specify which messages won't be affect by the rule. For example all the messages that you receive from your manager (condition) could be automatically move to a separate folder (action) except for the ones that are marked with high importance (exception).

Both of the email clients rely mostly on manually constructed pattern-matching rules that need to be tunes to each user's incoming messages. However, constructing and maintaining the rules is a burdensome task, especially so if the number of folders is large or the folder organization requires significant restructuring. A system that would learn automatically to filter out the Spam as well as classify the emails into the folders would, therefore present significant advantages.

2.3 Intelligent Method for Email Classification

Email data is considered a text data type, though with some specific feature. Thus, email classification is one of the functionalities of text classification. Several Machine-Learning (ML) algorithms have been applied to automate the text classification. These algorithms learn to classify emails into the user-defined folders, based on the textual content, after being training on the training data. Apart from that, Information Retrieval (IR) technique has been applied to preprocess the training data. An overview of ML and IR in Text Classification is given in the following sections.

2.3.1 Text Classification (TC)

Text Classification (a.k.a Text Categorization, TC) is the task of building software tools capable of classifying text [or hypertext] documents under predefined categories or subject codes. TC has witnessed a booming interest in recent times, due to the availability of ever larger numbers of text documents in digital form and to the ensuing need to organize them for easier use.

TC dates back to the early 60's, but only in the early '90s did it become a major subfield of the information systems discipline, due to increased applicative interest and to the availability of more powerful hardware. [Sebastiani, 2002] Until the late 80's, the most popular approach for TC was an expert system capable of taking TC decisions. This is called knowledge engineering (KE) techniques, which consisted in manually building a set of rules for TC [Sebastiani, 2002].

In the '90s this approach has increasingly lost popularity especially in research community. The dominant approach is nowadays one of building text classifiers automatically by learning the characteristics of the categories from a training set of pre-classified documents. State-of-the-art machine learning methods have recently been applied to the task, leading to systems of increased sophistication and effectiveness. This has encouraged the application of TC techniques to novel domains, such as Web page categorization under hierarchical catalogues, and spoken document categorization. It also leads to the progressive adoption of automatic or semi-automatic (i.e. interactive) classification systems in applicative contexts where manual work was the rule. [Sebastiani, 2002]

Current-day TC is thus a discipline at the crossroads of machine learning and information retrieval, and as such it shares a number of characteristics with other tasks such as information extraction from texts and text mining. [Sebastiani, 2002]

2.3.2 Machine Learning

Machine learning is a subfield of the field of artificial intelligence that deals with programs that learn from experience. Its main objective is to parallel the human learning abilities in computers [Mitchell, 2002]. It is said to occur in a program that can modify some aspect of itself, often referred to as its *state*, so that on a subsequent execution with the same input, a different [hopefully better] output is produced.

Learning can be defined as changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently the next time. In the case of a classification algorithm, learning is the ability to improve its performance at classifying the instances presented to it.

Machine Learning can be divided to supervised learning and unsupervised learning. Supervised Learning is learning by analyzing the training dataset, which the label of each training sample is provided. It can be classified to role learning, learning by being told, learning by analogy and learning from examples. Among these techniques, learning from examples, a special case of inductive learning, appears to be the machine learning technique for knowledge discovery or data analysis. It induces a general concept description that best describes the positive and negative examples.

Unsupervised Learning also known as learning from observation and discovery is a contrast to supervised learning [Han & Kamber, 2001]. It signifies a mode of machine learning where the system learns without consulting to a known class label. Instead, the system is left to find interesting patterns, regularities, or clusters among them. The is a very general form of inductive learning that includes discovery systems, theory – formation tasks, the creation of classification criteria to form taxonomic hierarchies, and similar tasks without the benefit of an external teacher.

2.3.2.1 Machine Learning in Text Classification

Classification is the process of finding a set of models (or functions) that describe and distinguish data classes and concepts, for the purpose of the being able to use the model to predict the class of objects whose class label is known [Han & Kamber, 2001]. It is a very useful operation in problems where predictions for new cases can be made by looking at the cases from past experience with known answers. The examples of such problems are fraud detection, marketing, healthcare outcomes, investment analysis, automatic article and image classification. Classification can be done over different types of data, such as relational, transactional, data warehouse, text and image. Email data is considered a text data type, though with some specific features.

In the case of Text Classification, a general inductive process (also called the learner) automatically builds a text classifier by learning, from a set of documents. [Sebastiani, 2002]. The advantages of this approach are an accuracy comparable to that achieved by human experts, and a considerable savings in terms of expert labor power, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories. The machine learning algorithms that used for Text Classification are Naïve Bayesian, Artificial Neural Network, Support Vector Machines, Decision Trees and K-Nearest Neighborhood [Itskevitch, 1997].

Most Text Classification approaches use supervised learning algorithms. In the case of email classification, the supervised learning algorithms require the users classify the emails into folders manually to construct the training data set. However, my thesis will focus on the unsupervised learning algorithm: clustering, in email classification, which eliminate the laborious manually classification process will be eliminated. An overview of the application of clustering methods in email classification will be given at section 2.6

2.3.3 Information Retrieval (IR)

Information Retrieval (IR) can be defined as the study of systems for indexing, searching, and recalling data, particularly text or other unstructured forms [Han & Kamber, 2001]. The information retrieval community has spent many years developing robust retrieval methods applicable to many retrieval tasks concerning text-containing documents. Although for many years the focuses have been primarily on retrieval tasks, here, too, the last decade has seen a significant increase in interest in the use of such methods for text-classification tasks.

Nowadays, Text Classification heavily depends on the basic machinery of information retrieval. The reason is that Text Classification is a content-based document management task, and as such it shares many characteristics with other IR tasks such as text search [Sebastiani, 2002]. In the case of email classification in my thesis, IR techniques are used in text analyzing such as stemming, stopwords removal in document preprocessing and weighting in Feature Selection.

2.4 Related Work

Research in intelligent method for email classification has been started since a few years ago. A number of systems have examined different ways of classifying email using Machine Learning and IR approaches. Some of these systems are described below:

- **Learning Rules that Classify E-mail, by Cohen [Cohen, 1996]**

  Cohen describes the RIPPER algorithm in this paper. Ripper is a rule –learning algorithm which is designed for efficient performance on large and noisy datasets. The paper compares this "keyword spotting" approach with an IR method, based on TF-IDF weighting. Both approaches show similar accuracy. However, Cohen argues that keyword spotting is more useful as it induces an understandable description of the email filter.

- **Support Vector Machine, by Jake D.Brutlag and Christopher Meek [Brutlag, & Meek, 2000]**

  Brutlag and Meek described three classification algorithms for email classifications that are linear Support Vector Machine (SVM), a TFIDF classifier and a simple language model called the Unigram Language Model. They preprocessed the data using a Zipf filter which removes the common words. They found that the classification accuracy is depending on the store of email used. They also research on the effect of folder size on performance of the algorithms. Result shows that TF-IDF offered the best performance for sparse folders while SVM was very accurate on dense folders.

- **Naïve-Bayes, by Jefferson Provost [Provost, 1999]**

  Provost compares the RIPPER algorithm and Naïve-Bayes algorithm for email classification and spam filtering. Result shows that Naïve-Bayes significantly outperformed RIPPER in both cases.

- **Email Classification with Co-Training by S. Kiritchenko and S. Matwin [Kiritchneka,& Matwin, 2002]**

  An attempt was done by Kiritchenko and Matwin to apply the co-training algorithm, a semi-supervised learning algorithm that uses unlabeled data along with a few labeled samples to boost a performance of a classifier. This algorithm allows one to start with just a few labeled examples to produce an initial weak classifier and then use only unlabeled data to improve the performance. The results show that the performance of co-training depends on the learning algorithm it uses. In particular, Support Vector Machines significantly outperforms Naive Bayes on email classification.

2.5 PIGEON – Prototype of Automatic Email Classification System

PIGEON is a prototype of automatic email classification system that implements Naïve Bayesian Classifier. Many attempts have been done in email classification or anti-Spam filtering and the result is quite impressed. The following subsection gives a brief description of Naïve Bayesian Algorithm.

2.5.1 Naïve Bayesian Classifier

Naïve Bayesian classifier is a probabilistic classifier based on Bayesian Decision Theory [Provost, 1999; Mitchell, 1996], which originated from Thomas Bayes, at 18th century cleric.

$$P\,[H|X] \quad = \quad \frac{P\,(H|X)\,P\,(H)}{P\,(X)}$$

In email classification, each email is represented by a vector of attribute value pairs, X depicting n values of n attributes. Given an unknown data, X , the classifier will predict that X belongs to the class C having the highest probability, conditioned on X.

$$H_{BAYES} = \mathrm{argmax}\ P(C|X)$$

It assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve'.

It was widely used by researchers in the field of information retrieval to classify document (e.g. web pages) into relevant categories. It also used in software to perform prediction, systems control and disease diagnosis. It assumes that variables are independent of each other given their classification information

2.5.2 Discussion

Since Naïve Bayesian algorithm is a sort of unsupervised learning, it requires the users to manually classify a large amount of emails to construct the training dataset at the initial stage. This is the main disadvantage of the prototype, as the user might feel tedious with the time-consuming and laborious process. This has led to research in another efficient way.

Therefore my thesis is an enhancement of PIGEON, in which unsupervised learning method [clustering] will be implemented to eliminate the manual classifying process. Two clustering algorithm that I research in will be discussed in the following section.

2.6 Email Classification Using Clustering Algorithm

Clustering techniques consider data as objects. The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity [Han & Kamber, 2002]. That is, clusters of objects are formed so that objects with a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other cluster. Similarity is commonly defined in terms of how " close" the objects are in space, based on a distance function. The "quality" of a cluster may be represented by its diameter, the maximum distance between any two objects in the cluster. Centroid distance is an alternative measure of cluster quality and is defined as the average distance of each cluster object form the cluster centroid.

Since clustering is a form of unsupervised learning, it does not rely on predefined classes and class-labeled training examples. Therefore, in the context of email classification, the laborious and time-consuming manual classification process can be eliminated.

In the following subsection, I will discuss two clustering algorithms that are potential to be used in email classification: Swarm Intelligence and K-Means algorithm.

2.6.1 Swarm Intelligence

Swarm Intelligence is a relatively new computational and behavioral metaphor for solving distributed problems. This is inspired from the collective emerged intelligence coming from the social insects like for instance ants and bees. The approach emphasizes distributedness, direct or indirect interactions among relatively simple agents, flexibility and robustness.

Two of the main principles in swarm intelligence are *self-organization* and *stigmergy*[Swarm, 2001]. *Self-organization* is the emergence of large structure based on simple mechanisms. Some of the simple control mechanisms are positive and negative feedback, which amplify and balance activity.

Stigmergy is a made up word and means stimulation by work. One of the main principles is that the environment is used as an external collective memory. The task carried out by the agent is regulated by the state of the environment.

One of the interesting and useful behaviour of ants was noted by Jean-Louis Deneubourg et al [Deneubourg, 1990], with the *Messor sancta* worker ants. This particular species of ants have been observed to carefully pile up their colony's deal to organize their nest. If the corpses were initially randomly distributed in space, the workers would cluster them into neat piles within a few hours. It is believed that the basic mechanism underlying this type of aggregation phenomenon is an attraction between dead items mediated by the worker ants. Figure 2.6 shows the experiment of the corpses clustering by the ants.

This behaviour of the ants suggests very interesting heuristics in data clustering. Jean-Louis Deneubourg has simulated this behaviour with a population of ant-like agents randomly moving on a discrete 2D board and pick or drop objects that scatter around the 2D board.

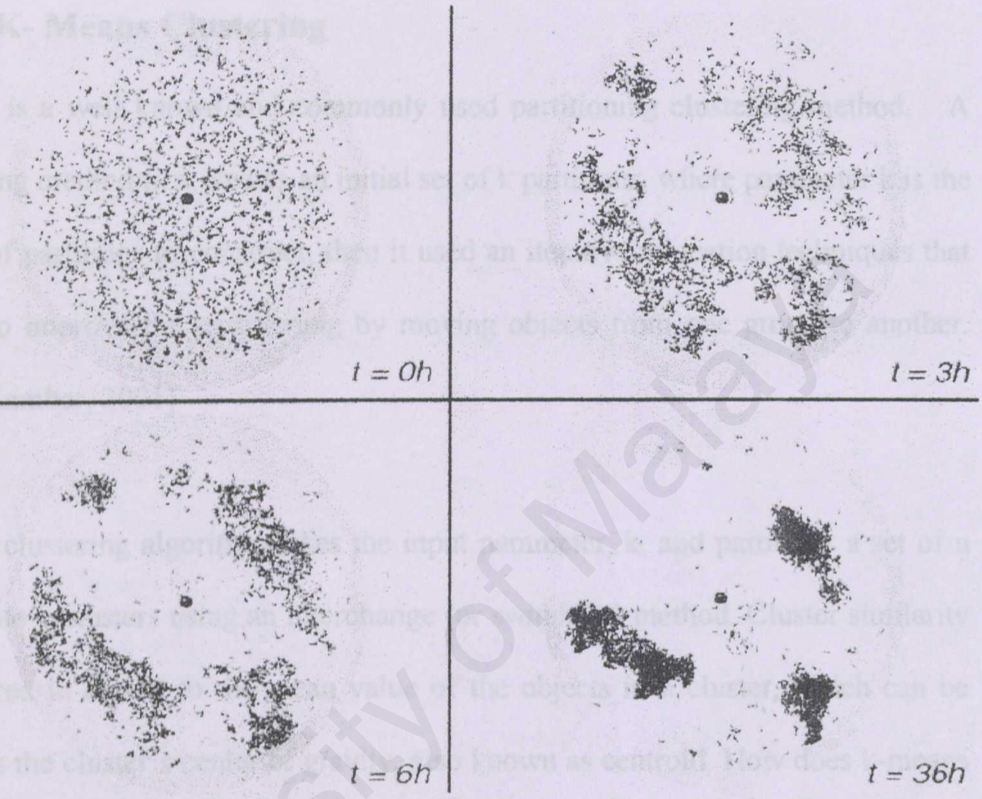


Figure 2.6 The Experiment of the Corpses Clustering by The Ants

[Ramos,2002]

Dropping or picking up an object depends on the similarity among the objects. The agents are only allowed to carry a single object and move only one step in any direction at a time. For each step, the agents will check with the its perceivable neighborhood (e.g.8 adjacent cells for a 2D grid) .The agents pick up an isolated item and drop the item where more similar items are present in its perceivable neighborhood.

The ant-like agents can be applied to clustering the email data. The general idea is that isolated items should be picked up and dropped at some other location where more items of that are present.

2.6.2 K- Means Clustering

K-means is a well known and commonly used partitioning clustering method. A partitioning method first creates an initial set of k partitions, where parameter k is the number of partitions to construct; then it used an iterative relocation techniques that attempt to improve the partitioning by moving objects from one group to another. [Han & Kamber, 2001]

K-means clustering algorithm takes the input parameter, k, and partitions a set of n objects into k clusters using an interchange (or switching) method. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's center of gravity, also known as centroid. How does k-means algorithm works?

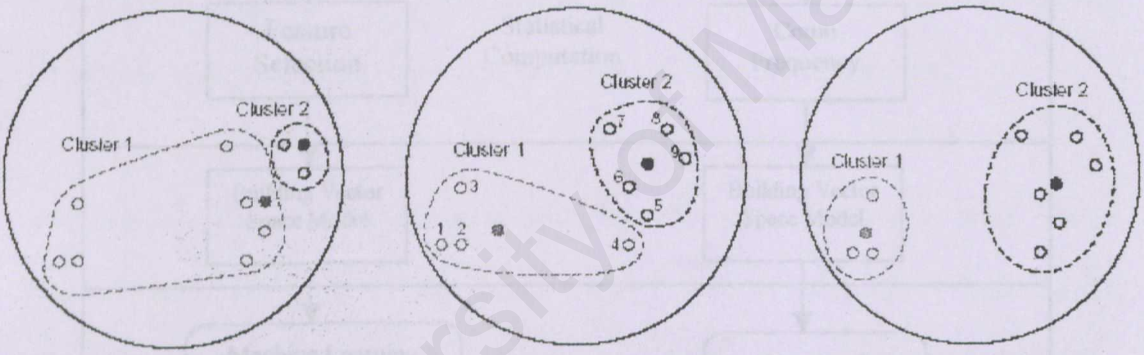
Generally, K-means involves two procedures [Faber 1994; K-Means, 1999]. The first assigns each object to a cluster and the second sets initial positions for the cluster centroids.

In the first procedure, the users are required to specify k, the number of cluster. *k* of the objects are selected to initially represent the centroid of a cluster. Each remaining object will be assigned to the cluster which has the highest similarity, or, in the other words, the cluster which its centroid is the nearest to the object.

In the second procedure, the new centroid of each cluster will be recalculated based on the objects in the cluster. Relatively to these new centroids, each object is visited to recalculate the distance to the updated clusters. If the closest cluster of the object is not the one it currently belongs, the object will switch to the new cluster domain which cluster center is the nearest.

The process iterates until no redistribution of the objects in any cluster occurs. The resulting clusters are returned by the clustering process.

The figure below illustrates the clustering of a set of object based on the k-means method



a. The initial state of a set of objects, which k = 2. The initial centroid of each clustering is denoted by the filled circle.

b. The iterative process of update the new centroid and redistribution of objects to new centroid correspondingly.

c. The iterative process terminated when no redistribution of the object, come out with two convergent cluster

Figure 2.7 Clustering of a Set of Object based on the K-Means Method

[Faber, 1994]

The method is relatively scalable and efficient in processing large data set. However the necessity for users to specify the k, the number of clusters, cause k-means is not suitable for discovering clusters with noncovex shapes or clusters of very different size.[Han & Kamber, 2001]

2.7 Classification Process

The figure below shows the framework of the Classification process:

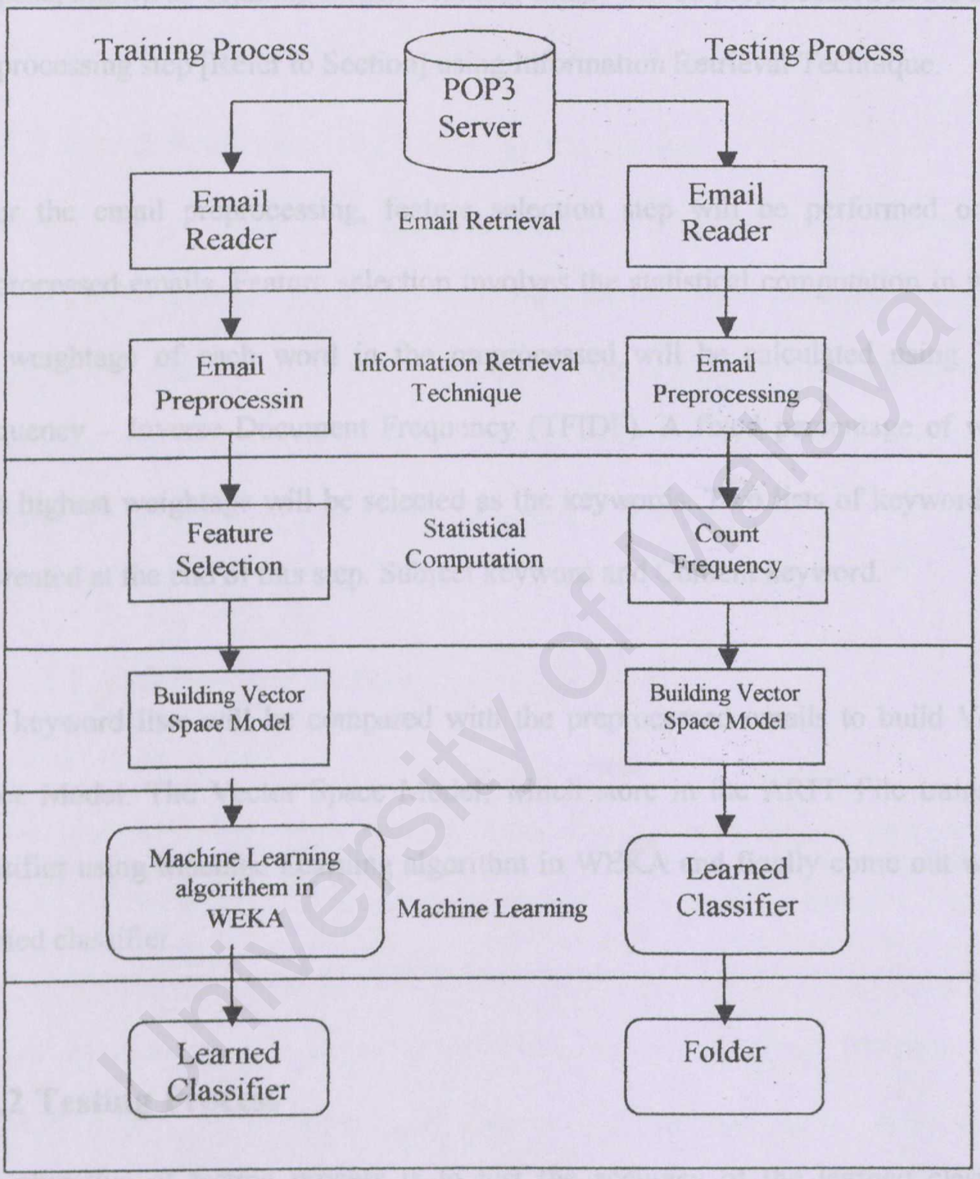


Figure 2.8 The framework of the Classification Process

Classification Process can be divided to Training Process and Testing Process.

The following subsection will give a brief description about these two processes:

2.7.1 Training Process

The main objective of training process is to generate a learned classifier.

A fixed amount of emails will be retrieved from the Pop3 Server for training purpose, known as Training Email. Training Email will be preprocessed in the Email Preprocessing step [Refer to Section] using Information Retrieval Technique.

After the email preprocessing, feature selection step will be performed on the preprocessed emails. Feature selection involves the statistical computation in which the weightage of each word in the preprocessed will be calculated using Term Frequency – Inverse Document Frequency (TFIDF). A fixed percentage of words with highest weightage will be selected as the keywords. Two lists of keyword will be created at the end of this step: Subject keyword and Content keyword.

The keyword lists will be compared with the preprocessed emails to build Vector Space Model. The Vector Space Model, which store in the ARFF File trains the classifier using Machine Learning algorithm in WEKA and finally come out with a learned classifier.

2.7.2 Testing Process

The objective of testing process is to test the accuracy of the learned classifier generated in the training process. In the testing process, another set of emails will be retrieved from the Pop3 server, known as Testing Email.

As performed in the training process, the Testing Email will be preprocessed in the Email Preprocessing step, using Information Retrieval Technique. A Vector Space

Model is built by comparing the keyword lists created in training process with the preprocessed email. An ARFF file will be created to store the Vector Space Model. The learned classifier analyzes the ARFF file, identifies the feature of each email and consequently assigns the emails into the corresponding folder.

2. 7. 3 Email Preprocessing and Representation

To classify the set of emails, we must first transform the emails into a representation suitable for machine learning algorithm. Emails are typically a sequence of character strings. Information Retrieval research suggests that an attribute value representation of the document is sufficient for many tasks. Details of the document preprocessing and representation are presented in the following subsections.

2.7.3.1 Email Representation

a. Vector Space Model

Vector Space Model is a representation of documents, where they are converted into vectors. In the context of email classification, each email is converted into an attribute-value pair vector [IR, 1997]. The attribute is the keyword that selected in feature selection part while the value is the number of times (frequency) the keyword occurs in the email for subject and message body. Besides, the corresponding email address for the sender and recipients will also be represented in an email vector.

b. Attribute-Relational File Format (ARFF)

An Attribute-Relational File Format (ARFF) is a common file format used for storing processed data as vectors [Witten & Frank, 1999]. It defines a data set in terms of relation (i.e. table) made up of data attributes (i.e. columns) and their values. This format provides the basic two-dimensional data set that machine-learning schemes typically require.

The figure below shows an example of ARFF file for an email data set

```
%email.arff-This is used for Email Classification
%Created on 12th August 2002

@relation email

@attribute sender {1.334.0f-
      C6zvgOKs98rR.1@techmail.techrepublic.com,…….zenath1@excite.com}
@attribute To {151@163.net, 2000@netsys.kaist.ac.kr,….zenath1@excite.com}
@attribute CC {@os.my, tac@mimos.my, ……lcy@hotmail.com }
@attribute action< frequency
@attribute advanc< frequency
.
.
@attribute x real
……
@data
wei.li@usa.net,Undisclosed.Recipients@ns.kyoshin-print.co.jp,?,0,0,…,0,1,0,?
tony@aknap.fsnet.co.uk,user@es.net,?,0,0,……………………..,0,0,0,0,0,?
…………
…………..
jan@VIRTUALUNLIMITED.COM,JMF-INTEREST@JAVA.SUN.COM,?,0,0, ..,,,0,0,0,0,0,0,?
```

Figure 2.9 An Example of ARFF file

ARFF files consist of two distinct sections. The first section is the Header information, which is followed the Data information. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types [ARFF, 2002].

i. The ARFF Header Section

The ARFF Header section of the file contains the relation declaration and attribute declarations.

- The @relation Declaration

  The relation declaration defines the name of the ARFF file.

- The @attribute Declarations

  Attribute declarations take the form of an ordered sequence of **@attribute** statements. Each attribute in the data set has its own **@attribute** statement, which uniquely defines the name of that attribute and it's data type. The order the attributes are declared indicates the column position in the data section of the file. For example, if an attribute is the third one declared then all that attributes values will be found in the third comma delimited column.

ii. The ARFF Data Section

The @data Declaration

The @data declaration is a single line denoting the start of the data segment in the file. The format is:

```
@data
jan@VIRTUALUNLIMITED.COM,JMF-INTEREST@JAVA.SUN.COM,?,0,0, 0,0,0,0,0,0,?
```

Each row represents an email vector on a single line, with carriage returns denoting the end of the email vector. Attribute values for each vector are delimited by commas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the n^th @attribute declaration is always the n^th field of the attribute).

2.7.3.2 Email Preprocessing

Email Preprocessing is the process of producing document representations. Conventionally, email preprocessing follows a standard procedure. It involve email data extraction where individual words from the message body and subjects will extracted. It consists of the three steps

a. Lexical Analysis
b. Stopwords Removal
c. Stemming

a. Lexical Analysis

Lexical Analysis is generally defined as the process of converting a stream of characters (the text of the documents) into a stream of words [Porter, 1980].

The main question of lexical analysis is what counts as a valid word or token. The choice should be made between selecting or discarding numbers, breaking hyphenated words like "one-on-one" into their constituents or not, preserving the case or not, etc. All the options are very domain-specific and depend on the purpose of lexical analysis. Below we give a list of properties of the lexical analyzer used in the thesis:

- *Letters [upper and lower case]*

  The lexical analyzer will convert all the letters into to lower case, for " Malaysia" becomes "malaysia".

- *Digits*

  All the digits like "1234"are discarded.

- *Hyphens*

This is a tradeoff between correcting inconsistent usage (like "state-of-the-art" and "state of the art") and loosing associations between connected terms by splitting them into often meaningless constituent parts. In this thesis, we choose to sacrifice the consistency and preserve valuable associations. Hyphenated words are kept as a single token.

- *Punctuation marks*

Punctuation marks are usually removed entirely as this does not affect the retrieval process and the risk of misinterpretation is very minimal. All punctuation and spacing are treated as token delimiters.

b. Stopword Removal

Stopwords are very commonly used English words – e.g. articles, pronouns, adjectives, adverbs and prepositions, which are known as common word. Common word normally has a high frequency across a collection. Many studies have shown that the most common 250 and 300 words in English may account for 50% or more of any given text. [Korfhage & Robert, 1997]

The common words have two impacts on Information Retrieval. First, in any measurements depending on word frequencies, the very high frequencies of these words tend to diminish the impact of frequency differences among the less common words. Second, as these words carry little meaning by themselves, they may result in a large amount of unproductive processing if left in the text.

For these reasons, it is common to define a stop list, or negative dictionary, consisting of the stopwords. When a text is initially processed, each word occurrence is checked against the stop list. If the word is found there, it is ignored or removed in any further processing.

c. Stemming

Stemming is the process of removing suffixes from words in a document or query in the formation of terms in the system's internal model [IR, 1997]. This is done to group words that have the same conceptual meaning. For example, the words such as WALKED, WALKER, and WALKING all resulting in the root word "WALK". Hence the user doesn't have to be so specific in a query. It is also essential to ensure there are no duplicate words that have the same meanings since this might disrupt the accuracy of classification. The Porter Stemmer is a well-known algorithm for this task.

2.7.4 Feature Selection

In the field of machine learning many have argued that maximum performance is often not achieved by using all available feature, but by using only a "good" subset of those. The problem of finding a "good" subset of features is called feature selection [Korfhage & Robert, 1997]. Applied to text classification this means that we want to find a subset of words, which helps to discriminate between classes. This can be achieved by using the weighting algorithm.

Weighting is the process of giving emphasis to the parameters for more important words [IR, 1997]. A popular weighting scheme is Term Frequency - Inverse Document Frequency (TFIDF).

The weightage [w] are calculated as a combination of statistics Term Frequency (TF) and Document Frequency (DF). The term frequency is the number of times word occurs in document d. The document frequency is the number of documents in which the term occurs at least once. The inverse document frequency (IDF) can be calculated from the DF as shown below

$$\mathrm{IDF} = \log\left(\frac{D}{DF}\right)$$

The weightage is the multiplication of Term Frequency and its Inverse Document Frequency as shown below:

$$w = TF \, x \, IDF$$

Those terms that occur frequently in a folder but rarely in the rest will have the higher weightage. The processed words will be ranked according to their weightage. A fixed percentage of those words with highest weightage will be selected as the keywords, which will become the attributes in Vector Space Model.

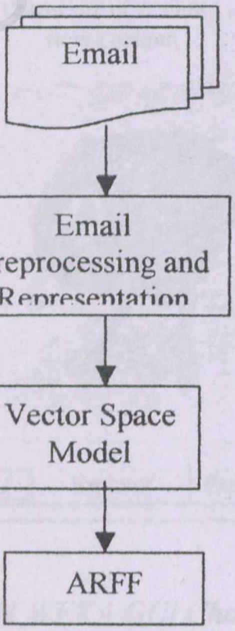
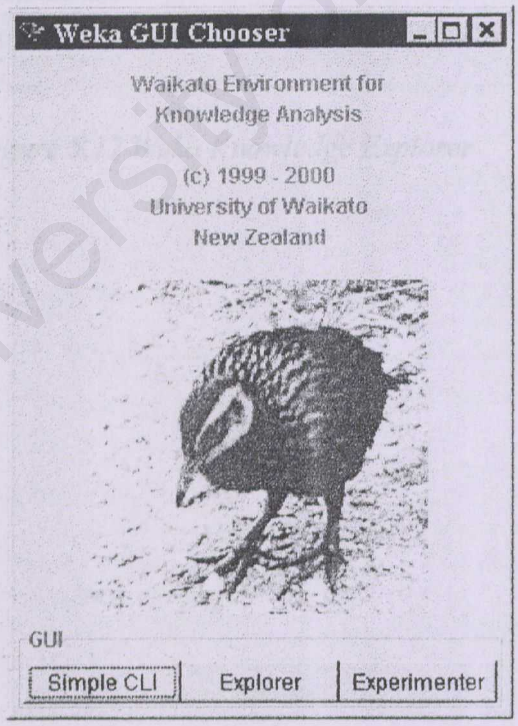


Figure 2.10: A graphical view of the email representation and preprocessing stage.

2.7.5 Waikato Environment for Knowledge Analysis (WEKA)

WEKA is a collection of Machine Learning algorithms (e.g. decision tree, Naïve Bayesian, K-Means, Support Vector Machine) for solving real-world data mining problems. It is developed in University of Waikato in New Zealand which is original applied in agricultural. In New Zealand, WEKA has already been used to help determine what information dairy farmers use in deciding which cows to keep in their herds. It has also been applied to mushroom classification and the analysis of milk production.[WEKA, 2002]

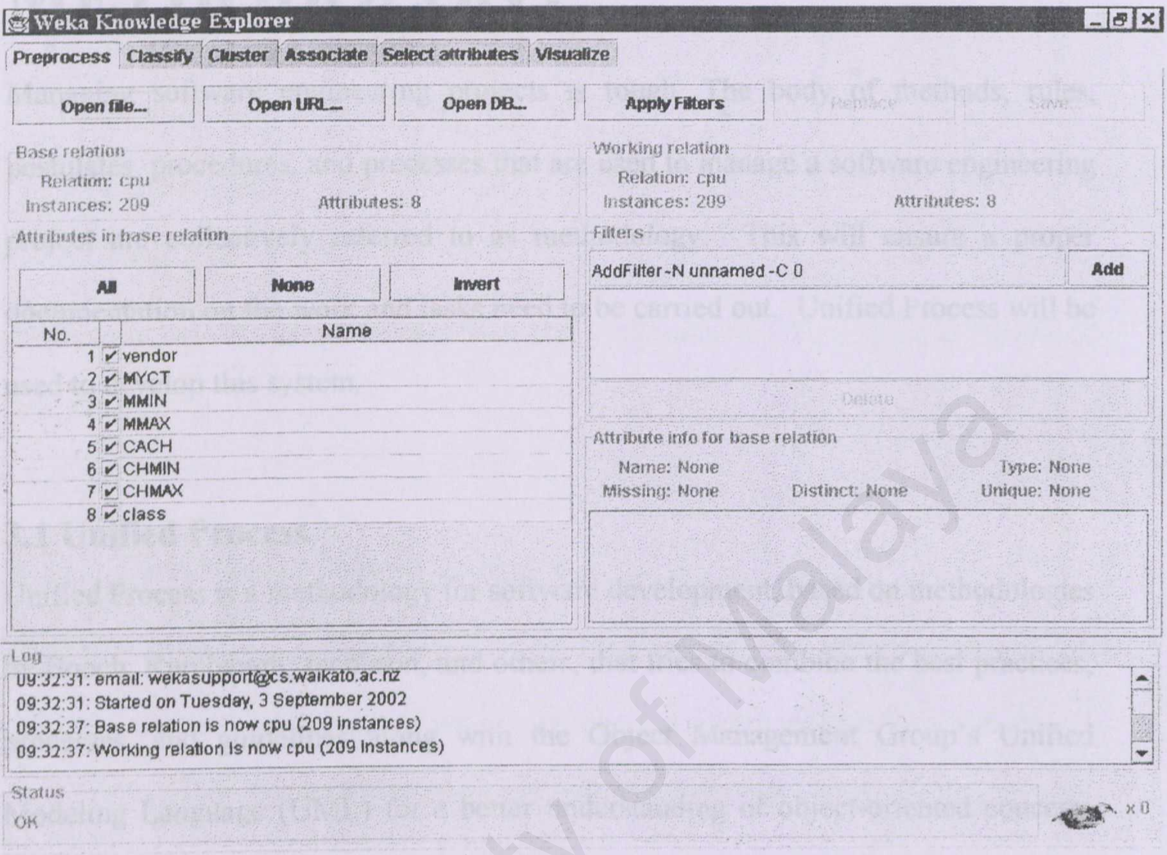
With it, the scientists are able to use Machine Learning to derive useful knowledge or information from databases that are far too large to be analyzed by hand. The software is written entirely in Java and includes an interface as shown as the figure below:



Waikato Environment for
Knowledge Analysis
(c) 1999 - 2000
University of Waikato
New Zealand

Figure 5.11 WEKA GUI Chooser

Since the system is open source, it can be integrated into the Automated Email Classification System easily.



Base relation
Relation: cpu
Instances: 209 Attributes: 8

Working relation
Relation: cpu
Instances: 209 Attributes: 8

Attributes in base relation

No.		Name
1	☑	vendor
2	☑	MYCT
3	☑	MMIN
4	☑	MMAX
5	☑	CACH
6	☑	CHMIN
7	☑	CHMAX
8	☑	class

Filters
AddFilter -N unnamed -C 0

Attribute info for base relation
Name: None Type: None
Missing: None Distinct: None Unique: None

Log
```
09:32:31: email: wekasupport@cs.waikato.ac.nz
09:32:31: Started on Tuesday, 3 September 2002
09:32:37: Base relation is now cpu (209 instances)
09:32:37: Working relation is now cpu (209 instances)
```

Status
OK

Figure 5.12 Weka Knowledge Explorer

CHAPTER 3

METHODOLOGY

Managing software engineering projects is tough. The body of methods, rules, postulates, procedures, and processes that are used to manage a software engineering project are collectively referred to as methodology. This will ensure a proper documentation on the work and tasks need to be carried out. Unified Process will be used to develop this system.

3.1 Unified Process

Unified Process is a methodology for software development, based on methodologies by Booch, Rumbaugh, Jacobson, and others, that tries to combine the best practices, processes, and guidelines along with the Object Management Group's Unified Modeling Language (UML) for a better understanding of object-oriented concepts and system development [Bahrami, 1999].

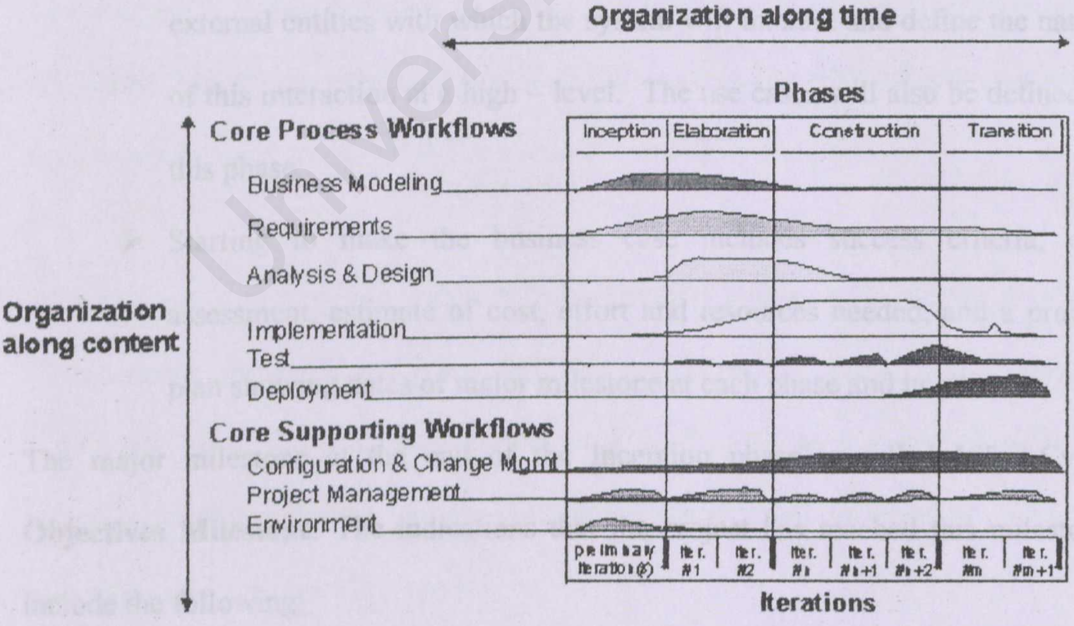


Figure 3.1 Unified Process- The Main Phases and Core Process Workflows

3.1.1 Four Main Phases in Unified Process

Refer to Figure 3.1 [Rational, 2002], there are four main phases in Unified Process:

1. Inception
2. Elaboration
3. Construction
4. Transition

Each phase will has a specific objective and a milestone will be defined at the end of each phase.

3.1.1.1 First Phase: Inception Phase

The main objective of this phase is to establish the case for the feasibility of the proposed system. The tasks that involved in Inception Phase include the following:

- ➢ Giving the problem definitions and identifying the objective of the system
- ➢ Defining the system scope. This can be accomplished by identifying all external entities with which the system will interact and define the nature of this interaction at a high – level. The use cases will also be defined in this phase.
- ➢ Starting to make the business case includes success criteria, risk assessment, estimate of cost, effort and resources needed, and a project plan showing dates of major milestone at each phase and iteration.

The major milestone at the end of the Inception phase is called **Life- Cycle Objectives Milestone**. The indications that the project has reached this milestone include the following:

- ➢ The major stakeholders concurrence on scope definition and cost/schedule estimates

- A set of critical high-level requirements addressed clearly by the use cases
- The business case the project is strong enough to justify a green light for continued development. This can be accomplished by ensuring the credibility of the cost/schedule estimates, priorities, risks, and development process

The project may be cancelled or considerably re-thought if it fails to pass this milestone.

3.1.1.2 Second Phase: Elaboration

The primary goal of the Elaboration phase is to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the highest risk elements of the project [Scott, 2001]. In other words, it is to establish the ability to build the new system given the constraints that the development project faces.

To accomplish these objectives, a thorough understanding of the whole system is necessary. The scope, major functionality and nonfunctional requirements such as performance requirements would be stated out clearly in this phase.

In this phase, an executable architecture prototype is built in one or more iterations and addresses the critical use cases identified in the inception phase, which typically expose the major technical risks of the project. Besides the business case for the project is finalized and the project plan that contains detail sufficient to guide the next phase of the project is prepared.

The major milestone associated with the Elaboration phase is called **Life-Cycle Architecture Milestone**. At this point, the detailed system objectives and scope, the choice of architecture, and the resolution of the major risks are examined. The indications that the project has reached this milestone include the following:

- ➢ Most of the functional requirements for the system have been captured
- ➢ The business case has been approved and an initial project plan that describes how the Construction phase will proceed is sufficiently detailed and accurate
- ➢ The vision and architecture of the product are stable
- ➢ The demonstration of the executable prototype show that the major risk elements have been addressed and credibly resolved.

3.1.1.3 Third Phase: Construction

The main purpose of the Construction phase is to build a system that is capable of operating successfully. In one sense, it is a manufacturing process where emphasis is placed on managing resources and controlling operations to optimize costs, schedules, and quality.

The tasks performed during Construction phase involve building the system iteratively and incrementally, making sure that the viability of the system is always evident in executable form.

At the end of this phase is the third milestone called **Initial Operational Capability Milestone**. At this point, the software, the sites, and the users should be ensured ready to go operational, without exposing the project to high risks. This release is often called a "beta" release.

The project has reached this milestone if the product release stable and mature enough to be deployed in the user community. Transition may have to be postponed by one release if the project fails to reach this milestone.

3.1.1.4 Fourth Phase: Transition

The primary goal of the Transition phase is to transition the software product to the user community. The transition phase is entered when a baseline is mature enough to be deployed in the end-user domain. This typically requires that some usable subset of the system has been completed to an acceptable level of quality and that user documentation is available so that the transition to the user will provide positive results for all parties. This includes:

- "Beta testing" to validate the new system against user expectations
- Parallel operation with a legacy system that it is replacing
- Conversion of operational databases
- Training of users and maintainers
- Roll-out the product to the marketing, distribution, and sales teams

The major milestone associated with the Transition phase is called **Product Release Milestone**. At this point, we have to make sure that all the objectives defined in Inception phase were met.

3.1.2 Core Workflows

Workflow is a sequence of activities that produces a result of observable value [Rational, 2001]. Within the Unified Process, there are nine core workflows cut across the set of four phases. The workflows are divided into six core process workflows and three support workflows [Refer to *Figure 3.1*].

Since Unified Process is a sort of iterative process, these workflows are revisited again and again throughout the lifecycle.

The following subsections describe the key features of five of the workflows, which is applied in the development of my project, in terms of the kinds of UML models associated with each workflow, and also the relationship of each workflow with each of the four phases.

3.1.2.1 Requirement Workflow

Description: Building the use case model, which captures the functional requirements of the system being modeled. The use case model also serves as the foundation for all other development works

The Requirement workflow cuts across the four phases of the Unified Process roughly as follows:

- Inception phase: A relatively bare – bones use case model developed in or order to capture critical high-level requirements
- Elaboration phase: The majority [around 80 percent] of the use case model gets built as functional requirements are addressed on a broad basis

- ➢ Construction phase: Completing the use case model since there might have some requirements weren't captured during Inception and Elaboration.
- ➢ Transition phase: Fine – tune the use case model

3.1.2.2 Analysis Workflow

This workflow involves building the analysis model based on the use case model. The analysis model is built by analyzing and structuring the functional requirements captured within the use case model. This model realizes the use cases that lend themselves better than the use cases to design and implementation work.

The Analysis workflow cuts across the four phases of the Unified Process roughly as follows:

- ➢ Inception phase: The analysis model is initiated as part of the effort to represent high-level requirements.
- ➢ Elaboration phase: A large majority of the analysis model gets built, as functional requirements are analyzed, refined and structured.
- ➢ Construction phase: Completing the analysis model and addressing unanalyzed requirements that arise after Elaboration
- ➢ Transition phase: Fine-tune the analysis model

3.1.2.3 Design Workflow

The main activities of the Design Workflow are focused on constructing the design model. The design model describes the physical realization of the use cases, which are defined in the use case model and described in the contents of the analysis model. The design model serves as an abstraction of the implementation model.

The Design workflow also focuses on the deployment model, which defines the physical organization of the system in terms of computation nodes.

The Design workflow cuts across the four phases of the Unified Process roughly as follows:

- Inception phase: The design model starts evolving as part of the effort to realize high-level requirements. The deployment model generally consists of a few broad sketches.
- Elaboration phase: The architecturally significant use cases are stated out within the design model. The deployment model starts taking shape if the system is going to be distributed to a noteworthy extent.
- Construction phase: The majority of the design model and the deployment model are built up, and the developers have to determine the compatible hardware node of software.
- Transition phase: Fine-tune the design model and deployment model

3.1.2.4 Implementation Workflow

Implementation workflow involves building the implementation model, which describes the implementation of the classes and objects in terms of components (source file, binaries, executable, etc). It also involves integration the results produced by individual implementers (or teams), into an executable system.

The Implementation workflow cut across the four phases of the Unified Process roughly as follows:

- ➢ Inception phase: The implementation model, if it exists, is a form of an executable prototype.
- ➢ Elaboration phase: The implementation model addresses the architecturally significant use cases.
- ➢ Construction phase: The large majority of the implementation model is built
- ➢ Transition phase: Fine-tune the implementation model

3.2.1.5 Test Workflow

The primary activity of the Test workflow is to build the test model. The objective of test model is to verify the interaction between objects, to verify the proper integration of all components of the software, to verify that all requirements have been correctly implemented and to ensure defects are addressed prior to the deployment of the software.

The test model consists test cases that are often derived directly from use cases. Testers perform black box testing using the original use case text, and white-box testing of the realization of those use cases, as specified within the analysis model

The Unified Process proposes an iterative approach, which means that you test throughout the project. The test workflow cuts across the four phases of the Unified Process as follows:

- ➢ Inception phase: Testing is done on the executable prototype is it exists
- ➢ Elaboration phase: the test model addresses the architecturally significantly use cases.

- Construction phase: Building the large majority of the test model and on performing suitable unit, integration, and system testing
- Transition phase: Fine-tune the test model, as ongoing testing helps to uncover flaws and defects.

3.2 Unified Modeling Language

The Unified Modeling Language (UML) is a general purpose modeling language for specifying, visualizing, constructing and documenting the artifacts of software systems (in particular object-oriented and component-based systems), as well as for business modeling and other non-software systems [Bahrami, 1999]. It includes many concepts and notations useful for the description and documentation of multiple models, and it enjoys a strong support from academic and industrial communities.

An important feature of UML, use cases are defined as sequences of actions a system perform that yield observable results of value to a particular user [actor]. Notations for scenarios and use cases, as well as design processes based on them, have become very popular over the last few years. These models describe partial representations of the system. UML allows the description of complex software-driven systems and models through the use of nine different diagram techniques. Each diagram provides a view of model from the aspect of a particular stakeholder, and each diagram must be semantically consistent with all the others. These diagrams are categorized into two sets. The first set, called behavioral diagrams, focuses mainly of functional and dynamic aspects of systems. It is comprised of five types of UML diagrams:

a) *User case Diagrams*: show actors and use cases together with their relationships, they describe system functionalities from the user's point of view

b) *Sequence Diagrams*: describe patterns of interaction among objects, arranged in a chronological order.

c) *Collaboration Diagrams*: show generic structure and interaction behavior of the system

d) *State Diagrams*: show the state space of a given context, the events that case the transitions of one state to another, and the actions that result

e) *Activity Diagrams*: capture the dynamic behavior of a system in terms of operations. They focus on flows driven by internal processing

The second set, called structural diagrams, relates more to components and static characteristics of systems. It includes these four types of UML diagrams.

a) *Class Diagrams*: capture vocabulary of a system. They show the entities in a system and their general relationships

b) *Object Diagrams*: snapshots of a running system. They show object instances (with data values) and their relationships at some point in time.

c) *Component Diagrams*: show the dependencies among the software components

d) *Deployment Diagrams*: show the configuration of run-time processing elements and the software components, processes and objects that live on them

3.3 Methods of collecting data

The methods used in collecting data required for developing electronic document delivery system are:

1. Searching the Internet
2. Interview
3. Document analysis

3.3.1 Searching the Internet

The Internet is a platform where a lot of information can be acquired. With development of search engines such as Googles, Copernic, Excite, Yahoo and Search, relevant information sites can be viewed with only a click away provided users key in the related keywords for search. The keywords used in searching relevant web pages are:

1. Email Classification
2. Email Filtering
3. Email Management
4. Machine Learning
5. Data Mining
6. K-Means
7. Swarm Intelligence
8. Information Retrieval Techniques

3.3.2 Interview

A few interview sessions have been conducted at the early stage of this project. The respondents of the interview include an electronic engineer from Agilent Technology and an editor from SinChew Jit Poh. Both of them are using their own company's email account, which has up to 100Mb of mailbox space and attach to the daily flow of email traffic.

The purpose if the interview is to get a better view about the problems and the current scenarios regarding the email management. From the interviews also, the requirements for a much more effective way to manage the email is being mentioned.

3.3.3 Document Analysis

Materials and writing about the research work on Automated Email Classification [or Text Classification] using intelligent method are found by searching the Internet. There are many reviews about the email classification and being used to as research materials. Some of the development tips and ideas are captured and will be used as guideline in developing the system.

CHAPTER 4
SYSTEM ANALYSIS

Requirement analysis is done during analyzing system needs to provide a guideline when developing s system. Requirement analysis activities include analyzing and determining functional requirements and non-functional requirements [Sommerville, 1996]

4.1 Functional Requirements

Functional Requirements are functions or features, which are expected by the user, and stated by them to be incorporated into the system [Sommerville, 1996]. The system is considered incomplete if any of the necessary functions is not included.

a. Account Management

Authenticator

An authenticator is vital to the system in order to protect the user's privacy. Users are required to enter their user name and password of their POP3 account to access the system.

b. Folder Management

Define and Rename folder

The users are required to define the number of folders before the training process start. In the training process the training emails will be clustered into a few groups corresponding to number of folders defined and store into the folders. User can rename the folder based on the content of email in each folder.

Add and Remove Folder

The users are allowed to add or remove folder.

Display Folder

Display the list of folders and the number of email in each folder in a tree structure. By clicking on the folder icon in the tree, the list of Sender and Subject of each email in that folder will be listed out in a table. The user can view a message by highlighting that particular row.

c. Email Management

Check Mail

A Graphic User Interface (GUI) is provided in this module to display the email message to the users. The users can select the message that they want to view by simply click on the header of the message.

Send Mail

Users can use the GUI to compose, reply or forward the email message.

Attachment

The system allows the users to send and receive attachment. The system can display the attachment file in various format include plain Text, Rich Text Format (RTF), Hypertext Markup Language (HTML) file, GIF file, JPEG file, Bitmap file and ZIP File.

Saving Message

The user can save the email message including the attachment.

Delete Mail

Users can delete the unwanted email or junk email. The deleted emails will be automatically moved to Trash folder.

d. Email Classification

Automatically Classify Email

The system can automatically classify the emails into the folders based on the textual content using the personalized classifier generated in training process.

Retrain email Classifier

As the content of emails might be changed over time, the user can choose to retrain the classifier so that the classifier is more consistent with the current email content

e. Additional Functions

Address Book

Address book is provided to allow the users to store the email addresses

4.2 Non – Functional Requirements

Non – functional specifications are the constraints under which a system must operate and the standards which must be met by the delivered system [Sommerwille, 1996]. The Automatic Email Classification System must ensure certain qualities such as accuracy, user – friendliness, functionality, reliability, security.

a. Reliability

Reliability is the extent to which a program can be expected to perform its intended function with required precision .It is closely related to the accuracy of classification. The overall performance of the system is determined by the classification accuracy. It means that the higher percentage of the emails that is being classified correctly, the greater performance of the system. . This quality is essential as it indicates how far the users will be confident in the implementation of the email system to handle daily email traffic.

b. User – Friendliness

User interfaces design creates an effective communication medium between a human and a computer. Therefore, it is very important to make sure that the interface fulfill user – friendliness so that it would not cause trouble to users.

c. Functionality

The functionalities stressed here are the retrieving capability, which is very important in any email client that deals with email retrieval from the email server.

d. Security

The proposed system also has security measures to minimize the risk of personal email accessed by invalid user.

4.3 Use-Case Driven Object-Oriented Analysis

The object-oriented analysis [OOA] phase of the unified approach uses actors and use cases to describe the system from the users' perspective [Bahrami, 1997]

The actors are external factors that interact with the system. The use cases are scenarios that describe how actors use the system. The definition of use case by Jacobson et al.: " *A Use Case is a sequence of transactions in a system whose task is to yield results of measurable value to an individual actor of the system"*. The use cases identified here will be involved through the development process. This is called use-case driven.

4.3.1 Actors and Use Cases

The relationships between the actor and the use cases or among the use cases are modeled in a use diagram. Diagram 4.1 shows the Use Case Diagram of Automatic Email Classification System.

Six actors are identified in this system: User, Pop3 Server, SMTP server, Email Reader Module, Classification Module and Text Analyzer. Each actor plays different roles and interacts with the use cases in various forms.

The following subsection will give a detail description for each actor and use case.

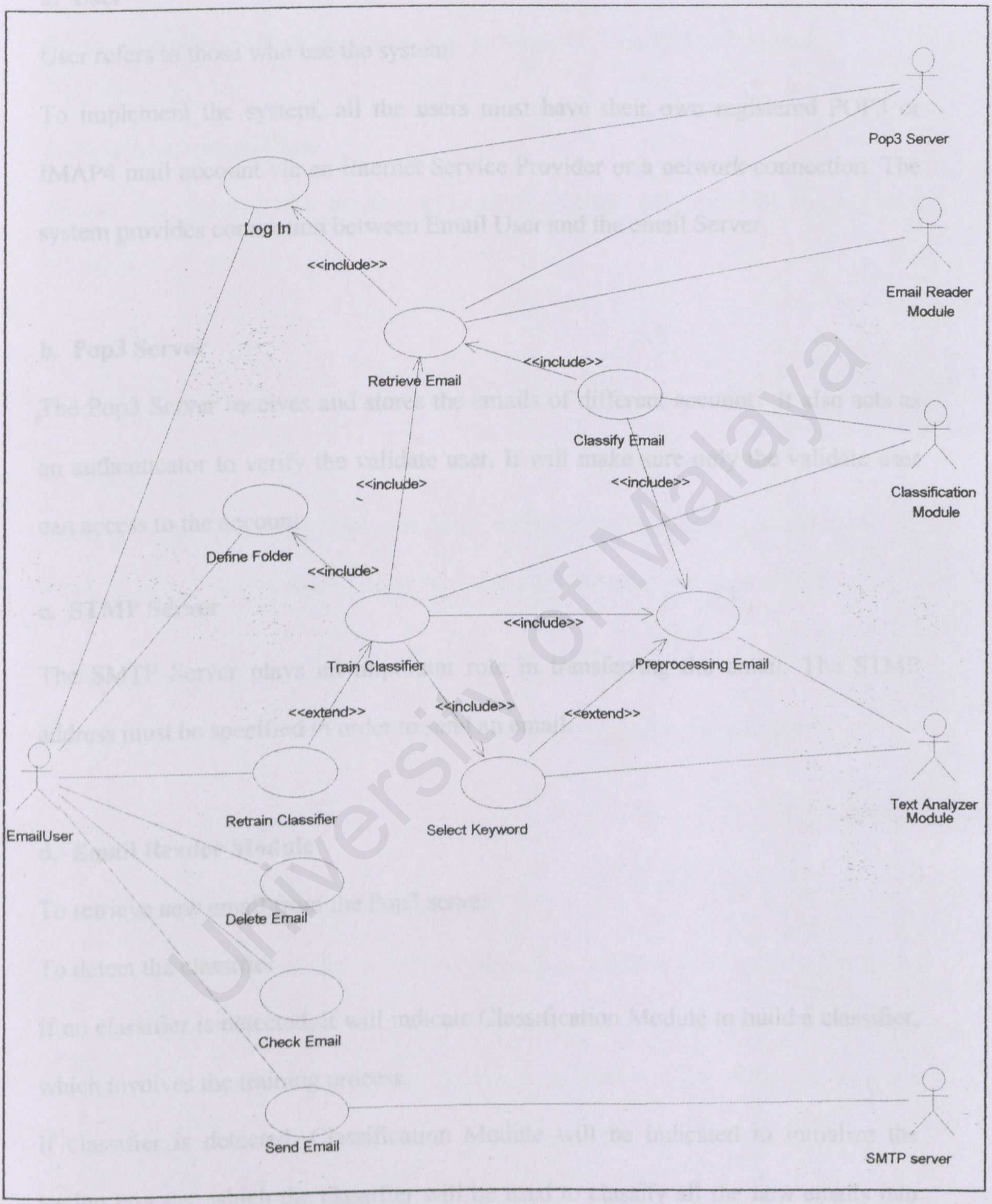


Diagram 4.1 Use Case Diagram

4.3.1.1 Actor

a. User

User refers to those who use the system.

To implement the system, all the users must have their own registered POP3 or IMAP4 mail account via an Internet Service Provider or a network connection. The system provides connection between Email User and the email Server.

b. Pop3 Server

The Pop3 Server receives and stores the emails of different account. It also acts as an authenticator to verify the validate user. It will make sure only the validate user can access to the account.

c. STMP Server

The SMTP Server plays an important role in transferring the email. The STMP address must be specified in order to send an email.

d. Email Reader Module

To retrieve new email from the Pop3 server

To detect the classifier

If no classifier is detected, it will indicate Classification Module to build a classifier, which involves the training process.

If classifier is detected, Classification Module will be indicated to initialize the testing process, which the classifier will be used to classify all the new emails into folders.

e. Classification Module

Receive message from Email Reader Module to invoke the classification process (Training process or Testing process) as indicated by Email Reader Module.

f. Text Analyzer Module

To start the processes that involve text analyzing: email preprocessing and keyword selection.

4.3.1.2 Use Case

a. Log In

Users are required to enter their user name and password in order to access to the POP3 Server. The system will provide access after the user is being verified.

b. Retrieve Email

System will automatically retrieve email from the server after the user log in.

c. Define Folder

The user is required to specify the number of folder.

In the clustering process, the number of clusters is required to be specified to initial the clustering process. The clusters are corresponding to the email folders in email classification. After the user defines the number of folders, each folder will be given a default name. The user can rename the folders after the classifier is generated, based on the content of the email in each folder.

d. Train Classifier

The system would able to build a classifier. This use case involves the training process, which produces an email classifier. This use case includes the "Select Keyword" use case, "Retrieve Email" use case, " Define Folder" use case and "Preprocessing Email" use case.

e. Select Keyword

The system must able to generate a set of keywords and store in a keyword list. The keyword list is used to build the Vector Space Model.

f. Classify Email

This use case refers to the Testing process. If a classifier is detected, the classifier will classify all the incoming email into the folders.

This use case includes "Retrieve Email" use case and "Preprocessing Email" use case.

g. Retrain Classifier

As the content of the emails might be changed over time, the user is allowed to retrain the classifier so that the classifier is suitable to classify the current emails. Since the use case extends Train Classifier use case. The training process will be invoked as in Train Classifier use case.

h. Email Preprocessing

All the incoming new emails must be preprocessed and convert to vectors, which is typically required by the machine-learning algorithm. Email Preprocessing is required in both Testing and Training Email

i. Delete Email

The user is allowed to delete the unwanted email(s).

j. Send Mail

The user can use the system to send a plain text email or attachment to multiple users (To, Cc) using SMTP.

4.3.2 Object Behavior Analysis:

Decomposing a Use-Case Scenario with a Collaboration Diagram

The UML specification recommends that at least one scenario be prepared for each significantly different use-case instance. Each scenario shows a different sequence of interaction between actors and the system, with all decision definite. In essence this process helps us to understand the behavior of the system's objects.

A collaboration diagram is a diagram that focuses on the organization of the objects that participate in a given set of messages. It shows the sequence and interaction of a given use case or scenario. In the process of creating the collaboration diagrams, we may find that objects may need to be added to satisfy the particular sequence of events for the given use case. Therefore, the process of creating the collaboration diagrams can assist us in identifying classes or objects of the system.

The following diagrams show the Collaboration Diagram for each use cases in the Use Case Diagram [Refer to *Diagram 4.1*]

4.3.2.1 Login Use Case

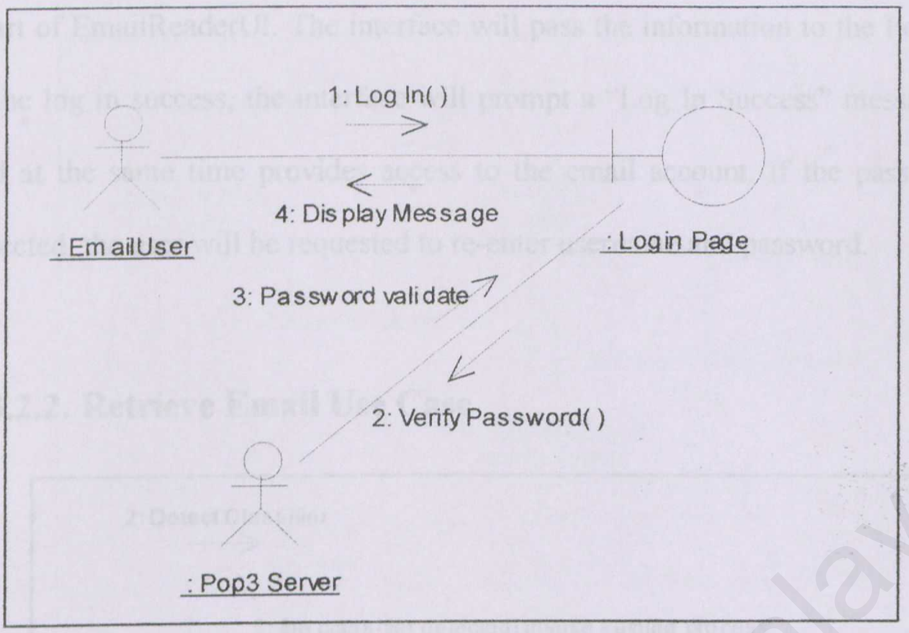


Diagram 4.2(a) Password Verified

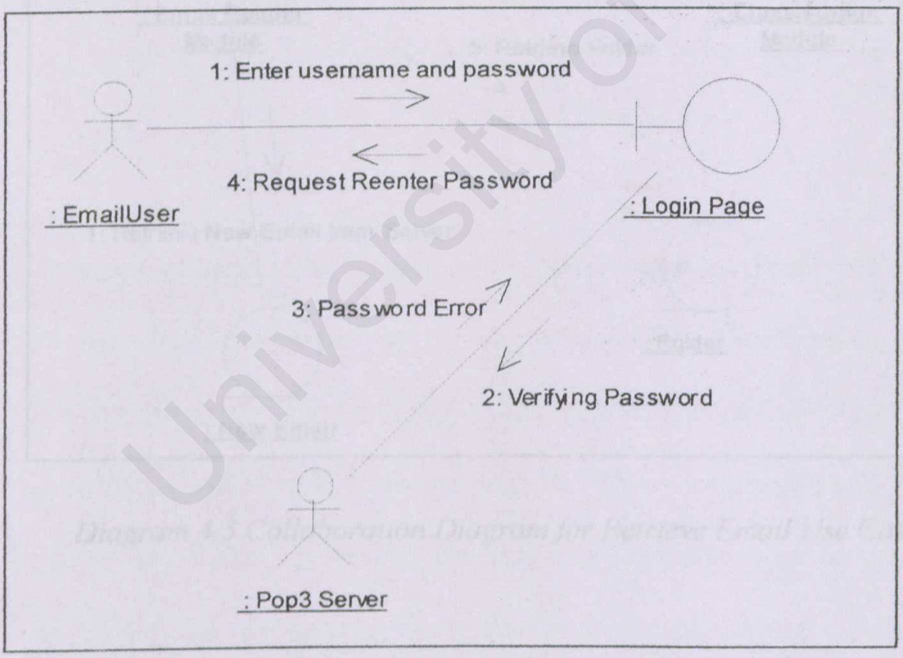


Diagram 4.2(b) Invalid Password

Diagram 4.2 Collaboration Diagram for Log In Use Case

In the log in process, the user enters the username and password at the log in page, apart of EmailReaderUI. The interface will pass the information to the Pop3 Server. If the log in success, the interface will prompt a "Log In Success" message to user and at the same time provides access to the email account. If the password error detected, the user will be requested to re-enter username and password.

4.3.2.2. Retrieve Email Use Case

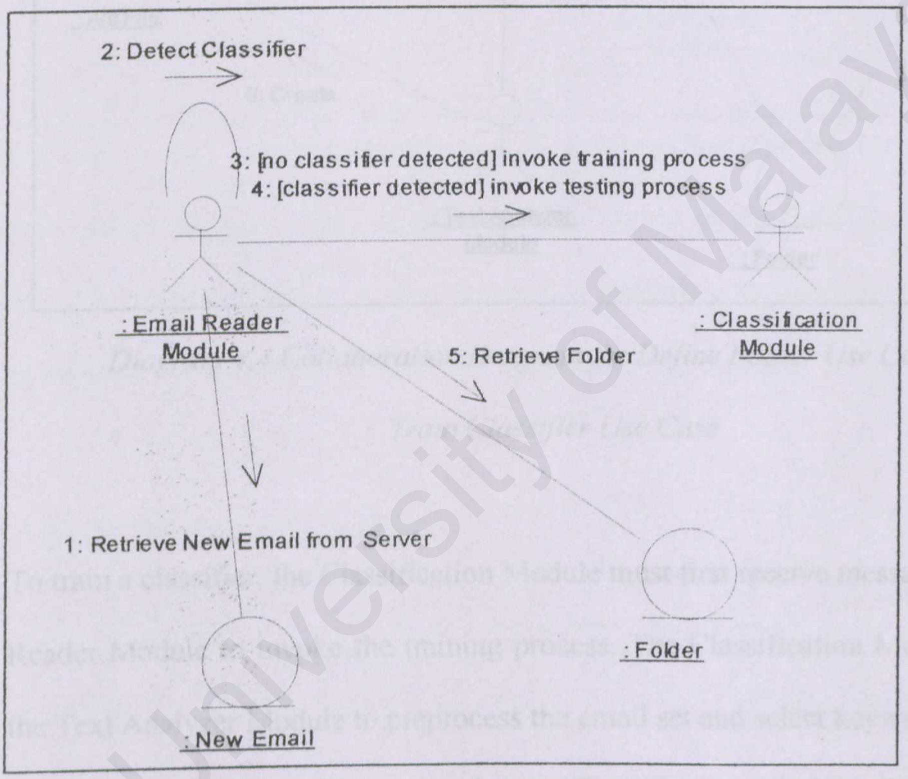

2: Detect Classifier
3: [no classifier detected] invoke training process
4: [classifier detected] invoke testing process
: Email Reader Module
: Classification Module
5: Retrieve Folder
1: Retrieve New Email from Server
: Folder
: New Email

Diagram 4.3 Collaboration Diagram for Retrieve Email Use Case

After the user is verified, the Email Reader Module will retrieve the new emails from the server. Consequently, Email Reader Module will try to detect the classifier from the user profile. If no classifier is detected, it will send a message to Classification Module to invoke the Training Process. Otherwise, Testing Process will be initialized.

4.3.2.3 Define Folder Use Case and Train Classifier Use Case

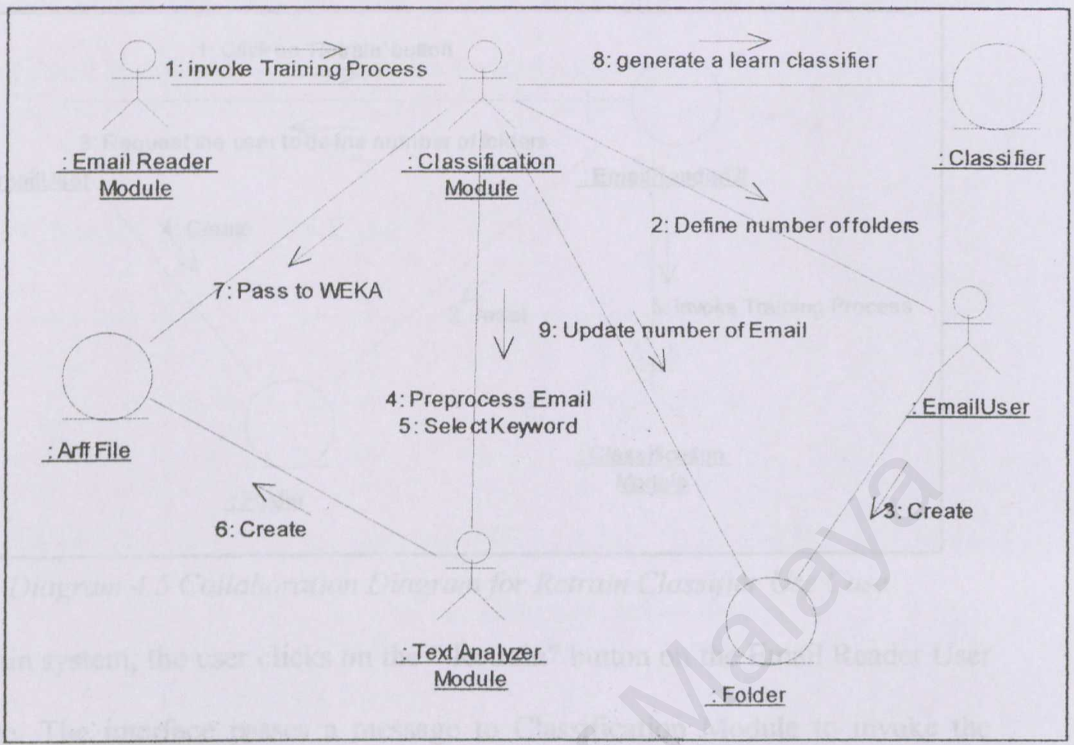


Diagram 4.4 Collaboration Diagram for Define Folder Use Case and Train Classifier Use Case

To train a classifier, the Classification Module must first receive message from Email Reader Module to invoke the training process. The Classification Module indicates the Text Analyzer Module to preprocess the email set and select keywords in order to generate an ARFF file. Consequently, the Classification Module directs the ARFF file to the WEKA. By using Machine Learning algorithm in WEKA, the data in ARFF file will be clustering into a number of groups, corresponding to the number of folders defined by user. A classifier will be trained using the clustered data and finally come out with a learned classifier. The emails will be stored into the folders as clustered before. The number of emails in each folder will be updated.

4.3.2.4 Retrain Classifier Use Case

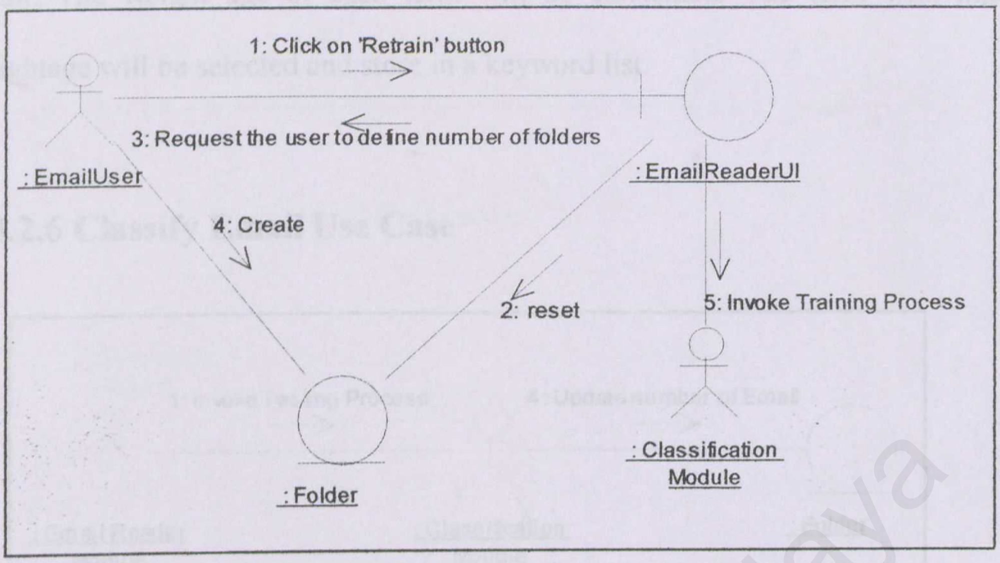

1: Click on 'Retrain' button
3: Request the user to define number of folders
: EmailUser
: EmailReaderUI
4: Create
2: reset
5: Invoke Training Process
: Classification Module
: Folder

Diagram 4.5 Collaboration Diagram for Retrain Classifier Use Case

To retrain system, the user clicks on the " Retrain" button on the Email Reader User Interface. The interface passes a message to Classification Module to invoke the training process. All the folders that created previously will be reset.

4.3.2.5 Select Keyword Use Case

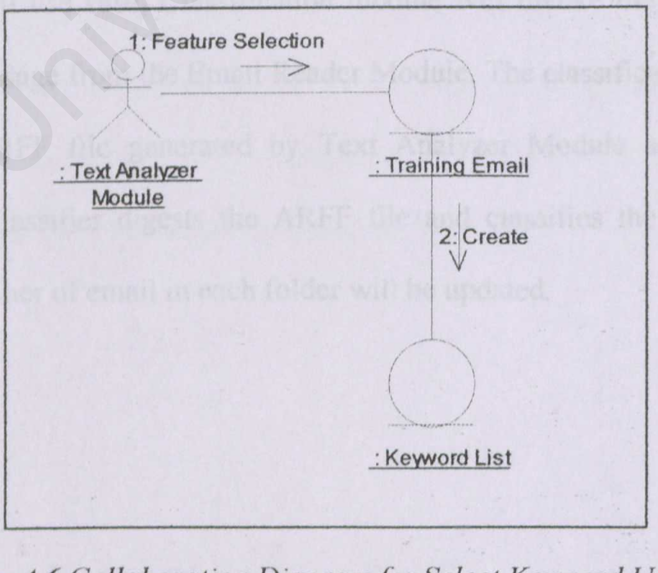


Diagram 4.6 Collaboration Diagram for Select Keyword Use Case

To select keyword, feature selection will be performed on the preprocessed training email. The weight age of each term will be calculated. The term with highest weightage will be selected and store in a keyword list.

4.3.2.6 Classify Email Use Case

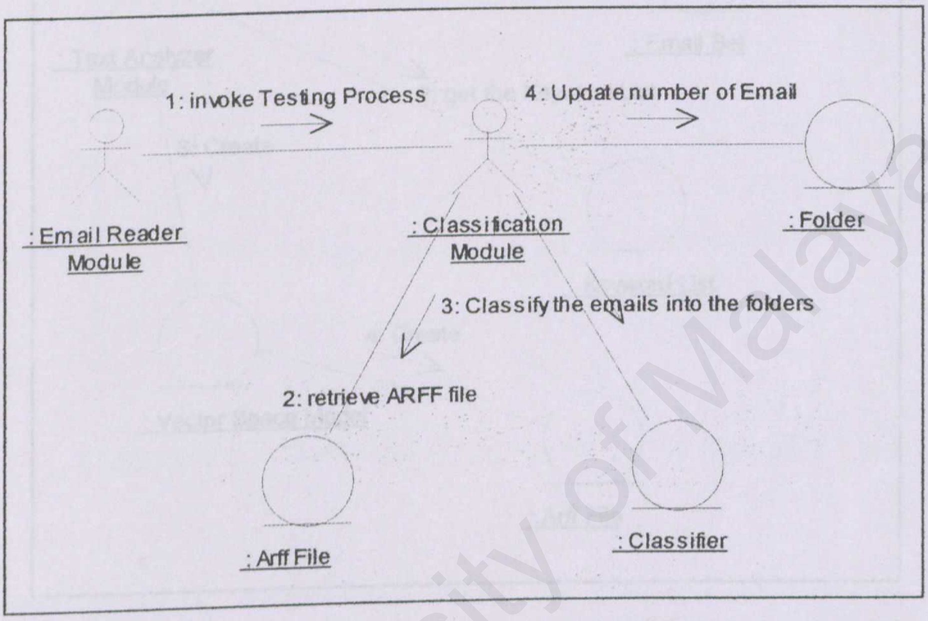


Diagram 4.7 Collaboration Diagram for Classify Email Use Case

In Classify Email use case, Classification module will invoke the Testing Process after receive message from the Email Reader Module. The classification module will retrieved the ARFF file generated by Text Analyzer Module and direct it the classifier. The classifier digests the ARFF file and classifies the emails into the folders. The number of email in each folder will be updated.

4.3.2.7 Preprocessing Email Use Case

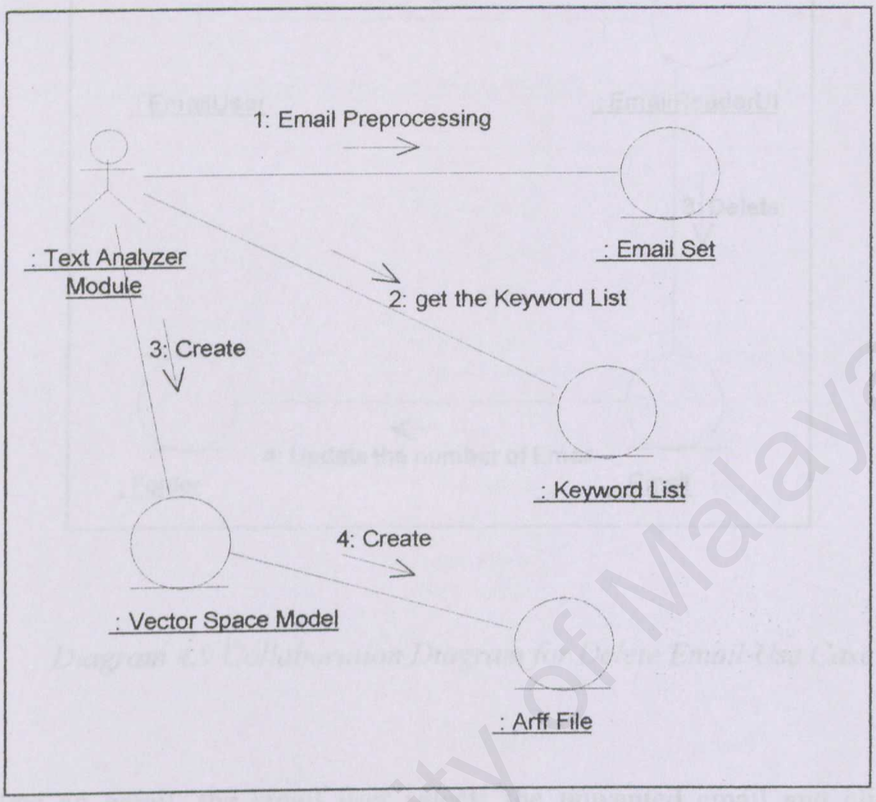


Diagram 4.8 Collaboration Diagram for Preprocessing Email Use Case

Text Analyzer Module will start preprocessing the emails after received message from the Classification Module. Email Preprocessing will be performed on the email set. The preprocessed email will be compared with the keyword list to create the Vector Space Model and consequently create an ARFF file.

4.3.2.8 Delete Email Use Case

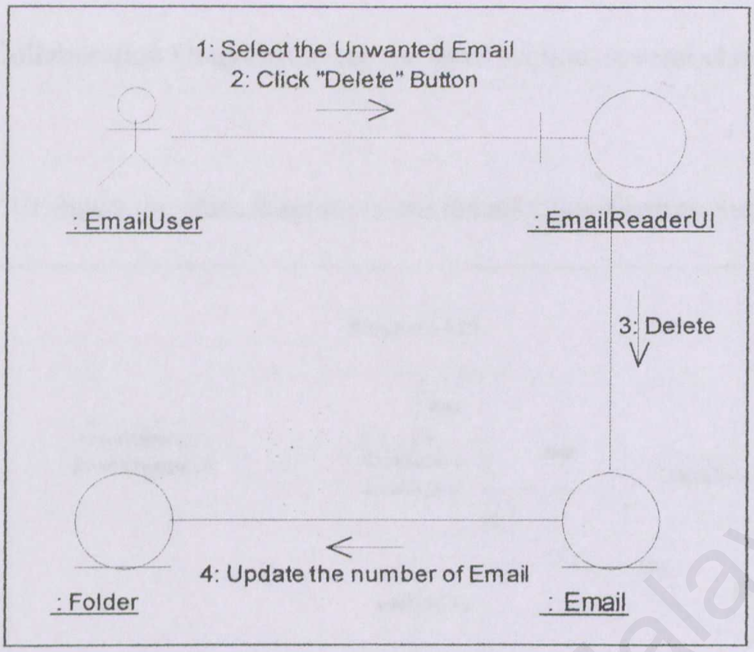


Diagram 4.9 Collaboration Diagram for Delete Email Use Case

To delete an email, the email user selects the unwanted email and click on the 'Delete' Button on EmailReaderUI. The EmailReaderUI send message to Email and delete the selected. The number of emails in the corresponding folder will be updated.

4.3.3 Identifying Class Relationships

From the Collaboration Diagrams in the previous section, several classes have been identified.

Diagram 4.10 shows the class diagram of the Email Classification System

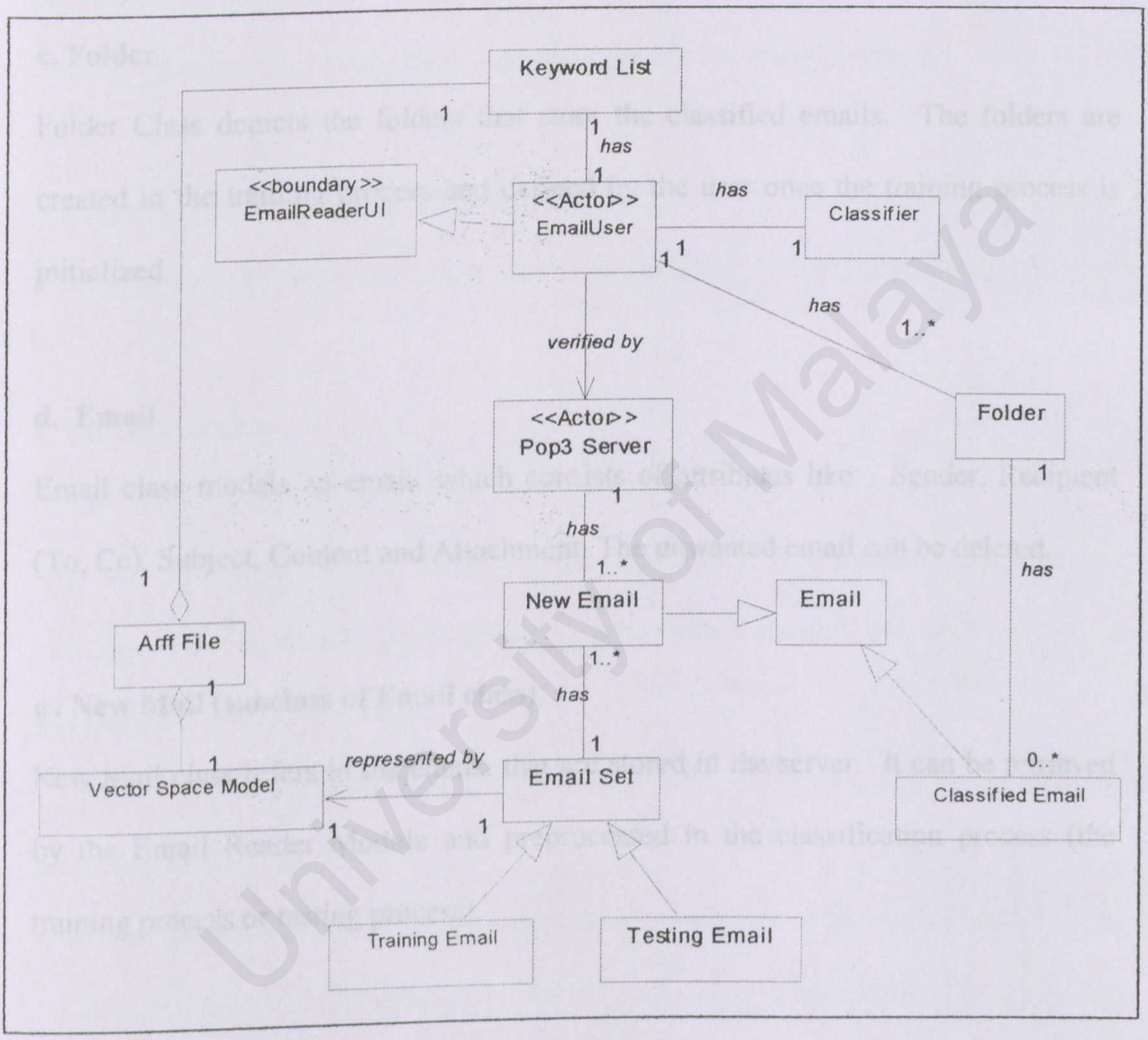


Diagram 4.10 Class Diagram

A brief description about each class will be given in the following:

a. Pop3 Server

Pop3 Server stores the emails of the users. Allow the users to access their account by entering password and username.

b. Classifier

Classifier depicts the email classifier that automatically classifies the emails into folders. The classifier is created in the training process by analyzing the ARFF file. In the testing process, the classifier will be used to classify the testing emails.

c. Folder

Folder Class depicts the folders that store the classified emails. The folders are created in the training process and defined by the user once the training process is initialized.

d. Email

Email class models an email, which consists of attributes like Sender, Recipient (To, Cc), Subject, Content and Attachment. The unwanted email can be deleted.

e . New Mail (subclass of Email class)

New Mail class refers to the emails that are stored in the server. It can be retrieved by the Email Reader Module and preprocessed in the classification process (the training process or testing process).

f. Email Set

Email Set refers to a set of new emails, which will be preprocessed in the classification process.

There are two types of Email Set: Training Email and Testing Email.

g. Training Email (subclass of Email Set class)

Training Email refers to the email set that used in training process. Feature selection will be performed on the Training Email to create the keyword list.

h. Testing Email (subclass of Email Set class)

Testing Email refers to the emails that processed in the testing process.

i. Classified Email (subclass of Email class)

Classified Email models the emails that are being classified and stored in the folder.

j. EmaiReaderUI

EmailReaderUI depicts the GUI of the system.

The interface allows the user to login to the Pop3 server, view the email message, compose and send emails. It also provides the buttons that allow the user to retrain the classifier and define folders.

k. Arff File

The class defines the Arff File that stores the Vector Space Model in the @data section and stores the keywords as attributes in the @attribute section [Refer to Section 2.6.3.1(b)].

The WEKA program will analyze the Arff File in the training process and consequently generate a classifier.

l. Keyword List

The Keyword List stores the keywords, which are selected in the training process using feature selection.

In the classification process, the keyword list will be retrieved by the Text Analyzer Module to create the Vector Space Model.

m. Vector Space Model

In the classification process, each email will be converted to a vector. Vector Space Model class depicts the class which is created by combining each email vector in a set of emails.

4.4 System Requirements

System requirements here are divided into two categories, namely the development environment and the runtime [user] environment.

4.4.1 Development Environment

1. Hardware Requirements

The recommended hardware requirements for development environment of the Automating Email Classification System are listed as follow:

Table 4.1 Hardware Requirements for Development Environment

Hardware Requirements
➢ IBM compatible PC with a Pentium 4 processor or higher
➢ 192 MB RAM or higher (256 recommended)
➢ 3.5 GB of hard disk space or higher
➢ Maxtor 6.4 GB hard disk
➢ Standard floppy disk drive
➢ Keyboard and mouse as input device

2. Software Requirements

The recommended software requirements for development environment of the Automating Email Classification System are listed as follow:

Table 4.2 Software Requirements for development Requirements

Software Requirements
➢ Microsoft 2000 Professional
➢ Java Second Edition Enterprise Platform
➢ JavaMail Package
➢ Setup Runtime Files

4.4.2 Runtime Environment

1. Hardware Requirements

The recommended hardware requirements for runtime environment of Automatic Classification System are listed as follow:

Table 4.3.Hardware Requirements for Runtime Environment

Hardware Requirements
➢ IBM compatible PC with a Pentium 300MHZ processor or higher
➢ 64 MB RAM or higher
➢ l20 MB of hard disk space or higher
➢ A SIS 6326 PCI graphics daughter card
➢ Keyboard and mouse as input device

2. Software Requirements

The recommended software requirements for runtime environment of Automatic Email Classification System listed as follow:

Table 4.4 Software Requirements for Runtime Environment

Software Requirements
➢ Windows 98 or Windows 2000 and higher

4.5 Machine Learning Algorithm Implemented

K- Means clustering algorithm will be applied in developing the Automatic Email Classification System.

[Please refer to Section 2.6.2 for further information about K-Means Clustering]

4.6 Programming Language Implemented

JAVA

Java is a programming language expressly designed for use in the distributed environment of the Internet [Whatis.com Inc, 1996]. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces a completely object-oriented view of programming. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build small application modules or applets for use as part of a Web Page.

The major characteristics of Java are:

- The programs created are portable in a network. The program is compiled into Java byte-code that can be run on any server or client in a network that has a Java Virtual Machine. The Java Virtual Machine interprets the byte-code into code that will run on the real computer hardware. This means that individual computer platform differences such as instruction lengths can be recognized and accommodate locally without requiring different versions of your program.
- The code is "robust", means that, unlike programs written in C++ and perhaps some other languages, the Java objects have no outside references that may case them to "crash".
- Java was designed to be secure, meaning that its code contains no pointers outside itself that could lead to damage to the operating system. The Java interpreter at each operating system makes a number of checks on each object to ensure integrity.

- Java is object-oriented, which means that, among other characteristics, similar objects can take advantage of being part of the same class and inherit common code. Objects are thought of as " nouns" that a user might relate to rather than the traditional procedural "verbs".
- In addition to being executed at the client rather than the server, a Java applet has other characteristics designed to make it run fast.

Java was introduced by Sun Microsystems in 1995 and instantly created a new sense of the interactive possibilities of the Web. Since then, almost all major operating system developers [IBM, Microsoft, and others] have added Java compilers as part of their operating system products.

CHAPTER 5

SYSTEM DESIGN

5.1 The Object Oriented Design Method

During the design phase the classes identified in object-oriented analysis must be revisited with a shift in focus to their implementation. Attributes and methods must be added for implementation purposes and user interfaces [Bahrami, 1999].

5.1.1 Defining Attributes and Operations

Attributes are things an object or class must store such as color, cost, and manufacturer. Identifying attributes of a system 's classes starts with understanding the system's responsibilities.

Operations (methods or behavior) refer to queries about attributes of the objects. In other words, methods are responsible for managing the value of attributes.

Both attributes and operations can be defined by analyzing the use cases and developing other UML diagrams such as sequence diagram, activity diagram and state diagram [Bahrami, 1999].

5.1.2 Sequence Diagram

The UML sequence diagram is a diagram that focuses on the time ordering of the messages that go back and forth between objects. The following shows the sequence diagram for each use case, which is transformed from the Collaboration Diagram developed in Analysis Phase [Refer to Section 4.3.2], but with further details of the messages exchanged among the objects.

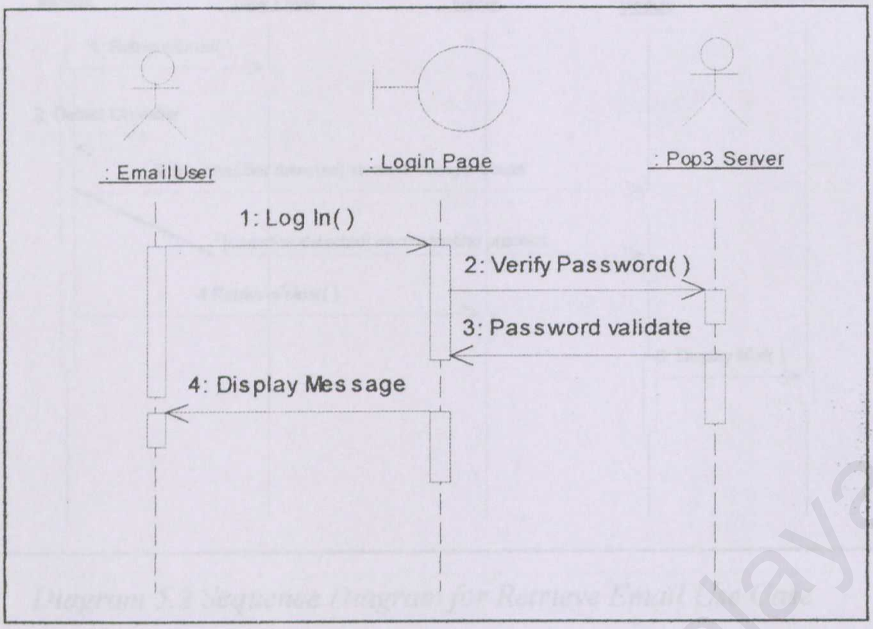


Diagram 5.1 (a) Login Success

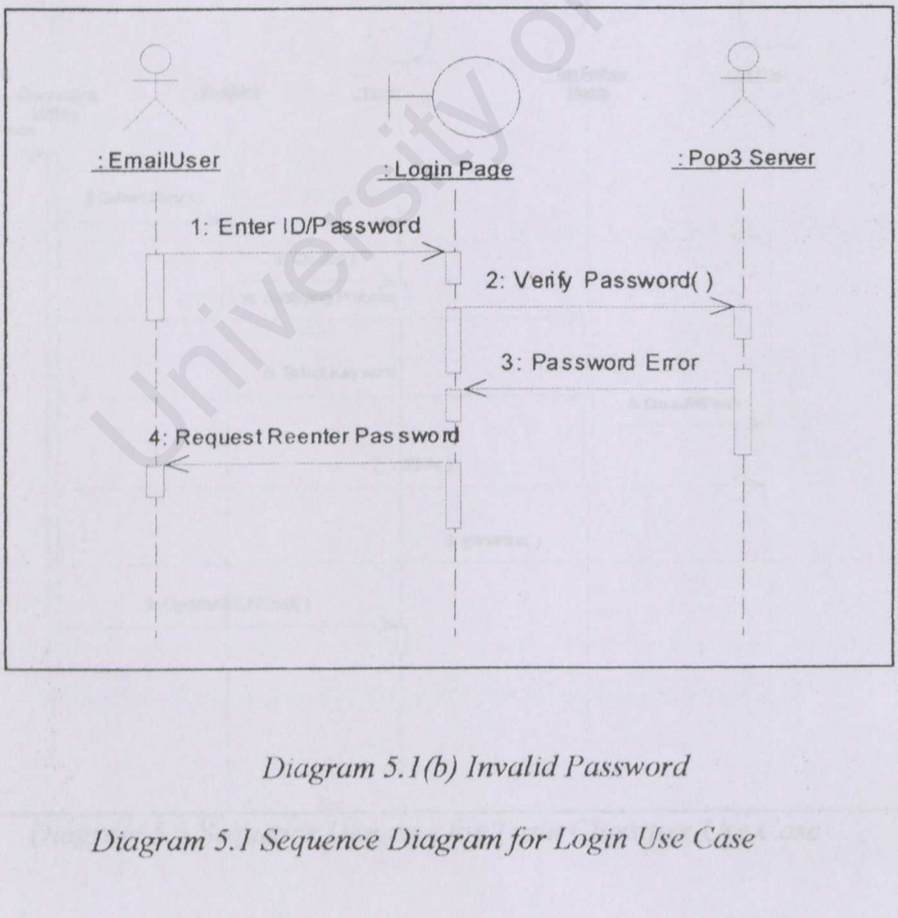


Diagram 5.1(b) Invalid Password

Diagram 5.1 Sequence Diagram for Login Use Case

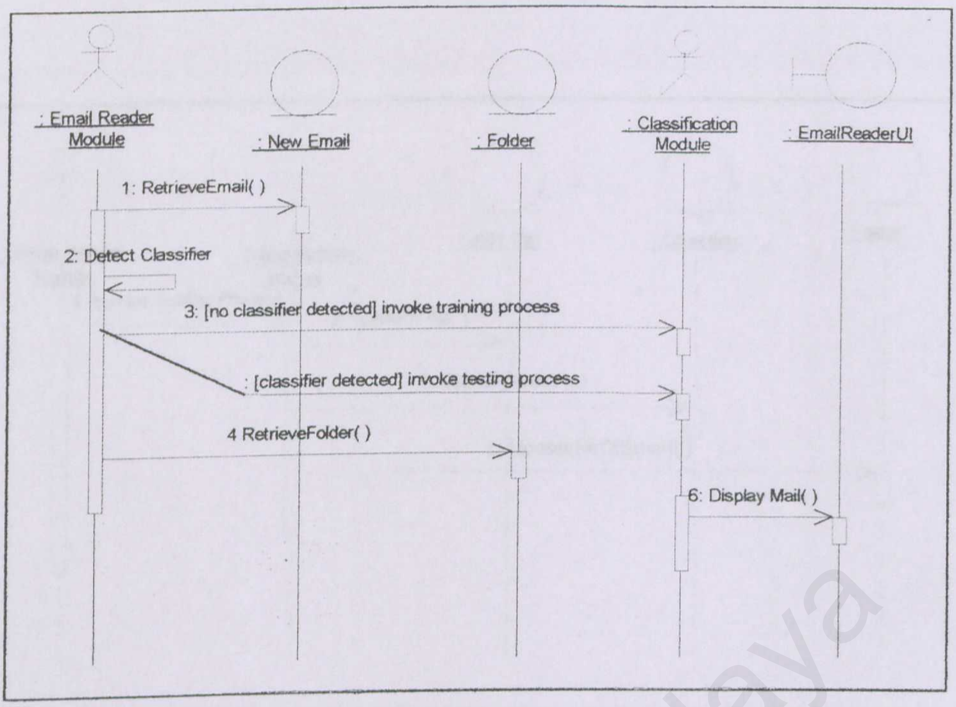


Diagram 5.2 Sequence Diagram for Retrieve Email Use Case

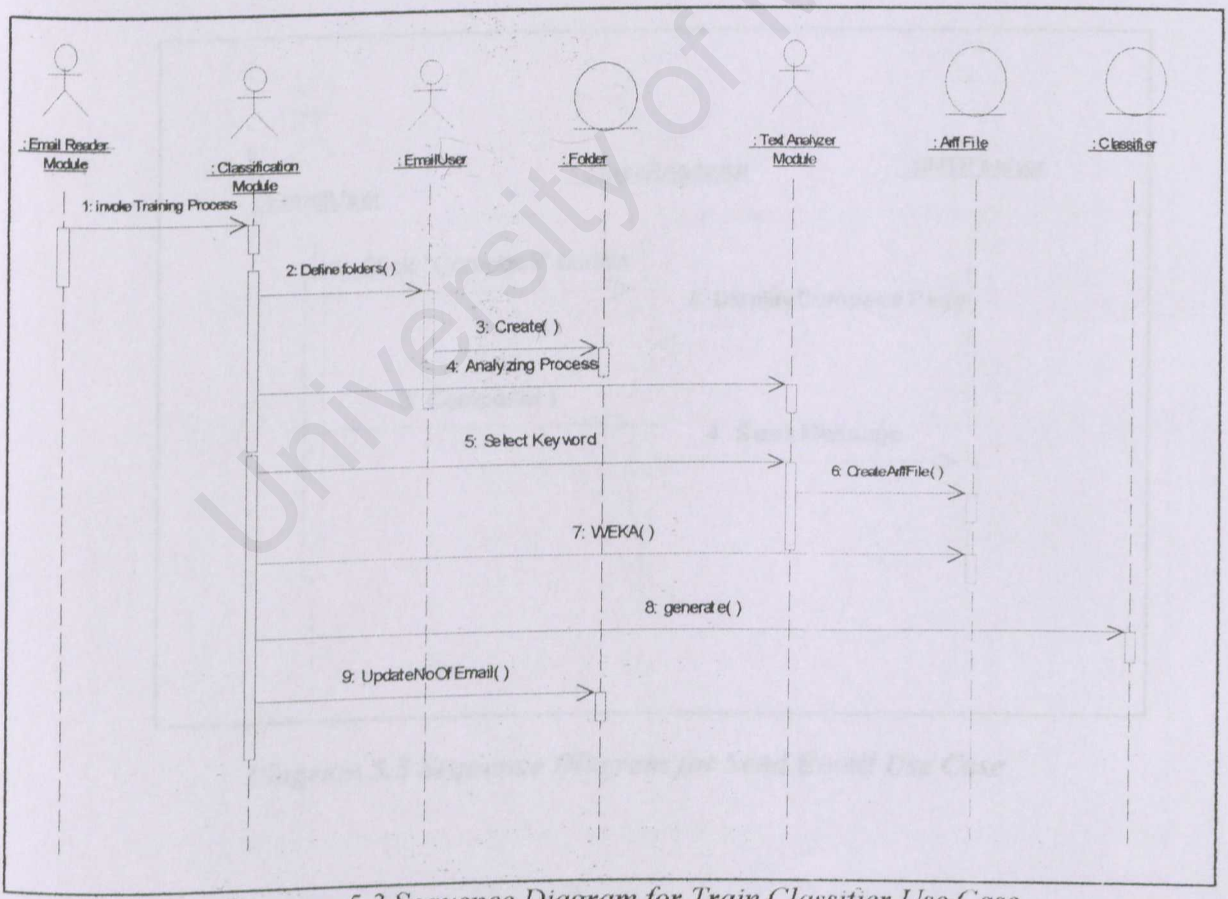


Diagram 5.3 Sequence Diagram for Train Classifier Use Case

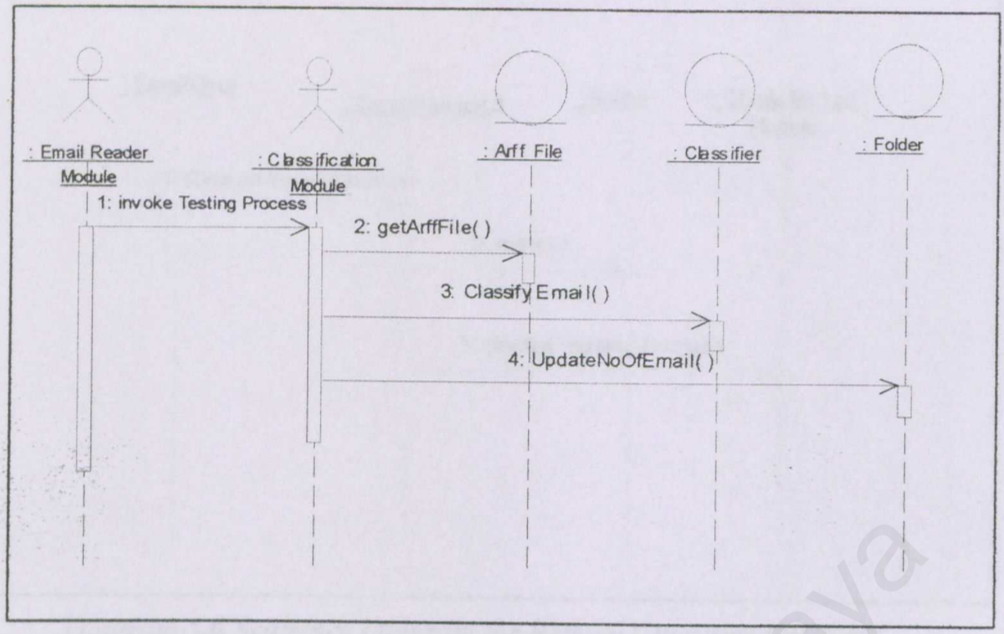


Diagram 5.4 Sequence Diagram for Classify Email Use Case

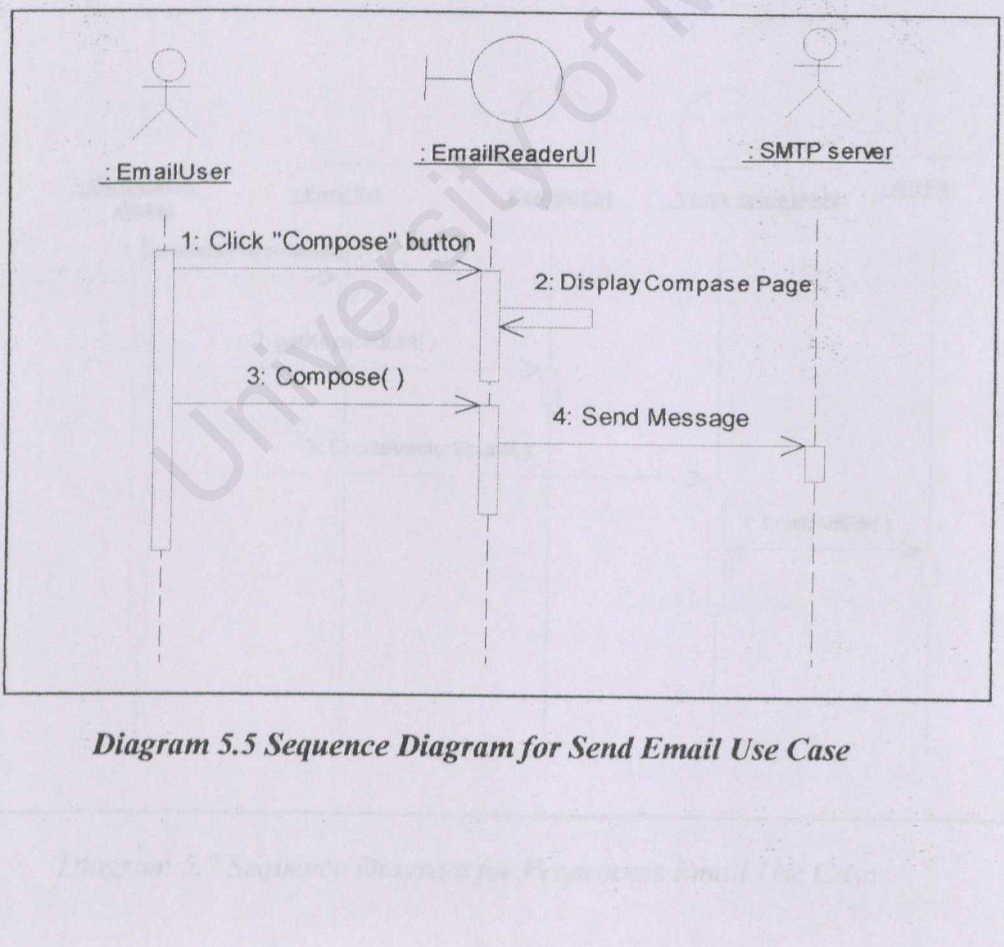


Diagram 5.5 Sequence Diagram for Send Email Use Case

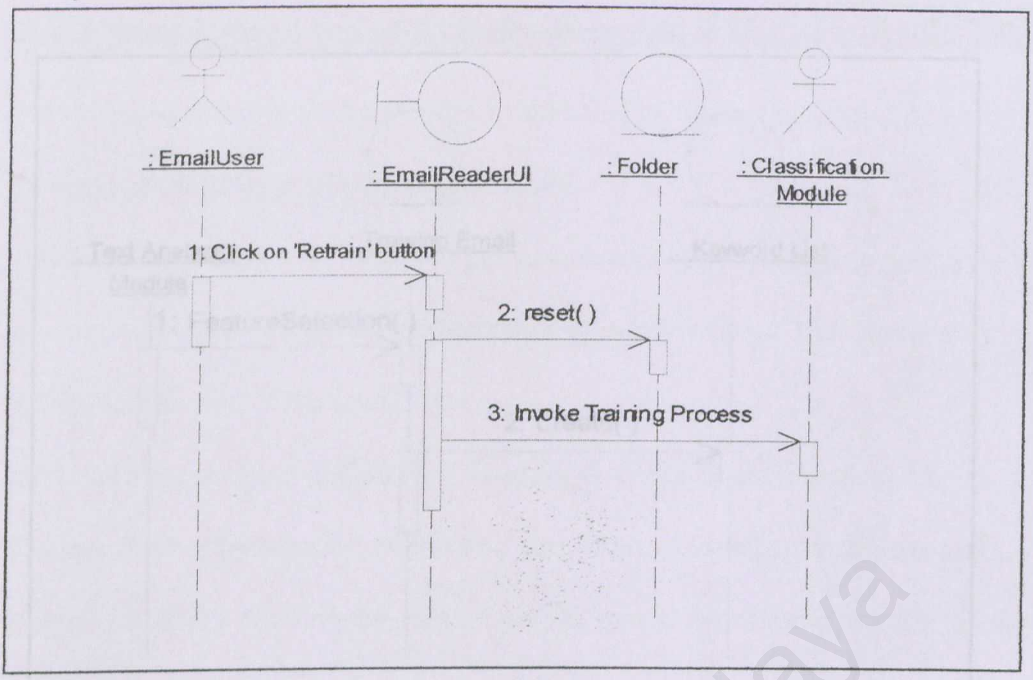


Diagram 5.6 Sequence Diagram for Retrain Classifier Use Case

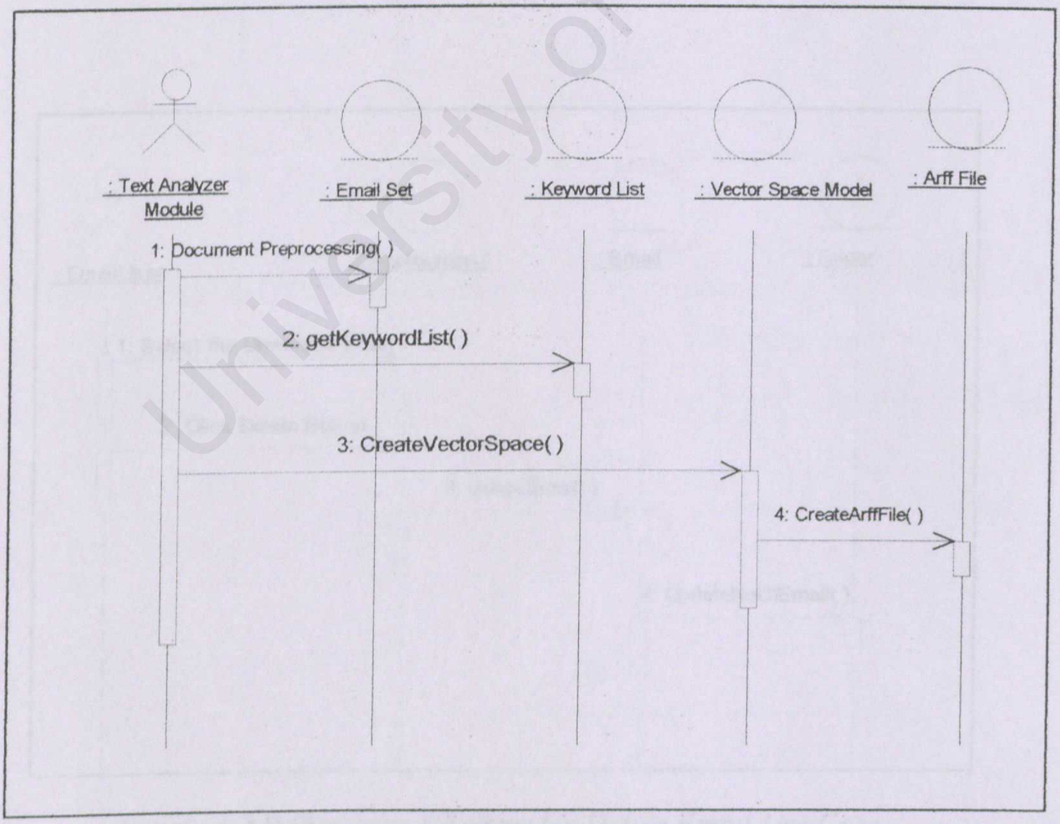


Diagram 5.7 Sequence Diagram for Preprocess Email Use Case

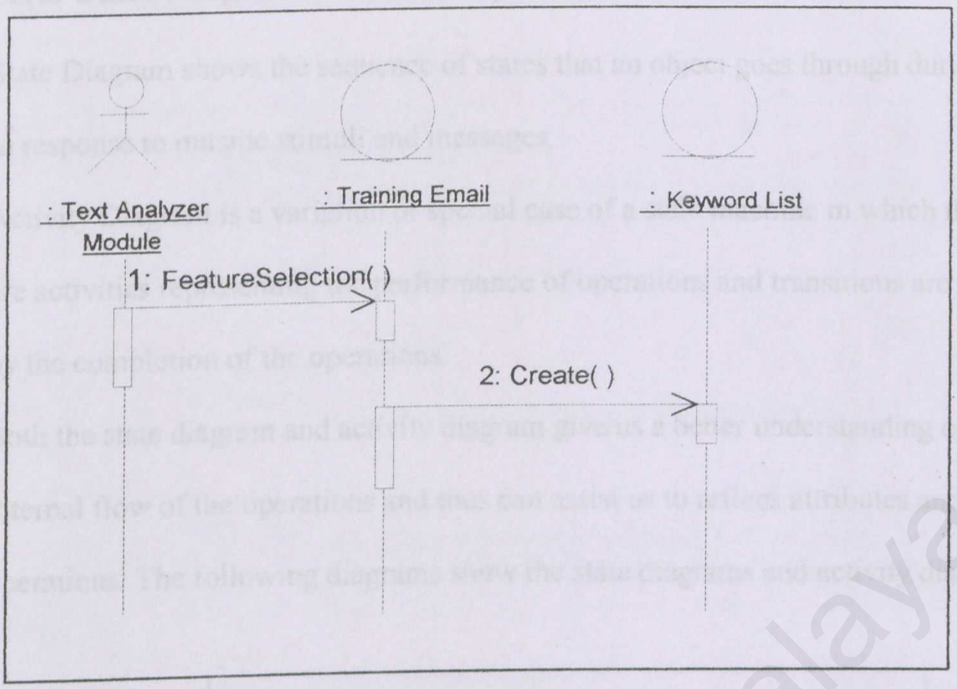


Diagram 5.8 Sequence Diagram for Select Keyword Use Case

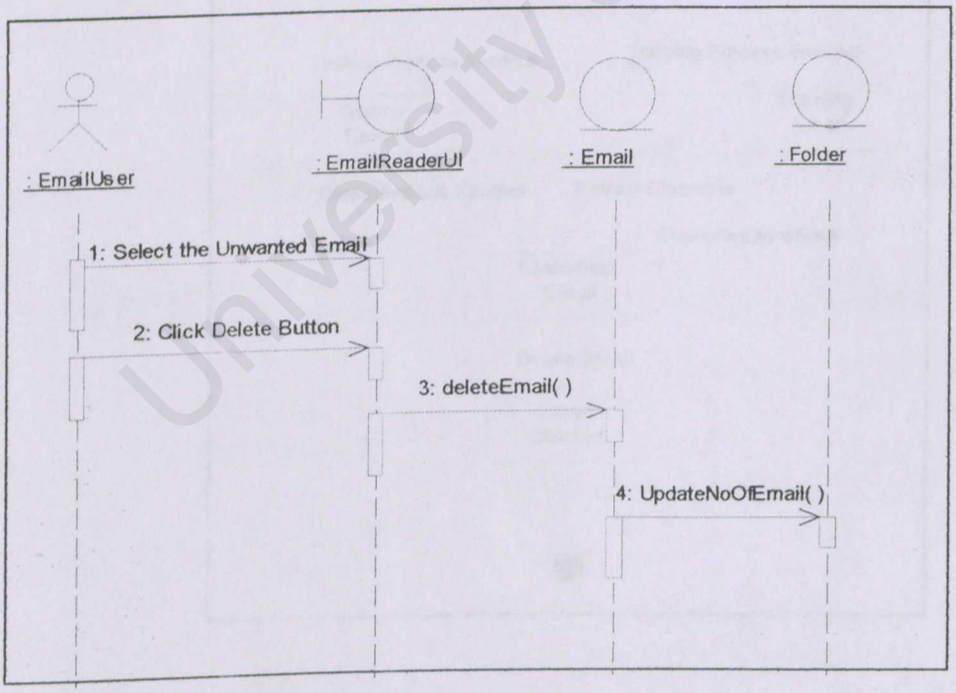


Diagram 5.9 Sequence Diagram for Delete Email Use Case

5.1.3 State Diagram and Activity Diagram

State Diagram shows the sequence of states that an object goes through during its life in response to outside stimuli and messages.

Activity Diagram is a variation or special case of a state machine in which the states are activities representing the performance of operations and transitions are triggered by the completion of the operations.

Both the state diagram and activity diagram give us a better understanding of the internal flow of the operations and thus can assist us to refines attributes and operations. The following diagrams show the state diagrams and activity diagrams.

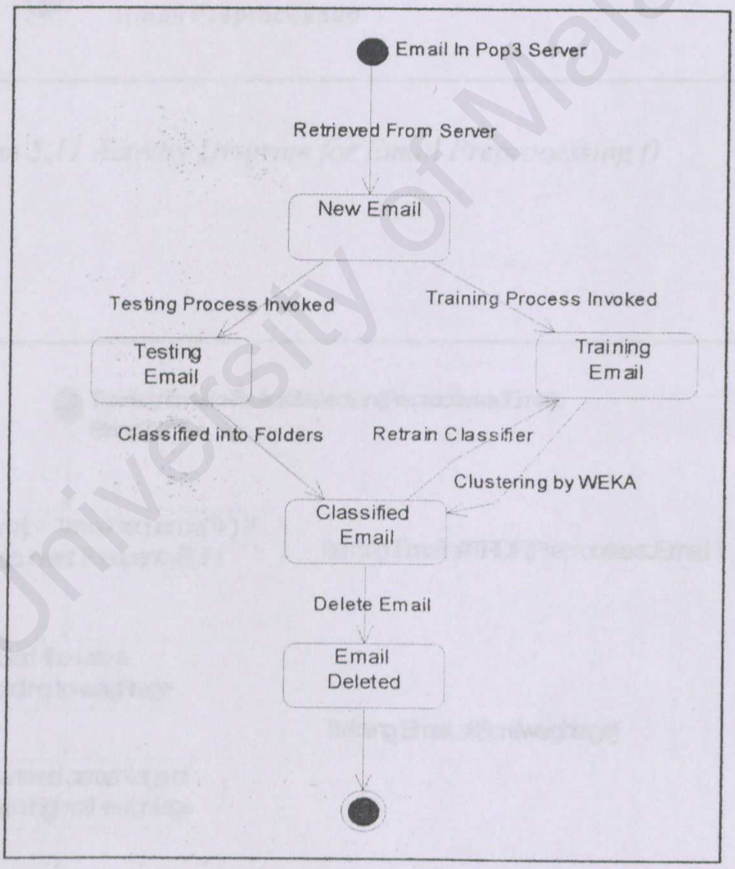


Diagram 5.10 State Diagram for Email Object

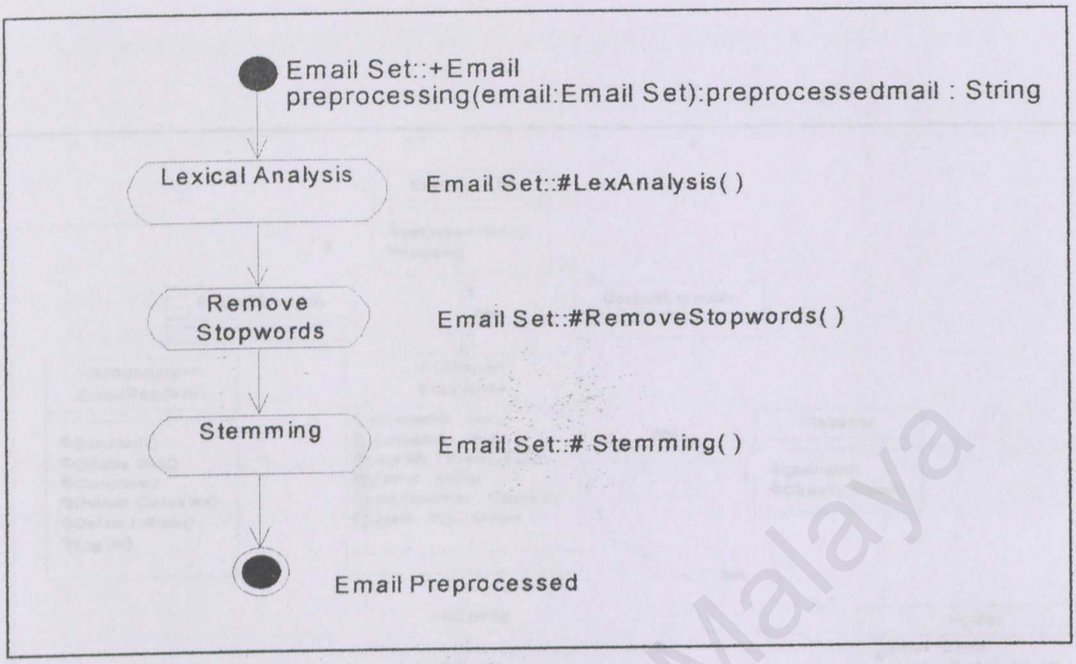

Email Set::+Email preprocessing(email:Email Set):preprocessedmail : String
Lexical Analysis
Email Set::#LexAnalysis( )
Remove Stopwords
Email Set::#RemoveStopwords( )
Stemming
Email Set::# Stemming( )
Email Preprocessed

Diagram 5.11 Activity Diagram for Email Preprocessing ()

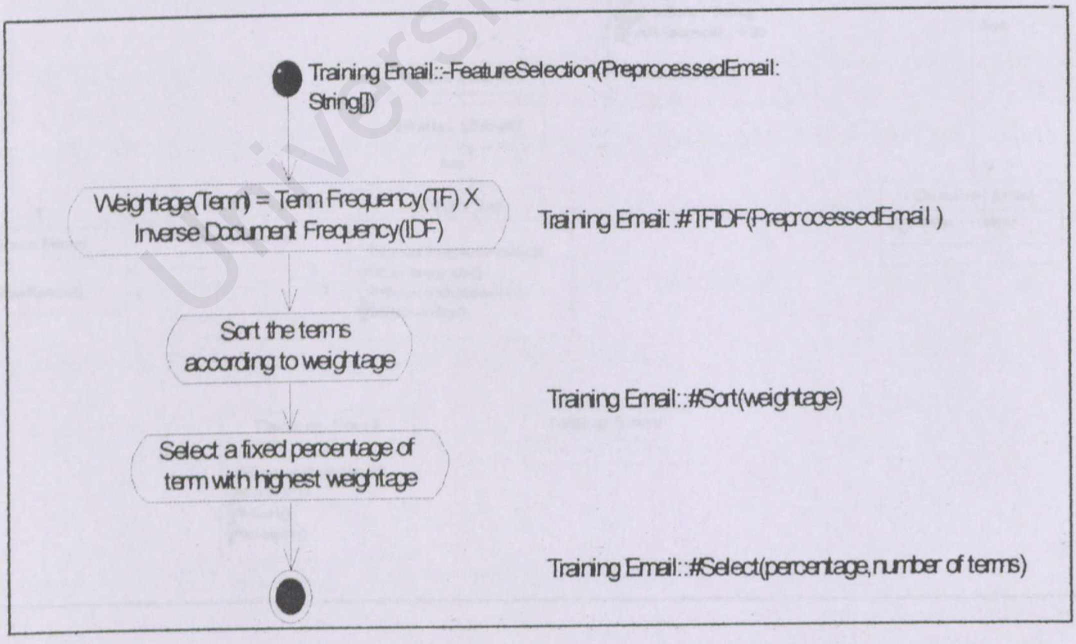

Training Email::-FeatureSelection(PreprocessedEmail: String[])
Weightage(Term) = Term Frequency(TF) X Inverse Document Frequency(IDF)
Training Email: #TFIDF(PreprocessedEmail)
Sort the terms according to weightage
Training Email::#Sort(weightage)
Select a fixed percentage of term with highest weightage
Training Email::#Select(percentage,number of terms)

Diagram 5.12 Activity Diagram for Feature Selection ( )

5.1.4 Class Diagram

Diagram below shows the complete class diagram of the system.

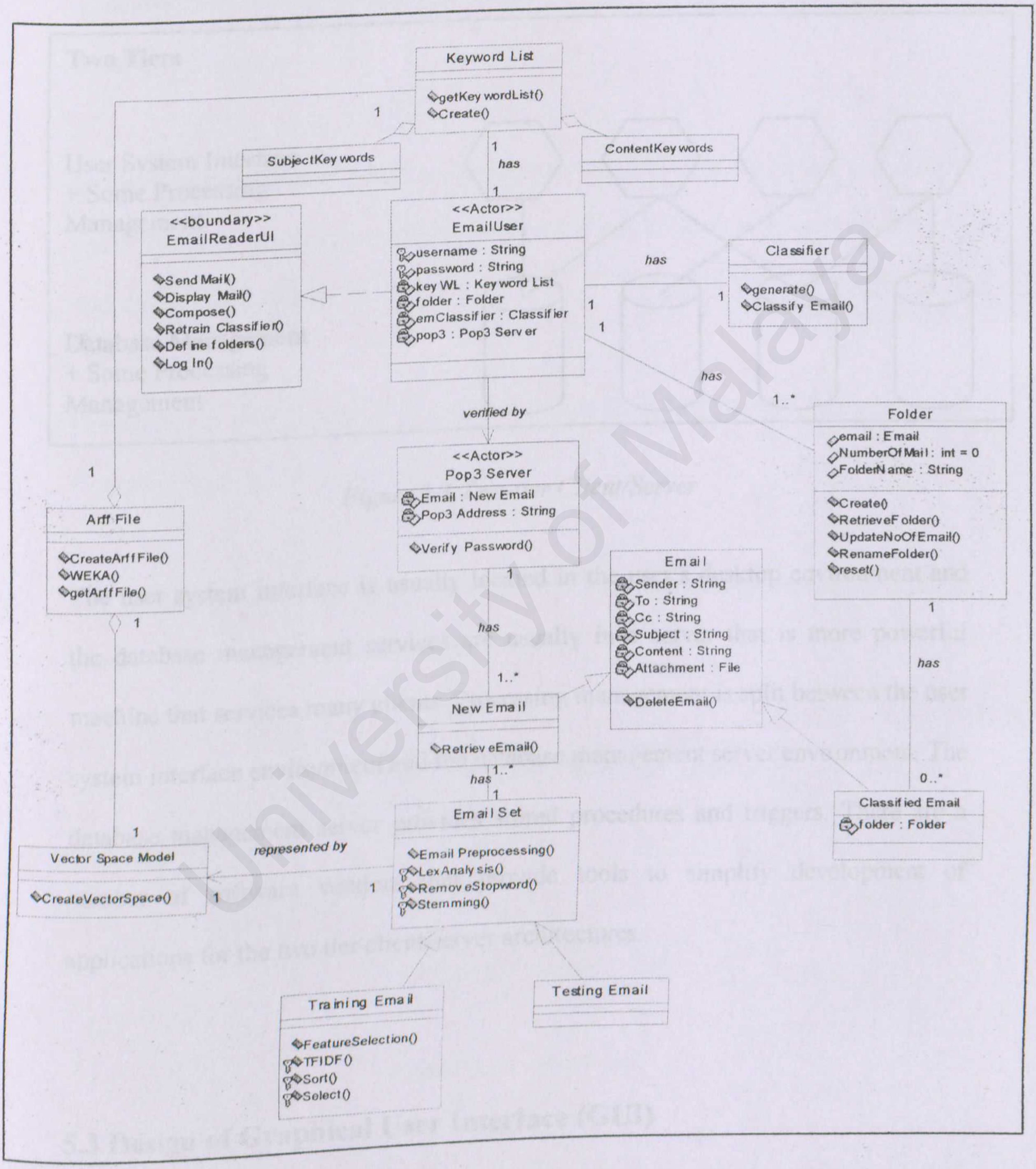


Diagram 5.13 Class Diagram of Automatic Email Classification System

5.2 System Architecture - Two-Tier Client/Server

The client server architecture model is chose as Automatic Email Classification System's network architecture. Refer the two-tier client/server architectures below:

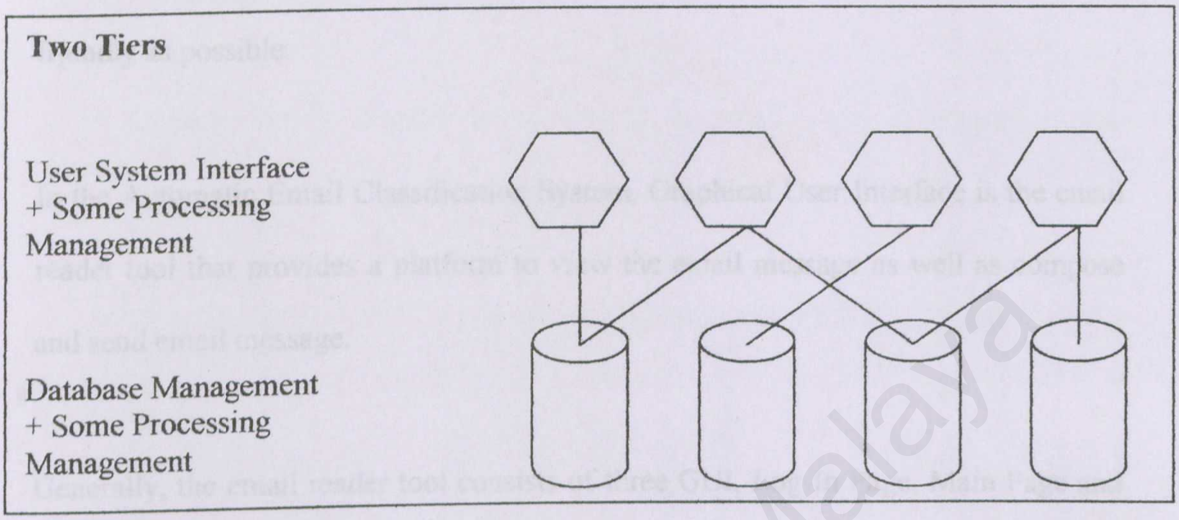


Figure 5.1 Two-tier Client/Server

The user system interface is usually located in the user's desktop environment and the database management services are usually in a server that is more powerful machine that services many clients. Processing management is split between the user system interface environment and the database management server environment. The database management server provides stored procedures and triggers. There are a number of software vendors that provide tools to simplify development of applications for the two tier client/server architectures.

5.3 Design of Graphical User Interface (GUI)

The interface of a system works as a central communication between the processing functions and the user requests. The objective of an interface is to enable the user to grab the information that they need or to act as a medium for them to supply more information to the system. The interface is aimed to improve the efficiency and effectiveness of the user when using the entire system. In addition, a good interface shall not cause the user to remember a large number of commands or codes but are as friendly as possible.

In the Automatic Email Classification System, Graphical User Interface is the email reader tool that provides a platform to view the email message as well as compose and send email message.

Generally, the email reader tool consists of three GUI, Log In Page, Main Page and Compose Page, as shown as figure below.

Log In Page provides two text fields that allow the users to enter their username and password, in order to access to the Pop3 server.

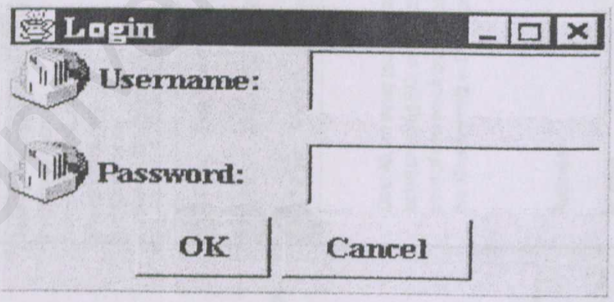
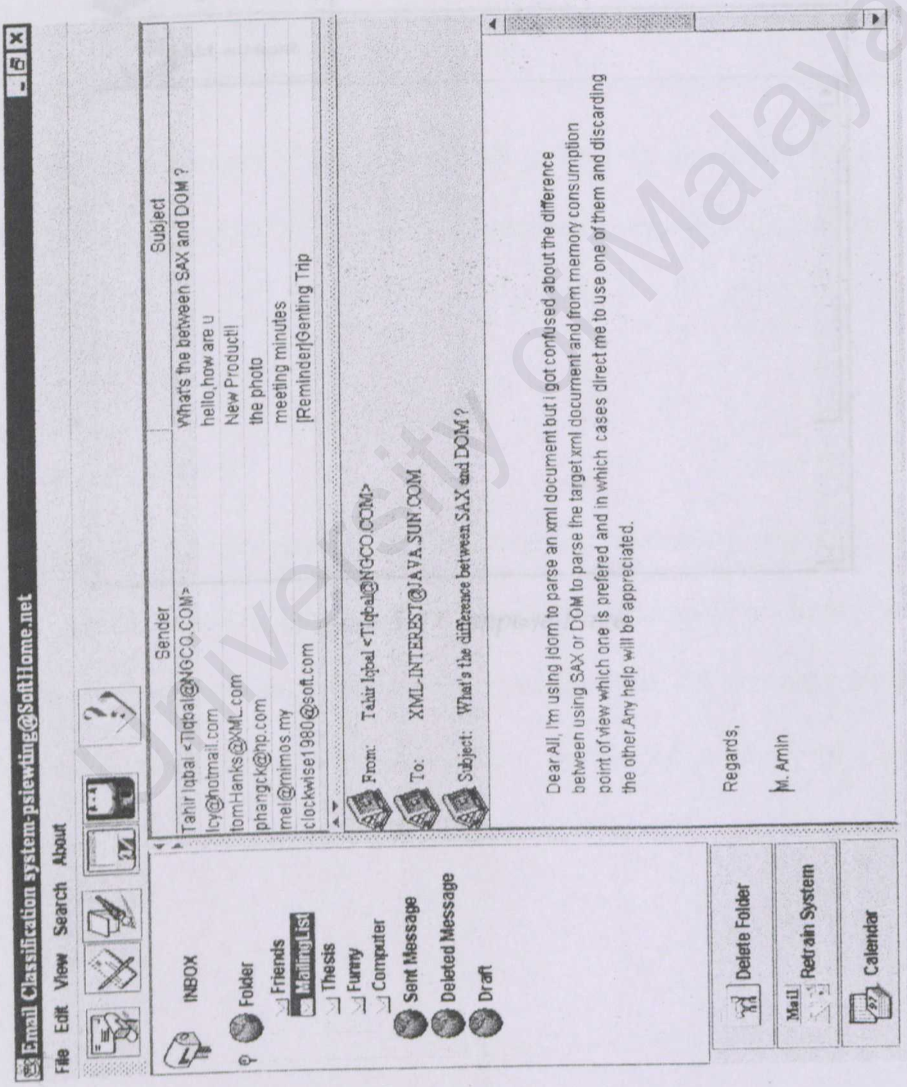


Figure 5.2 Log In Page

Main Page is the main interface of the system, which consists of three parts.

- Folder Pane – display the existing folders in a tree structure
- Preview Pane – lists the sender and subject and date of all the email messages in a selected folder and sorts them in ascending or descending order in preview pane
- Message Pane – display the content of the email message, selected in the Preview Pane.



Sender	Subject
Tahir Iqbal <TIqbal@NGCO.COM>	What's the between SAX and DOM ?
lcy@hotmail.com	hello,how are u
tomHanks@XML.com	New Product!!
phangck@hp.com	the photo
mel@mimos.my	meeting minutes
clockwise1980@soft.com	[Reminder]Genting Trip

From: Tahir Iqbal <TIqbal@NGCO.COM>

To: XML-INTEREST@JAVA.SUN.COM

Subject: What's the difference between SAX and DOM ?

Dear All, I'm using Jdom to parse an xml document but i got confused about the difference between using SAX or DOM to parse the target xml document and from memory consumption point of view which one is prefered and in which cases direct me to use one of them and discarding the other.Any help will be appreciated.

Regards,

M. Amin

Figure 5.3 Main Page

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

System implementation is a process that converts the system requirements and system designs into workable program codes. The initial stage of system implementation involves setting up the development environment which involved installing proposed development tools to facilitate the system implementation.

Each module in Automatic Email Classification System developed separately and later integrated into a fully functional system once every module has been tested successfully.

6.1.1 System Design

Rational Rose is used in preparing Unified Diagram for developed system. Although system design is being stated clearly in Chapter 5, nevertheless, during the initial stage of system development, a number of considerations, adjustments and addition of functions were done to the initial system design when the earlier proposed techniques are found not suitable and also in order to match the actual needs and requirements.

6.1.2 System Development

The basic tools used for the system development are:

i. Microsoft Windows 2000 Server (Operating System)

ii. Java Development Kit 1.4.0 (JDK1.4)

iii. Textpad 4.0 (Editor for java coding)

iv. Adobe Photoshop 7.0 (Icon Creation Tool)

6.1.3 Report Writing

Microsoft Word 2000, Microsoft PowerPoint 2000 and Rational Rose are used for preparing proposal and other requirement representations. All the problems encountered, together with solutions found throughout the processes (from system implementation until system evaluation) are recorded as well as result from system testing and system integration.

6.2 System Coding

6.2.1 Coding Approach – Object Oriented Programming (OOP)

OOP is programming paradigm that is fundamentally different for traditional procedural programming styles. It is centered on the concept of objects-programming constructs that have properties and the procedures for manipulating those properties. This approach models the real world much more closely than conventional programming method and is ideal for the simulation-type problems commonly encountered in Automatic Email Classification System.

To successfully use Java to develop a system, embracing the OOP into the implementation phase allows coding effort to be focused on the states and behaviors of the objects involved. Another key benefit in OOP is code reuse. This definitely helps in achieving the goal of reusing as much as the code as possible. Thus, the project would concentrate wholly on the code building in and OO manner.

6.2.2 Coding Principle

Throughout the development of Automatic Email Classification System, several guidelines and principles are followed during the implementation phase:

- *Reuse*

  Reuse has long been touted as a method for improving product quality through out the software development process. It is important to create components (classes) to be reused in subsequent and related applications. Productivity can increased not only by reducing testing and documentation times.

- *Readability*

  A readable programming style supports ease of future enhancement by other developers without having to verify its implementation line by line. Thus, several strategies are used in preserving readability in the codes, including meaningful variables and labels name, header comment blocks, comments and proper indentation.

- *Robustness*

  This is another important factor that determines the quality of a system. A program can preserver its robustness in terms of handling exception errors. In this project, control structures are embedded to capture all predictable exceptions. When an exception occurs, the normal flow of system execution is stopped and control is transferred to a class that has been designated to handle the exceptional condition and display relevant exception messages.

- *Maintainability and Ease of Testing*

Program codes that perform functions for one module should be grouped together. In other words, high cohesion and loose coupling should be achieved.

6.2.3 Coding Methodology

The coding methodology of this project is the bottom-up approach. The bottom-up approach starts coding at the lower level modules before the higher modules. The higher modules act as an empty shell that calls the lower modules. The completed lower level modules will then be integrated with the newly completed higher-level modules.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Testing is an important process in developing a system. All of the system's newly written or modified application programs-as well as new procedural manuals, new hardware, and all system interfaces must be tested thoroughly. Testing of a system does not actually come at the end of the system development, but should be carried out during the development phase.

The purpose of testing is to ensure that the resulting component of program as well as the program as a whole fulfills the requirement specification and to eliminate faults in the program. Due to the errors that has been done during the system development or system design, faults and failures may happen even when the entire system has been developed. Therefore, the main idea of testing is to demonstrate correctness of the program, identify the errors in the system coding or the system design. The faults that are discovered during the testing procedures will be corrected.

7.2 Types of Testing

Although the testing process involved a lot of methods and testing levels, but basically there are 3 major stages of testing involved in the Automatic Email Classification System.

1) Module Testing
2) Integration Testing
3) System Testing

The figure below depicts the flow of testing stages involved:

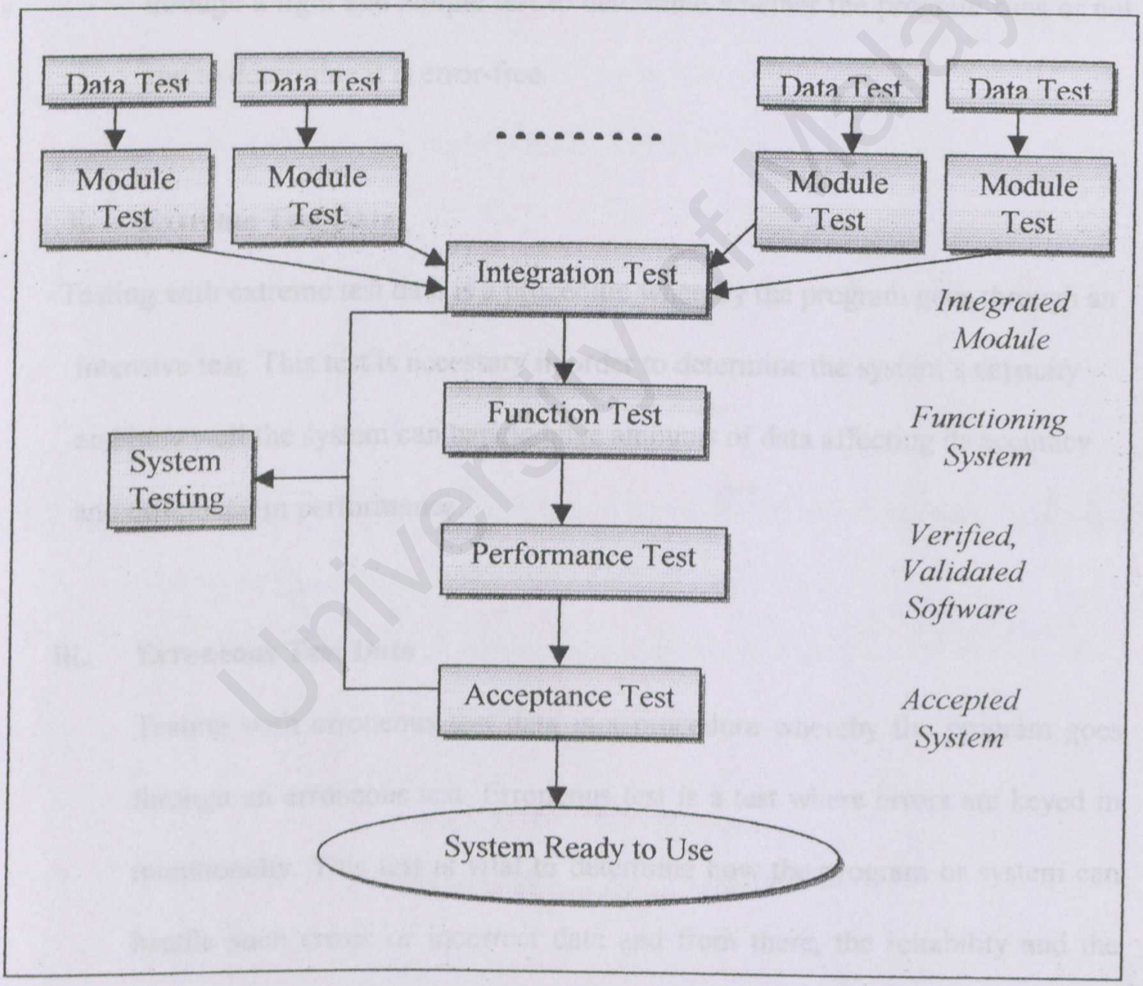


Figure 7.1 The Flow of Testing Stage

7.2.1 Data Testing

Test data was used in the execution of the program. For this system, a series of tests was conducted with data that are individually designed to represent the real environment as closely as possible. 3 categories of data were used to execute the program and they are:

i. **Normal Test Data**

Testing with normal teat data is a procedure whereby the program goes through a light and simple test to determine whether the program runs or not and to determine it is error-free.

ii. **Extreme Test Data**

Testing with extreme test data is a procedure whereby the program goes through an intensive test. This test is necessary in order to determine the system's capacity and how well the system can handle huge amounts of data affecting its accuracy and efficiency in performance.

iii. **Erroneous Test Data**

Testing with erroneous test data is a procedure whereby the program goes through an erroneous test. Erroneous test is a test where errors are keyed in intentionally. This test is vital to determine how the program or system can handle such errors or incorrect data and from there, the reliability and the efficiency of the system can be predicted.

7.2.2 Module Testing

Module testing is performed without other system modules. A module consists of a collection of dependent components to perform a particular task or function. Different possible test cases are applied to the module and the test results would be verified. Unusual results will be analyzed and they would help in debugging sub modules in order to produce the desired output.

The test is dynamically done. Dynamic test require modules to be executed on a machine. To do this, white-box testing is conducted. White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. It can be conducted in parallel for multiple modules.

The steps for module testing are:

i. Manually examine the code simply just form reading through it, trying to spot algorithm and syntax errors.

ii. Comparing the codes with the specification defined and also with the design is necessary to ensure all relevant cases are considered.

iii. Compile the code and eliminate remaining syntax faults.

iv. Develop test cases to show that the input is properly converted to the desired output.

Testing in this Automatic Email Classification System is focused on the three main modules: Email Reader Module, Text Analyzer Module and Classification Module. The following section discusses some of the sub-modules testing in detail:

i. Email Reader Module

- Log in to the Pop3 Account with valid username and password. Validated users are allowed to access to Pop3 server and retrieve email messages from the account.
- Log in with either incorrect login username or password. The system will alert the users that either the username or password is incorrect.
- Test if the all the emails in various MIME format, such as *.zip, *.jpg, *.bmp, *.html, *.txt, can be displayed properly.
- Test if the Email Reader Module can send or reply email to other email addresses and make sure the sent email messages can be received and viewed properly by the recipients.

ii. Text Analyzer Module

- Test if this module can analyze the plain text data properly
- In preprocessing part, make sure that all the common words in the stoplist have been removed and the remaining words are being stemmed to its root word.
- In calculation part, test if the Term Frequency-Inversed Document Frequency (TFIDF) is calculated correctly
- Test if the two keyword lists for both subject and content are generated and make sure keywords in the keyword list are not redundant.
- Test if the vectors for each email are generated correctly.

iii. Classification Module

- Make sure the Attribute Relation File Format (ARFF) file is in the right format.
- Test if the generated classifier is generated correctly.
- Test if the generated classifier cluster or classify the email messages correctly.

7.2.3 Integration Testing

The integration testing is carried out after the module testing process has been done. When the individual components or modules are working in satisfactory and meeting the system objectives during the module testing, those modules are then being combined into a whole working system. Several independent modules combined into a single system may cause some unpredicted and unexpected errors that relates to the integration of these modules. Therefore, integration testing is a systematic approach for constructing the application while conducting tests to uncover errors associated with interfacing of different components or modules.

There are many approaches that can be used to do the integration testing. There are the Bottom-Up Integration, Top-Down Integration, Big-Bang Integration and Sandwich Integration. For this system, the Bottom-Up Integration has been used. By using Bottom-Up Integration, each component or module at the lowest level of the system hierarchy is tested individually at first. Then, the next components to be tested are those that call the previous components. This approach is followed repeatedly until all components or modules are included in the testing. After the integration test is completed, those errors and faults discovered are being corrected as soon as possible in order to proceed to the system-testing phase.

The Figure 7.2 below shows an example of constructed component hierarchy, whereas Figure 7.3 depicts the sequence of tests and their dependencies of Bottom-Up Testing.

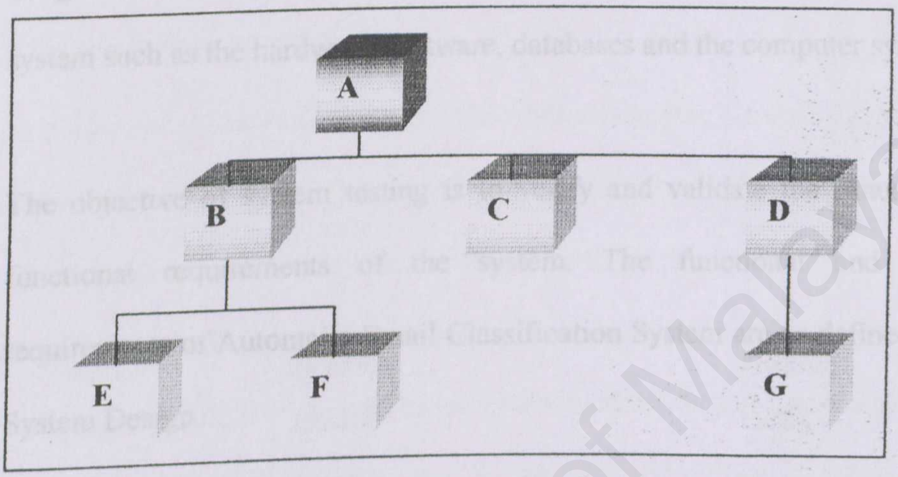


Figure 7.2 An example of constructed component hierarchy

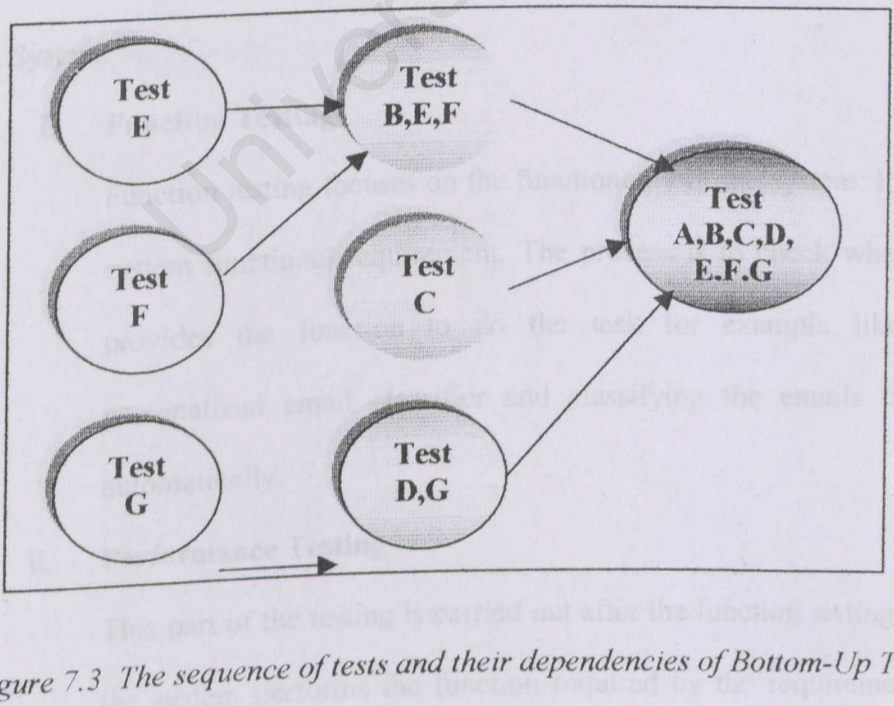


Figure 7.3 The sequence of tests and their dependencies of Bottom-Up Testing.

7.2.4 System Testing

After all the modules are completed, the entire system must then be validated. Carrying out the system testing process does the validation of the system. Testing the whole system is very different from module and integration testing. When the system testing process is being carried out, the major difference compared to module and integration testing is that one needs to work with the entire environment of the system such as the hardware, software, databases and the computer systems.

The objective of system testing is to verify and validate the functional and non-functional requirements of the system. The functional and non-functional requirements of Automatic Email Classification System are as defined in Chapter 5: System Design.

There are several types of system testing that can be used to test a software system. But only 3 types of system testing are applied on the Automatic Email Classification System.

i. **Function Testing**

   Function testing focuses on the functionality of the system. It is based on the system functional requirement. The process is to check whether the system provides the function to do the task for example like generating a personalized email classifier and classifying the emails into the folders automatically.

ii. **Performance Testing**

   This part of the testing is carried out after the function testing process. When the system performs the function required by the requirements, the testing process then turn to test the way in which those functions are performed. Thus, the performance testing addresses the non-functional requirements. The purpose of this testing is to test the run time performances of the software within the context of an integrated system. It involves both hardware and software instruments.

7.2.5 Acceptance Testing

The final stage of testing process before the system is being accepted by the users is the acceptance testing. Testing by users will reveal the errors and omission in the system requirements definition because the acceptance testing involves testing from the users. This will also reveal the requirement problems where the systems facilities do not really meet the user's needs or the system performances is unaccepted.

Acceptance testing for Automatic Email Classification System is being conducted by asking the users to experience themselves with the system and test the accuracy or efficiency of the generated email classifier.

CHAPTER 8

EVALUATION AND CONCLUSION

8.1 Introduction

This chapter is the final phase in the life cycle of the Automatic Email Classification System project. During the periods of coding and implementation, various problems were encountered. So this chapter will highlight some of the problems faced throughout the project duration and also with the solutions that have been taken to solve those problems. Besides that this chapter will also includes the evaluation of the system to identify its strengths and limitations. Possible ways to enhance the system are also being explored as suggestions to further improve the system.

8.2 Problems and Solution

8.2.1 Lack of Experience Implementing External Java Package

In developing the Automatic Email Classification System, I have implemented three external java packages: WEKA, mailpuccino and Oyoaha LookAndFeel. Lack of experience dealing with the external java packages especially WEKA and Mailpuccino has become the major problem in developing the system. The following states out the problems faced in implementing the WEKA package and Mailpuccino package.

- **Extract Source Code from WEKA package**

WEKA is a collection of Machine Learning algorithms for solving real-world data mining problems (Refer to Section 2.6.4). The structure of the package is very complex and consists of several hierarchies. In my system, only the K-

Means clustering algorithm is needed. So it is necessary to extract the source code from package before I can integrate it into my system. Lack of experience dealing with java packages and lack of knowledge in data mining has become the biggest obstacle in developing the system.

Solution

With referring to the PIGEON –the prototype of Automatic Email Classification System, which applies Naïve Bayesian algorithm in generating the classifier, and the searching information from the book shop, I successfully extract the source code from the package and integrated it into my system.

- **Modify and Enhance the Mailpuccino Package**

Mailpuccino is a package of a complete email client. The structure of this package is much more simple if comparing to WEKA. However, many of the classes in the package are not compatible with my system and the functions inside are not enough to fulfill the system requirements. Furthermore the some of the classes in the package have been deprecated. A lot of modifications are required to be done on the package and a lot functions need to be added into the package. It is not an easy task to modify the package as extensive studies necessary to understand thoroughly the source codes before any modifications can be done on the package.

Solution

I have spent more than one-week time to understand and test the package before I can start modifying the package. A lot of additional functions has been added. Several classes of others external packages such as Pooka, Javamail, and Netscape Messenger are being integrated into this package in order to produce a more complete and comprehensive package to fulfill the requirement of my system.

8.2.2 Lack of Email Messages in my Pop3 Server

The system is designed to analyze the plain text data with only the pure English. This is because English stemming algorithm and English stoplist are used in the preprocessing step. However, most of the emails in my Pop3 account are the forwarded emails with attachment. Furthermore, not all the emails are in the pure English language. Some of the emails are actually in Chinese or Malay language and some are in a mixture of two different language. Using these email messages will affect the accuracy of the classifier.

Solution

As a solution to solve this problem, I have subscribed a lot of newsletters and mailing lists from Internet. All the subscribed newsletters and mailing lists can guarantee the pureness of English language. I have chose to receive the newsletter/mailing list in plain text format during the subscription so that the content of email messages can be preprocessed by the system.

8.3 System Evaluation

The main objective of this thesis was to develop accurate and efficient system to automate the process of filing e-mail messages into folders. Therefore, the accuracy of the classification will almost determine the performance of the system. The following subsections discuss about accuracy of the classification result and system evaluation done by the end users. Due to the time constraint, only three users are invited to test the system.

User 3

Table 8. 3 The Email Classification Result - User 3

Cluster	Folder Name (Content)	Emails	Correct Classify	Incorrect Classify
1	Friends	15	10	5
2	Article	14	9	5
3	Entertainment	11	8	3
4	Mailing List	15	15	0
5	Miscellaneous	10	5	5
	Total	65	47	18

Correctly Classified Instances	47	72.31%
Incorrectly Classified Instances	18	27.69%

8.3.1 Discussion

From the tables above, the system proves the average accuracy is more than 70% correctly classified emails.

Generally we can conclude that the accuracy of the classification is determined by the content. From the results above, the folders that consist of newsletter or mailing lists such as Washington Newsletter, Flashkit, E-book Mailing and Embarrassing, from user 1, Mailing list folder from user 2 and user 3, normally have the higher classification accuracy. This is possible due to the contents in newsletters and mailing lists has more precise and specific topics, thus make the K-Means algorithm easier to converge the email messages with the same topics into clusters. For the other folders like Friends folder from user 1, Miscellaneous folder from user 2 and user 3, these folders show the lower accuracy. This might be caused by the variety and unspecific of the topics that cause ambiguous in clustering. Besides, many of the emails in these folders were not purely in the English language but a mixture of

Malay and English. No attempt was made to differentiate the English and Malay words. This could be a reason why the classification accuracy is lower for these kinds of folders.

8.4 System Strength

The system is observed to successfully provide the following functions and criteria:

i. **Efficiency automated email classification process**

The system assist the user handle high volume of email messages by automating the classifying process using intelligent method. A classifier will be generated in training process and all the incoming emails will be classified on the fly into the corresponding folders

As the content of the new emails and users' mailing habits might be changed over time, the system provides the Retraining function to allows the users to regenerate a classifier.

ii. **Interactive and User-friendly Interface**

Dynamic and well-organize folder structure

There are three default folders in the system: Inbox, Trash and Sent

Inbox: The default location to store all the emails retrieved from server if the classifier is not generated or as if the user doesn't want to classify the emails.

Trash: The deleted email messages will be moved automatically into the Trash folder

Sent: To store the copy of the email messages that sent by the user

The folders are represented in a dynamic tree structure whereas the user can **Add** new folder, **Remove** unwanted folder, and **Rename** the folder. (Except the default folders cannot be removed)

Since the folders created in the training process are given the default name: “Cluster0, Cluster1, Cluster2…” the **Rename** function is vital so that the user can change the default name of the folder based the content in each folder.

Well-organized Email Messages

By simply clicking on the folder tree, the email messages in the selected folder will be listed in a table. The user can sort the message list based on Sender name, Subject or Received date.

From the table the user can delete the unwanted message or move the message from one folder to another.

The Move function is necessary especially in this system as the classification done by this system is not 100% accurate. The user might have to move the message to the right folder using the Move function

Email Display

The preview pane consists of two tab panes: Body and Attachment

Body: Display the textual content of emails (includes the full header of a message, if required by the user)

Attachment: Display the MIME type attachment in an email message. Each attachment in an email message will be depicted as an icon in attachment pane. The user can view or save the attachment by simply double-clicking on the icon.

The system supports the following MIME type of attachment:

- plain text
- html

- JPEG
- GIF
- ZIP
- Bitmap

Theme selection

Ten different themes are installed in the system for user selection.

Effective of Using Icon and Short-Cut Key

The using of icon and short-cut key enables the users are able to navigate smoothly through the system

iii. Password Security

The users can choose to save the password of their Pop3 Account in the system user profiles so that the users do not need to enter the password for every time they want to fetch email from the server. The system guarantees the security of the password by encrypting the password.

iv. Import Tool

The system provides an import tool that allows the users import emails from other email client application. This tool will encourage the new user using the system as the users can easily switch their current email client program to Automatic Email Classification System.

v. **Easy-to-use Address Book**

The system provides an address book that allows the users to create the directories to store the email addresses or details for individual

vi. **A full functioning Email Client**

The Automatic Email Classification System is actually a full functioning email client that provides all the functions to handle the emails transportation over the Internet. It provides functions for composing, sending or receiving emails.

8.5 System Constraints

Can only analyze pure English email

Malaysia is a multiracial country where Malay is the national language and

English the second and thus some of our emails are not written in pure English but a mixture of both Malay and English. For our classifier, we assumed that all the emails are in pure English. We can thus enhance the classifier into a bilingual classifier by developing appropriate stemmers and stop word list for the Malay language.

Accuracy

The accuracy of the system is not hundred percent accurate. This is due to the variety of the email content makes the system difficult to determine the class of the emails. For the emails that are wrongly classified, the users are allowed to move the emails to the correctly folder using the Move menu provided.

Cannot analyze image

Currently the system classifies the email based on the textual content of an email classifier. All the attachments are not taken account in determining the classes. However, the attachment actually can determine which class an email should belong to.

The Manual Retraining Function

The system allows the user to regenerate the classifier.

For every training process, folders with the same default name (logical name) will created for each cluster. Therefore, in order to regenerate the classifier, the user are required to delete the folders generated in previous training process to avoid duplicated folders exist in the system. The duplicated folders will cause ambiguous in classifying the email message.

8.6 Future Enhancements

The following directions seem promising on the way to even more effective and efficient automatic email classification. Some suggestions to further automate e-mail processing are also given.

Enhance into a Bilingual Classifier

Malaysia is a multiracial country where Malay is the national language and English the second and thus some of our emails are not written in pure English but a mixture of both Malay and English. We can thus enhance the classifier into a bilingual classifier by developing appropriate stemmers and stop word list for the Malay language.

More flexible Retraining Function

The system should provide a more flexible and easy-to-use retraining function so that the user can retrain the classifier as needed easily.

Image attribute

The attached image attribute should be taken account to improve the classification accuracy. Technology for image analyzing can be researched and developed in email classification.

Time attribute.

The time and date when a message was sent might be very useful in reflecting periodical or seasonal changes in mailbox content. For example, some messages are automatically sent out at the same time every day, or other messages can be expected only when final exams are happening or during a ski season. Employing the time and date attributes with suitable granularity can improve the classification accuracy.

Incremental algorithm.

This is a challenging but very important task. An average user mailbox is very dynamic, i.e. both contents of new messages and the user's mailing habits constantly change. This requires creating, deleting or splitting folders to reflect the change. Retraining the classifier every time a change is needed is not feasible. A much better approach is to incrementally update the existing classifier by modifying the existing centroid of each cluster. A user's corrections to the system classification must be taken into account for this.

Message importance classification (or prioritizing)

Handling a big volume of e-mail messages can be facilitated further if the system not only automatically files the messages into folders but also ranks them according to their importance or urgency to the user. Each message's topic can be used as one of the attributes whose value is extracted to build such an \e-mail importance classifier". Other interesting features for this kind of classification are elements of the user's behavior while handling messages. These can be the time he/she spends on reading it, immediate or postponed reply or deletion, the order in which the messages are opened, etc.

Customer Relation Management (CRM)

Customer relation management is the ability to organize and maintain a connection with clients, customers and service agents with regards to business relationships and customer satisfaction. Customer relation management is a vital issue in the day-to-day business world. Providing customer service and having positive relations will help to ensure successful dealings with clients in every aspect of the business market. The data mining algorithm implemented in the system can be enhanced to be used Customer Relation Management.

8.7 Conclusion

The project has achieved it objectives to develop an application, the Automatic Email Classification System, which not only provides time saving for user but allows greater user interactivity and personalization elements.

As an extension of the previous email classification system prototype, the Automatic Email Classification System perform an enhancement over the previous prototype as the following:

a. The system successfully eliminates the laborious manual classification by using K-Means Clustering algorithm.
b. The system achieves much better accuracy in the classification.
c. The system has successfully enhanced the prototype to be an email client like Netscape Mail Messenger and Microsoft Outlook.

In the process, invaluable insight was gained into the complexities and intricacies of application programming. Knowledge gained throughout the life cycle of project development, from the planning of the project, studies on the subject and technologies, setting up of servers, programming, to implementing the system proves to be a valuable experience. At the same time, theories and knowledge, gained throughout the course of Information Technology studies and Industrial Training were put into practice. The experience will definitely prove useful in future software development projects.

There is till much room for improvement in the email classification system. The successful development of this system is the first step towards the future development of similar systems. It is hoped that it can provide a foundation and basis for the concept of data mining, which is increasingly getting popular in modern database system.

APENDIX

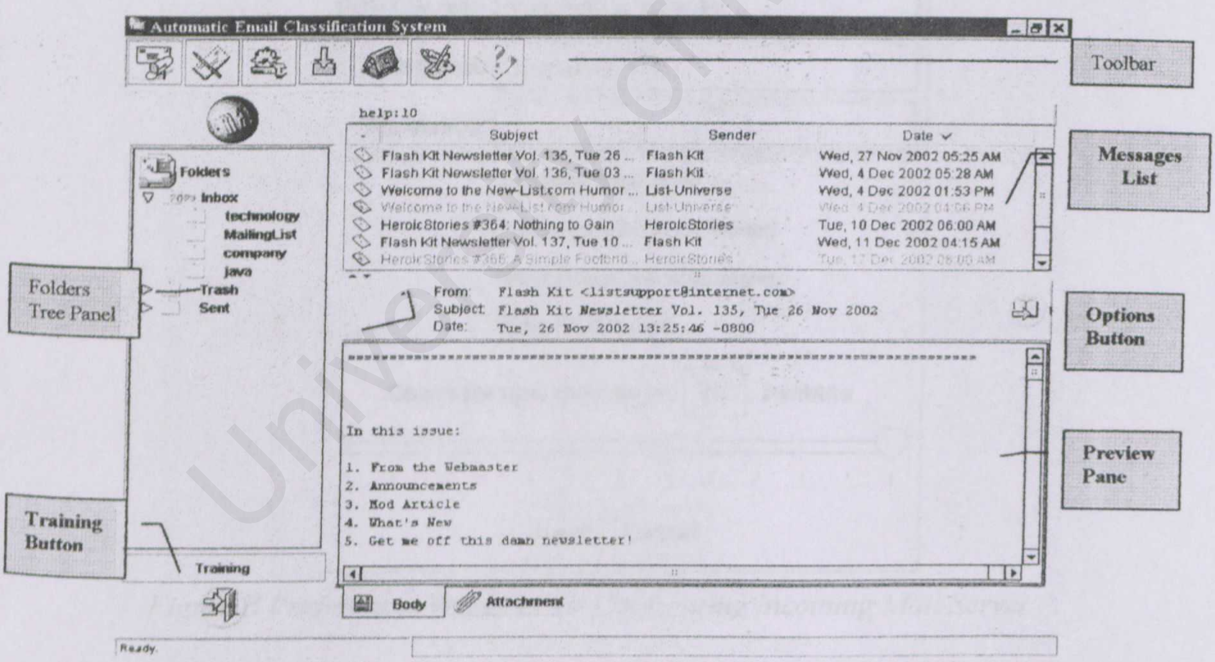
USER MANUAL

User Manual

Figure A shows the names of the each component in the main interface of Automatic Email Classification System.

Generally the interface consists of

- Folders Tree Panel – depicts the folders structure
- Messages List – A table that displays the list of email messages in the folder selected in Folders Tree Panel
- Preview Pane - view the content of an email messages selected in Message List
- Toolbar – consists of seven buttons. (Check Email, Compose Email, Import Tool, Address Book, Select Theme, and About)



help:10

Subject	Sender	Date ✓
Flash Kit Newsletter Vol. 135, Tue 26 ...	Flash Kit	Wed, 27 Nov 2002 05:25 AM
Flash Kit Newsletter Vol. 136, Tue 03 ...	Flash Kit	Wed, 4 Dec 2002 05:28 AM
Welcome to the New-List.com Humor ...	List-Universe	Wed, 4 Dec 2002 01:53 PM
Welcome to the New-List.com Humor ...	List-Universe	Wed, 4 Dec 2002 04:06 PM
HeroicStories #364: Nothing to Gain	HeroicStories	Tue, 10 Dec 2002 06:00 AM
Flash Kit Newsletter Vol. 137, Tue 10 ...	Flash Kit	Wed, 11 Dec 2002 04:15 AM
HeroicStories #365: A Simple Footbrid...	HeroicStories	Tue, 17 Dec 2002 08:00 AM

From: Flash Kit <listsupport@internet.com>
Subject: Flash Kit Newsletter Vol. 135, Tue 26 Nov 2002
Date: Tue, 26 Nov 2002 13:25:46 -0800

```
==========================================================================

In this issue:

1. From the Webmaster
2. Announcements
3. Mod Article
4. What's New
5. Get me off this damn newsletter!
```

Figure A The Main Interface of the Automatic Email Classification System

Configuring Mail Servers

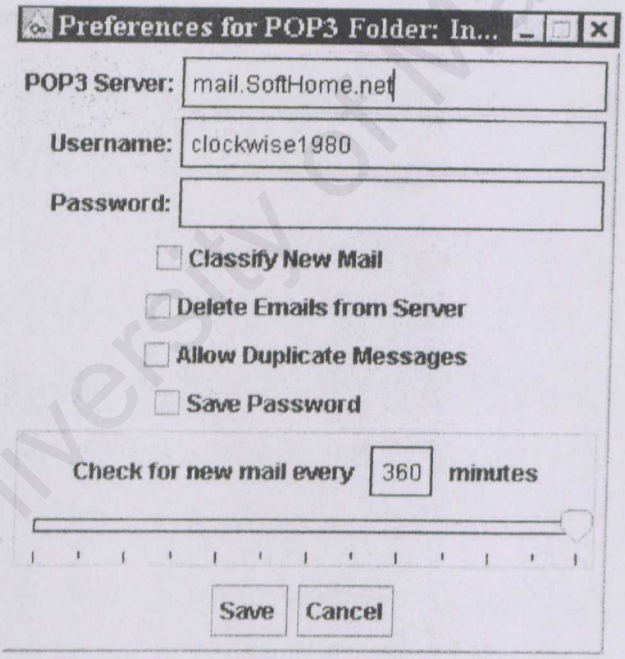
To configure the email server, click the configure button, third from the left, on the toolbar (Refer to Figure A).

From the popup menu, select **Configure Incoming Mail Server** or **Configure Outgoing Mail Server**. Follow the instructions below to do the configurations.

Configure Incoming Mail Server

Figure B shows the Preferences windows for configuring incoming mail server.

Be sure to type the following information exactly as it's given to you.



Preferences for POP3 Folder: In...

POP3 Server: mail.SoftHome.net

Username: clockwise1980

Password:

□ Classify New Mail

□ Delete Emails from Server

□ Allow Duplicate Messages

□ Save Password

Check for new mail every 360 minutes

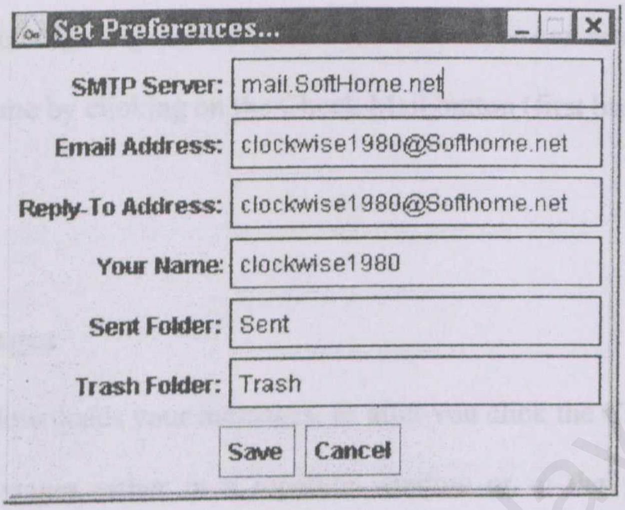
Figure B Preferences Windows for Configuring Incoming Mail Server

- *POP3 Server*: The name of the mail server that delivers your messages. Currently the system only supports Pop3 Server.
- *Username*: The name that identifies you to your POP3 Server.
- *Password*: Your password for the mail server

- *Classify New Mail*: Check if you want to classify the incoming new mail using the system generated email classifier.
- *Delete Email From Server*: Check it if you want to delete the emails from the server after retrieving it and being stored in the system local folder
- *Allow Duplicate Message*: Check if you allow duplicate message storing in your folder
- *Save Password*: Check if you don't want to re-enter your password each time you start the system.
- Select the box at "*Check for new mail every ___ minutes* " and then specify the number of minutes between mail checks. You might also use the slider to specify the minutes.

- Click **Save**.

Configure Outgoing Mail Server


Set Preferences...

SMTP Server: mail.SoftHome.net

Email Address: clockwise1980@Softhome.net

Reply-To Address: clockwise1980@Softhome.net

Your Name: clockwise1980

Sent Folder: Sent

Trash Folder: Trash

Figure C Preferences Windows for Outgoing Mail Server

Figure C shows the Preferences Windows for configuring outgoing mail server (SMTP Server). To configure the outgoing mail server, fill up the following text field

- *SMTP Server*: The name of your SMTP server
- *Email Address*: Your email address, which is verified by your SMTP server.
- *Reply-To Address*: The email address that the recipient of the email message will reply to.
- *Your Name*: The name that will be displayed as Sender in your sent message.
- *Sent Folder*: Specify the folder that store the sent email messages (default: Sent)
- *Trash Folder*. Specify the folder that store the deleted email messages (default: Trash)

Click **Save.**

Getting New Messages

The system automatically downloads new messages to your Inbox at timed intervals as you've set it in configuring incoming mail server, but you can retrieve them manually at any time by clicking on the Check Mail button (first button from left in the toolbar).

Reading Messages

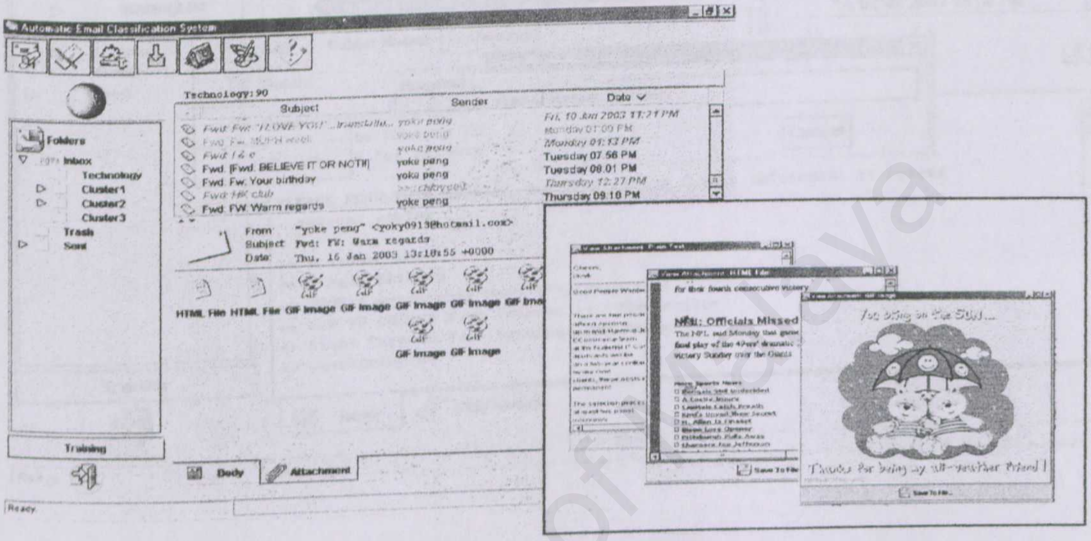
After the system downloads your messages, or after you click the Check Mail button, you can read messages either in a separate window or in the preview pane by following the instructions below:

- Click the **Inbox** icon on Folders tree Panel.
- All the email messages in the **Inbox** will be listed in message list (Refer to Figure A)
- To view the message in the preview pane, click the message in the message list.
- To view all the headers of a message, click on Options button, and check the View All Header button.
- To save the message in your file system, click the Options button and click **Save as.**
- To add the Sender or Recipient(s)'s address(es) into your address book, click on Options button and check the " Add Sender" or " Add Recipient(s)" option.

To View or Save Attachment

To view the message in a separate window or to view the attachment, double-click on the icons in Attachment pane.

To save the message or attachment, click the Save button below the separate window.



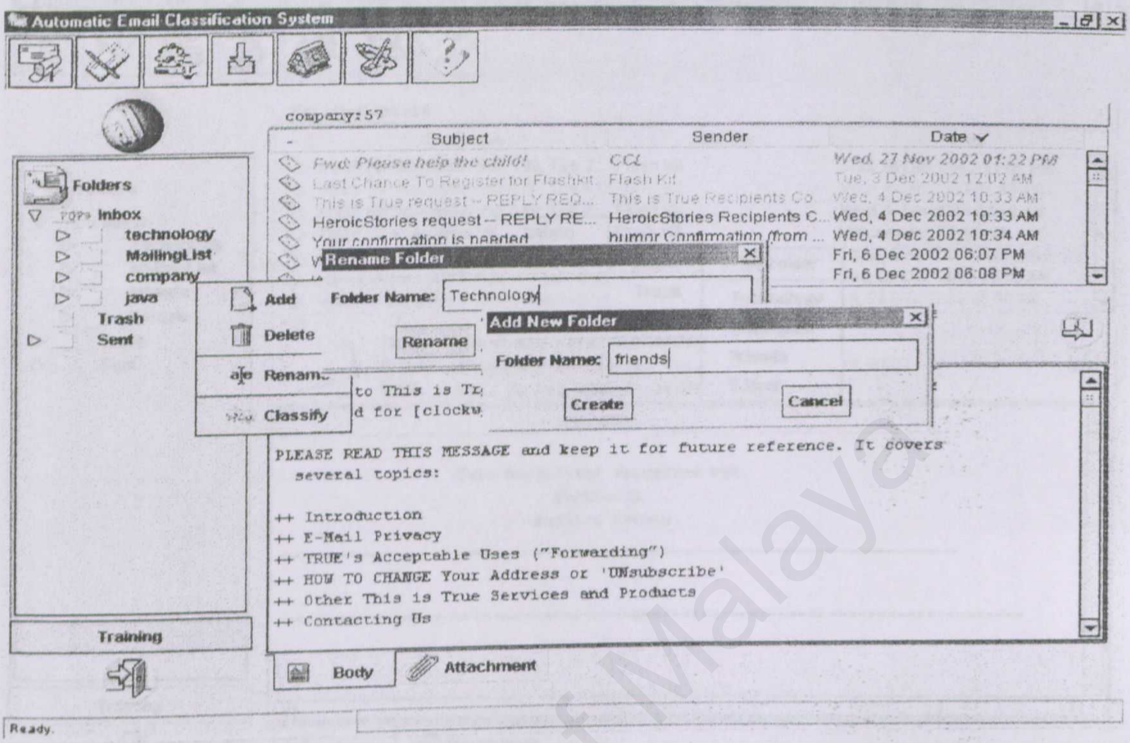
Technology:90

Subject	Sender	Date
Fwd: Fw: "I LOVE YOU" ...translatio...	yoke peng	Fri, 10 Jan 2003 11:21 PM
Fwd: Fw: MURH week	yoke peng	Monday 01:09 PM
Fwd: I & c	yoke peng	Monday 01:13 PM
Fwd. [Fwd. BELIEVE IT OR NOT!!]	yoke peng	Tuesday 07:56 PM
Fwd. Fw. Your birthday	yoke peng	Tuesday 08:01 PM
Fwd: HK club	>>chkyee	Thursday 12:27 PM
Fwd. FW. Warm regards	yoke peng	Thursday 09:10 PM

From: "yoke peng" <yoky0913@hotmail.com>
Subject: Fwd: FW: Warm regards
Date: Thu, 16 Jan 2003 13:10:55 +0000

Figure D The Attachment Preview Pane

Add, Delete, or Rename folder



company:57

Subject	Sender	Date
Fwd: Please help the child!	CCL	Wed, 27 Nov 2002 01:22 PM
Last Chance To Register for Flashkit...	Flash Kit	Tue, 3 Dec 2002 12:02 AM
This is True request -- REPLY REQ...	This is True Recipients Co...	Wed, 4 Dec 2002 10:33 AM
HeroicStories request -- REPLY RE...	HeroicStories Recipients C...	Wed, 4 Dec 2002 10:33 AM
Your confirmation is needed	humor Confirmation (from ...	Wed, 4 Dec 2002 10:34 AM
W		Fri, 6 Dec 2002 06:07 PM
		Fri, 6 Dec 2002 06:08 PM

Rename Folder — Folder Name: Technology

Add New Folder — Folder Name: friends

```
to This is Tr
d for [clockw

PLEASE READ THIS MESSAGE and keep it for future reference. It covers
  several topics:

++ Introduction
++ E-Mail Privacy
++ TRUE's Acceptable Uses ("Forwarding")
++ HOW TO CHANGE Your Address or 'UNsubscribe'
++ Other This is True Services and Products
++ Contacting Us
```

Figure E Add, Remove, and Delete Folder

To Add Folder

Click on a folder node in Folders tree panel, right-click and click **Add** from popup menu. In the Folder Name text box, type the name and click **Create**.

To Remove Folder

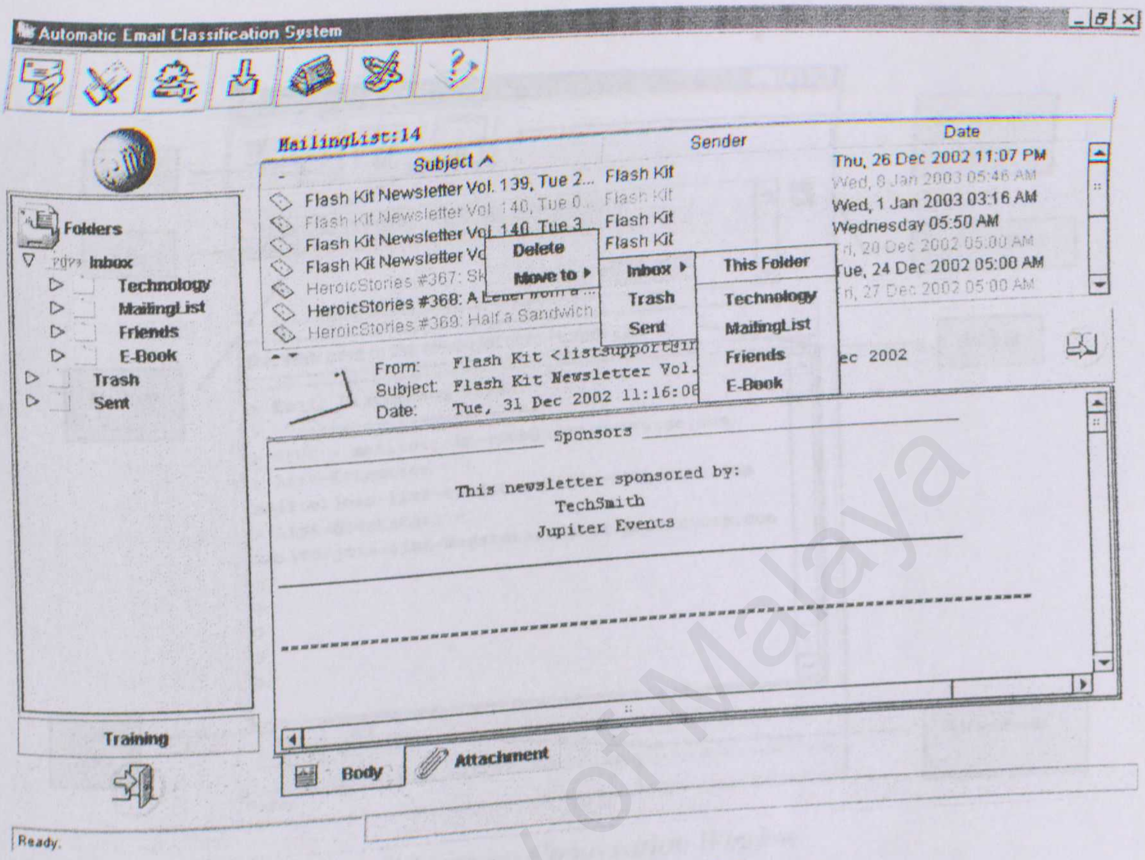
Click the folder in the Folders tree panel. Right-click and click Remove from the popup menu.

To Rename Folder

Click the folder in the Folders tree panel. Right-click and click **Rename** from the popup menu. In the Folder Name text box, type the name and click **Rename.**

NOTES: You cannot delete **Inbox**, **Trash** and **Trash** folders.

Delete, Move and Restore Message



MailingList:14

Subject	Sender	Date
Flash Kit Newsletter Vol. 139, Tue 2...	Flash Kit	Thu, 26 Dec 2002 11:07 PM
Flash Kit Newsletter Vol. 140, Tue 0...	Flash Kit	Wed, 8 Jan 2003 05:46 AM
Flash Kit Newsletter Vol. 140, Tue 3...	Flash Kit	Wed, 1 Jan 2003 03:16 AM
Flash Kit Newsletter Vo...	Flash Kit	Wednesday 05:50 AM
HeroicStories #367: Sk...		Fri, 20 Dec 2002 05:00 AM
HeroicStories #368: A Letter from a ...		Tue, 24 Dec 2002 05:00 AM
HeroicStories #369: Half a Sandwich		Fri, 27 Dec 2002 05:00 AM

From: Flash Kit <listsupport@fl...
Subject: Flash Kit Newsletter Vol. ...
Date: Tue, 31 Dec 2002 11:16:08 ... ec 2002

Sponsors

This newsletter sponsored by:
TechSmith
Jupiter Events

Figure F Delete, Move or Restore Message

To delete a message

In the message list, select the message. Right-click and click **Delete** from the popup menu. The deleted message will be stored in the **Trash** folder.

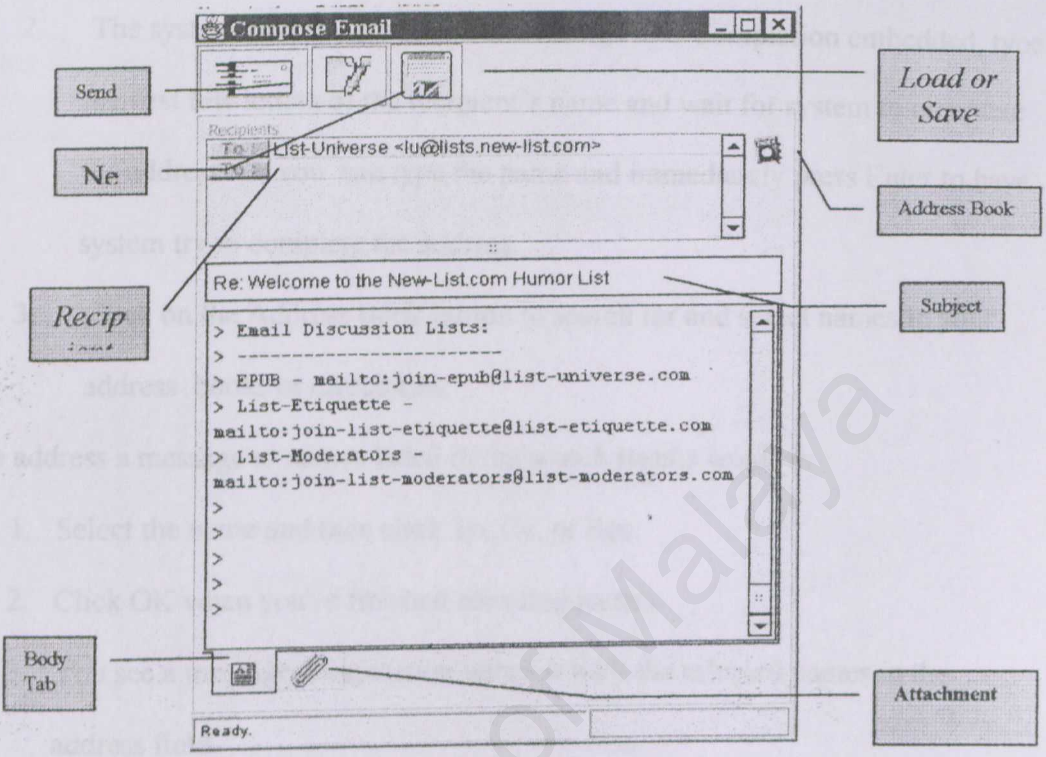
To move a message to another folder

In the message list, select the message you want to move. Right-click and click **Move to**, and then select the destination from the popup menu.

To restore a deleted message

Open the **Trash** folder, and move the message back to the Inbox or other folders.

Creating and Sending Message



To: List-Universe <lu@lists.new-list.com>

To:

Re: Welcome to the New-List.com Humor List

```
> Email Discussion Lists:
> -----------------------
> EPUB - mailto:join-epub@list-universe.com
> List-Etiquette -
mailto:join-list-etiquette@list-etiquette.com
> List-Moderators -
mailto:join-list-moderators@list-moderators.com
>
>
>
>
>
```

Ready.

Figure G Message Composition Window

Use the message composition window to address, compose, and send email.

The message composition window contains the following buttons :

- Send: Send a completed message.
- New: To compose a new message
- Load or Save Message: To Load, Delete or Save Message

To save the current message as message draft in the message composition window, click **Load or Save Message** button and choose **Save** from the popup menu.

The saved message(s) will be appended to the bottom of popup menu.

To load the saved message, click on the message list in the popup menu.

Addressing a Message

1. Type the name in the address field.
2. The system have the function Addressing Auto-Completion embedded, type the first few letters of the recipient's name and wait for system to complete the address. Or you can type the name and immediately press Enter to have system try to complete the address
3. Click on the Address Book button to search for and select names in your address books or directories.

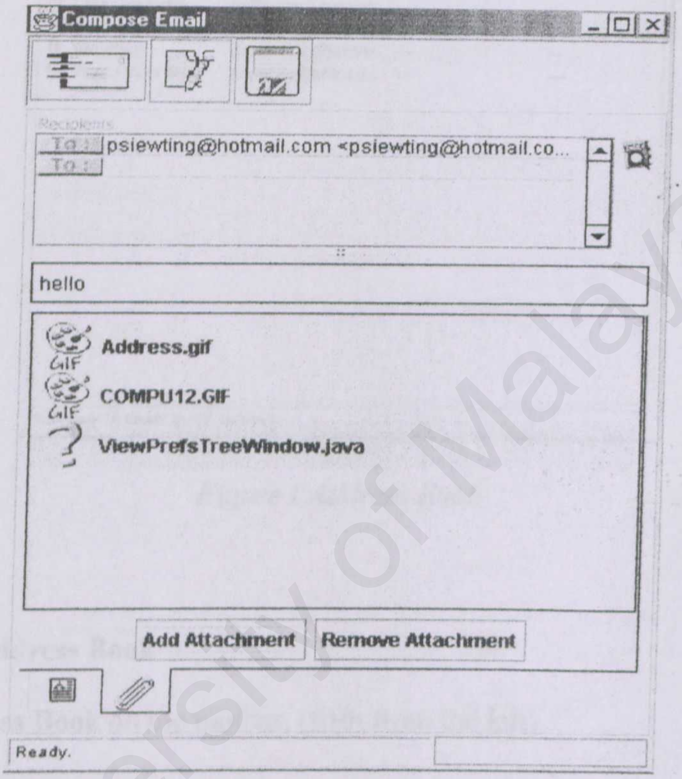
To address a message to names listed in the search results window:

1. Select the name and then click To, Cc, or Bcc.
2. Click OK when you've finished selecting names.
3. You see a message composition window with the selected names in the address field.
4. If necessary, click "To:" to choose a different recipient type after you enter the recipient email address.

To:	Primary recipients of your message.
CC:	Carbon copy, for secondary recipients.
BCC:	Blind carbon copy, for secondary recipients not identified to the other recipients, including those in the Cc list.

Sending Attachment

Click on the attachment tab, the attachment pane displays the attachment that is being attached to your email message. You can add or remove attachment using **Add Attachment** and **Remove Attachment** Button.



To : psiewting@hotmail.com <psiewting@hotmail.co...

To :

hello

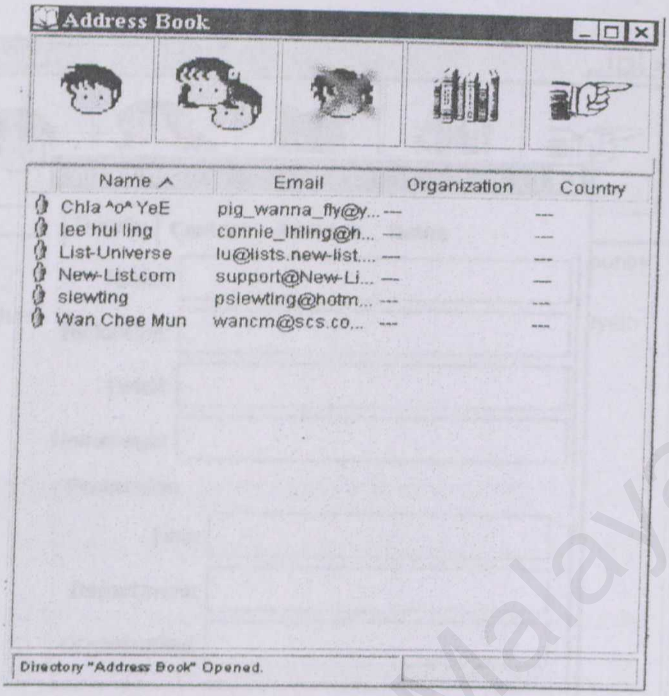
Address.gif

COMPU12.GIF

ViewPrefsTreeWindow.java

Figure H Email Composition Window – Sending Attachment

Using Address Book



Name ^	Email	Organization	Country
Chia ^o^ YeE	pig_wanna_fly@y...	---	---
lee hui ling	connie_lhling@h...	---	---
List-Universe	lu@lists.new-list....	---	---
New-List.com	support@New-Li...	---	---
siewting	psiewting@hotm...	---	---
Wan Chee Mun	wancm@scs.co...	---	---

Directory "Address Book" Opened.

Figure 1 Address Book

To Open the Address Book

Click the **Address Book** on the toolbar, (fifth from the left)

To Open the Address Book from within an Email Editor Window

Click the **Address Book** icon in Email Editor Window.

To Add Names Directly from Email Messages to Your Address Book

Click on the **Option** button at the up-right corner of the preview pane. Click **Add Sender** or **Add Recipient(s)** from the popup menu.

Adding a New Contact to Your Address Book

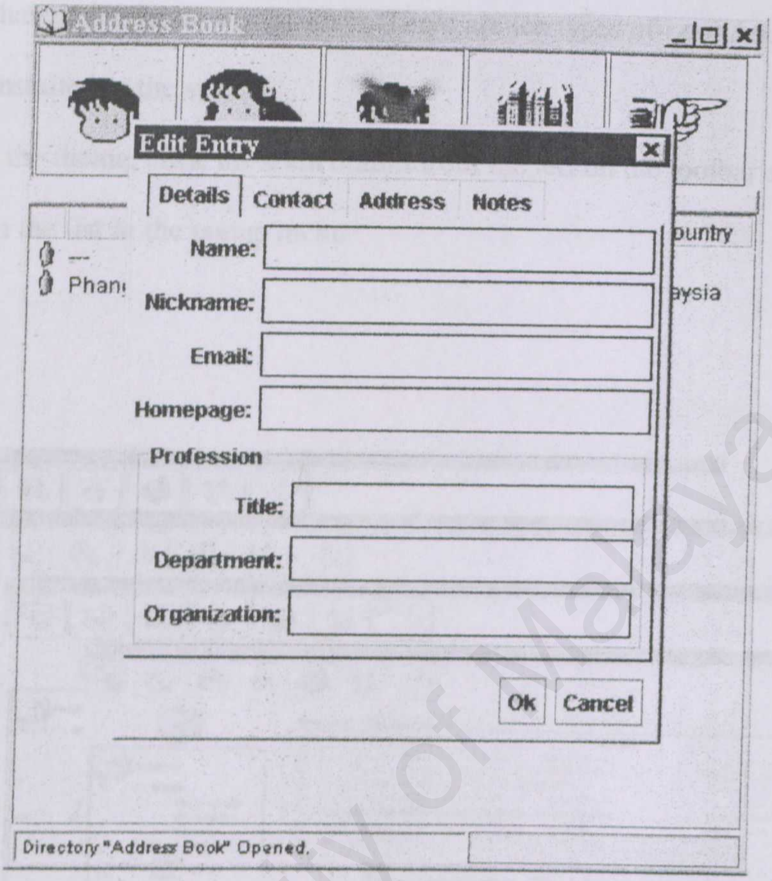


Figure J Address Book-Add A New Contact

Click on the first button from the left in the Address Book Window.

An Edit Entry Window will be prompt out. Insert the details for an individual, include the Name, Nickname, Email Address, Homepage, Profession, Contact, Address and Notes. Click Ok to add the entry to your address book.

Deleting a Name from Your Address Book

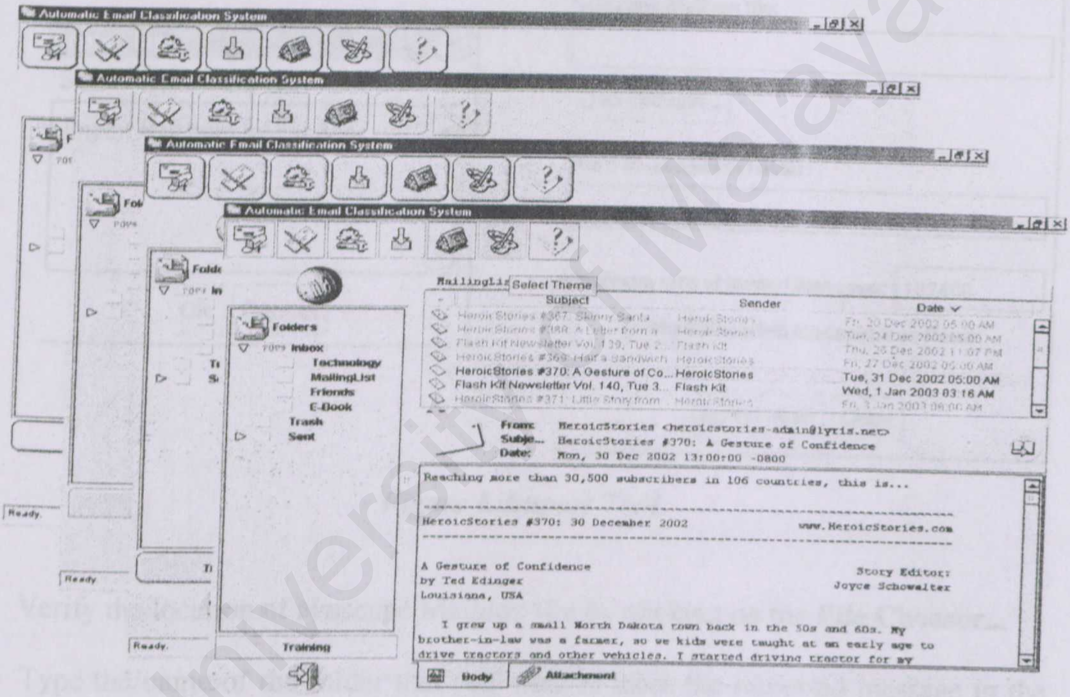
Select the name that your want to delete in your address book.

Click the delete button (Second from the left)

Selecting Theme for Your User Interface

You can change the theme as you wish. There are ten types of LookAndFeel currently installed in the system.

To change the theme, click the sixth button from the left on the toolbar and select the theme from the list in the popup menu.


Automatic Email Classification System
Automatic Email Classification System
Automatic Email Classification System
Automatic Email Classification System
MailingLis
Select Theme
Folders
Inbox
Technology
MailingList
Friends
E-Book
Trash
Sent
Training
Ready
Ready
Ready
Body
Attachment

Subject	Sender	Date
HeroicStories #367: Skinny Santa	HeroicStories	Fri, 20 Dec 2002 05:00 AM
HeroicStories #368: A Letter from a	HeroicStories	Tue, 24 Dec 2002 05:00 AM
Flash Kit Newsletter Vol. 139, Tue 2...	Flash Kit	Thu, 26 Dec 2002 11:07 PM
HeroicStories #369: Half a Sandwich	HeroicStories	Fri, 27 Dec 2002 05:00 AM
HeroicStories #370: A Gesture of Co...	HeroicStories	Tue, 31 Dec 2002 05:00 AM
Flash Kit Newsletter Vol. 140, Tue 3...	Flash Kit	Wed, 1 Jan 2003 03:16 AM
HeroicStories #371: Little Story from ...	HeroicStories	Fri, 3 Jan 2003 05:00 AM

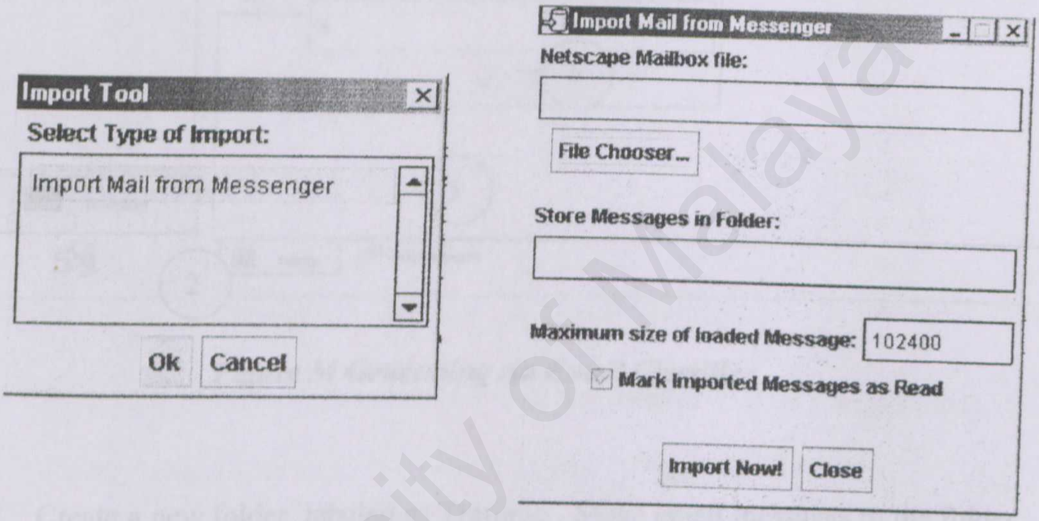
```
From:     HeroicStories <heroicstories-admin@lyris.net>
Subje...  HeroicStories #370: A Gesture of Confidence
Date:     Mon, 30 Dec 2002 13:00:00 -0800
```

```
Reaching more than 30,500 subscribers in 106 countries, this is...

------------------------------------------------------------------
HeroicStories #370: 30 December 2002           www.HeroicStories.com
------------------------------------------------------------------

A Gesture of Confidence                                Story Editor:
by Ted Edinger                                      Joyce Schowalter
Louisiana, USA

   I grew up a small North Dakota town back in the 50s and 60s. My
brother-in-law was a farmer, so we kids were taught at an early age to
drive tractors and other vehicles. I started driving tractor for my
```

Figure K Variety of Themes for User Selection

Import Messages from Other Mail Programs

1. Using the import tool, you can easily import mail messages from Netscape Messenger.
2. On the toolbar, click **Import Tool** button.
3. Select the e-mail type you want to import messages from (So far the system only support import messages from Netscape Messenger).



Select Type of Import: Import Mail from Messenger

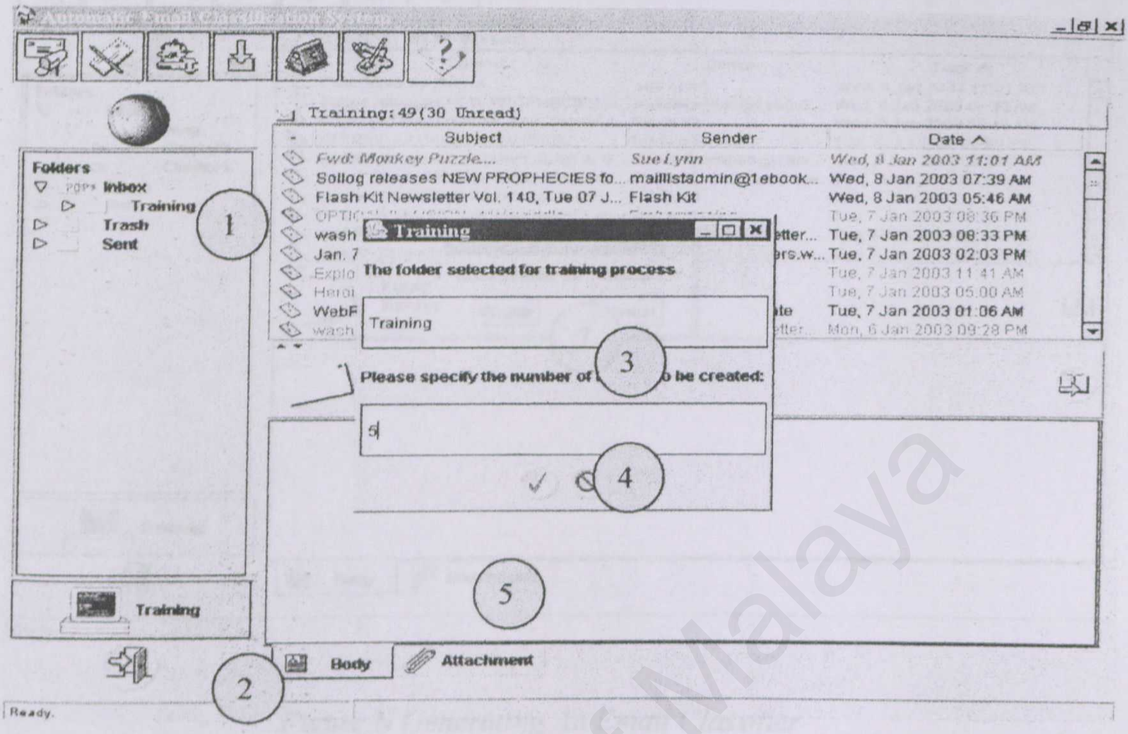
Maximum size of loaded Message: 102400

☑ Mark Imported Messages as Read

Figure L Import Tool

4. Verify the location of Netscape Mailbox file by clicking on the **File Chooser...**
5. Type the name of the folder that you wish to store the imported message in the **Store Messages** in Folder text box.
6. Type the maximum size of load Message, the default value is 1024000
7. Check **Mark imported Messages as Read** if you want the imported messages to be marked as read.
8. Click **Import Now!**.

Generating An Email Classifier



Training: 49(30 Unread)

Subject	Sender	Date ^
Fwd: Monkey Puzzle....	Sue Lynn	Wed, 8 Jan 2003 11:01 AM
Sollog releases NEW PROPHECIES fo...	maillistadmin@1ebook...	Wed, 8 Jan 2003 07:39 AM
Flash Kit Newsletter Vol. 140, Tue 07 J...	Flash Kit	Wed, 8 Jan 2003 05:46 AM
OPTIC		Tue, 7 Jan 2003 08:36 PM
wash	etter...	Tue, 7 Jan 2003 08:33 PM
Jan. 7	ers.w...	Tue, 7 Jan 2003 02:03 PM
Explo		Tue, 7 Jan 2003 11:41 AM
Heroi		Tue, 7 Jan 2003 05:00 AM
WebF	te	Tue, 7 Jan 2003 01:06 AM
wash	etter...	Mon, 6 Jan 2003 09:28 PM

Training

The folder selected for training process

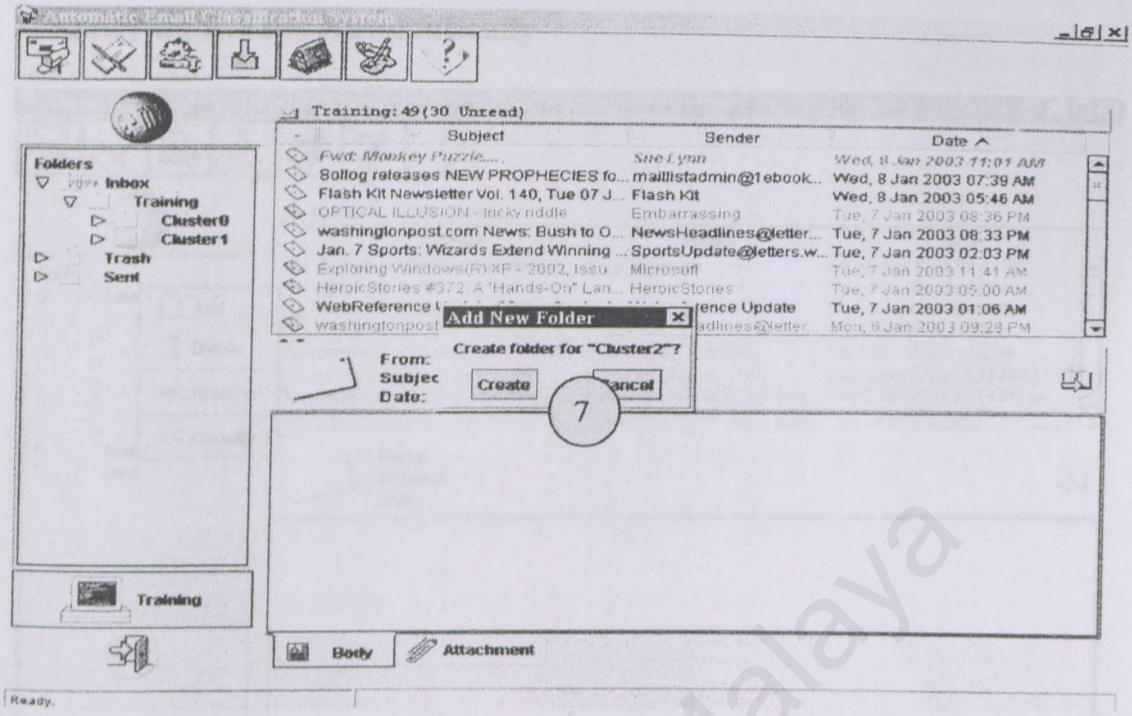
Training

Please specify the number of o be created:

5

Figure M Generating An Email Classifier

1. Create a new folder, labeled as Training. Move email messages to the folder to construct a training data set. The number of emails in training data set is unlimited. But the more training emails you use, the more accurate the classifier generated. It is recommended to insert at least 100 emails for training purpose
2. Click the **Training** button.
3. A training dialog will be prompt out
4. Type the name of your training folder
5. Type the number of folder that you want to be created from the training folder
6. Click the '**Tick**" button



Training: 49 (30 Unread)

Subject	Sender	Date ^
Fwd: Monkey Puzzle....	Sue Lynn	Wed, 8 Jan 2003 11:01 AM
Sollog releases NEW PROPHECIES fo...	maillistadmin@1ebook...	Wed, 8 Jan 2003 07:39 AM
Flash Kit Newsletter Vol. 140, Tue 07 J...	Flash Kit	Wed, 8 Jan 2003 05:46 AM
OPTICAL ILLUSION - tricky riddle	Embarrassing	Tue, 7 Jan 2003 08:36 PM
washingtonpost.com News: Bush to O...	NewsHeadlines@letter...	Tue, 7 Jan 2003 08:33 PM
Jan. 7 Sports: Wizards Extend Winning ...	SportsUpdate@letters.w...	Tue, 7 Jan 2003 02:03 PM
Exploring Windows(R) XP - 2002, Issu...	Microsoft	Tue, 7 Jan 2003 11:41 AM
HeroicStories #372: A "Hands-On" Lan...	HeroicStories	Tue, 7 Jan 2003 05:00 AM
WebReference U	ence Update	Tue, 7 Jan 2003 01:06 AM
washingtonpost	adlines@letter...	Mon, 6 Jan 2003 09:28 PM

Add New Folder

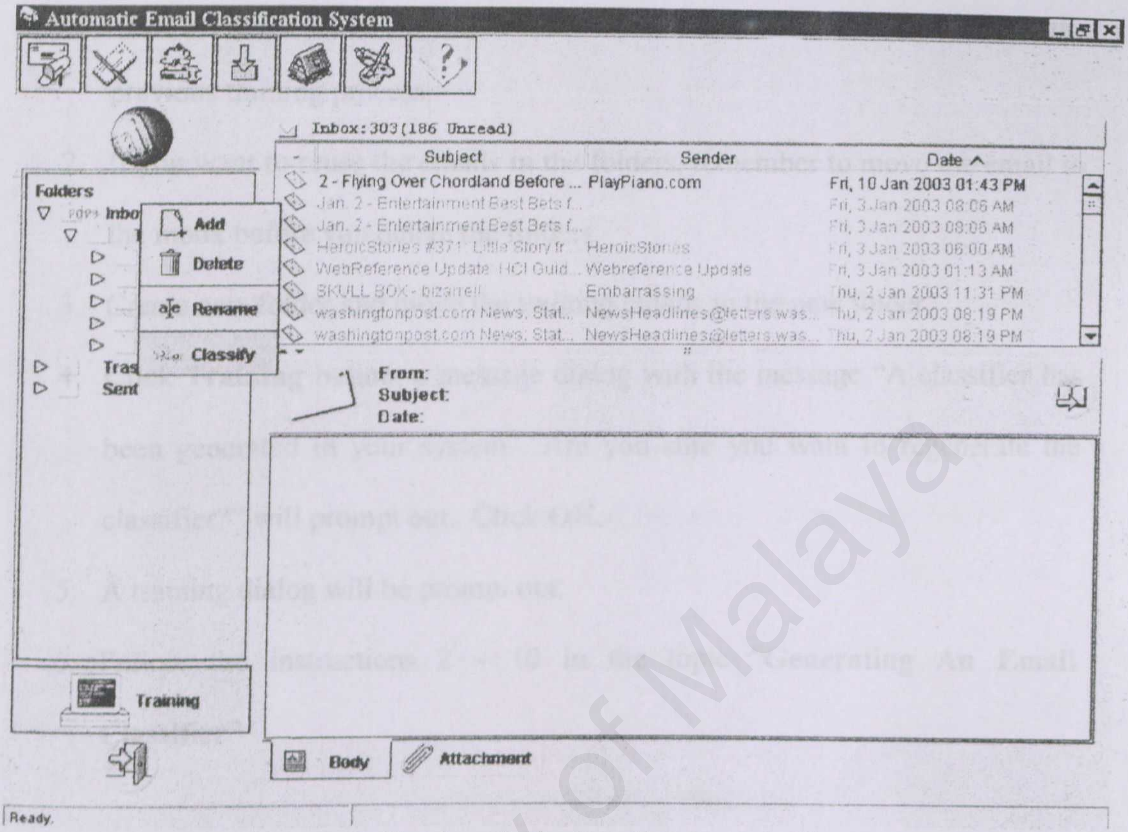
Create folder for "Cluster2"?

Create | Cancel

Figure N Generating An Email Classifier

7. A message will prompt out to confirm that folders will be created for each cluster, with default name "Cluster0, Cluster1, Cluster2…"
8. Click **Create** to start the training process.
9. A message dialog will be prompt out to tell you that an email classifier is generated when the training process is completed
10. You will find that all the emails in the training folder have been classified into the "Cluster0, Cluster1, Cluster2…" folders.

Classifying Emails Automatically



Subject	Sender	Date ^
2 - Flying Over Chordland Before ...	PlayPiano.com	Fri, 10 Jan 2003 01:43 PM
Jan. 2 - Entertainment Best Bets f...		Fri, 3 Jan 2003 08:06 AM
Jan. 2 - Entertainment Best Bets f...		Fri, 3 Jan 2003 08:06 AM
HeroicStories #371: Little Story fr...	HeroicStories	Fri, 3 Jan 2003 06:00 AM
WebReference Update: HCI Guid...	Webreference Update	Fri, 3 Jan 2003 01:13 AM
SKULL BOX - bizarre!!	Embarrassing	Thu, 2 Jan 2003 11:31 PM
washingtonpost.com News, Stat...	NewsHeadlines@letters.was...	Thu, 2 Jan 2003 08:19 PM
washingtonpost.com News, Stat...	NewsHeadlines@letters.was...	Thu, 2 Jan 2003 08:19 PM

Figure O To classify the Email Automatically

1. After the email classifier is generated, right-click on the folder in Folders Tree panel, and click **Classify.**
2. Or you can also classify the emails "on the fly" when the emails downloaded from the server. Check the "Classify New Mail" checkbox in the Incoming Email Server Configuration. The incoming emails will be classified into the corresponding folders without go through the Inbox.

Re-generate the Classifier

1. To regenerate the email classifier, delete all the folders that created in the previous training process.
2. If you want to reuse the emails in the folders, remember to move the email to the inbox before you delete the folders.
3. Create new folder and move the training emails to the new folder.
4. Click **Training** button, a message dialog with the message "A classifier has been generated in your system. Are you sure you want to regenerate the classifier?" will prompt out. Click **OK.**
5. A training dialog will be prompt out.
6. Follow the instructions 2 – 10 in the topic "**Generating An Email Classifier**"

REFERENCES

[23m.com, 2002] *http://www.123m.com/pop3email.asp*

[AmikaNow.com , 2002] *tp://www.amikanow.com/corporate/email_overload.htm*

[ARFF, 2002] Attribute-Relation File Format *http://www.cs.waikato.ac.nz/~ml/weka/arff.html*

[Bahrami, 1999] Bahrami, A. (1999) *Object Oriented Systems Development.* Irwin/McGraw-Hill, Singapore

[Boetriger, 2002] Boetriger.A, (2002) What can I do about Spam? *http://www.email911.com*

[Brutlag & Meek, 2002] Brutlag, J.D and Meek,C.(2002) Challenges of the Email Domain for Text Classification. In *Proc. of the 17th International Conference on Machine Learning*, pages 103–110, Stanford University, USA

[Cohen, 1996] Cohen, W.W (1996). Learning rules that classify e-mail. In *Papers from the AAAI Spring Symposium on Machine Learning in Information Access.*

[Deneubourg,1990] Deneubourg, J. L., Goss, S., Franks, N.R., Sendova-Franks, A., Detrain,C. & Chretien, K.(1990) The Dynamics of Collective Sorting: Robot-like Ants and Ant-Like Robots. *In Meyer, J-A, & Wilson, S., Eds, Simulation of Adaptive Behaviour: From Animals to Animals, MIT Press, Massachusetts*, 356-365

[Email, 2000] Introduction to Email : University of Birmingham *http://www.bham.ac.uk/is/email/intro.shtml*

[Faber,1994] Faber, V. (1994) Clustering and the Continuous *k*-Means Algorithm, *Los Alamos Science,* number 22.

[Han & Kamber, 2001 Han, J. and Kamber, M.(2001) *Data Mining: Concepts and Techniques*. Morgan Kaufmann,. 7, 284-301.

[IDC, 2001] International Data Corporation (IDC) Email Usage Forecast and Analysis, 2001-2005 *http://www.idc.com/getdoc.jhtml?containerId=25335*

[IR, 1997] Glossary for Information Retrieval, 1997.
http://www.cs.jhu.edu/~weiss/glossary.html

[Itskevitch, 1997] Itskevitch, J(1997) *Automatic Hierarchical Email Classification Using Association Rule.* B.Sc thesis. Belorussian State Polytechnic Academy
http://www.cs.sfu.ca/~itskevit/personal/thesis

[JavaMail, 2001] Fundamental of JavaMail API
http://developer.java.sun.com/developer/onlineTraining/JavaMail

[Kiritchenko & Matwin, 2002] Kiritchenko, S. and Matwin, S. (2002) *EmailClassification with Co-Training*, University of Ottawa

[K-Means,1999] *http://obelia.jde.aca.mmu.ac.uk/multivar/kmeans.htm*

[Korfhage & Robert, 1997] Korfhage and Robert R., (1997) *Information Storage and Retrieval,* John Wiley & Sons, Inc, Canada

[Lemos, 2002] Lemos, R. (2002) Spam hits 36 percent of email traffic (Article at August 29, 2002)
http://zdnet.com.com/2100-1106-955842.html

[Mitchell, 1996] Mitchell, T.M. (1996). *Machine Learning.* McGraw Hill, New York

[Netscape, 1999] Netscape.com (1999) How can I filter unsolicited commercial mail("spam") *http://help.netscape.com/kb/consumer/19990412-7.html*

[Ong, Seah, Sundraraj, 2001] Ong, C. S., Seah C. Y. M and Sundraraj, K. (2001), Classification Tool for Email Filtering, *Proceeding of the Third MIMOS R&D Symposium on ICT and Microelectronics.* Kuala Lumpur, Malaysia

[Pastore1, 2001] Pastore, M.(2001) E-Mail Continues Dominance of Net Apps

http://cyberatlas.internet.com/big_picture/applications/article/0,1323,1301_808741,00.html

July 25, 2001 cyberatlas.com

[Pastore2, 2001] Pastore, M. (2001) More Mailboxes on the Way *http://cyberatlas.internet.com/big_picture/applications/article/0,,1301_885551,00.html*

September 17, 2001

[Porter, 1980] Porter, M.F (1980) *An algorithm for suffix stripping Program*, Vol.14 (3), 130-137.

[Provost, 1999] Provost, J. (1999) Naive Bayesian VS rule- learning in classification of email

[Ramos, 2002] Ramos. V (2002) Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. *Submitted to ICEIS'2002,*Ciudad Real, Spain.

[Rational, 2002] Rational Unified Process:Best Practices for Software Development Teams (2002) *http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.html*

[Scott, 2001] Scott, K(2001) *UML Explained.* Addison-Wesley

[Sebastiani, 2002] Sebastiani F. (2002). Machine Learning in Automated in Text Categorization , *ACM Computing Surveys,* **Vol.34,**No.1(March) *http://faure.iei.pi.cnr.it/~fabrizio/*

[Sommerville, 1996] Sommerville, I.(1996) *Software Engineering.* 5th Ed, AddisonWesley Publishing Company.

[Swarm, 2001] Swarm Intelligence
http://imv.au.dk/~thomasr/Swarm/swarm.htm

[Turner, 2001] Turner, R. (2001) Congressional Email Overload
http://www.techsoup.org/articlepage.cfm?ArticleId=288&topicid=5

[WEKA , 2002] *http://www.cs.waikato.ac.nz/~ml/*

[Whatis.com Inc., 1996] *http://www.whatis.com/cgi-bin/redirectexe/java*

[Witten & Frank, 1999] Witten, I.H.and Frank,E (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann
http://www.cs.waikato.ac.nz/ ml/weka/.