

ASSIGNMENT BOX ALERT FOR STUDENT (ABA)

AZIZUL BIN AZIZ

A report submitted in partial fulfilment
of the requirements for the award
of the degree of
Bachelor Of Computer Science (Computer Systems & Networking)

Faculty of Computer Systems & Software Engineering
University Malaysia Pahang

MAY 2011

PERPUSTAKAAN UNIVERSITI MALAYSIA PAHANG	
No. Perolehan 068656	No. Panggilan QA 76.9
Tarikh 30 NOV 2012	.A95 2011 r5 8c.

ABSTRACT

Client-server is a network model for computer networking that utilizes client and server devices each designed for specific purpose. The client-server model can be used on the internet as well as local area network (LAN). Client-server networking grew in popularity many years ago as personal computers (PC) became alternative to older mainframe computers. The client-server model distinguishes between applications as well as devices. Network client make requests to a server by sending messages, and servers responds to their clients by acting on each request and returning results. Before this, the submission assignments from the students were manually submitted using the hardcopy (printing material) or by softcopy (Compact Disc (CD)), which is taking time and very easily being damaged or misplace. In this project, this client-server application will help the student in their assignment submission. It also will make the submission faster than ever. Therefore, it will make the data of the participant can be view more systematically by the lecturer. Besides that, with the notification system, the student can alert the uploaded assignment by their lecturer.

ABSTRAK

Client-server model rangkaian ialah rangkaian komputer yang menggunakan peranti client dan server dimana masing-masing direka untuk tujuan tertentu. Model client-server boleh digunakan pada internet serta rangkaian kawasan tempatan (LAN). client-server rangkaian wujud dalam populariti bertahun-tahun yang lalu sebagai komputer peribadi (PC) menjadi alternatif untuk komputer mainframe yang lebih tua. Model client-server adalah berbeza antara aplikasi dan peranti. Rangkaian client membuat permintaan ke server dengan menghantar mesej dan client merespon server mereka dengan bertindak berdasarkan permintaan masing-masing dan mengembalikan hasil. Sebelum ini, penyerahan tugas dari pelajar disampaikan secara manual dengan menggunakan hardcopy (bahan cetak) atau dengan softcopy (Compact Disc (CD)), yang memakan masa dan sangat mudah menjadi rosak atau salah letak. Dalam projek ini, aplikasi client-server akan membantu pelajar dalam penyerahan tugas mereka. Ini juga akan membuat permohonan lebih cepat dari sebelumnya. Oleh kerana itu, akan membuat data peserta dapat melihat secara lebih sistematik oleh pensyarah. Selain itu, dengan sistem notifikasi, pelajar dapat waspada tugas dimuaturun oleh pensyarah mereka.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	SUPERVISOR'S DECLARATION	
	TITLE PAGE	
	DECLARATION	
	DEDICATION	
	ACKNOWLEDGEMENT	
	ABSTRACT	
	ABSTRAK	
	TABLE OF CONTENTS	
	LIST OF TABLES	
	LIST OF FIGURES	
	LIST OF APPENDICES	
1	INTRODUCTION	1
	1.1 Introduction	2
	1.2 Problems Statements	3
	1.3 Objectives	3
	1.4 Scope	3
	1.5 Thesis Organization	4
2	LITERATURE REVIEW	5

	2.1 Introduction	6
	2.2 Concept Use In ABA Application	7
	2.3 Client-server architectures	9
	2.3.1 Two-tier client-server architecture.	9
	2.3.2 Two-tier limitations	11
	2.3.3 Three-tier	11
	2.4 Notification system	13
	2.5 Remote Method Invocation (RMI) for ABA system	15
	2.6 Comparisons between Remote Method Invocation (RMI) and Web services (WS)	17
3	METHODOLOGY	19
	3.1 Introduction	20
	3.2 Method	21
	3.3 RAD overview	24
	3.4 ABA system	25
	3.5 Flow chart ABA system	26
	3.6 Data Flow Diagram	37
	3.7 RMI Architecture	42
	3.7.1 The Stub/Skeleton Layer	44
	3.7.2 The Remote Reference and Transport layer	45
	3.8 Requirement Technology	46
	3.9 Gantt Chart	47
	3.10 Method and Technology justification.	48
	3.10.1 Advantages of RAD Software Development.	48

	3.10.2 Disadvantages of RAD Software Development.	49
4	IMPLEMENTATION	50
	4.1 Introduction	51
	4.2 ABA Notification System (lecturer)	52
	4.3 ABA Notification System (student)	62
	4.4 ABA File Transfer System (student)	64
	4.5 ABA File Transfer System (lecturer)	71
	4.6 RMI Stub and Skeleton	74
	4.7 Summary	78
5	DISCUSSION AND RESULT	79
	5.1 Result analysis	80
	5.1.1 Advantages of the system	80
	5.2 Constrain	81
	5.2.1 Development constraints	81
	5.2.2 System Constraints	81
	5.3 Suggestion on to Improve the System	82
	5.4 Summary	83
6	CONCLUSIONS	84
	REFERENCE	86
	APPENDIX A - D	88 - 93

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	Differential between two-tier and three-tier	12
2.2	The different between RMI and WS that use in distributed client-server application	17

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Dynamic classes loading from client-server wiki that use in RMI method	7
2.2	Architecture of client-server that took from journal Distributed Application Architecture	10
2.3	Three-tier architecture took from client-server Wikipedia.	11
2.4	Email Announcers use by Blind Bat Software.	13
2.5	Email Announcer by Blind Bat Software – agent interface and configuration environment.	14
2.6	The basic architecture of RMI client-server from journal <i>a performance evaluation of rpc, java rmi, mpi and pvm,</i>	15
2.7	Four-layer RMI system from journal <i>A Meteorological Data Distribution System Using Remote Method Invocation Technology.</i>	16
3.1	Rapid Application Developments life cycle	22

illustrate from Wikipedia.

3.2	The Flow chart for ABA notification (student).	26
3.3	The flow chart for notification (lecturer).	28
3.4	The flow chart for ABA file transfer (lecturer).	31
3.5	The flow chart for ABA file transfer (student).	33
3.6	The DFD diagram level 0 for the ABA Notification System.	38
3.7	The DFD diagram level 1 for the ABA Notification System.	39
3.8	The DFD diagram level 0 for the ABA File Transfer system.	40
3.9	The DFD diagram level 1 for the ABA File Transfer system.	41
3.10	RMI architecture took from journal <i>RMI architecture</i> .	43
4.1	The Interface of the IDE tool use.	51
4.2	The login interface for ABA notification system for lecturer.	52
4.3	modules that responsible to call out login function is in highlight blue color.	53
4.4	The main interface in ABA notification for lecturer.	55
4.5	The step on how to connect to the user (student) through the address book in the system.	57

4.6	The address book when the lecturer wants to add more users.	59
4.7	The directory that the lecturer have to access to execute the notification application.	60
4.8	The login interface for the student system.	62
4.9	The ABA notification system for student.	63
4.10	The system is connected with lecturer system.	63
4.11	The login interface for ABA files transfer (student).	64
4.12	ABA File Transfer main page.	66
4.13	address book use in ABA system.	67
4.14	add user form.	67
4.15	The login interface for lecturer.	71
4.16	The main page of the server.	73
7.0	The estimate time and schedule in documenting the project.	88
7.1	The estimate time and schedule in developing system for PSM 2.	89
7.2	The form that will be shown in student computer for the notification system.	92
7.3	The acceptance letter from World Conference Educational Technology Researches (WCETR) for the ABA proposal.	93

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Gantt chart for PSM I and PSM II	88
B	How to install and use ABA.	99
C	Notification Application Form.	92
D	Acceptance letter from World Conference Educational Technology Researches (WCETR) for the ABA proposal.	93

CHAPTER 1

INTRODUCTION

This chapter briefly discusses about introduction for the purpose project. There are five main sections in this chapter. The first main section is introduction of this chapter. Then, the next main section describes problems statements. After problems statements, the next section will briefly discuss about the objectives for this project to achieve. Next is the scope for this project. The last main section reviews the thesis organization for each chapter in this report. The thesis organization will discuss briefly about each chapter in this report.

1.1 INTRODUCTION

Client-server is a network model for computer networking that utilizes client and server devices each designed for specific purpose [1]. The client-server model can be used on the internet as well as local area network (LAN). Client-server networking grew in popularity many years ago as personal computers (PC) became alternative to older mainframe computers. The client-server model distinguishes between applications as well as devices. Network client make requests to a server by sending messages, and servers responds to their clients by acting on each request and returning results. A client computer and server are usually two separate devices, each customized for their designed purpose [2]. The projects propose file transferring between student and lecture. It is based on the client-server model application. With this system, lecture can sent any exercise or assignment to the student based on the client-server model. Both the lecture and student have to install the system into their Personal Computer or Note Book so that both can communicate with each other as well transfer file. When student received any file from their lecture, the system will alert the student by some of reminder system in the system. For this system, we will implement on the use of Remote Method Invocation (RMI) as a method for communication between server and client. The RMI architecture is based on one important principle; the definition of behaviour and the implementation of that behaviour are separate concepts. RMI allows the code that defines the behaviour and the code that implements the behaviour to remain separate. The RMI implementation is essentially built from three abstraction layers. The first is the Stub and Skeleton layer, which lies just beneath the view of the developer. This layer intercepts method calls by the client to the interface reference variable and redirects these calls to a remote RMI service. The next layer is the Remote Reference Layer. This layer understands how to interpret and manage references made from client to the remote service objects. The stub and skeleton layer of RMI lie just beneath the view of the Java developer. The stub class plays the role of the proxy. A skeleton is a helper class that is generated for RMI to use. The skeleton understands how to communicate with the stub across the RMI link. The skeleton carries on a conversation with the stub.

1.2 PROBLEMS STATEMENTS

- i. Some student misplaced their assignment due to manual system.
- ii. Some of the hard copy that student sent which mostly by paper or printing material might easily rip up.
- iii. The using of email which is not reliable due to connection problems as well limitations on file size and type of file that can be submit or transfer causes the use of email is not efficient for file transfer.

1.3 OBJECTIVE

- i. Develop Assignment submission system based on the client-server model application.
- ii. To apply Remote Method Invocation (RMI) in developing the system.

1.4 SCOPE

- i. The users of this system are UMP (University Malaysia Pahang) lectures and UMP FSKKP last year students.
- ii. Using campus LAN network.
- iii. System builds up based on RMI method.
- iv. Using windows as operating system.
- v. Transfer word file, data file, image file, video file, web file and compressed files.

1.5 THESIS ORGANIZATION

For this project, there are consists of five chapter. Chapter 1 will discuss on introduction to the system or research briefly. The discussion includes the introduction, problems statement, objective, and scope. Chapter 2 describes the literature review. In Chapter 2, it describe the review of existing system that we will refer when develop the system. The discussion will be include the manual system, concept of the Assignment submission system, existing system, technology of client-server, technology use in RMI and its extension, and the comparisons between different system. Meanwhile Chapter 3, it is about the methodology or the requirement for the whole system that we will develop in future. For methodology, there are introduction on how the project will be executed, project methodology, and reason choosing the methodology, equipment and software use in this project. Chapter 5 discuss on the conclusion based on the project implementation, planning, result and important information about the project.

CHAPTER 2

LITERATURE REVIEW

This chapter briefly discusses about the literature review of Assignment Box Alert (ABA) using client-server model based on Remote Method Invocation (RMI). There are seven main sections in this chapter. The first main section is introduction of this chapter. Then, the next main section describes the concept and method use to develop the system. After introduction, the next section will briefly discuss about the notification system. Next, is the comparison between method uses to in the client-server model. The next main section discusses the existing system that using Peer-To-Peer (P2P) model and client-server as well the comparison between them. While the last main section reviews the methodologies used to develop the application.

2.1 Introduction

In this section, a brief review about the introduction of the Assignment Box Alert (ABA) and concept using for this system. As technology getting advance, the using of manual system in assignment submission will slow down work and mark evaluation. For that, this section will discuss on the new system to replace the manual system and the advantages and disadvantages of the system. The ABA system will use the client-server model as the distribute model. The client-server model is a distributed application structure that partitions workloads between the provider of resource and resource requesters. This provider resource and resource requester also know as server and client. Usually clients and server communicate over computer network on separate hardware but both server and client may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. The client-server characteristic describes the relationship between the cooperating programs in an application. Meanwhile the server component provides a function or service to one or many clients, which initiate requests for such services [3]. The method use for the distributed client-server is RMI. RMI is one of the method uses in the distributed network. One of the ABA application function is by sending a notification to student about event or assignment that they received. This notification tells the student that that had received an assignment or task.

2.2 Concept use in ABA application

The concept use in ABA application is client-server model based on RMI method. In RMI client-server architecture, clients can communicate with the remote object only when the server side stub is available with the client. The stub can be loaded on the client's side dynamically by an external web server as shown in Figure. 3 [4].

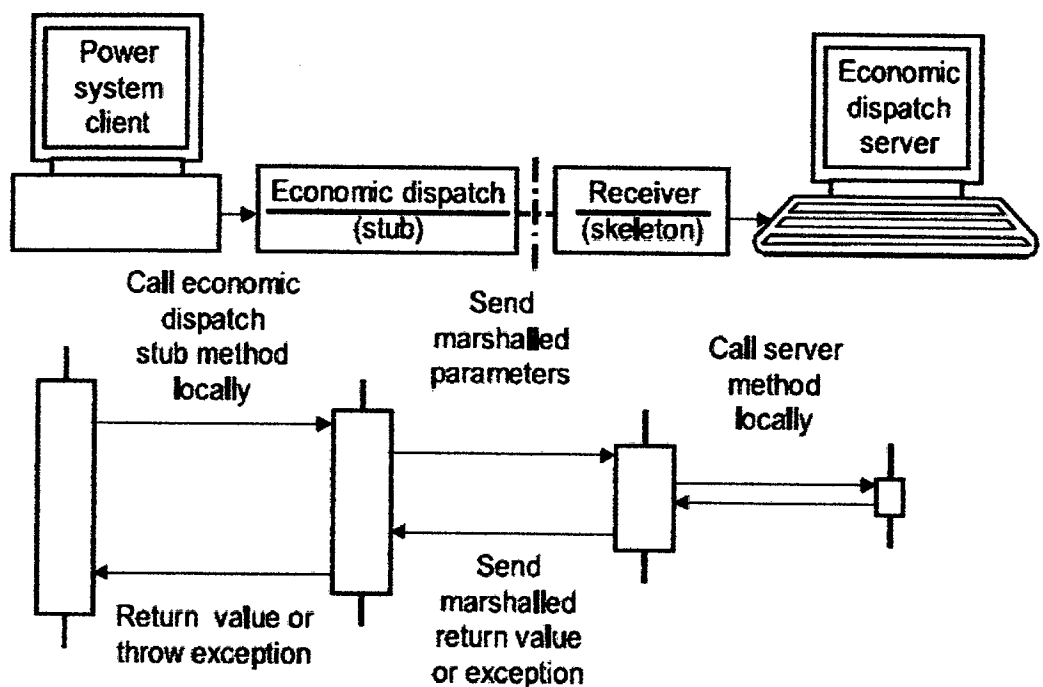


Figure 2.1 Dynamic classes loading from client-server wiki that use in RMI method [4].

The steps involved in downloading RMI stubs are as follows [4]:

- I. The remote object's codebase is specified by the remote object's server by setting the `java.rmi.server.Codebase` property.
- II. The power system client makes a request for a reference to a named remote object. The reference to the remote object's stub instance is what the client will use to make remote method calls to the economic load dispatch server object.
- III. The RMI registry returns the stub instance reference to the requested class.
- IV. The codebase which the power system client uses, is the URL that is annotated to the stub instance when the stub class was loaded by the registry.
- V. The class definition for the stub is downloaded to the client dynamically. Now the power system client has all the information that it needs to invoke remote method on the economic load dispatch server object. The stub instance acts as a proxy to the remote object that exists on the server.

2.3 Client-server architectures

There are basically two major kinds of modern architectures which are two-tier client-server and three-tier also commonly called n-tier. For the ABA application client-server architectures use is two-tier client-server. Each one has many variations. At a high level, these architectures focus on the partitioning system processing. They decide on what machine and in what process space a given bit of code executes. Client-server is often a generic umbrella term for any application architecture that divides processing among two or more processes, often on two or more machines. Any database application is a client-server application if it handles data storage and retrieval in the database process and data manipulation and presentation somewhere else. The server is the database engine that stores the data, and the client is the process that gets or creates the data. The idea behind the client-server architecture in a database application is to provide multiple users with access to the same data [5].

2.3.1 Two-tier client-server architecture

The simplest shape client-server architecture takes is called two-tier architecture. In fact, most client-server architectures are two-tier. The term "two-tier" describes the way in which application processing can be divided in a client-server application. A two-tier application ideally provides multiple workstations with a uniform presentation layer that communicates with a centralized data storage layer. Most Internet applications such as email, Telnet, FTP, gopher, and even the Web are simple two-tier applications.

Without providing a lot of data interpretation or processing, these applications provide a simple interface to access information across the Internet. When most application programmers write their own client-server applications, they tend to follow this two-tier structure. Figure 4 shows how two-tier systems give clients access to centralized data. If you use the Web as an example, the web browser on the client side retrieves data stored at a web server.

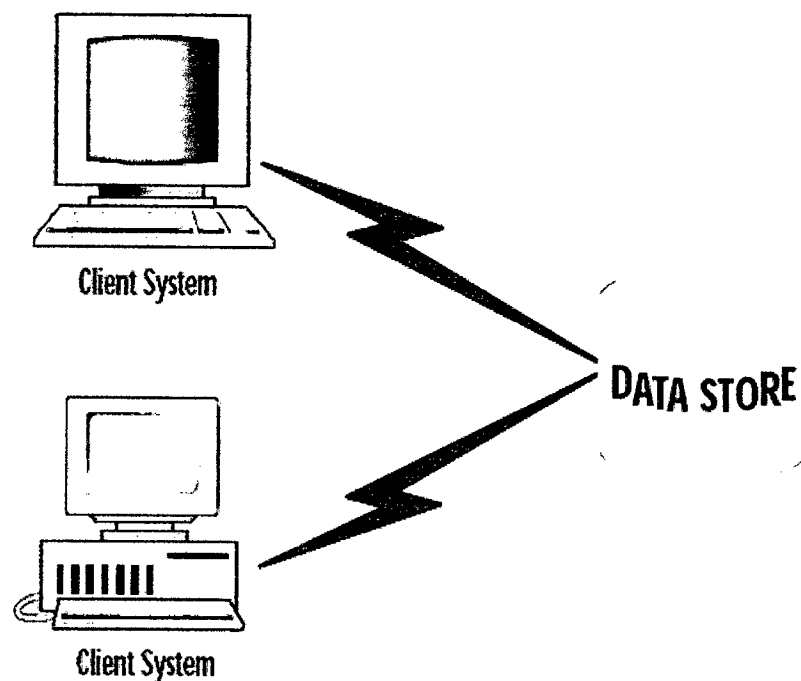


Figure 2.2 Architecture of client-server that took from journal Distributed Application Architecture [5].

2.3.2 Two-tier limitations

One of the problems with two-tier model is limited for processing simple data validation. For complex data validation, it needs to outstrip the abilities of the two-tier architecture. One of the solutions is using fat-clients. Two-tier, fat-client systems are notorious for their inability to adapt to changing environments and scale with growing user and data volume. Even though a client's primary task is the presentation of data, a fat client is loaded with knowledge completely unrelated to that task.

2.3.3 Three-tier

Unlike two-tier, three-tier architecture adds to the two-tier model a new layer that isolates data processing in a central location and maximizes object reuse. Figure 5 shows how this new third layer might fit into application architecture.

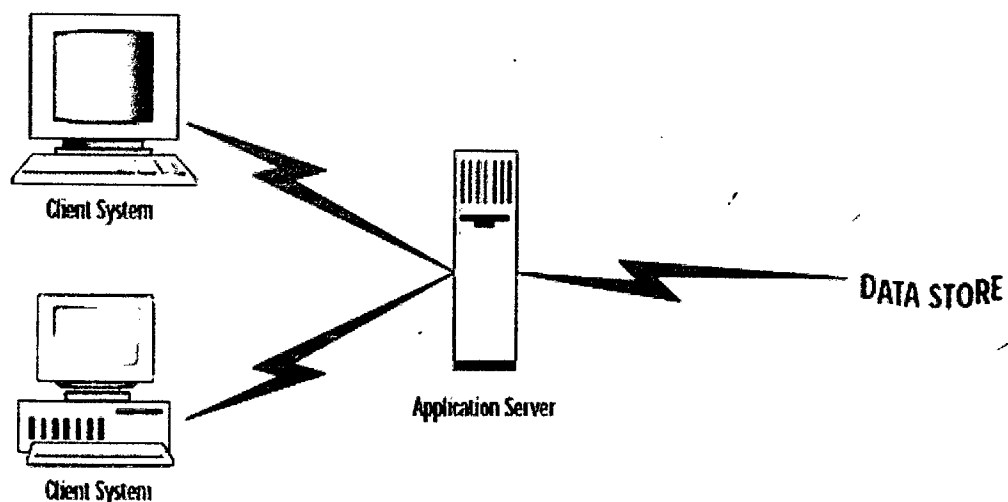


Figure 2.3 three-tier architecture took from client-server Wikipedia [3].

Comparison between two-tier and three-tier**Table 2.1** Differential between two-tier and three-tier

	Two-tier	Three-tier
Architecture	Consist of client system and data store [5].	Consist of client system, application server and data store [5].
Network layer	Logic and data tire are combined [6].	Three different layers like presentation tier, logic tier, and data tier [6].

2.4 Notification system

For this ABA system, the system application is adapt from the email notification system. In this journal [7], the findings indicate that email notification *interface agent users are innovative individuals who have adopted and utilized this technology for a relatively long period of time. With respect to the role of computer playfulness, it is argued that it positively influences a user's perceptions of enjoyment with email interface agents. As theorized, those individuals who tend to interact with computer systems more flexibly, imaginatively, creatively, playfully, originally, and inventively in general tend to enjoy using email interface agents to a greater extent than people with a lower degree of computer playfulness [7].* Figure 4 show some of the email notification interface. Figure 4.1 show the configuration setup for the email notification.

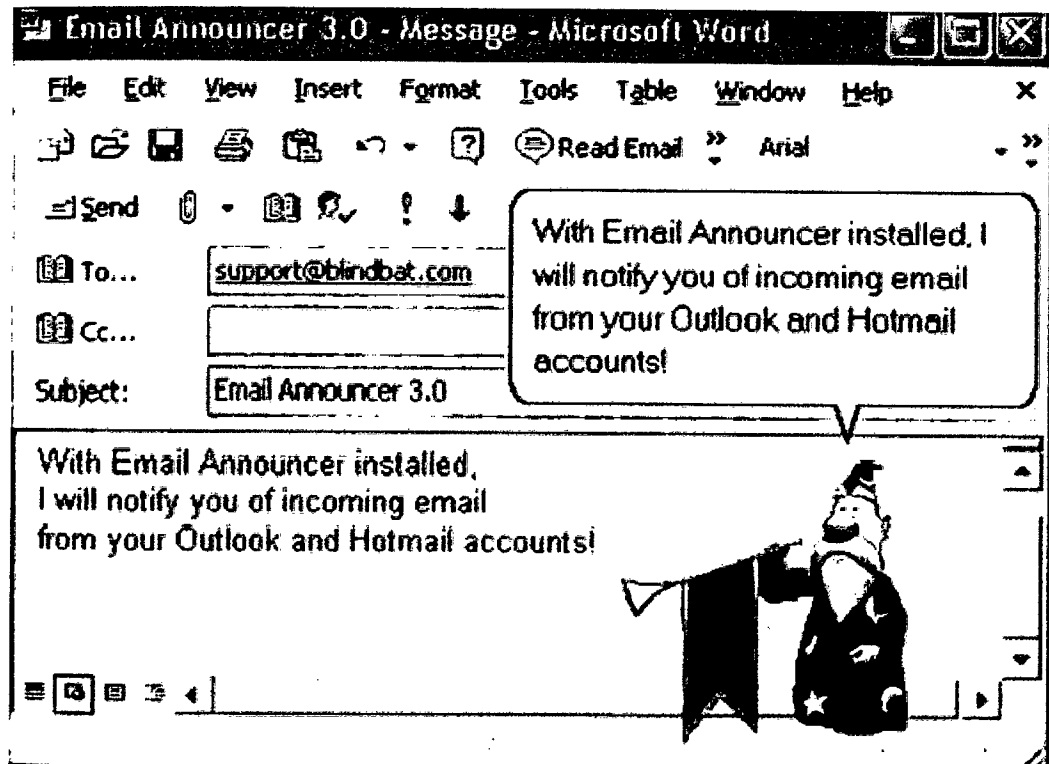


Figure 2.4 Email Announcers use by Blind Bat Software.