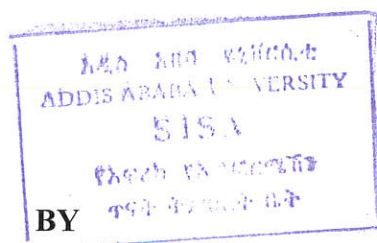


ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

**THE APPLICATION OF INTERFACE AGENT TECHNOLOGY FOR
PERSONALISED INFORMATION FILTERING: THE CASE OF ILRIALERTS**



ABEBE CHEKOL HUNEGNAW

**A THESIS SUBMITTED TO THE GRADUATE STUDIES OF ADDIS ABABA
UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE**

JUNE 2001

ADDIS ABABA UNIVERSITY
LIBRARIES
P.O. BOX 1176
ADDIS ABABA ETHIOPIA

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

**THE APPLICATION OF INTERFACE AGENT
TECHNOLOGY FOR PERSONALISED INFORMATION
FILTERING: THE CASE OF ILRI Alerts**

By

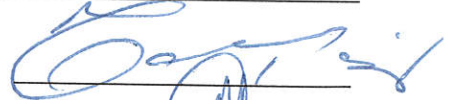
ABEBE CHEKOL

Name and Signature of Members of the Examining Board

Ato Getachew Jemaneh, Chairman, Examining Board



Ato Tesfaye Biru, Advisor



Ato Ermias Abebe, Advisor



Dr. Lars Asker, External Examiner



DEDICATION

to

My parents Wzo Dinkayehu Tafesse

and

Ato Chekol Hunegnaw

ACKNOWLEDGEMENT

First and foremost, my greatest thanks go to my first advisor, Ato Tesfaye Biru without his continuous follow-up, guidance and advice that this thesis would not have reached this stage. Apart from encouraging me to take up this area of research for my research thesis, his unreserved follow-up, keen insight and pushes has a great role on this research. I am also greatly indebted to my second advisor Ato Ermias Abebe for his invaluable time in critically commenting on this research work and his overall advise was so important throughout my thesis work.

I am grateful to all staff members of the ILRI information section particularly to Wzo Azeb, Ato Getachew, Wzo Maria and Wrt Rebecca. I would also like to thank all staff members of the School of Information Studies for Africa (SISA), all classmates and others for their all rounded co-operation and support in the preparation of this thesis and throughout my study at the School.

I am much indebted to the British Council Ethiopia and the Director, Ms Rosemary Arnott for supporting my study at the Addis Ababa University. I also thank my colleagues at the British Council for their support and cooperation in one way or another in the preparation of this thesis.

Last but not least, big thanks go to my family (brothers and sisters) and friends who assisted me in all aspects during my study programme. I thank Kassahun Zewdie, particularly, for quickly responding and acquiring the AgentBuilder toolkit that I used in this thesis.

ABSTRACT

Selective Dissemination of Information (SDI) service is one of the personalised information filtering method of delivering current information based on predefined user profile. However, this service has been found to be one of the resource-intensive and time-consuming both for the service provider and the user. Particularly, in developing countries, it is one of the rarely implemented information service components.

User profile is fundamental to information filtering in SDI systems. However, as previous studies have repeatedly stated, the creation and maintenance of users profiles is the most difficult area. This thesis research investigated into the SDI user profile updating problem domain with particular emphasis to the application of interface agent technologies. Towards addressing the stated problem, the SDI user profile has been studied by taking a case SDI system known as ILRIAlerts of the International Livestock Research Institute (ILRI) in Addis Ababa. This has helped to investigate the various approaches and methods that agent technology can be applied in improving information filtering in SDI systems. Literature reviews showed that this technology has already been employed in similar systems, particularly in using the Internet information resources.

Based on these findings, agent-oriented modeling of the information filtering process of the SDI system through methods and procedures for updating the user profile has been investigated and some of the basic methods and procedures proposed have been demonstrated by developing a prototype agent. Applying this agent, some small-scale tests have been made on the performance of the proposed methods taking few test data from ILRIAlerts users

feedback records. The prototype and tests have indicated the applicability of using agent methods to SDI information filtering problems.

TABLE OF CONTENTS

	Page
DECLARATION.....	ii
DEDICATION.....	iii
ACKNOWLEDGEMENT.....	iv
ABSTRACT.....	v
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xi
LIST OF EXAMPLES.....	xii
 CHAPTER I	
INTRODUCTION	
1.1 Background.....	1
1.2 Statement of the problem.....	3
1.3 Objectives of the study.....	6
1.3.1 General Objectives.....	6
1.3.2 Specific Objectives.....	7
1.4 Methodology.....	7
1.4.1 Review of related literature.....	7
1.4.2 Data collection methods.....	8
1.4.3 Programming techniques.....	8
1.4.4 Testing	8
1.5 Justification of the Study.....	9
1.6 Scope and limitations of the Study.....	10

1.7	Organisation of the thesis.....	11
-----	---------------------------------	----

CHAPTER II

LITERATURE REVIEW

2.0	Personalised information filtering systems.....	13
2.1	Previous works.....	15
2.1.1	Information retrieval and information filtering.....	16
2.1.2	Selective Dissemination of Information (SDI)	17
2.2	Modeling Users' Interest for Information Filtering.....	18
2.3	Software agents	19
2.3.1	What are intelligent agents?	19
2.3.2	The concept of agent and intelligent agent.....	20
2.3.3	What can an agent do?	21
2.3.4	The attributes of intelligent agent.....	22
2.3.5	Types of intelligent agent.....	23
2.3.6	Characteristics of agent.....	28

CHAPTER III

THE ILRIAlerts System

3.1	Introduction.....	30
3.2	The Experimental case: ILRIAlerts.....	30
3.2.1	The ILRIAlerts Users Profile	32
3.2.2	The ILRIBIB: the bibliographic database source for the ILRIAlerts service.	34

3.2.3	The Matchmaking process.....	36
3.2.4	The ILRIAlerts output and the relevance feedback process.....	38
3.3	Methodology.....	43
3.3.1	Preparation of test data.....	44
3.4	Technical representation of users' profile.....	46
3.5	Limitations of the existing system.....	47

CHAPTER IV

THE PROPOSED INTERFACE AGENTS-BASED INFORMATION FILTERING SYSTEM

4.1	Introduction.....	49
4.2	Previous model	50
4.3	The New General Model.....	53
4.4	The New Agent-oriented model for information filtering.....	58
4.5	The Proposed Agent Design.....	60
4.5.1	Design considerations.....	60
4.5.1.1	The Feedback Handling Agent Problem Domain	64
4.5.1.1.1	Document ranking.....	66
4.5.1.1.2	Relevance judgment analysis.....	67
4.5.2	Agent Role Models.....	71
4.5.3	List of Agent Responsibilities.....	73
4.5.3.1	The Delivery, Reminder and Feedback Handling Chain.....	73

CHAPTER V

THE EXPERIMENT

5.1 Introduction.....	78
5.1.1 Experimental objectives and scope.....	78
5.1.2 Tools and Methodologies.....	79
5.2 The AgentBuilder toolkit: An Overview.....	80
5.3 The Prototype Delivery and Feedback Handler Agents.....	86
5.4 Testing.....	91
5.4.1 The test.....	92
5.4.2 Communication between agents.....	94
5.4.3 The performance of removing non-content bearing words, term weighting, term selection and setting threshold.....	94
5.4.4 The performance of filtering operations – recall and precision.....	100
5.5 Discussion.....	100

CHAPTER VI

CONCLUSION AND RECOMMENDATION

6.1 Conclusion.....	103
6.2 Recommendations.....	106
REFERENCES.....	108
Annex 1.....	114

LIST OF TABLES

		PAGE
Table 3.1	User feedback record.....	44
Table 4.1	Agent roles.....	72
Table 4.2	The Delivery Agent Social Role.....	74
Table 4.3	The Delivery Agent Domain Role.....	74
Table 4.4	The Reminder Agent Social Role.....	75
Table 4.5	The Reminder Agent Domain Role.....	75
Table 4.6	The Feedback Handler Agent Social Role.....	76
Table 4.7	The Feedback Handler Agent Domain Role.....	77
Table 5.1	Number of abstracts examined from users' relevance feedback.....	93
Table 5.2	List of potential key-words that represent 'relevant' and 'non-relevant' documents of users in the test cases.....	96

LIST OF FIGURES

Figure 4.1	A client/server SDI model.....	51
Figure 4.2	A Model for Agent Oriented SDI the System.....	53
Figure 4.3	Level 0 Overview Diagram of the SDI service.....	56
Figure 4.4	Alert/Remind User to give Feedback Level 1 Diagram.....	57
Figure 4.5	Follow-up Profile Performance Level 1 Diagram.....	57
Figure 4.6	SDI Search and Delivery Level 1 Diagram.....	58

Figure 4.7	A New Model of Agent Oriented SDI System.....	59
Figure 4.8	The Structure of the SDI Client System.....	62
Figure 5.1	AgentBuilder Components.....	82
Figure 5.2	The Feedback Handler Agent Implementation Architecture.....	87

LIST OF EXAMPLES

Example 3.1	User Profile.....	33
Example 3.2	ILRIBIB - bibliographic database record format.....	35
Example 3.3	ILRIAlerts output format.....	40
Example 3.4	Feedback receiving format.....	41

CHAPTER I

INTRODUCTION

1.1 Background

There has been a massive increase in the amount of information available to people as a result of the printed media in the past and now the electronic media; and information seekers are expected to be in the position to identify and evaluate this information according to their needs. Even if users carry out search using various search tools, as stated by Haverkamp and Gauch (1998), they will typically be in either of the three possible outcomes: *'they are buried under an information avalanche; or they are unable to locate any useful information at all; or they find what they need in roughly the amount they need'*.

The Internet literature in particular refers to the problem of information overload and to difficulties in finding accurate and pertinent information. This has created a demand for information filtering systems that are meant to deliver personalised information by building and applying profiles based on the user's information preferences. (Quiroga : 1999).

As a response to this problem, and with the development of information service provision, information service providers / systems have also been introducing a variety of services to provide users with access to information in their area of interest, under different levels and to different target users. One of the current awareness services which has been around since the mid-60s is SDI (Selective Dissemination of Information). SDI (a phrase coined by H.P. Luhn) is aimed at helping the user to keep up-to-date with current publications which is also intended

to be selective enough for an individual or small group, based on a 'profile' of the interests of the individual or group. (Vickery : 1994).

SDI mainly refers to a computer-assisted subscription-based personalised service that notifies a user about new literature and data in accordance to his/her statement of interest (Leggate : 1975). Users upon receiving updates either on diskettes, via e-mail or as a computer printout send their feedback on the relevance of the information for consideration of further refining of their profile. This has, however, been a very time consuming task. As quoted by Dawit (1998), Leggate (1975) states the objective of SDI as a reduction in the users effort in selecting from a mass of information sources, while Torkelson (1973) views it as a method aiming at preventing information overload.

This and other problems related to the operation of SDI outlined in the section below attracted researchers to identify technology solutions. One of the solutions identified by previous researcher, on which the current research is based, is the application of interface agent technology. Locally, research in this area is very rare. The first of its kind was undertaken by Dawit (1998), who was a student of SISA, for his MSc Thesis in 1998.

According to Russell (1995), an agent is *'anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**'*. This definition shows that agent is something that has a sensor to perceive or understand (learn), and effector to act on a specified objective.

Sheth (1994), states that information filtering meets the two requirements for applicability of agents - repetitive computer interaction and differences in individual preferences. This is the particular interest of this research in terms of SDI's similarity to the application of agents.

Since the last study undertaken by Dawit, it is found that there is no improvement in the ILRIAlerts SDI service and the problems identified still persist. A new development introduced to this service since 1998 is the distribution of SDI outputs by e-mail to those having e-mail address. Out of 125 users of the ILRIAlert, 45 receive it by e-mail.

1.2 Statement of the Problem

The users' profile of the SDI system, which is actually reduced to keywords and boolean-based search languages for information filtering, has to be refined based on feedback on the performance of the previous profile. Such cases lead to the execution of a number of filtering attempts to be run by database experts. At the same time, when there are changes in the interests of users, the database experts need to reformulate the users' profile. It is assumed that if every attempt of information filtering for SDI requires further refining of the profile and filtering strategy, it means that the objective of SDI, which as Leggate (1975) states is a reduction in the users effort in selecting from a mass of information sources, could not be achieved.

Hence, such problems necessitate the need for automatically constructing and refining a user's interest profile based on user feedback about documents which can simplify the task of programming an information filtering rule. As stated by Soltysiak and Crabtree (1998), there

is no question that user profiles are vitally important, but furthermore that these profiles should be acquired automatically with minimal user input.

Another problem which could occur on the part of the SDI service provider, as it may also occur with the user is the frustration with the system if the relevance and precision rate is below expected. Furthermore, according to Sheth (1994), unlike the users of Information Retrieval (IR) systems, users of Information Filtering (IF) systems might not be highly motivated in their information seeking and may not have well defined interest so that IF systems need to be more user friendly.

SDI's major problem in its implementation, maintenance and management are reported to relate to the filtering of information component which depend mostly on maintaining a quality users profile (Dawit : 1998). Maintaining refined users profile which is in the hands of end-users, database experts or other intermediaries is a difficult task in the existing conventional SDI system, maintaining and managing them have also proved to be challenging. It is so due to the need for carrying out a great number of search runs as the database experts may or may not understand users statement of interests. Hence the user has to supply a feedback for each run to the database expert, consult with him/her, modify his/her profile, etc. When there is a need to update user profile due to change of interest, the previous cycle will be repeated. This, therefore, proves to be that the SDI system is demanding of effort both on the part of users and SDI system managers. Baeza-Yates and Ribeiro-Neto (1999) stated that the most common measures of information retrieval system performance are time and space, the shorter the response time, the smaller the space used, the better the system is considered to be. Information filtering system is a time saving system. If one had an infinite amount of time to

handle all information and information filtering activity, then the information filtering system would not be necessary. One of the problems of the system under study (our case, i.e. ILRIAlerts) is that it takes three or more days to process the SDI output for their hundred and twenty active users. Each month, each user's profile is matched one by one to a batch of 1500 to 2000 newly added bibliographic records and a personalised SDI output is then produced. This has occurred due to a shift from the formerly CDS/ISIS based bibliographic database to INMAGIC a new DBMS. In the earlier CDS/ISIS based database, they had a small programme that produces SDI output automatically for each of their users.

In addition, despite the contribution of the above service, according to the 'Strategy and Programme for ILRI Information Resources 1998 - 2010' document, 59% of ILRI scientists spend 30-40% of their time searching for and using information. A further 13% of scientists devote more than 40% of their time to these activities. (ILRI : 1998) This shows that there is a great need for a time saving mechanism in searching for information such as an information filtering system tailored to users' needs and which could also be maintained and run by them.

In addition to the above-mentioned problems related to users interest profiling for information filtering, this research is conceived based on the problems dealt with by agent technology which has greater similarity to the stated SDI problems. Hence the role of agents in information filtering in the works of Sheth (1994), Quiroga (1999), Chambers (1999) and Dawit (1999) indicates that the problem domain dealt in their work is similar to problems of SDI information filtering mentioned above.

It is, therefore, believed that with the development of artificial intelligence, particularly software agent technology, these problems could be resolved.

Moreover, it was tried to identify some of the limitations of Dawit's research in which the current research is conceived to deal with. SDI is mainly a filtering system rather than a retrieval system due to the existence of a relatively static user interests as opposed to retrieval system which deals with a dynamic and short term needs (Chambers : 1999). Therefore, learning and adaptation is of much greater importance in SDI in terms of applying an agent support to its processes. However, it was less emphasised and dealt with in Dawit's research. Furthermore, according to Sheth (1994) information filtering system must satisfy three requirements i.e. specialisation, adaptation and exploration. Dawit's research dealt with the first aspect to some extent. Lastly, the findings of Dawit's research showed that the existing ILRIAlerts SDI system is unable to provide complete services using human intermediary which needed an automated assistance in both user directed and system directed operations. However, his research dealt with only the user directed operation to some extent. Therefore, the current tries to fill some of these gaps and deal with the aforementioned limitations.

1.3 Objectives of the study

1.3.1 General objective

The general objective of this study is to investigate into the approaches and techniques of agent technology, mainly interface agents, and test its applicability in addressing the information filtering problems in SDI systems and service provision by taking the case of ILRIAlerts.

1.3.2 Specific objectives

The specific objectives are:

- 1.3.2.1 to refine the statement of the personalised information filtering problems (SDI problems) by critically reviewing the previous study and findings
- 1.3.2.2 to undertake a further analysis of the SDI problem domain related to information filtering
- 1.3.2.3 to explore further the use of vector space model in document and profiles representation to address the SDI's information filtering problems
- 1.3.2.4 to define agency architecture, agent roles, communication protocol, and agent behaviours
- 1.3.2.5 to develop and implement a prototype user interface
- 1.3.2.6 to test the prototype agent

1.4 Methodology

The methods employed to undertake this study involved the following:

1.4.1 Review of related literature

In order to establish the basis for information filtering, the previous study has been critically assessed and reviewed. In addition, past studies and literature on information filtering, interface agents, SDI and user profile modeling, data structure and algorithms, etc. were extensively reviewed.

1.4.2 Data Collection Methods

Interviews, observation and review of documents on the function of the ILRIAlerts has been conducted to gather information on the filtering mechanism of the system under study. Interview with ILRI information service staff and with few selected users has been conducted. Close observation of ILRIAlerts was done while staff run the monthly SDI output in order to understand how the system functions, and examine its operation and problems.

1.4.3 Programming techniques

In order to develop the prototype agent, an exploratory programming approach was followed using *AgentBuilder: An Integrated Toolkit for Constructing Intelligent Software Agents* by Reticular Systems, Inc. (Reticular Systems, Inc. : 1999), the academic version of which has been purchased for this research purpose.

To develop the prototype agent and to code the required agent components, C⁺⁺ programming language was used.

In addition, the use of vector space models was explored for the modeling and representation of document and user profiles to information filtering.

1.4.4. Testing

To test the suggested agent-oriented model, an updated and comprehensive data, based on the previously collected data from the real users, was used. Mainly the performance of stop-word

removal, threshold setting, precision and recall have been tested including how communications between agents is formed was also the focus of the testing.

1.5 Justification of the Study

Agent technologies are believed to have contributed to resolve the client/server problems and the need for online information retrieval. The time required by the users refining their profile and the database experts executing the SDI matching process can be reduced through the implementation of agent technologies which are autonomous by their very nature to perform given tasks.

As argued by several scholars, the lack of skilled manpower in most developing countries' information systems who are continually involved in the process would not be easily resolved in the near future, hence, the assistance of technology would contribute a lot. In this respect, ILRI will benefit from the results of this study.

Apart from this, the following opportunities can be envisaged from the application of agent technologies:

- Intelligent information filtering systems reduce the users' and database experts communication burden in refining users interest in order to maximise the performance of SDI by automatically eliminating incoming data predicted to be irrelevant;
- Must be read*

Must be read.

- Adapting an internal user model which is based upon user interaction to increase performance of SDI;
- It enables autonomous updating of the profile based on the information that it gathers about users interest changes which could save time and efforts of the user and database experts;
- It could enable end users create their own SDI profile, run them and make changes with little or no effort.

1.6 Scope and Limitations of the Study

This study has tried to accomplish three major tasks. Firstly, it tried to refine the personalised information filtering problem i.e. the SDI problem by critically reviewing the previous work (case) including by investigating the current state of the SDI system - the ILRIAlerts. Secondly, it tried to model and describe how interface agents can be applied in user profile maintenance based on relevance feedback. Finally, it attempted to demonstrate this model by developing and testing a prototype. The second and third tasks have not been dealt in full due to the limited time available for the study. The prototype aimed to test and demonstrate how agents perform tasks delegated to them by communicating to each other. In the relevance feedback process, three roles (agents) have been proposed, however, due to the limited time available for the study, it has been tried to demonstrate in the prototype the feedback handling process to some extent and the delivery and reminding activities. As the agent building toolkit used can only support single agent application development, the demonstration or prototype for the delivery-feedback processes has been tested on a console-based communication rather than on a client-server environment. The feedback handling activity is the most important

activity in maintaining and updating the user profile. This study has, therefore, focused on user profile maintenance based on feedback from users. Due to the limited data obtained on relevance feedback and the time constraint, a simplified approach to analysing the feedback has been followed to demonstrate the prototype agent solutions as opposed to the proposed range of steps for feedback handling presented in chapter 4.

Furthermore, testing the performance of the updated user model based on relevance feedback could not be achieved with a number of users as most if not all users are outside the country and those in Ethiopia are based outside Addis Ababa. Consulted users are only those accessible by e-mail. It is, however, believed that for better outcome of the evaluation of updating the user model based relevance feedback, an extended experiment need to be done. This makes possible experimenting a learning approach to user modeling and updating using a large set of relevance feedback data.

Finally, the other limitations to this study include the scarce availability of literature in this topic although Internet searches produced valuable literature, the limitations of the agent building toolkit for the application proposed and the extended time required to study and understand the toolkit.

1.7 Organisation of the thesis

This thesis is divided in to six chapters. The first chapter provides the background to the study including the refined statement of the problem and the basic introduction to the study. In the second chapter, we dealt with the personalised information filtering systems with particular

emphasis to reviewing previous works in this topic, describing the experimental case and reviewing agent application to information filtering. The third chapter is devoted to the explanation of the methodology employed in this study with particular emphasis to data structures and algorithms used which gives the basis to the proposed agent models for information filtering presented in chapter four. The fourth chapter presents the proposed agent-oriented models from problem design to application realisation. The fifth chapter presents the experimental case which also describes the agent building toolkit employed to develop the prototype agent models and the results of the experiment. Finally, the sixth chapter provides a conclusion to the study and recommendations as an indication to further research in this area.

CHAPTER II

LITETATURE REVIEW

2.0 Personalised Information Filtering Systems

The need for information filtering and personalised information filtering systems has increasingly becoming an area of research and attention in IR with the tremendous amount of information created and delivered to users from various sources and in various formats. As frequently referred, the Internet literature refers to the problem of information overload and to difficulties in finding accurate and pertinent information. As stated by Quiroga (1999) this problem has created a demand for information filtering systems that are meant to deliver personalised information by building and applying profiles based on the user's information preferences.

Information filtering systems can help users by eliminating irrelevant information and by bringing the relevant information to the user's attention. Hence, filters are mediators between the sources of information and their end users. (Sheth : 1994). These systems are capable of identifying the documents potentially relevant for users information needs. An essential aspect of an information filtering system is its ability to represent the user information preferences. (Asnicar, Fant, Tasso: 1999)

Quoted by Quiroga (1999), Bowman et al, stated that obtaining the right information is important given that, "at least 99% of the available data is of no interest to at least 99% of the users". This is why users spend everyday a great amount of their time. Therefore, in such cases there is a great demand for specialisation of information retrieval which is leading to the design of personalised information systems which will act as personal information agents. In this line Maes (1994) stated that information filtering is one of the many applications that an agent can assist in, alleviating information overload of their clients.

According to Quiroga (1999) personalisation of information delivery relies on systems that selectively weed out irrelevant information based on user preferences. Thus, profiles, i.e. representations of users' information preferences, are applied as filters to streams of documents.

The problem of information overload has favoured the development of systems capable of automatically identifying the subset of the available information, which is potentially relevant to the user information needs. More specifically, filtering systems have been used, and are aimed at automatically evaluating the potential relevance of incoming information on the basis of an explicit description of the user information interests (user profile). However, while the need for these systems has been widely recognised and adequate techniques for their implementation have emerged -- we mainly refer to the intelligent agent technology (Hendler, 1996) and (Petrie, 1996) -- two basic problems still remain open and need further investigation: (i) the mechanisms for learning, representing and updating the user's information preferences, and (ii) the processing algorithms to be adopted to extract the

information content of the incoming documents and the matching algorithms to be explored to assess their potential relevance. (Asnicar, Fant, Tasso: 1999)

Asnicar, Fant, and Tasso (1999) also show that a systematic evaluation activity, which support the claim that a specific user-modeling component improves the performance with respect to other filtering system reported in the literature.

In summary, therefore, as Sheth (1999) stated that the personalised information filtering problem is to design a system that can provide information that match user needs in consistent and timely manner.

2.1 Previous Works

To lay the basis for this thesis, the research tried to see previous studies in the areas that are of topical interest to this study i.e. information retrieval, information filtering and agent technology. According to Quiroga (1999), there are four related concepts that form the core of information filtering (IF) research: filtering, profiles, user models, and agents.

In addition, SDI as a filtering system and as the particular case in this study, it will be dealt with. Information retrieval is a well-established field of information science which addresses issues of retrieval from a large collection of documents in response to user queries. (Sheth : 1994). However, when compared to IR, research in agent is a relatively new field of study which has grown out of Artificial Intelligence (IA). Agent research is concerned with the issues of designing intelligent and autonomous software for a variety of tasks. Thus, this research will draw up on work from both these areas.

2.1.1 Information retrieval and information filtering

The fact that various definitions of the concept IF given in various literatures may show that this field is in the early stages of development, where many basic concepts still are not consistently defined. Some of the definitions, as quoted by Quiroga (1999), found in the literature describe information filtering (IF) as:

"a field of study designed for creating a systematic approach to extracting information that a particular person finds important from a larger stream of information" ; "tools... which try to filter out irrelevant material"; "...systems that determine the interestingness of or relevance of a text based on an analysis of the relationship of the content of the input to the user's interests"; "as a process of selecting things from a larger set of possibilities, then presenting them in a prioritised order"

As compared to the generally accepted definitions of information retrieval (IR), we can see little differences between these two processes. Both processes involve a large set of documents and a process of selection or matching documents to an information need.

Baeza-Yates and Ribeiro-Neto (1999) in their taxonomy of information retrieval models presented two retrieval tasks i.e. browsing and retrieval and that retrieval as being termed as *ad hoc* and *filtering*. They stated that in a conventional information retrieval system, the documents in the collection remain relatively static while new queries are submitted to the system which has been termed as *ad hoc* retrieval. In operations where queries remain relatively static while new documents come into the system (and leave), for instance wire news, are referred to as *filtering*.

However, the distinction between information filtering and the more established field of information retrieval has proven to be the source of some confusion as well. Information retrieval broadly deals with the selection of information, and many of the features of information retrieval system design (e.g. representation, similarity measures or boolean selection, document space visualisation) are present in information filtering systems as well. If one considers information retrieval from a very general "information selection" viewpoint, information filtering is simply a special case in which the information space is very dynamic. If, on the other hand, our personal definition of information retrieval involves selection of relatively static information in response to relatively dynamic queries, then information filtering is best viewed as the dual problem to information retrieval.

Information filtering has been applied to several domains using a variety of technical approaches. The original methods were manual alerting services that brought new information to the attention of users of research and special libraries. At the time, this was referred to as Selective Dissemination of Information (SDI), a name which fell from favor about the time the Strategic Defense Initiative (SDI) was introduced in the United States. A few modern systems have adopted this remarkably descriptive name for the filtering process, however, and the interest in information filtering that has resulted from the present research thrusts in digital libraries arises at least in part from this tradition. (Oard : 1995).

2.1.2 Selective Dissemination of Information (SDI)

SDI (a phrase coined by H.P. Luhn) as one of the current awareness services has been around since the mid-60s. SDI is aimed at helping the user to keep up-to-date with current

publications which is also intended to be selective enough for an individual or small group, based on a 'profile' of the interests of the individual or group. (Vickery : 1994). Garcia-Molina and Yan (199) stated the increasingly importance of SDI particularly in wide area information systems in order to help users cope with information overload. In an SDI service, users post their long-term queries, called profiles, at some SDI servers and continuously receive new, filtered documents. As indicated above, traditionally, libraries and databanks provide such kind of information filtering service, named Selective Dissemination of Information (SDI).

2.2 Modeling Users' Interest for Information Filtering

Quoted by Dawit (1998), Kemp (1979), defined the user-interest profile as "an expression of the current interests and needs of a client, stated in a way which enables it to be used to identify the records of documents which will meet those interests and needs". Pao (1989) on the other hand describes it as "a listing of key words joined by Boolean operators characterising a user's (topical) interest".

As indicated above an information filtering system is located between a document source and a user which is capable of identifying the documents potentially relevant for users information needs. Therefore, an essential aspect of an information filtering system is its ability to represent the user information interests. A number of user modeling approaches have been stated in the literature. Some of which as quoted by Quiroga (1999) include Malone's (1983) approach to model user's organisational habits in order to overcome information overload, Kwasnik's (1989) of understanding how individuals organise their own information space.

Soltysiak and Crabtree's (1998) approach to user profiling aims at discovering whether user interests could be automatically classified through the use of several heuristics. The result of their investigation highlights the need for minimal user feedback, and the need to consider the implications for the role of machine learning in user profiling.

In the review of the three prototypes for information filtering (IFTTool, PIFT and ifWeb) undertaken by Asnicar, Fant and Tasso (1999), the user model includes information which represents both the interests and the 'not interests' of the user. This is constituted by a weighted semantic network whose nodes correspond to terms (concepts) found in documents and where arcs link together terms which co-occurred in some document.

2.3 Software agents

2.3.1 What are Intelligent Agents?

Various scholars and writers gave several definitions of the terms agent and intelligent agent. In fact some of the definitions given are contextual used in different literatures.

Russell (1995) defines an agent as 'anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**.

Agents employ artificial intelligence techniques to assist users with daily computer tasks, such as reading electronic mail, maintaining calendar, and filing information. Agents learn through example-based reasoning and are able to improve their performance over time (Janson : 1996).

As quoted by Meek (1995), Roesler and Hawkins in 'Intelligent Agents: Software Servants for an Electronic Information World (And More!)', (1994: 19), described intelligent agents as follows:

Intelligent agents are autonomous and adaptive computer programs operating within software environments such as operating systems, databases or computer networks. Intelligent agents help their users with routine computer tasks, while still accommodating individual habits.

This technology combines artificial intelligence (reasoning, planning, natural language processing, etc.) and system development techniques (object-oriented programming, scripting languages, human-machine interface, distributed processing, etc.) to produce a new generation of software that can, based on user preferences, perform tasks for users.

2.3.2 The Concepts of Agent and Intelligent Agent

According to Meek (1995), the idea of agents originated with John McCarthy in mid 1950s, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology (MIT). They had in view a system which carry out the

details of appropriate computer operation when given a goal, which could ask for and receive advice and offered in human terms when it was stuck. Then this idea was also forwarded by Negroponte who was sometimes credited with its origin with elements emerging in his publication of *The Architecture Machine: Toward a more human environment*, in 1970. Jansen (1996) has also mentioned that Nicholas Negroponte and A. Key were among the first to recognise the potential value of agents.

Looking at the concepts of agents, Gadomski (1997) stated that agent in the subject matter literature is used in two contexts:

Software world where an agent is a "softagent" interacting only with software entities in a computer software world while the second context is the **physical world** where agent is a cognitive and engineering attempt to the explanation, modeling and simulation of human mental functions.

2.3.3 What can an agent do?

The application of agents has become enormous ranging from information gathering and filtering to mail management, meeting scheduling and selection of books, music, movies, etc. including running an autonomous process on a remote node in the network. Based on a very broad definition, however, agents tend to be found in the 'information' and 'work' categories as shown below (Meek : 1995).

Functions for Agents?

INFORMATION

Navigation and browsing
Information retrieval
Sorting and organising
Filtering

WORK

Reminding
Programming
Scheduling
Advising

LEARNING

Coaching
Tutoring
Providing help

ENTERTAINMENT

Playing against
Playing with
Performing

Thus, as shown above, an agent could be used in a variety of applications ranging from information gathering and filtering to being a wizard to function as expert assisting and guiding the novice, etc.

2.3.4 The Attributes of Intelligent Agents

Research in the area of agent technology is new and broad which includes research in different software engineering fields such as interface design, communication and co-ordination, adaptation and learning algorithms, etc. Many researchers believe that to be 'intelligent', an

agent must satisfy several interrelated criteria. Hence, according to Chen, Chung, and Ramsey (1998), the attributes of intelligent agents include the following:

Integrated - the agent must support an understandable, consistent interface;

Expressive - the agent must accept requests in different modalities;

Goal-oriented - the agent must determine how and when to achieve a goal;

Co-ordinated - the agent must collaborate with the user; and

Customised - the agent must adapt to different users.

2.3.5 Types of Intelligent Agents

Several writers described the types of agents in different ways. As quoted by Meek (1995), Paul Harmon (ed.) in the article 'Software Agents' appeared in *Intelligent software strategies*, 11(1), 1-13 (January 1995), classifies agents into three types in terms of how they do their work and what they do, as follows:

Rule-based approach - the rule-based approach features a collection of user-programmed rules for processing information related to a particular task. An end user program allows users to program rules for processing information for a task;

Knowledge base approach - this interface agent has domain-specific background knowledge about the application and user to recognize plans and contribute to users' tasks. This approach has several problems as well. Where an end-user program approach requires a great deal of work on the part of the user, this approach requires a huge amount of work for a knowledge

engineer. The knowledge of the agent is fixed and cannot be customized to individual users, so its use by users may be limited.

Machine learning approach - this approach addresses problems encountered by the rule-based and knowledge-engineered approaches. This approach requires less initial work, and adapts over time. The agent acts as a personal assistant to cooperate with a user on a task, but makes allowances for user override. The agent learns by:

- observing and imitating user
- adapting based on user feedback
- trained by user by example
- ask for advice from other agents

Machine learning uses memory-based reasoning combined with rules to model each user's habit. This approach achieves a level of personalisation, previously impossible, available except through user intervention. However, these agents have their problems as well. Learning agents have slow learning curve, requiring a sufficient number of examples before it can make accurate predictions. These agents also encounter problems when dealing with completely new situations. To address this problem, agent may learn from existing agents to get up to speed quickly.

Another mode of classification of agents, according to Meek (1995), came from Kurt Indermaur who acknowledges that agents are being developed today in a number of application areas. Accordingly Indermaur lists three major types:- **advisory agents** which offer instruction and advice to help you do your work; **assistant agents** which can be more ambitious than advisory agents because they often act without direct feedback from users; and

Internet agents most of which are information gatherers, some of which attempt to make sense of the information they find on the Web.

Finally, Miller (1996) categorises agents into three types by demonstrating the products in which software companies adding agent feature in their applications to help users with such as sorting out their information overload. These products, according to Miller (1996) fall into the following categories:

Personalisation - software that selects relevant or interesting information for a user based on profile information. Examples include Pointcast and Quote.com.

Notification - software that actively notifies users about new information or changing conditions. Examples include Surfbot and HotPage.

Offline browsers - applications that act like a user on the network, participating in or monitoring the Web, Usenet and chat traffic and returning interesting information to the user. Examples include Webcompass, Web browser intelligence agent and WebWhacker.

Therefore, the above explanation of the types of agents by different authors varies only in the way that the mode of classification they use to categorise intelligent agents such as by how and what they do; their architecture; application area, and products.

However, Nwana and Wooldridge (1996) has identified the different types of agents which he has presented them in terms of their essential metaphors, their hypotheses/goals, motivations, roles, prototypical examples, potential benefits, key challenges, and some other general issues about particular agent type which in one way or the other covers the types of agents discussed above by different scholars. This typology of agents will also help us present why the particular type of agent, i.e. interface agent we selected to see its applicability to information filtering in terms of our experimental case i.e. the ILRIAlerts SDI service. The types of agents identified by Nwana and Wooldridge (1996) are collaborative agents, interface agents, mobile agents, information agents, reactive agents, hybrid agents, heterogeneous agents systems, smart agents. Of these types of agents, some of them are more applicable for information filtering than others as described in his overview. However, collaborative agents, interface agents, mobile agents and information agents are more or less handles information retrieval tasks. For the purpose of clarification of the choice of the type of agent in this study, we briefly describe some of these types of agents below as presented by Nwana and Wooldridge (1996):

- *Collaborative agents* - these agents emphasise autonomy and cooperation (with other agents) in order to perform tasks for their owners. This type of agents may learn, but this aspect is not typically a major emphasis of their operation. They may have to negotiate in order to reach a mutually acceptable agreements on the some matters in order to have a coordinated set up of collaborative agents. In the area of retrieval, one of the motivations for this agents is to provide solutions which draw from distributed

information sources, e.g. for distributed on-line information sources, it is natural to adopt a distributed and collaborative agent approach. An example of collaborative agents is the Pleiades project at CMU which applies collaborative agents in the domain of Organisational Decision Making over the "InfoSphere" (which refers essentially to a collection of internet-based heterogeneous resources).

- *Interface agents* - these agents emphasis autonomy and learning in order to perform tasks for their owners. The key metaphor underlying interface agents is that of a personal assistant who is collaborating with the user in the same work environment. We can see the subtle emphasis and distinction between collaborating with the user and collaborating with other agents as is the case with collaborative agents. Some of the motivation to interface agents are managing junk mail, scheduling and rescheduling meetings, searching for information among heaps of it, etc. An example of these agents include Eager assistants (Kozierok & Maes,1993), Guides (e.g. Liebermann, 1995), Memory aids (e.g. Rhodes & Starner, 1996), Filter/critics (e.g. Sheth & Maes, 1993), Matchmaking/referrals (e.g. Chavez & Maes, 1996), Buying and selling on your behalf (e.g. Chaves & Maes, 1996) and Entertainment (e.g. Maes, 1995b).
- *Mobile agents* - these agents are computational software processes capable of roaming wide area networks (WANs) such as the

WWW, interacting with foreign hosts, gathering information on behalf of its owner and coming 'back home' having performed the duties set by its user. The duties may range from flight reservation to managing telecommunications network.

- *Information / Internet agents* - these agents have come about because of the sheer demand for tools to help users manage the explosive growth of information currently experiencing. Information agents play the role of managing, manipulating or collating information from many distributed sources. Quoted by Nwana and Wooldridge (1996), Etzioni and Weld (1994) describes a state-of-the-art agent called the Internet Softbot (software robot).

Looking at the different types of agents discussed above and as presented by the key proponent of this class of agents (Pattie Maes), interface agents are so far the widely applicable type of agents in the area of information filtering.

2.3.6 Characteristics of Agents

Similar to the types of agents described above, several scholars have also described the characteristics of agents in many ways from different perspectives. However, we will give the one that is relevant to information retrieval.

According to Haverkamp and Gauch (1998) from artificial intelligence point of view, agent possesses a long list of properties including autonomy, social ability, reactivity, pro-

activeness, benevolence and rationality. However, for an agent used in information retrieval, all of these properties may not be necessary. Quoted by Haverkamp and Gauch (1998), Wooldridge and Jennings in their article 'Intelligent agents: theory and practice' appeared in *Knowledge Engineering Review*, 10(2), 1995, stated that from information retrieval point view, the characteristics of intelligent agents can be limited to the following:

Autonomy - agents operate without direct human intervention and have control over their own actions;

Social ability - agents communicate through a common language to other agents and even to humans;

Reactivity - agents perceive their environment and react to changes in it;

Pro-activeness - agents are capable of exhibiting goal-directed behaviour by taking the initiative.

CHAPTER III

The ILRIAlerts System

3.1 Introduction

In this chapter, the ILRIAlerts system and the methodology to be employed will be discussed in more detail. As discussed in the previous chapter, the user profile is an important component to the information filtering system. As the ILRIAlerts' users profile is the experimental data for this study, it would be appropriate to give a detailed description of how it is formed, its structure and how the study uses the data in the experiment. Then, the source of information under use for the filtering process, its structure and content would be described as well since the output of the ILRIAlerts relies on these sources. Following this, an outline of the limitations of the existing system will be presented.

3.2 The Experimental Case: ILRIAlerts

The mission of ILRI's information services has been to ensure that ILRI researchers and NARS (National Agriculture Research Systems) partners are provided with information support in a comprehensive and responsive manner as they plan, develop, and implement their research programmes, and as they deliver and exchange technologies arising from research development in ILRI and NARS (ILRI : 1998).

ILRI provided a free selective dissemination of information (SDI) service to 400 NARS scientists in sub-Saharan Africa (SSA) using procedures which were time intensive (ILRI : 1998). These services were delivered directly to NARS scientists reinforcing the view that access to the ILRI library is easier and more useful than to NARS libraries. However, the new strategy is that the SDI service would not be extended beyond Africa, including, potentially, to hundreds more scientists in India and thousands more in China. According to the ILRI strategy document (ILRI : 1998), ILRI will replace its SDI service in SSA by net-based technologies, and will not extend beyond SSA. It aims to provide NARS libraries with its databases and the software used to generate SDI output. This new vision of devolving the SDI services go along with Garcia-Molina and Yan's (199) idea of distributed SDI service with multiple servers.

The ILRI bibliographic database, which is the documentary sources of the ILRIAlerts, contained records from ILRI, AGRIS and CABI. The AGRIS and CABI records were transferred from tapes which takes unnecessary time and effort since the AGRIS and CABI data is now on the Internet. Thus the new direction in this service according to ILRI strategy document (1998) is to change and adopt new technologies that will allow them to do more with less which is also based on services and products not technology only to meet the needs of Intra- and extramural users. One of the major priority activities in this strategy is to deliver the best possible information services to Intra- and extramural users using net-based technologies (ILRI : 1998).

3.2.1 The ILRIAlerts User Profile

The procedure for maintaining users profile for the ILRIAlerts involves a number of steps. Primarily, users are required to complete the ILRIAlerts Service Request Form. In this form the users provide the following information:

- *Personal information* - personal particulars such as name, higher degree acquired, field of specialisation, university, country, year of highest degree; Category of affiliated organisation; full address
- *Statement of area of interest* - field / subject of interest and on-going work, in narrative form; title of current research project(s); citation of at least two documents relevant to the requested topic (if known by user); aspects of the topic(s) particularly important (e.g. geographic coverage, sub-topics, breeds, species); aspects of the topics that are not important (e.g. geographic coverage, sub-topics, breeds, species); statement of expected level of coverage, i.e. comprehensive or specific (which dictates whether the search terms are restricted or not to a specific portion(s) of the records).

Information staff will then convert the above into a user profile, i.e. a series of actual search expressions, using CAB International Thesaurus, AGROVOC and ILRI list of key words. In this process, according to the ILRI information policy manual (ILCA : 1991), consideration is given to the specific research projects that the user is involved as opposed to the description of his/her interest.

The user profile established in this manner will then be run on a sample database based on the results the necessary refinements will be undertaken and passes on to testing it on actual databases. The user profile is structured using the query language of the database under operation i.e. INMAGIC database management software.

The profiles constructed from the analysis of the above information obtained from users differ in their depth as we can see in the following examples of an active user profiles which could be as long as user profile number 007 and as short as user profile 120-T given below:

Set Name: 'SDI profile 007'

(natural resource / natural resources / agricultural resources / ecosystem / ecosystems / animal resources / energy resources / feed resources / fishery resources / food resources / forest resources / genetic resources / marine resources / nonrenewable resources / land resources / grazing lands / rangelands / water resources / plant resources / renewable resources / soil resources / (resource / resources) & (natural / animal / animals / energy / feed / feeds / fishery / fisheries / food / foods / forest / forests / forestry / genetic / germplasm / germplasm / marine / nonrenewable / nonrenewables / land / lands / farmland / farmlands / pasture / pastures / ley / leys / irrigated / irrigation / grazing / rangeland / rangelands / range land / range lands / peatland / peatlands / scrubland / scrublands / water / plant / plants / renewable / soil / soils / wilderness)) & (resource management / resources management / resource allocation / resource conservation / resource substitution / (resource / resources) & (management / allocation / conservation / substitution) / economic / economics / agricultural economics / production economics / welfare economics / land economics / models / modelling / mathematical models / econometric models / linear models / simulation models / dynamic models / policy / policies / agricultural policy / agricultural policies / land policy / land policies / development policy / development policies / economic policy / economic policies / production policy / production policies / support measure / support measures / energy policy / energy policies / environmental policy / environmental policies / fishery policy / fishery policies / food policy / food policies / forestry policy / forestry policies / social policy / social policies / structural policy / structural policies / valuation / valuation) / watershed management / (watershed w2 management / watersheds w2 management / shed w2 management / sheds w2 management)

Set Name: 'SDI Profile 120-T'

({feed legumes}) & ({cattle} / {sheep} / {goats} / small w2 ruminant / small w2 ruminants) & (animal feeding / animal nutrition / nutriti* / feed* / grazing / L02* / L5* / L3* / Q5*)

Example 3.1: User profile

The profile set name with T as in the above user (120-T) distinguishes users receiving their SDI output by postal mail as opposed to those receiving by e-mail identified by a set name (number) without T.

3.2.2 The ILRIBIB: the bibliographic database source for the ILRIAlerts service

The ILRIAlerts is provided based on an in-house database, known as ILRIBIB (developed on the INMAGIC database management software), that integrates input from external databases with data from locally generated input. The external databases are CAB International abstracts and AGRIS (of the FAO), two of the big international agricultural databases of the world. ILRI receives every month updates to these databases and search for records that are relevant to ILRI. The results of the search will then be downloaded to ILRIBIB after modifications to meet the needs and formats of the ILRI. The local input to the ILRIBIB database includes records on books and other documents that ILRI adds into its collection. The collection consists of ILRI publications and other acquisitions of the ILRI Library and Documentation Unit.

An example of the bibliographic record from the ILRIBIB database is given below to show the content and structure from which the ILRIAlerts output is produced.

```
ID 84355
Database_MFN ILRIB 085020
Record Created_Date 1996-02-29
Sources CABI
Indexer code 06
Bibliographic level AS
Author Niamir, M.
Title Indigenous systems of natural resource
```

management among pastoralists of arid and semi-
 arid Africa

Author Source Warren, D.M. (ed.)
 Slikkerveer, L.J. (ed.)
 Brokensha, D. (ed.)

Title Source The cultural dimension of development:
 Indigenous knowledge systems

Languages_Text En
 Languages_Summary En

Publisher Intermediate Technology Publications Ltd
 Place London (UK)

Pub Date 1995

Physical Description p. 245-257

Collation Notes 1 table

ISBN ISBN 1-85339-251-0

Subject code P01
 E50

Record Type_Form doc Article

CABI descriptors AFRICA
 ARID REGIONS
 ENVIRONMENTAL MANAGEMENT
 PASTORAL SOCIETY
 RANGELANDS
 PASTORALISM
 RANGE MANAGEMENT
 INDIGENOUS KNOWLEDGE

CABI abstract Although individual pastoral societies in
 Africa have been the object of numerous studies
 during this century, no systematic effort has
 yet been made to amalgamate the information to
 allow an analysis of the importance of
 indigenous natural resource management systems
 for the development process. The paper is based
 on a study carried out by Niamir for the FAO in
 1990 [Local knowledge and systems of natural
 resource management in arid and semi-arid
 Africa] that attempts to answer such a need. It
 considers a wide range of issues centred around
 natural resource management, including
 environmental knowledge and management
 techniques, but only the institutional aspects
 are summarized here. These are considered under
 three headings: informal and formal social
 controls over rangeland utilization; natural
 resource tenure; and means of enforcement of
 rules and regulations.

Example 3.2: ILRIBIB - bibliographic database record format

Each month the individual query (user profile) is matched to a range of 1500 to 2000 records in the bibliographic database. The range of records (i.e. 1500 to 2000 records) is determined for matching from the recently added records since the last SDI run.

3.2.3 The Matchmaking process

ILRIAlerts service alerts its users to the most recent world-wide literature in topics of users' interests. As mentioned above, the information is drawn from two international databases, viz. AGRIS and CAB International to which information acquired by ILRI library is added. The ILRIAlerts service to individual users is based on statement of interest maintained against individual user is based (i.e. users profile). Therefore, the ILRIAlerts is the result of the matchmaking process of the two profiles - the bibliographic database records and the user profile created as mentioned above.

The current ILRIAlerts offers its usual service to around hundred and twenty-seven users of which around forty five receive by e-mail. Since a change from the CDS/ISIS based database (which has had a small program to run the matching of user profile with the bibliographic database to produce the ILRIAlerts) to INMAGIC database management software, the ILRIAlerts matchmaking process involves a lot of manual operations. These steps involve the following:

1. Creating a folder (directory) for this month's SDI run using the current month and year which will be maintained for a maximum of six months after which will be purged

2. Loading this month's batch by taking the last record in the previous month's SDI run and adding 1 to this value and then determining the range of the current batch of 1500 to 2000 records. The extracted records will be exported to an output file
3. Refreshing referenced sets found in the Tropics and ILRI databases
4. Processing outputs for profiles found in the Tropics and ILRI databases which involves the following steps:
 - a. Refresh an SDI profile
 - b. Add the profile number to the SDI Recipients field in the records retrieved by this refresh.
 - c. Print the output for those recipients who get their SDI output by post. These profiles have a -T in the profile number, such as 396-T). For those who have e-mail, the output is written to two files i.e. one for the list and the other for the feedback.
 - d. Delete the profile number added in (b) above.
 - e. Repeat steps 4a through 4d for every profile.
5. Send the SDI outputs by e-mail by writing a covering letter and attaching the two files produced in 4c above.
6. Produce labels for postal addresses and send outputs using the address database and searching those whose profile number has -T and then print the label to dispatch output by post.
7. Record statistics i.e. the number of records sent to each recipient which is shown besides each profile in a spreadsheet created for this purpose containing the SDI number, name including the total output, relevant, irrelevant and can't say for each month.

According to information staff, the section is currently working to automate the matchmaking process in order to resolve the above time consuming and tasking manual operations.

3.2.4 The ILRIAlerts output and the relevance feedback process

Following the procedures described in section 3.2.3 above, the ILRIAlerts output is produced each month. The output has two major components:- the first is that the ILRIAlerts output list (which consists of the name and address, the profile, total items (output) to the user this month, the list consisting of full bibliographic information including abstract if any and where the hard copy of this document could be found) in one column; and the second, in the other column, which is a tear-off sheet consisting of the feedback information required of users mainly consisting of the following two statements in addition to the bibliographic details and user profile and addresses:

- Did you find this item relevant? [Yes/ No/Can't say]
- Do you want a copy of this item? [Yes/ No]

An example of ILRIAlerts output look like the following:

```

                                Information Services
          International Livestock Research Institute
                P.O. Box 5689, Addis Ababa, Ethiopia
                E-mail: ILRISDI@CGIAR.ORG
=====
User ID: 005
Ato Abate Tedla
Forage agronomist
ILRI
P.O. Box 5689
Addis Ababa, Ethiopia
```

Profile

```

=====
({feed grasses} / {feed legumes}
OR
Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct pasture / pastures
/ forage /
forages / fodder / fodders / tree / trees / browse / browses /
browsing /
herbaceous / herbage / feed crop / feed crops)
AND
Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct cropping system /
cropping systems /
farming system / farming systems / rotation / rotations /
rotational
/intercropping / plant production / crop production / crop
yield / crop yields
/herbage yield / herbage yields / cultivation / agronomy /
agronomic / seed
production / seed technology / seed certification / seed
characteristics / seed
cleaners / seed cleaning / seed collection / seed crops / seed
dispersal / seed
dressing / seed drilling / seed extraction / seed filling /
seed harvesting /
seed industry / seed longevity / seed mixtures / seed pelleting
/ seed stands /
seed storage / seed testing / seed treatment / seed weight

=====
Total items output to user this month: 7
User must provide feedback to remain on the service
=====

```

1. Teste de envelhecimento precoce para sementes de azevem, aveia preta e milheto.
 [Accelerated aging test for ryegrass black oat grass and pearl millet seeds] 1999. Garcia, D.C.; Menezes, N.L. de. Ciencia Rural (Brazil). v. 292. p. 233-237. 19 ref. ISSN 0103-8478.

Key words: LOLIUM MULTIFLORUM; AVENA; PENNISETUM GLAUCUM; SEEDS; GERMINATION; SEED LONGEVITY; AGING; RIO GRANDE DO SUL.

Did you find this item relevant? [Yes/No/Can't say]

2. Effect of nitrogen and phosphorus on the fodder yield and quality of two sorghum

cultivars (*Sorghum bicolor* L.) 1999. Ayub, M.; Tanveer, A.; Kahmud, K.; Ali, A.; Azam, M. Pakistan Journal of Biological Sciences (Pakistan). v. 21. p. 247-250. 1 table; 24 ref. ISSN 1028-8880.

Response of two sorghum cultivars namely Hegari and JS-263 to different NP levels of 0-0, 50-0, 100-0, 50-50, 100-50 and 100- 100 kg ha⁻¹ was studied under field conditions. The cultivar Hegari gave significantly higher green fodder and dry matter yield due to greater plant height, stem diameter and leaf area plant⁻¹. The crude fiber percent and ash percent were also significantly higher for cultivars Hegari. A progressive increase in yield was observed with fertilizer application up to 100 kg N + 50 kg P₂O₅ ha⁻¹ whereas, the quality parameters such as protein content, crude fiber and ash percent were significantly higher with NP application of 100 + 100 kg ha⁻¹. The cultivars Hegari and fertilizer level of 100 kg N + 50 KG P₂O₅ ha⁻¹ seems to be the best combination for harvesting higher fodder yield of sorghum under Faisalabad conditions.

Key words: SORGHUM BICOLOR; VARIETIES; NITROGEN FERTILIZERS; PHOSPHATE FERTILIZERS; FERTILIZER APPLICATION; APPLICATION RATES; CROP YIELD; PROXIMATE COMPOSITION.

Did you find this item relevant? [Yes/No/Can't say]

For a reprint, contact first author at : University of Agriculture, Faisalabad(Pakistan). Dept. of Agronomy

Example 3.3: ILRIAlerts output format

The total number of filtered outputs (for instance, seven of the documents for the above particular user) will be listed on a separate column (a feedback form) that is printed for the user to provide a feedback on each of the outputs provided. The feedback output is similar to the ILRIAlerts output given in Example 3.3 above but is an abridged form of the output (i.e. bibliographic details only) as shown in the example below:

Information Services
International Livestock Research Institute
P.O. Box 5689, Addis Ababa, Ethiopia
E-mail:ILRISDI@CGIAR.ORG

=====

User ID: 005
Ato Abate Tedla
Forage agronomist
ILRI
P.O. Box 5689
Addis Ababa, Ethiopia

=====

1. Teste de envelhecimento precoce para sementes de azevem, aveia preta e milheto.
[Accelerated aging test for ryegrass black oat grass and pearl millet seeds] 1999. Garcia,
D.C.; Menezes, N.L. de. Ciencia Rural (Brazil). v. 292. p. 233-237. 19 ref. ISSN 0103-8478.

Did you find this item relevant? [Yes/No/Can't say]
Do you want a copy of this item? [Yes / No]

2. Effect of nitrogen and phosphorus on the fodder yield and quality of two sorghum
cultivars (Sorghum bicolor L.) 1999. Ayub, M.; Tanveer, A.; Kahmud, K.; Ali, A.; Azam, M.
Pakistan Journal of Biological Sciences (Pakistan). v. 21. p. 247-250. 1 table; 24 ref. ISSN
1028-8880.

Did you find this item relevant? [Yes/No/Can't say]
Do you want a copy of this item? [Yes/No]

3. Homegarden agroforestry in Bangladesh. A case study in Rangpur district. 1998. Islam,
M.N. 79p. tables, fig.; 106 ref. Norges Landbrukshoegskole, Aas (Norway). Aas (Norway) :
NLH. M.Sc.

Did you find this item relevant? [Yes/No/Can't say]
Do you want a copy of this item? [Yes/No]

Example 3.4: Feedback receiving format

The user is expected to fill in his/her response, tear the printout into two (separating the two columns) and mail back the feedback copy (i.e. the column containing the feedback questions as shown above in Example 3.4).

Upon receiving feedback, which has been found to be rare from our investigation in the interview with staff, a feedback statistical database maintained by the information section will be updated by counting the number of relevant and non-relevant items as judged by the users in order to compute and record the performance of the output in terms of precision and recall measures to update the users profile in return. The feedback forms will then be forwarded to the library to supply a photocopy of the articles for users requesting a copy on their feedback.

According to the Information Policy and Procedure Manual (ILCA : 1991), on receiving feedback from users, the Precision Ratio for each feedback should be determined using the formula $(a/a + b) * 100$ where **a** is the number of items marked 'relevant' by the user in the Feedback Copy, and **b** is the number of items marked 'non-relevant'. The acceptable Precision Ratio, according to the policy, is in the range of 60 to 70 percent. A ratio less or greater than this range is an indication of profile inefficiency telling either the profile is too broad or too narrow, respectively. In the policy, it is also stated that for each user, Precision Ratios are calculated for the first 4 to 6 months. Thus, in order to conclude that the user profile is effective, a user should receive output which provides an Average Precision Ratio of 60 to 70 percent, calculated over the first six months. If this condition is not fulfilled, the profile will be modified based on an analysis of those items marked 'relevant' or 'not-relevant' in the different outputs. In this case, according to the policy, the user may be contacted in person or in writing if necessary. If the user changes his/her area of interest, the profile will also be modified. A follow-up of the feedback provided by the users is also made to ascertain whether they are providing feedback on a continuous basis. If a user fails to provide feedback for six consecutive times, he/she would be asked for the reason requiring the change of his/her

profile, this is done according to the new interest he/she provides. Otherwise, he/she will be cancelled from the list of SDI users and a letter to this effect would be sent to the user.

Although the Information Services at ILRI started sending SDI outputs by e-mail for those that have e-mail addresses, the majority of the recipients are still receiving by postal mail. Thus, the filtered documents will then be produced in one of the two ways depending on the delivery mechanism:- for e-mail recipients the outputs are written to a file in order to be forwarded by e-mail; and for postal recipients the outputs are directly printed to be mailed to individual postal addresses.

Currently, a database is maintained to keep a statistics of the feedback received and to calculate and record the Precision Ratio for each user sending a feedback. This database is also being used to track users not providing feedback.

3.3 Methodology

The previous study by Dawit (1998) focuses on the maintenance of the user profile with particular emphasis on updating users profiles through tracking new users interests by scanning through the project documents that the users are currently working on. The main focus of the current study, however, is on the maintenance and updating of the users profiles based on an analysis of the users' relevance feedback records. It is, therefore, important to give a brief description of how the researcher undertakes the use of relevance feedback in personalised information filtering in the following sections.

3.3.1 Preparation of test data

As indicated above in this chapter, the ILRIAlerts output sent out and feedback received are kept for a maximum of six months and then they will be purged. Therefore, the feedback data available for use by the current research was the records of the last six months' (October 2000 to March 2001) ILRIAlerts output data and the feedback copy received at the library as the result of users requesting a photocopy of articles. The population of the feedback record would be our experimental data as shown in the table below:

User ID	Feedback for the month	Number of abstracts judged by the user as	
		Relevant	non-relevant
087	December	4	5
173	December	16	9
522	December	3	2
522	February	4	5
522	March	4	3
529	November	8	9
515-T	October	11	16
515-T	November	7	12
267-T	December	14	9
532-T	January	2	4
516-T	February	11	4

Table 3.1 User feedback record

Among these users, the feedback records of the users ID with a suffix "T" was not available due to the fact that these users receive their monthly SDI output by postal mail and their output is directly printed and forwarded to them by post. The records of those users receiving ILRIAlerts output by E-mail were only available of the last six months as stated above.

This research has therefore used the feedback records of four of the above users (user ID no. 087, 173, 522 and 529).

For the purpose of the filtering operation (testing the updated model as per the feedback analysis), 1500 abstracts have been used from the recently added bibliographic records into the ILRIBIB database. The experimental feedback data of about 72 abstracts of the documents from 4 users' feedback record of the last six months' ILRIAlerts service.

In previous studies also such as Asnicar, Fant and Tasso's (1999) on evaluating IFTool (an Information Filtering Tool), they experimented by performing a filtering operation of 2000 abstracts of technical reports on the basis of the information preferences of 4 subjects.

Data were collected first by going through the completed feedback outputs returned from the users and then looking at the judgments of the users, those documents judged as 'relevant' or 'non-relevant' and 'can't say' were identified and from the electronic copy in feedback folder files available with ILRI information staff, the individual documents' details were identified and the abstracts of the documents under each users' folder were extracted. Each new abstracts of relevant and non-relevant documents added to the users' folder were examined using vector space model to identify key terms that represent the documents judged as 'relevant' or 'not-

relevant'. The key terms will therefore be used to update the user model. The detail of the selection of key terms and updating the user model is described in chapter 5.

3.4 Technical representation of users' profile

As indicated above, in this study, the whole idea of obtaining feedback is to be able to continually update and refine the user profile so that the system retrieves more and more relevant items and less and less of the non-relevant items. Therefore, it would also be important to look at how the information obtained from the feedback can be used to represent user interests and update the user profile. This is what will be described below.

A traditional method of automatic indexing text consists of recognising individual words, eliminating the commonly used words included on a word-exclusion list (stop list) and using the remaining words for content identification of the texts. As the terms are not all equally important for content representation, importance for the text content identification should be given. The text is then representable as a vector of terms.

Research in information retrieval has given rise to many retrieval models, notably the boolean model, the vector space model, and the probabilistic model, that are applicable to information filtering. (Garcia-Molina and Yan : 1993). In this study, in representing the user interests (based on relevant and non-relevant judgments obtained from users) from the feedback i.e. abstracts of documents, we consider the vector space model (VSM), which is widely recognised as an effective retrieval model. (Baeza-Yates and Ribeiro-Neto : 1999). Our

methods are thus for the incoming 'relevant' and 'non-relevant' documents (abstracts) to be represented in VSM.

3.5 Limitations of the existing system

The general concept of any SDI system is that users define subject topics in which they are interested and these are automatically matched against items that are added to the database and users are automatically alerted. (Tedd and Yeates : 1998). However, the general process of filtering and refining / updating users' profiles dynamically of the existing system (ILRIAlerts), according to my observation and survey is a time consuming, difficult and costly operation. The reason for this is many but some of these include the following:

- 3.5.1 Feedback from users, which is a very essential part in keeping user preferences up-to-date is rarely obtained. Currently, very few users, around 5 of 127 active users regularly provide feedback. This may be due to the lack of close interaction between the system and the users as they are, in fact, widely distributed (i.e. Sub-Saharan Africa). On the part of the system, there is no continuous follow-up to maintain / update the users' interests and new areas of preferences due to the few number of intermediaries. On the other hand, the time consuming feedback procedure of item-by-item relevance feedback analysis may be tasking and time consuming for the users.
- 3.5.2 There is no simple mechanism to chase the users provide their feedback on the SDI outputs they receive. With the widening of users receiving SDI outputs on e-mail, it may be necessary to alert users to proactively use the service.

- 3.5.3 The task of analysing the limited feedback received is restricted to only keeping statistics on a MS Access database which records the number of items sent, the feedback statistics (number of items rated as relevant or non-relevant), number of item for which a copy was requested, and the precision ratio calculated, etc. This information is kept solely to track the number of active users rather than updating the profiles based on this information. It is found out that most of the user profiles have not been updated for more than three years now. The system has not tried to investigate the new possibility of obtaining regular feedback and then using this information update the user profile.
- 3.5.4 The researcher observed that a huge list of items (SDI output) containing more than a 100-page printout for a monthly output. This may show that the optimisation of the user profile needs to be done. It shows that the profile consists of broad areas of the users interests. Thus, elimination of unnecessary terms from the user profile has to be done.
- 3.5.5 Due to a shift of the ILRIBIB database on which the SDI output is produced, from the CDS/ISIS based database to INMAGIC database management software, the matching is not now done automatically. Intermediaries cut the keywords from the profile file and paste it to the INMAGIC database query form to search and do the matching one-by-one for each user. The filtering activity is now, therefore, time consuming and tasking on the part of the system. The intermediaries said that they are working to automate the matchmaking process.

CHAPTER IV

THE PROPOSED INTERFACE AGENT-BASED INFORMATION FILTERING SYSTEM

4.1 Introduction

In the previous chapter we tried to describe the operation of the existing ILRIAlerts system and the methodology that this research undertakes including the limitations of the system under study which we deal with in our proposed agent-oriented model in this chapter. It is clear from the previous investigation as well as our findings in this research that the existing SDI systems are unable to offer a complete service using human intermediaries. Thus, it requires automation in both user and system side operations. In this chapter, a review of the previous proposed system - model which is also the basis for this study is done and is further expanded and refined to include and entertain the problems identified in my investigation. The researcher, therefore, presents a proposed model integrating the previously proposed model and adding the components that accomplish the problems and activities that the proposed agent-oriented model undertakes. A number of other user-focused information filtering models have also been consulted in which some of them are found to be relevant to be adopted in this particular case. Although most of the previous studies on information filtering are based on the resources of the World Wide Web, particularly USENET and news articles, the concept and approach would also be useful for conventional document filtering and retrieval systems like the ILRIAlerts system. Some of these systems include Sheth's (1994) 'Newt' system which is

designed as a collection of information filtering interface agents which learn users' preferences and act on their behalf using a key word based filtering algorithm; Tedd and Yeates' (1998) 'NewsAgent' is a project which aims to create a user-configurable electronic news and current-awareness service for library and information professionals with a mixture of content streams, including metadata are described; Soltysiak and Crabtree (1998) have in their study also mentioned some examples of personal agents that are capable of operating autonomously in order to provide timely and relevant information for an individual. Some of these examples of such agents (Soltysiak and Crabtree : 1998) include GRAPEVINE, an interest matchmaking agent, RADAR, an information presentation agent, and PANDORA, an agent that suggests topics that may be of interest to a user. In all these agent-based information filtering systems, the central framework is the user profile which describes the interests of a user.

In the previous study as well, the central framework for the proposed agent-oriented model was the user profile. To start with, therefore, the generalised model of the previous study is presented and then by focusing on the user-side of the system, the new model of handling the overall information filtering activity of the ILRIALerts system and the prototype Feedback Handling Agent model is presented in detail.

4.2 Previous Model

As discussed above, Dawit's (1998) model tried to incorporate the components of the SDI system that are amenable to agent application. It however focuses on the model for the use of agent technology for particular problems pertaining to user profile management with special emphasis to continuous updating of the user profiles and their optimisation i.e. based on new

interest of the users. Soltysiak and Crabtree (1998) also emphasise on the user modeling for information filtering, they strongly affirm that user profiling is fundamental to the personalisation of agent technology.

Emphasizing the importance of improving the user profile management through distributed processing, the approach to build up on the model of client/server is well argued in Yan and Garcia-Molina (1994). Dawit (1998) mentioned that such architecture is widely implemented indicating specific cases such as the EBSCO Information Service's System and Web SPIRS of the SilverPlatter. The SDI system itself requires human-intermediaries that liaise between the information sources (databases) and the information users. Therefore, the resulting client/server architecture of the SDI environment was suggested to look like as follows (Dawit : 1998):

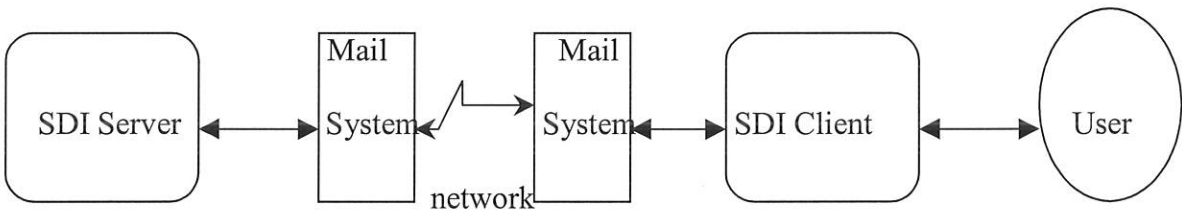


Figure 4.1 A Client/Server SDI Model

On the client side, a number of operations some of which are shared with the user are taken which include providing the user with an easy interface to the SDI system, delivery format conversion facilities, ranked display of outputs, effective browsing of deliveries, feedback gathering and processing, detection of user interest changes, alerting users to give feedback and many other that need close interaction with users.

The server side of the SDI system is as important as the client-side which mainly deals with the information filtering operation. It performs the conversion of user requests in the form of search expressions of the database or other information source and also conducts search on the database including delivering the results to the user.

Implemented in agent-oriented model, the previous study identified six basic functions, these are *formulation of initial profile, testing and refining profile, searching and delivering results, following-up of performance of profile, tracking user's new interests* and *updating user's profile*. (Dawit : 1998). From the user profiling point view, the three functions namely *follow-up profile performance, track user's new interests* and *update user's profile* involve continuous interaction with the user and, hence, require closeness to user related information sources. Therefore, this feature makes up the client side for the automated system that assist these operations. The other three being operations on the server side. The previously proposed agent-oriented model was the following:

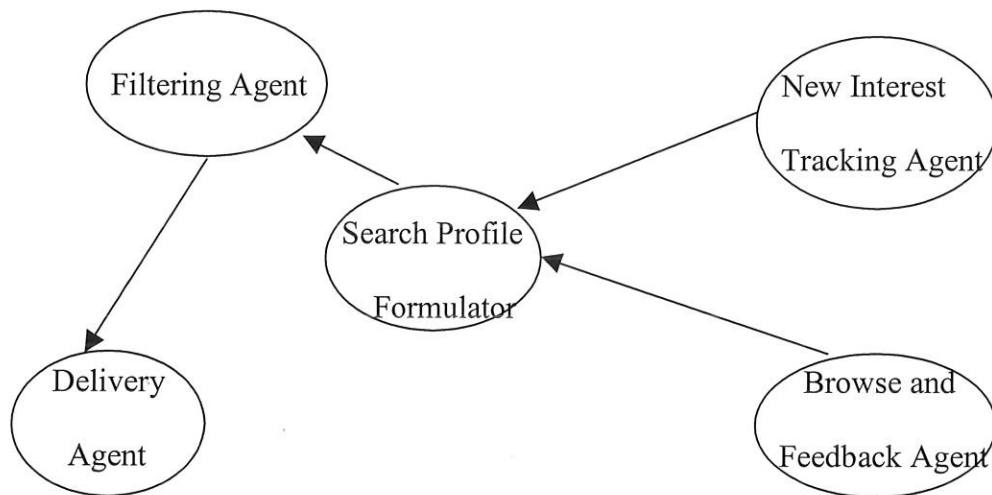


Figure 4.2 A Model for Agent Oriented SDI System

4.3 The New General Model

Apart from the basic problems related to filtering in general and updating user profiles based on new interests / change of interest area or feedback, one of the problems as discussed in the previous chapter is the problem of users not providing regular feedback. If the users don't provide feedback for whatever SDI output they receive, it is no use to develop a mechanism for refining and updating user profile based on relevance feedback. Therefore, one of the main additions in the new proposed system would be a 'reminder / remembrance agent' role within the client-side agent architecture and integrated within the Feedback Handling Agent role so that users are reminded of receiving SDI outputs and asked to give feedback for. This agent could also be empowered to learn from the way users give feedback and simplify the provision of feedback in future. This leads to the automatic classification of user interests to reduce the

need for user feedback as investigated by Soltysiak and Crabtree (1998) i.e. the automatic generation of user interest profiles with minimal user input.

The other addition to the previously identified agent components in terms of feedback handling is specifying the responsibilities of the Browse and Feedback Agent to include the receiving of internal representation of the corresponding documents of which the user gives feedback on and update the user model accordingly. The updating mechanism which we will be described in detail in the sections below will be constituted by changes in the weights associated to terms in the user model.

Based on the above general modifications, the basic functions previously identified can be expanded so that on the user-side model, three basic function can be achieved, namely *track user's new interests*, *Alert / remind user to give feedback*, *update user's profile* which involve continuous interaction with the user. All these functions confirm for the position of the client side operations that needs automated assistance. On the other hand, systems consisting of those identified previously, namely *formulate initial profile*, *test and refine profile* and *search and deliver* results which are technical operations involving database and intermediary expert inputs, are believed to be placed on the server side. However, with the additional capability of the Feedback Handler Agent described above, the testing and refining operation on the profile by the intermediary can be minimised. Once we formulate an initial profile in consultation with the user, then it could be tested and refined through the relevance feedback process. As mentioned earlier in chapter 2, this has been argued by Baeza-Yates and Ribeiro-Neto (1999) where in user profiling process instead of the user to be asked to provide his interests using key words and the database expert using this information formulate a profile to undergo a

testing and refining process, they give a more elaborate alternative method which is to collect information from the user about his preferences and to use this information to build the user profile dynamically. This, according to Baeza-Yates and Ribeiro-Neto (1999) can be achieved as follows:

... In the very beginning, the user provides a set of key words which described an initial (and primitive) profile of his preferences. As new documents arrive, the system uses this profile to select documents which are potentially of interest and shows them to the user. The user then goes through a relevance feedback cycle in which he indicates not only the documents which are really relevant but also the documents which are non-relevant. The system uses this information to adjust the user profile description such that it reflects the new preferences just declared.

This operation is, therefore, classified to be included in the Feedback Handling Operation at the user-side operations. This reduces the operations to, namely *formulate initial profile* and *search and deliver results* to be placed on the server side. It is not, however, believed such categorization would by no means be considered as exclusive as several implementations of similar studies categorise operations in this or the other side and also depend on the environment's priority problem areas. For modeling the client side agents, Reticular Systems Inc.'s AgentBuilder agent structure and Kendall et al's (1995) methodology have been employed.

The above explanation will better be elaborated by the general overview diagram below:

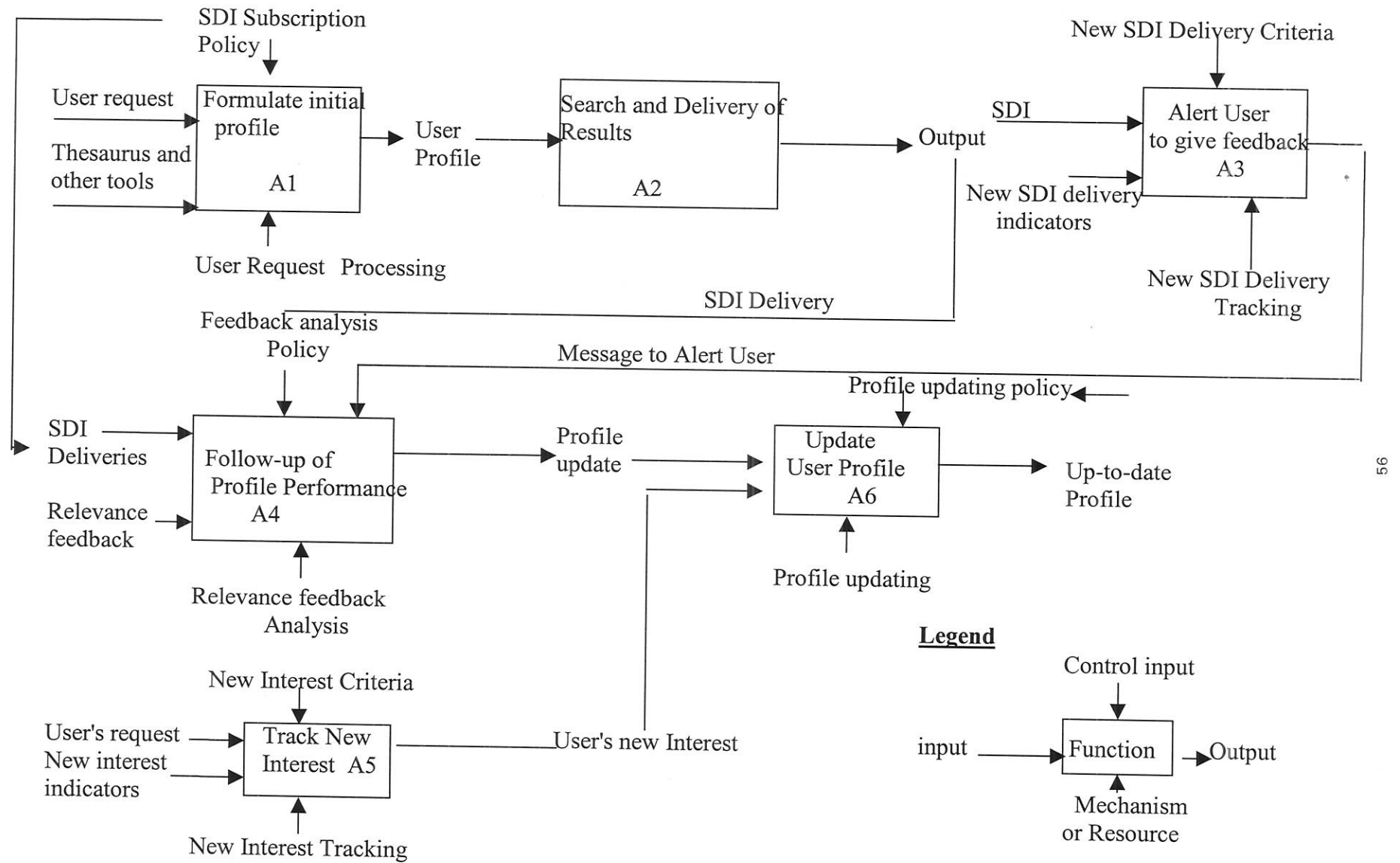


Figure 4.3 Level 0 Overview Diagram of the SDI Service

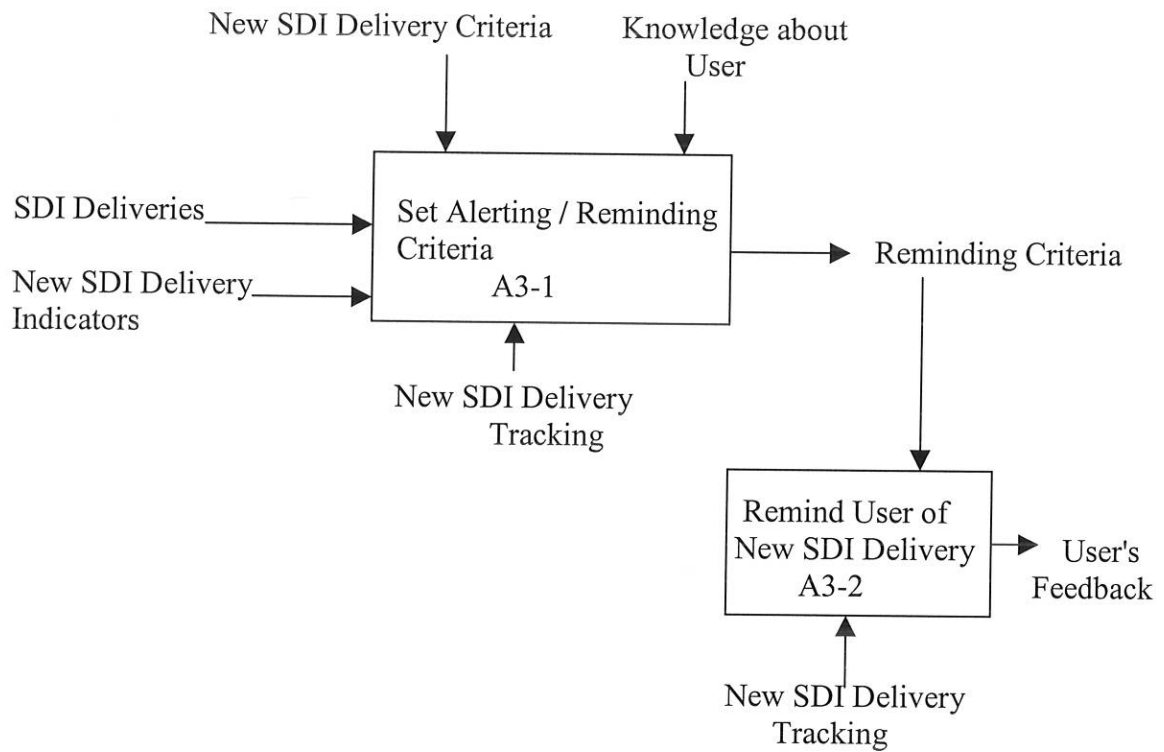


Figure 4.4 Alert / Remind User to give Feedback Level 1 Diagram

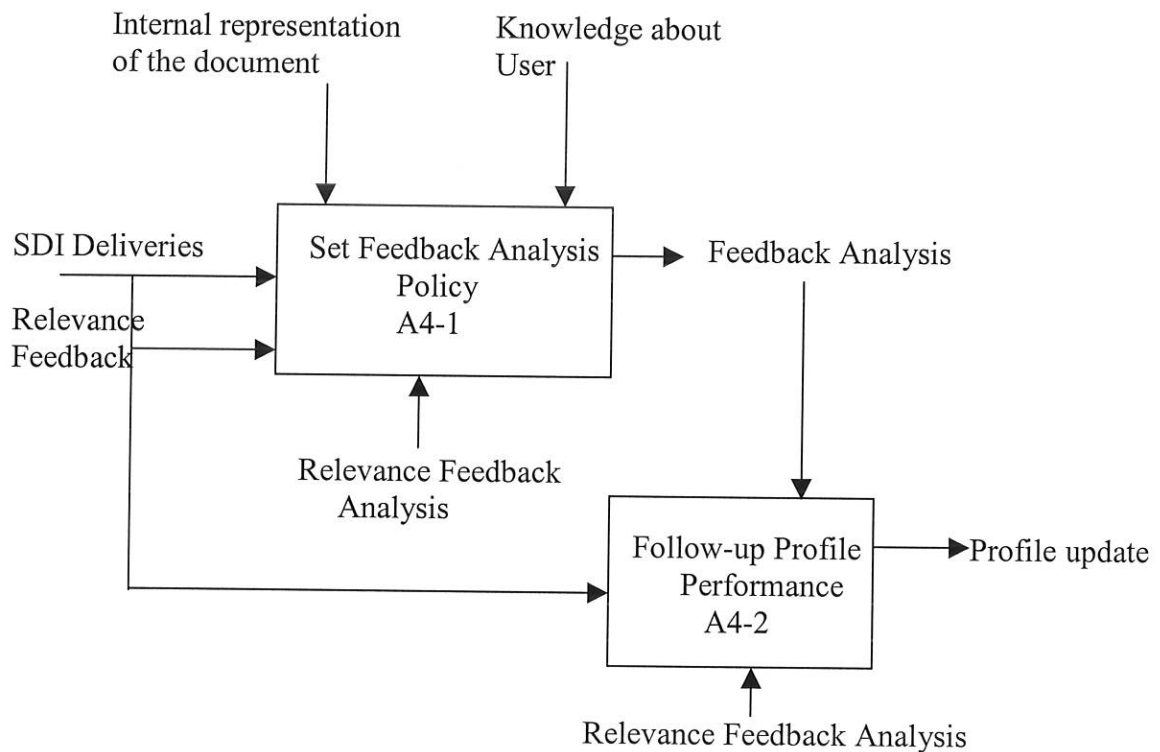


Figure 4.5 Follow-up Profile Performance Level 1 Diagram

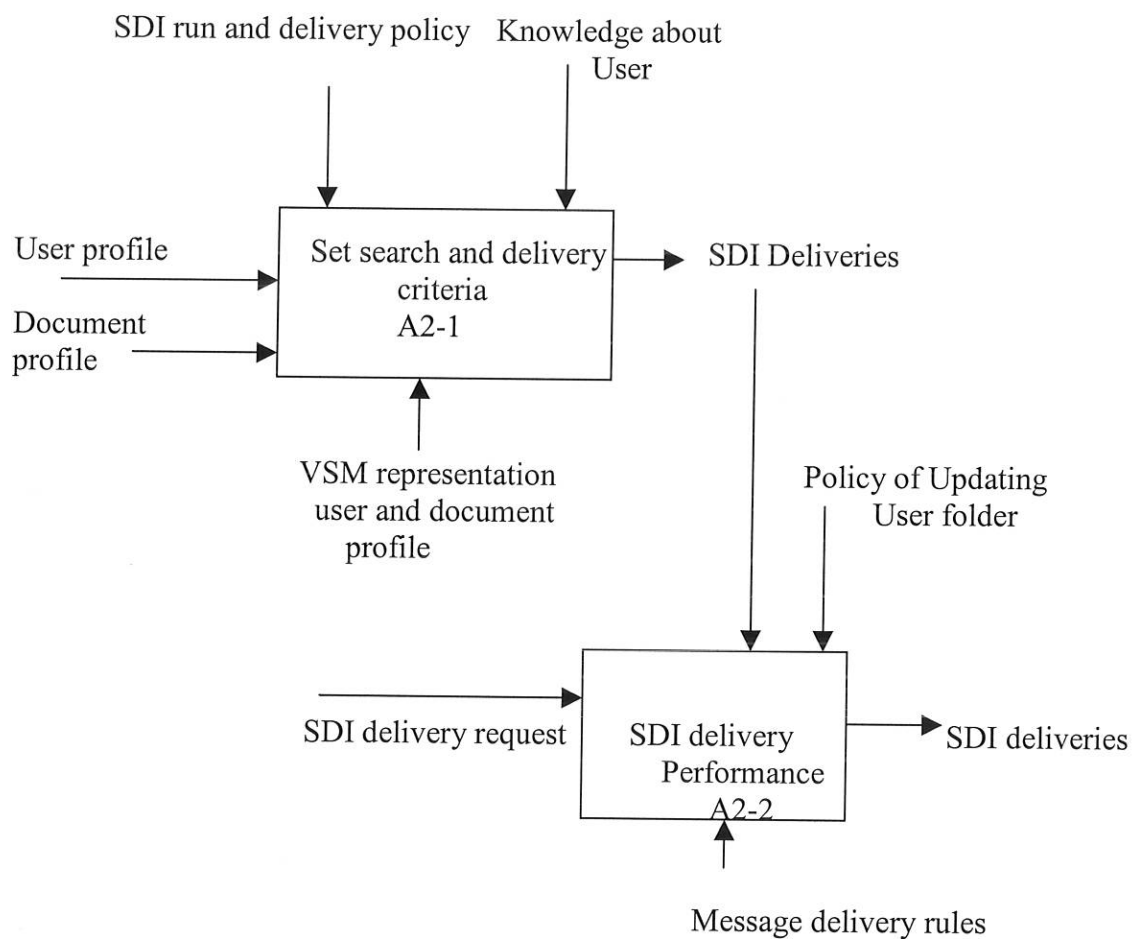


Figure 4.6 SDI search and delivery Level 1 Diagram

4.4 The New Agent-Oriented Model for Information Filtering

The above analysis has shown a new and detailed agent-oriented modeling of the existing SDI system. Based on the above analysis, therefore, six agents have been identified to assist the user in various decision-making and monitoring operations. These agents are named as *Reminder Agent*, *Feedback Handling Agent*, *New Interest Tracking Agent* on the user side (client side) and *Search Profile Formulator*, *Filtering Agent* and *Delivery Agent* on the system

(i.e. server side) as shown below. Several other specialist agents may be drawn from the respective side operations such as the internal document representation agent, etc.

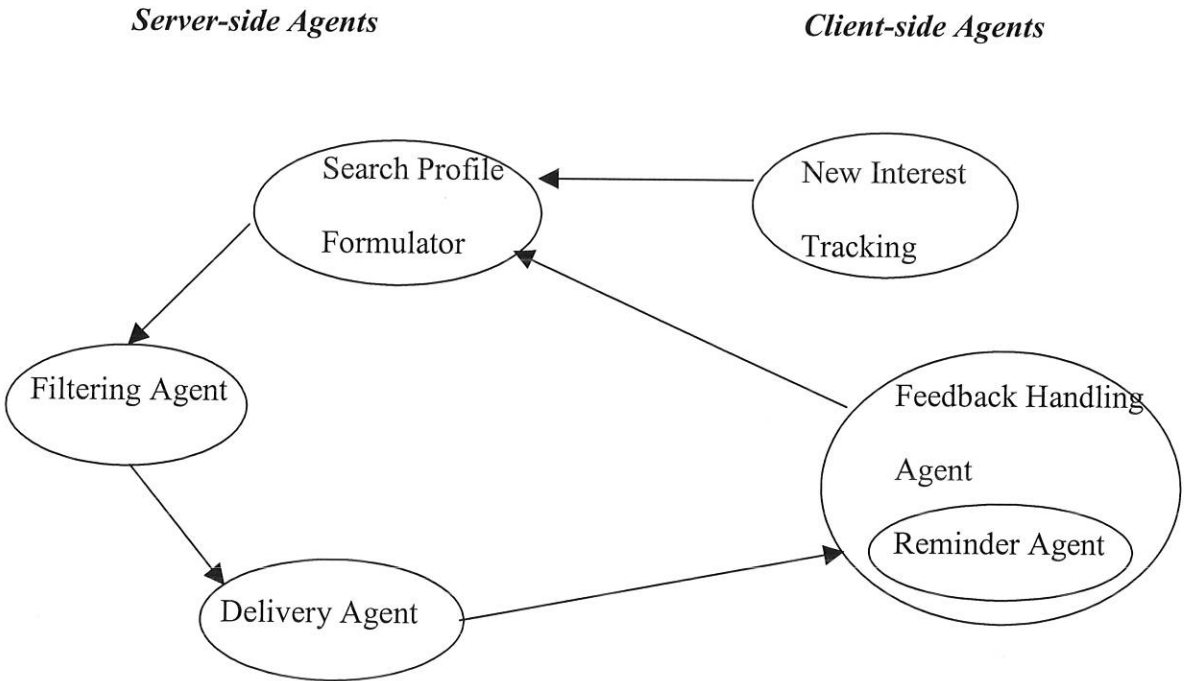


Figure 4.7 A New Model of Agent Oriented SDI System

Although the intention of this research is to focus on the client side of the SDI operation as it is fundamental to agent-based filtering of information, it is useful to give a brief description of the server side operations. The actual work of filtering should therefore be delegated to the server side operations various approaches of the implementation of which has been discussed in several literatures reviewed in this study. For the purpose of demonstration in this work, communicating (two) agents as opposed to the case in the previous one, autonomous agent development i.e. non-human-intermediary-based client-server agent communication is assumed and all the outputs of the server-side interface agents are intended to be used by the client-side agents.

4.5 The Proposed Agent Design

For a full-fledged agent-based information filtering system, it is required to implement all the client and server side agent components. However, due to the limited time available for this research, we only deal with some of the client-server side agent components which haven't been dealt in the past study by Dawit (1998) but which are, as discussed above, fundamental for information filtering system. In this section, we present the assumptions, requirements and technical specifications of the agents that are incorporated partially or in full in the prototype with the exception of the New Interest Tracking Agent which has been dealt with in greater detail in the previous study by Dawit (1998).

4.5.1 Design Considerations

In the previous section, the general client/server framework sketch has been given which can be noted that components depend with one another. As shown in figure 4.7, three agents have been identified, namely, the *Reminder Agent*, the *Feedback Handler Agent* and the *New Interest Tracking Agent*. The New Interest Tracking Agent, which aims to track the new interests of the user and predicts the likely interests of the user, has been the main area of investigation for the previous study (Dawit : 1998) and therefore, we will not discuss here. The Feedback Handling Agent has been decomposed to handle a number of roles i.e. incorporating a reminder role but mainly the handles the feedback functions both positive and negative feedbacks. The Feedback Handling Agent activated by the Reminder Agent displays SDI deliveries and receives as input the judgments possibly provided by the user and the

representation of the document for which feedback has been given and then analyse the relevant and non-relevant documents to identify potential key words to help update the user model accordingly. Updates are performed by changes in the weights of the associated terms in the user model taking into account the frequency of occurrence of terms and of co-occurrence of pair of terms in the considered document based on a certain threshold comparison. The setting of threshold for feedback handling is discussed below in this section. The Feedback Handler Agent allows to better follow the temporal evolution of the user information preferences and this is obtained by means of an increase or decrease of weights associated to terms in the user model. This allows to highlight the terms that better represent the user preferences. The Feedback Handling Agent is activated periodically on the user model and they identify the changes which have to be performed on the model so that all the terms which have a weight lower than a given threshold are recommended to the user for deletion from the user model.

The structure of the agents for the client system are shown in figure 4.8 below:

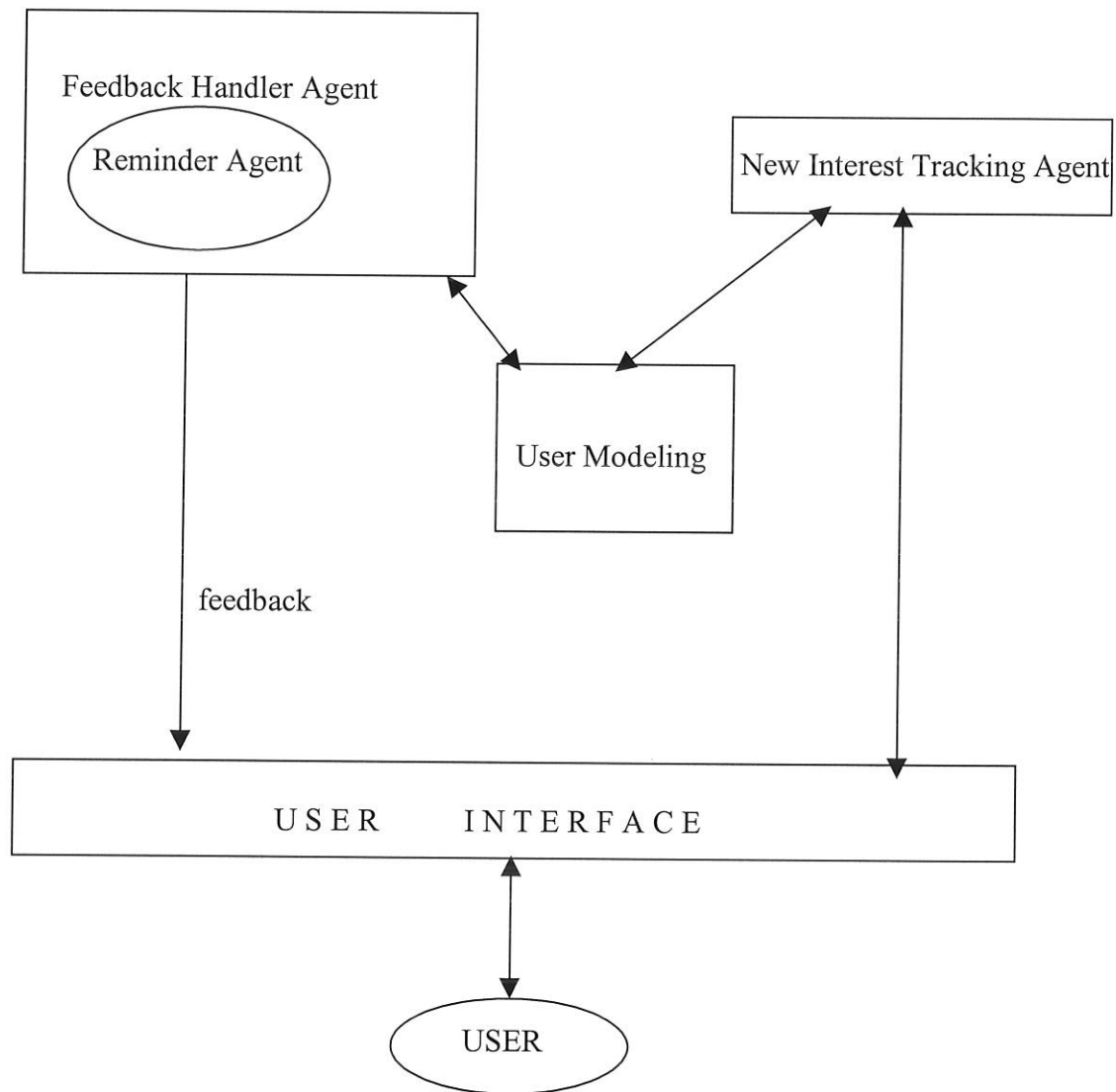


Figure 4.8 The Structure of the SDI Client System

The above SDI client system structure shows the applications that assists or act on behalf of the SDI service user in matters related to providing information needed by the SDI system to update the profile of the user based on the actual and current needs and based on the relevance

feedback. Through this interface which automates the process of feedback provision and then updating the user model, it also assists the SDI service providers:

- by relieving the intermediary from the tasks of follow-up or reminding the user to give feedback looking at the SDI deliveries
- by relieving the intermediary from the tasks of user-interest follow-up as has been tried to develop the New Interest Tracking System in Dawit's (1998)
- by refining the user profile with minimal input from the user particularly in terms of the evolution of user interests through the elimination of the terms which are no longer representing users needs and vice versa.

The client system can do these including assisting the user by reminding them the receipt of the SDI deliveries as these functions are close to the user and it would be easy for the system to draw information from various network available resources to users.

The proposed agent-based client system has also employed various techniques and procedures which have been reported to perform better in various other researches in information filtering and software agents including the previous study (by Dawit, 1998) undertaken on the same case. With the exception of the New Interest Tracking operation which has been dealt in the previous study, we describe below the assumptions, techniques and procedures used in designing the client interface agent-based system.

- A vector space model based processing is employed for analysing the abstracts of documents on which the relevance feedback has

been received. As opposed to the existing boolean based server for documents databases, the client is designed to serve as a vector space based interface. The consideration of this approach is to take advantage of the various ranking and weighting techniques made possible using the vector space model. These techniques are applicable both for analysing the documents to identify terms and change their weights accordingly. Thus, it is designed to include an enhanced relevance feedback analysis in addition to the currently practiced precision ratio calculation (as in the case of ILRIAlerts).

- Each user's SDI output is saved under individual folder by the intermediary for the delivery agent to regularly scan through the folders for new output to deliver it to the respective users.
- the Reminder Agent assumes that users have personal computer and e-mail address and, therefore, on receiving the e-mail message, the agent activates the user interface for feedback provision.

4.5.1.1 The Feedback Handling Agent Problem Domain

As stated by Allan (1996), relevance feedback techniques provide a means for automatically correcting a query (in our case a user model) to more accurately reflect the user's interests: a set of 'good/bad' relevance judgments on documents are 'fed back' into the query (user model) to generate a better query (user model). It is a supervised learning method used to improve the effectiveness of information retrieval systems.

Therefore, the Feedback Handler Agent is an interface agent to handle an improved filtering mechanism by providing the user with a user friendly interface in order the user look at the SDI deliveries and provide feedback. It, as previously suggested, uses the user model to rank the SDI deliveries according to relevance to user based on a ranking algorithm. As indicated in Haines and Croft (1992) the Feedback Handler Agent receives the relevance feedback, i.e. the judgments of users based on a few and highly-ranked documents, which is then used to modify the queries (user model in our case) in order to increase the performance of information filtering system.

The Feedback Handler Agent, therefore, achieves three goals by employing relevance feedback for information filtering:

- Receiving the SDI deliveries i.e. the SDI search results delivered by the server, it ranks and displays to the user
- Upon receiving the relevance judgments of the user, it analyses the relevant document records and identifies the terms (key words) occurring in the relevant documents that are to be added to the user model.
- Analysing the documents judged by the user as 'irrelevant', it identifies the terms (key words) that are to be deleted from the user model.

As we have seen above and as opposed to the use of vocabulary excerpts and term frequency information or citation to the source documents previously identified as relevant by the user to

construct improved queries (user model), we are employing an alternative user model alteration process. This process follows the display of information i.e. SDI deliveries ranked against similarity measures with the user model. Below, we will give a brief explanation on how the ranking of documents i.e. SDI deliveries is done.

4.5.1.1.1 Document Ranking

We also propose what previously published literature reports that among the methods usable for improving the display of output (SDI deliveries) is the incorporation of weighted instead of binary terms and the presentation of the output in decreasing order of the query (user model)-document similarity. According to Salton and McGill (1983), the generation of ranked document output increases user satisfaction and retrieval precision. Therefore, for simplicity, it is proposed to employ a statistical model that orders (ranks) the SDI outputs according to statistical similarity between the user model and of each of the records represented in vector space model. This approach has been used before in a number of systems such as the wide-area information system (WAIS), the SMART system, the SIRE (Syracuse information retrieval experiment) system and the CITE system. (Salton and McGill : 1983).

One of the common similarity measures, i.e. the **cosine measure**, measures the cosine of the angle between the documents vector (i.e. SDI deliveries) and the query vector (i.e. the user model) when these are viewed as vectors in the multidimensional term space of dimension t . Thus, the ranking function defines the similarity between a document DOC_i (i.e. a record in the SDI delivery) and a query Q (in our case the user model) defined as (Salton and McGill : 1983):

$$\text{COSINE}(\text{DOC}_i, \text{QUERY}_j) = \frac{\sum_{k=1}^t (\text{TERM}_{ik} * \text{QTERM}_{jk})}{\sqrt{\sum_{k=1}^t (\text{TERM}_{jk})^2 * \sum_{k=1}^t (\text{QTERM}_{jk})^2}}$$

where QTERM_{jk} is the weight of term k assigned in the user model which is also defined in the same way as TERM_{ik} (i.e. the weight of the term).

The normalisation factor as reported in various implementations in (e.g. Salton and McGill, 1983) and as quoted by Dawit (1998) (in Lee, Chuang and Seamons, 1995) to perform well for queries (to which the user model can be categorised), both in terms of accuracy and computational efficiency was considered. This normalisation factor is also much easier to compute than the original one given above. The formula for this approach quoted by Dawit (1998) from Lee, Chuang and Seamons (1995) is given as follows:

$$\text{Sim}(Q, D_i) = \frac{\sum_{j=1}^n wQ_j * w_{ij}}{\sqrt{\text{number of terms in } D_i}}$$

where Q is query (user model), D_i is document i , wQ_j is weight of query term j .

4.5.1.1.2 Relevance judgment analysis

The other major function that the Feedback Handler Agent performs, is receiving the relevance judgments (feedback) of users to analyse their feedback. In this process, the agent

employs the feedback analysis and feedback based learning. Relevance feedback has been employed in a number of agent-based implementations, such as in (Sheth, 1994; Armstrong, et al 1995; Lang 1994; Edwards, et al 1996).

Thus, the Feedback Handler Agent analyses the relevant and non-relevant documents judged by the user in order to identify the potential terms (key words) to be added to the user model and key words that should be removed. To this effect, an initial vector space run is done with the user model, running on the "learning document set" (the records set periodically received by the user). The Cornell 'lrc' weighting scheme that is commonly used with the vector space model of text retrieval is proposed. Each user model term, q_k , receives a weight that increases as the number of times the term (q_k) occurring in the user model increases (term frequency contribution), but decreases as the proportion of documents the term occurring in increases (the idf or inverse document frequency contribution). Thus, the entire query (user model) vector is normalised using the cosine length of the vector:

$$q_{ik} = \frac{(\log(f_{ik}) + 1.0) * \log(N_D/n_k)}{\sqrt{\sum_{j=1}^t [(\log(f_{ij}) + 1.0) * \log(N_D/n_j)]^2}}$$

where f_{ik} is the number of times term k occurs in Q_i , N_D is the collection size, and n_k the number of documents with the term k assigned.

While the documents (i.e. SDI deliveries) are weighted with the "lrc" weighting scheme which is the same as the "lrc" query (user model) scheme, except no idf factor is used as is suggested

in Buckley, Salton and Allan (199). This "lnc" document weight, "ltc" query weight combination proved very effective in TREC 1 and TREC 2.

Based on this weighting, the Rocchio's similarity-based relevance feedback algorithm is suggested to learn from each SDI delivery and produce a vector (called prototype) which will be suggested to the user, modified and used to update the user model. This algorithm is one of the important query reformation methods in information retrieval, which is essentially an adaptive supervised learning algorithm from examples. It is also chosen here in our case as it is used in wide range of information retrieval agents (for instance, as quoted by Dawit (1998) include Armstrong, et al 1995; Lang 1994; Edwards, et al ; Balabonovic and Shoham 1995). The **training set** in our case will be the SDI deliveries (records) periodically received by the user from which good and bad examples are taken for training. The range of records on which a monthly SDI run (searching against user profile) takes place is considered as the **test set** for the prototype produced by Rocchio algorithm. Therefore, what is learnt from the SDI deliveries is used to update the user profile (search profile) and tested on the database (range of records for the next SDI run). Thus, each SDI delivery will be a new learning source for the Feedback Handler Agent.

We also employ as previously suggested in Dawit's (1998) the expanded query weighted using the modified Rocchio formula used by Cornell in TREC 2 as follows:

$$Q_i^{new} = \alpha Q_i^{old} + \beta \frac{1}{|rel\ docs|} \sum_{rel\ docs} w_{ti} - \gamma \frac{1}{|nonrel\ docs|} \sum_{nonrel\ docs} w_{ti}$$

Query terms (terms in user model) with overall negative weights are dropped.

The constants α , β and γ control the relative impact of positive and negative examples on the prototype. For these, the standard constant values $\alpha=8$, $\beta=16$ and $\gamma=4$ are suggested by Salton and McGill (1983). As suggested by Dawit (1998) normally the prototype that results from the Rocchio algorithm may directly be used to compare with the test set (range of records for SDI run) in the current agent, but instead the terms are suggested to the user as the matching with the database is Boolean and that the user's approval should be obtained before a term is sent to the SDI server.

One of the problems of using current best-match methods in a filtering system and task is that of setting appropriate thresholds, which are required in order to force a binary decision on notifying a user of a document. (Robertson and Walker: 2000). One of the characteristics that distinguishes filtering from ad hoc searching is the requirement for a threshold for decision-making purposes. It is, therefore, needed to employ a method for setting such thresholds and adapting them as a result of feedback information on the performance of the profile.

For setting a threshold, unlike probabilistic model or empirical methods to adjust the threshold, we employ a model free threshold adjustment method that the current query formulation (user model in our case) searched against the accumulated collection at the time (Robertson and Walker: 2000). Under this method, output is ranked, and the known relevance judgments are used to evaluate the appropriate utility function at every rank position. The score of the document at the rank which maximised utility is taken as the threshold (the last such document if more than one gave the same maximum utility. For each topic, the utility of

a particular outcome (set of documents retrieved the specified period) was defined as (Robertson and Walker: 2000):

$$\text{Utility} = u_1r - u_2(n - r)$$

where r is the number of relevant documents retrieved, and n is the total number retrieved. u_1 and u_2 are, respectively, the unit utility associated with retrieving a relevant document and the penalty (negative utility) for retrieving a non-relevant document.

According to Robertson and Walker (2000), the two specific TREC utility functions were F1 ($u_1=3$ and $u_2=2$) and F3 ($u_1=4$ and $u_2=1$). It can be shown that the criterion of optimising F1 is equivalent to applying a rule that says 'retrieve if $P(R/D) > 0.4$ '. The probability of relevance is denoted by $P(R/D)$, here D is taken to be all the information we have about a document – its description (a specific query is assumed). Similarly, the probability corresponding to F3 is 0.2. (Roberts and Walker: 2000)

4.5.2 Agent Role Model

From our initial specifications, we notice that the client-server side SDI system is based on a matchmaking / delivering and receiving feedback chain to continually refine output to users' needs. This kind of chain of supplying SDI system output based on user profile has been modeled as simple market model for buyer and seller, in this case the item being sold or purchased is information items (document articles). Such models have been implemented in Ferguson and Karakoulas' (19). The general roles of the agents under consideration is described in the table below:

Agent Name	Roles
New Interest Tracking	Tracks the new interests of the user from user documents by regularly scanning through personal folders
Delivery	Delivering the results of the matchmaking i.e. SDI output to individual folders (addresses) - (Output request receiver), (Messenger)
Reminder	Reminds users on delivery of SDI output and facilitate feedback provision back to the system - (Output requester), (Messenger), (Chaser)
Feedback handler	Ranked display of SDI output and receiving and analysing documents based on feedback to suggest terms to be added to or deleted from the user model - (Displayer), (Feedback Receiver), (Feedback Analyser), (Update recommender), (Modifier)
Search Profile Formulator	Maintains the user profile based on initial user needs and continuously updating it based on suggestion from New Interest Tracker and Feedback Handler agents
Filtering	Undertakes the matchmaking of the user profile and the range of newly added records each month

Table 4.1 Agent roles

This general role indicates that there is a communication between at least two agents or agents and user or the intermediary which signifies the exchange of information items. For instance the reminder agent requests the delivery agent to provide the recent SDI output and on receipt reminds user to view SDI deliveries and provide feedback. On provision of feedback from the user interface, the feedback handler will be activated to handle its role.

Having identified what roles should exist within the application domain, the next stage is to think how each agent realises each role which will be described in the list of agent responsibilities below.

4.5.3 List of Agent Responsibilities

Each role played by agent entails some responsibilities, for example resources that will be needed to be produced or consumed, interaction with external systems, etc. We will therefore use the role description to list the responsibilities of each agent. In the next sections, we will only deal with the agent components that will be considered in the prototype design in the next chapter.

4.5.3.1 The Delivery, Reminder and Feedback Handling Chain

With the above assumptions and design considerations, the delivery agent is part of the server-side agents responsible to deliver SDI outputs on a regular basis. It resolves the problem of human-intermediary to individually delivering SDI outputs to the respective users' folders and then forwarding to each address. The following tables show the social and domain

responsibilities, the former involving interaction with other agents, and the latter involving some local application-specific activity; this results in the following:

The Delivery role - social responsibilities	
Origin	Responsibility
Output request receiver	To be activated to scan individual folder
Messenger	To send SDI deliveries to user personal folders (addresses)

Table 4.2 The Delivery Agent Social Role

The Delivery role - Domain responsibilities	
Origin	Responsibility
Output request receiver	To receive demand
Messenger	To update personal folder i.e. deleting files already delivered output on six monthly basis

Table 4.3 The Delivery Agent Role

The next role to consider in the delivery, reminder and feedback handler domain is the role played by those agents who both request SDI output from other agents and remind user to provide feedback.

The Reminder role - social responsibilities	
Origin	Responsibility
Output requester	To send a request to supply SDI output (deliveries)
Messenger	To send a reminder notice to the user to give feedback
Chaser	To negotiate with user the time to remind him/her again to see SDI deliveries and give feedback

Table 4.4 The Reminder Agent Social Role

The Reminder role - Domain responsibilities	
Origin	Responsibility
Output requester	To initiate a request
Messenger	To compose a reminder notice for the specific user address
Chaser	To record time and user id to initiate the next reminder notice

Table 4.5 The Reminder Agent Domain Role

The third role in the delivery, reminder and feedback handler chain of the SDI system is the feedback handler role whose role is slightly more complicated as it involves responsibilities from its five constituent roles.

The Feedback Handler role - social responsibilities	
Origin	Responsibility
Displayer	To display to the user the ranked list of SDI output
Feedback Receiver	To receive feedback from user on each item delivered
Update recommender	To compose message and recommend terms to be added to the user model
Update recommender	To compose message to recommend terms to be deleted from the user model
Modifier	Compose message to perform term addition to user model
Modifier	Compose message to perform term deletion from the user model

Table 4.6 The Feedback Handler Agent Social Role

The Feedback Handler role - Domain responsibilities	
Origin	Responsibility
Displayer	To rank SDI deliveries to be displayed
Feedback Analyser	To analyse relevant documents and identify terms to recommend to update user model
Feedback Analyser	To analyse non-relevant documents and identify terms to recommend to update user model
Modifier	Add terms to user model
Modifier	Delete terms from user model

Table 4.7 The Feedback Handler Agent Domain Role

CHAPTER V

THE EXPERIMENT

5.1 Introduction

As discussed in the preceding chapter, the agent-oriented modeling has adopted many procedures and algorithms from previous works and has suggested some useful approaches applicable to information filtering. A great deal of work has been done in the area of using the relevance feedback in query expansion in *ad hoc* retrieval and user profile updating in information filtering. A few of the current generation of search engines provide a percentage figures for the user to optimise his/her retrieval which is one example of using feedback to improve retrieval. In this chapter, therefore, the experimental case will be presented in the following order. First, the application of agent in relevance feedback handling will be discussed in order to demonstrate the use of agent technology for information filtering with particular reference to developing agent application using AgentBuilder (a toolkit employed under this research). Following this, the application area and experimental results will be discussed. This includes the description of the use of the implementation procedures, the agent-building tool and the specific proposed application model.

5.1.1 Experimental Objective and Scope

The main objective of the experiment was to demonstrate and test the applicability of agent-based solutions for the SDI's information filtering problem domain with particular emphasis to

updating and optimising the user model based on user feedback in the particular case area. Particularly, the prototyping and implementation focuses on the fundamental components and algorithms that the agents constitute rather than developing a fully operational agent application for the problem domain explored in the previous chapters. The prototype developed is only for the feedback handling process and that of the delivery agent whose input is required by the former. The prototype also emphasises to demonstrate the communication between agents which is one of the fundamental features of a software agent. In addition, this thesis doesn't aim to conduct a complete experiment on the proposed procedures, assumptions and algorithms in the previous chapter. Furthermore, since the whole research work is primarily an academic exercise, the focus of the experiment was on learning how the agent methodologies can be applied in solving the personalised information filtering problem in general and the provision of SDI and maintaining user profile based on user feedback in particular.

5.1.2 Tools and methodologies

The prototype for the Feedback Handler Agent and Delivery Agent was developed using the AgentBuilder - *An Integrated Toolkit for Constructing Intelligent Software Agents by Reticular Systems Inc.* In order to develop the prototype, the approach employed by Dawit (1998) using Bischofberger and Pomberger (1992) which they referred to as 'Experimental Prototyping' whose purpose was to experimentally validate the applicability of system component specification, architecture models including ideas for solutions for individual applications is used.

After the development of the prototype agent, various tests were made using test data mainly from the feedback records of the active users from the experimental case (i.e. ILRIAlerts) which is described in detail in section 5.4. In addition, to analyse the agent results information on the agents behaviour and mental model is added in order to demonstrate the full functionality of the prototype agent which may otherwise not be needed by the user. Finally, a statistical analysis of some of the data from the test case is done in order to describe the performance of the proposed agent-based filtering methods.

5.2 The AgentBuilder Toolkit: An Overview

The Reticular Systems Inc. AgentBuilder toolkit is designed to provide agent developers with an integrated environment to construct intelligent agents. AgentBuilder consists of two major components - the **toolkit** includes tools for managing the agent-based software development process, analysing the domain of agent operations, designing and developing networks of communicating agents, defining behaviours of individual agents, and debugging and testing agent software. The **Run-Time System** includes an agent engine that provides an environment for execution of agent software.

As opposed to the IBM Agent Building Environment that the previous researcher used (Dawit: 1998) which was more applicable in applications with a lot of file manipulation, the current researcher has found the current toolkit less user-friendly in this area which was quiet essential for the case under study. In addition, not all components (capabilities) of the AgentBuilder toolkit described below are available in the version that the current research has used i.e.

AgentBuilder Lite 1.2.1 which was obtained at a discounted price given for academic institutions for academic purposes.

All components of both the AgentBuilder toolkit and the Run-Time System are implemented in Java. This means that agent development can be accomplished on any platform that supports Java and has a Java development environment although it supports integration and use of existing software libraries such as Java, C, and C⁺⁺. Therefore, the agents created with the AgentBuilder toolkit are Java programs so they can be executed on any Java virtual machine. The toolkit and the run-time system are described in the following paragraphs. Figure 5.1 below shows the major components of AgentBuilder and their relationship to each other.

Project Control Tools are provided to help the agent developer manage the overall agent development process. These tools include the Project Manager and the Project Repository Manager. The Project Manager allows the developer to identify and manage multiple agent development efforts.

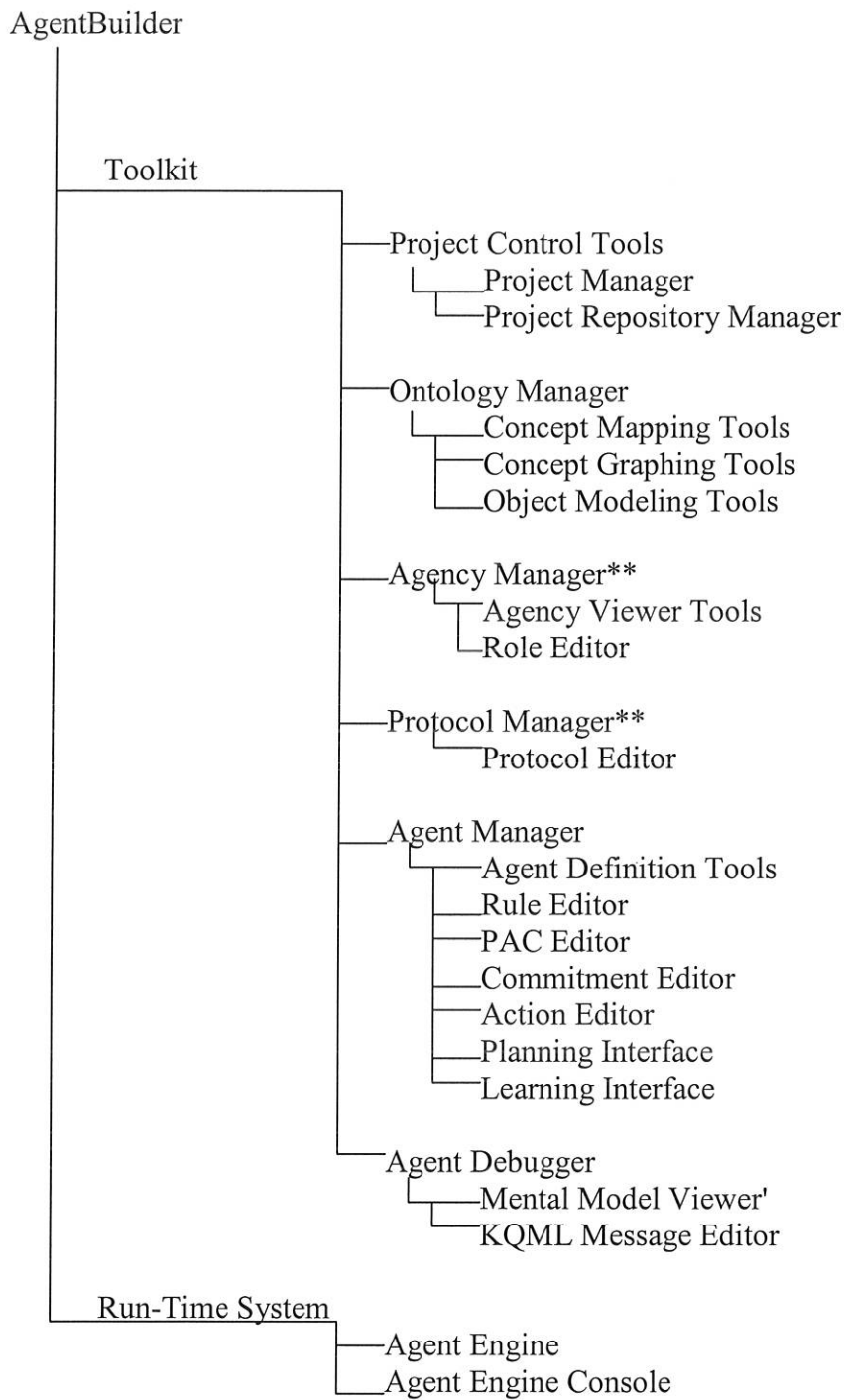


Figure 5.1 AgentBuilder Components

**Not available in the version of AgentBuilder used in this study (i.e. AgentBuilderLite 1.2.1)

The Project Repository Manager controls access to all of the data created using the AgentBuilder toolkit. This data include domain knowledge, agent definitions, agency specifications, etc. These tools allow information to be stored and used by multiple projects.

Ontology Manager assists the developer in analysing the agent application problem domain and in identifying the concepts relevant to the agent's operation in that domain. An ontology is simply a (formal or informal) specification of a conceptualisation. The Ontology Manager provides tools for visualising domain concepts and relationships, graphically representing these concepts and relationships, and codifying them for storage and subsequent retrieval. AgentBuilder provides a number of tools for ontology development. These include a *conceptual mapping tool* that graphically shows the major domain concepts and their relationships. An *object modeling tool* provides a mechanism for the developer to take the concepts identified using the conceptual mapping tool and express these concepts in terms of objects and relationships between objects defined in the domain.

Agency Manager is designed to help the developer construct an *agency*. An agency consists of two or more agents that communicate or cooperate with each other to perform some task. The agent may be identical or specialised for performing different functions. The Agency Manager allows the developer to identify and characterise all of the agents and agent types in the system under development.

Protocol Manager provides tools to specify the messages and conversational protocols between agents. This allows the developer to dictate the high-level information flow that occurs as part of agent-to-agent transactions. The specification of interagent communication

messages and conversational protocols provides the basis for specifying an individual agent's behaviour.

Agent Manager provides tools for defining an individual agent's initial mental model and behaviour. The agent definition tools include graphical editors for defining the various mental constructs that make up the agent: initial beliefs, initial commitments, capabilities and behavioural rules. The Agent Manager is used to create an agent definition file written in the Reticular Agent Definition Language (RADL). The RADL file is a complete specification of the agent's behaviour and initial mental model; this file is interpreted and executed by the Run-Time System. The *Commitment Editor* portion of the Agent Manager is used to specify the agent's commitments. By specifying commitments, the developer instructs the agent to pursue actions at certain time. All commitments, including initial commitments, have a time field that specifies when the agent will perform the action specified by the commitment. The *Action Editor* in the Agent Manager allows the developer to associate agent actions with methods in software libraries. The developer can browse the previously defined object model and choose objects and methods to associate with agent actions. The *Rule Editor* defines how the agent operates given a particular internal and external environmental state. The behavioural rules are made up of message conditions, mental conditions, private actions, communicative actions and mental model changes. The rule editor allows the developer to graphically build up the WHEN-IF-THEN portions of a rule. The LHS editor builds the WHEN and IF portions and the RHS editor builds the THEN portions of the rule. Each message condition consists of a KQML property and an associated value. The editor for constructing the RHS of the rule allows the developer to specify Operators, Actions, Built-in Actions, Defined variables, Return Variables, Instances and Values. The Built-in Actions button allows the developer to

select the private actions that will execute when the rule fires. Since there is no single learning algorithm or planning algorithm that is best for use in all agent applications, the toolkit facilitates for the developer to add the learning or planning algorithm that is most appropriate for the problem domain.

The Run-Time System, Agent Engine - the Run-Time System consists of the Agent Program and Reticular's run-time agent engine. The Agent Program is a combination of the Agent Definition defined in the RADL file and the classes defined in the agent's Project Accessory Class (PAC) library (i.e. Agent Actions and User Interface libraries). The Agent Program is executed by Reticular's run-time agent engine; the combination of the Agent Program and the agent engine produces an executable agent. At start-up, the Run-Time System initialises the agent engine using information stored in the Agent Definition file and links the required components from the PAC Library.

Project Accessory Class Libraries - An agent's PAC contains all of the domain-specific code an agent requires for operation in its domain. The PAC is the mechanism the agent uses to interact with its environment; by calling methods the agent executes its private actions. Generally, agents operate in unique domains, and every domain requires its own PACs. A PAC can be made up of classes and packages from a variety of sources. The developer may write part of the library, other classes may come from commercial off-the-shelf (COTS) packages, and other classes might be freeware downloaded from the Internet.

5.3 The Prototype Delivery and Feedback Handler Agents

As proposed to the previous chapter, the Delivery and Feedback Handling Agents were developed to deliver the monthly SDI output to the respective user and in return to receive and analyse the feedback from the user respectively. As mentioned earlier in the design part, the Feedback Handler Agent has several and complex roles which has not been implemented due to the limited time available to this research. The Delivery's agent role is self-explanatory which is responsible for forwarding the SDI outputs to the user. This agent need not be described in more detail. The Feedback Handler Agent has also incorporates a newly proposed role as a result of our survey that is to remind users of the delivery of SDI output and request feedback from. This is demonstrated in this prototype with a simple message communication between the Feedback Handler and the User Interface Agent. But our main focus in this experiment will be on some of the functions / roles of the Feedback Handler Agent. This agent receives feedback from user, analyses them and determines whether a term (key term representing the documents) from the feedback information or document and the user profile based on which the agent re-weights the terms in the user model depending on the analysed document (i.e. whether a document belongs to the non-relevant set or relevant set of documents). The structure of this agent when it is implemented using the AgentBuilder toolkit will be as follows:

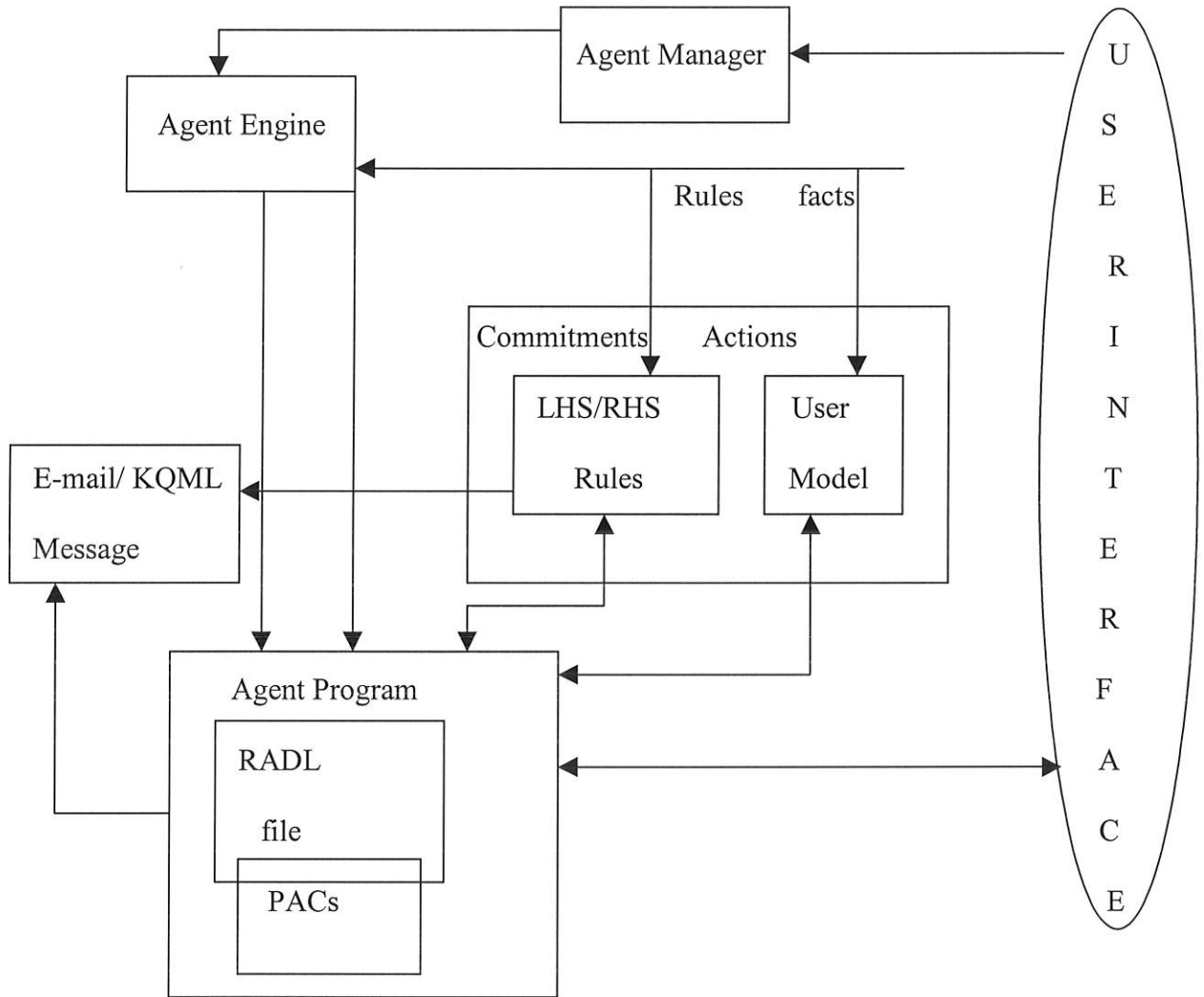


Figure 5.2 The Feedback Handler Agent Implementation Architecture

Some of the above components are developed and the others are provided by the AgentBuilder toolkit. These components provided by the AgentBuilder toolkit and the developed components are described below:

The above architecture is intended to demonstrate how two agents communicate with each other. The Feedback Handler Agent communicates with the Delivery Agent in order to receive the SDI output each month and in return reminds user to give feedback through the User

Interface agent (module) responsible for trapping the user's actions and passing them on to the appropriate agent component or agent. The toolkit's built-in interface is used as the user-interface for prototyping the above agent.

As mentioned above in section 5.2, the agent development process involves the analysis of the problem domain, the definition of agency architecture, the specification of the agent behaviour, the creation of agent application and finally the debugging, testing and integration of agents and agencies. Following this procedure, below is the step-by-step description of the development of the prototype agent.

1. Creation of the *Feedback Handler Ontology* - the domain analysis is facilitated using conceptual mapping tools and object modeling tools. Here all the objects in the domain feedback handler agent are specified, i.e. the ontology for this particular domain. An ontology gives meaning to the symbols and expressions used to describe the domain so that one agent can properly understand the meaning of the message from another agent. Thus both agents ascribe to the same meaning to the symbols (constants) used in that message. The UserModel object is defined here to contain the user profile i.e. the terms and their respective weights in the user model which are manipulated in the feedback handling process.
2. Creation of the *Feedback Handler Agent* - the feedback handler agent and its properties are defined. These include specifying the agent icon, the specifying the engine cycle time from the built-in options, the agency that this agent belongs, the ontologies that the agent subscribes to and the communications properties. Since the version of the AgentBuilder toolkit used in this research can only support single

standalone application, we used the console communication by running the agents in different ports in order to test communication between agents and effect their delegated tasks i.e. in this case the Feedback Handler Agent to re-weight the terms in the user model based on the analysis made on the feedback collected from the real users as described in section 5.4 below.

3. Importing *PAC Object and Create initial Java instance* - the object which is the UserModel has been defined in using Java and imported. This contains the user model for the test-case which has also been used in the previous research undertaken by Dawit (1998). The following class definition shows the UserModel object definition (containing a sample users model of the terms and weights) which has been created in a hashtable and imported to our application domain:

```
import java.util.Hashtable;

public class UserModel {
    public static void main (String args []) {
        UserModel e = new UserModel();
        e.usermodel();
    }

    public void usermodel () {
        Hashtable x = new Hashtable();
        x.put("cattle", "0.0109664");
        x.put("dairy", "0.0452365");
        x.put("farm", "0.0109664");
        x.put("livestock", "0.0219328");
        x.put("market", "0.0219328");
        x.put("methodology", "0.00959561");
        x.put("milk", "0.0123372");
        x.put("policy", "0.013708");
        x.put("production", "0.0246744");
        x.put("smallholder", "0.0246744");
    }
}
```

4. Creation of the *LHS and RHS rules* - the set of conditions are referred to as the *left-hand side*, or *LHS* and the actions and mental changes are the rule's *right-hand side*, or *RHS*. The rule allows the creation of two kinds of rules. One type of condition is based upon external events and is termed a message condition. A message condition is a test performed on any new messages arriving in the agent's input buffers (e.g., comparing the name of the message sender against an expected sender name). The second type of condition is based upon the agent's initial state and is termed as a mental condition. A mental condition is a test performed on beliefs in the mental model of the agent. The right-hand side of a rule allows to specify the actions that fires when the rule's conditions are satisfied.
5. The *Knowledge Query and Manipulation Language (KQML)* is a language and a protocol that supports network programming and intended to support interoperability among intelligent agents in distributed application like the case under study. KQML is most useful for communication among agent-based programs, in the sense that the programs are autonomous and asynchronous. There are several levels at which agent-based systems must agree, at least in their interfaces, in order to successfully interoperate (Finin, et al, 1994). Therefore, we use the KQML message communication facility to establish communication between the agents and to effect action. The KQML message consists of a performative, its associated arguments which include the real content of the message, and a set of optional arguments.

Every month, when the Feedback Handler Agent receives feedback from the user, the program which undertakes written in C++ and integrated with the agent application is activated to perform the analysis of the relevant and non-relevant documents based on the judgments of the user and to identify terms that represent both the relevant and non-relevant documents analysed. The information obtained will be used to build a potential 'interesting' and 'not-interesting' list of terms and then based on these update the user model. Updating is constituted by increasing or decreasing the weight of terms in the user model. After this a decision could be taken to remove terms (words) in the user model below a given threshold or if it is a new term, a decision could be to add into the user model. The operation of this activity by the agent and the application is described in the section 5.4 below.

6. The *RADL file* - At start-up, the Run-Time System initialises the agent engine using information stored in the RADL file (Feedback Handler Agent.radl) and links required components from the Project Accessories Library (UserModel class). Both the agent definition and the project accessory classes (PACs) are needed as the agent definition supplies the agent with a reasoning capability and an initial mental model and the PACs are the objects used to represent the problem domain and provide the agent the ability to interact with its environment.

5.4 Testing

Testing how well the Feedback Handler Agent performs the handling of feedback from users and updating the user model has been undertaken several times. These include aspects such as:

- establishing communication between two agents and a user to do reminding, receiving feedback and updating user model based on simple console communication using the toolkit's capabilities
- the performance of removing non-content bearing words and/or stop words including application of simple method of removing pluralizations.
- the performance of the suggested weighting method in identifying terms representing user's interest
- the performance and method of setting thresholds
- the performance of filtering operations - Recall and precision measures

5.4.1 The Test

For the purpose of the test, 4 ILRI scientists (out of about 9) who are also active users of the ILRIAlerts were selected from the existing feedback record of the last six months whose complete feedback record was found. As mentioned in Chapter 3, section 3.3.1, information staff keeps relevance feedback records only for six months and after which will be purged. Therefore, during the research period it was only possible to gather the relevance feedback records of the last six months (from October 2000 to March 2001). These records again consist only users who receive their SDI output by e-mail. Hence, the selection of 4 users out of the 9

active users who provide feedback relatively regularly. Table 5.1 shows the number of documents analysed to select the test data.

Table 5.1 Number of abstracts of documents examined from users' relevance feedback

User ID	Feedback for the month of	Number of abstracts judged by the user as	
		Relevant	non-relevant
087	December	4	5
173	December	16	9
522	December	3	2
522	February	4	5
522	March	4	3
529	November	8	9
Total Abstracts		39	33

Each document contains, the bibliographic description of the documents (author, title, place of publication, date, descriptors, abstracts (if any, etc.)). For the purpose of identifying terms that of 'interest' or 'not-interest' to the user, the titles, descriptors and abstracts of the documents have been extracted from the relevance feedback record given in the table above and they were analysed.

5.4.2 Communication between agents

Software agent has been defined as a software component that executes autonomously; communicates with other agents or the user; and monitors the state of its execution environment. (Reticular Systems, 1999). The prototype agents have been tested for being initiating communication with each other and for monitoring the state of its execution environments in that they act on the messages that one received from the other. For instance the Delivery Agent at start-up fires the *'Wait for incoming message'* rule and waits for the request from the feedback handling agent about the current SDI output. On receipt of the message that requests for SDI delivery, the Delivery Agent fires the *'Respond to the incoming request'* rule in which it forwards the SDI output to the requesting agent. Since the operation of the responding agent depend on receiving requests from the requesting agent, and that the response depend on whether the responding agent's engine being on run, the agent should be executed and wait for request. The above tests indicated that the Feedback Handler Agent and the Delivery Agent can be developed as a memory resident program so that they can autonomously initiate communication and perform their tasks.

5.4.3 The performance of removing non-content bearing words, term weighting, term selection and setting threshold

As the Feedback Handler Agent is processing a document, intermediary results such as the elimination of articles, prepositions, conjunctions, and other non-content bearing words listed in the stop list, term lists after stop word removal were kept for all of the test cases. Such removal of non-content bearing words helped to identify a list of terms that represent the

contents of the documents found both in the relevant and non-relevant documents from the feedback test data being examined. For the removal of non-content bearing words, the list of stop words of WAIS's (Wide Area Information Server) by Jonathan (1993) have been used. The words found after removal of the non-content bearing terms and implementation of simple de-prularisation (i.e. removing the suffix 's' or 'es') in order to reduce terms appropriately to reduce redundancy were found to be performing well in the identification of terms that represent the contents of documents. Although stemming the terms would be appropriate for better results, as the test case has been performed on the existing bibliographic database, the forms of query terms of which are almost similar to the ones used in the user profile, it was not implemented in this small-scale test. However, it should be utilised to obtain a better filtering results.

For identifying the potential key words, the top five percent of the high frequency terms were selected as suggested in Pao's (1989) which are believed to represent the content of the documents. After identifying the potential key words both from the documents judged as 'relevant' and 'non-relevant', they were weighted in order to do the next task, i.e. setting a threshold.

The testing undertaken include the identification of the terms that potentially identify those documents that were judged by the user as 'relevant' and 'non-relevant'. The relevance-feedback record for the following users and the potential key words representing the relevant and non-relevant documents were as follows:

Table 5.2 List of potential key-words that represent 'relevant' and 'non-relevant' documents of users in the test cases

USER-ID 087 - Relevance Feedback based on the December 2000 SDI output delivered (total output 9 abstracts)	
Potential keywords representing non-relevant documents (5 abstracts)	Potential key terms representing the relevant documents (4 abstracts)
Weaning	Cattle
Weight	Simulation
System	Animal
Model	Beef

USER-ID 173 - Relevance Feedback based on the December 2000 SDI output delivered (total output 25 abstracts)	
Potential keywords representing non-relevant documents (9 abstracts)	Potential key terms representing the relevant documents (16 abstracts)
Cow	Weight
Semen	Birth
Production	Mortality
Bull	Cow

USER-ID 522 - Relevance Feedback based on the December 2000 SDI output delivered (5 abstracts)	
Potential keywords representing non-relevant documents (2 abstracts)	Potential key terms representing the relevant documents (3 abstracts)
Disease-Prevalence	Feed
	Crossbreed
	Mortality

USER-ID 522 - Relevance Feedback based on the February 2001 SDI output delivered (9 abstracts)	
Potential keywords representing non-relevant documents (5 abstracts)	Potential key terms representing the relevant documents (4 abstracts)
Fat	Cow
Sheep	Animal
Production	
Growth	

USER-ID 522 – Relevance Feedback based on the March 2001 SDI output delivered (7 abstracts)	
Potential keywords representing non-relevant documents (3 abstracts)	Potential key terms representing the relevant documents (4 abstracts)
Calving	Calving
Cow	Weight
	Dam

In order to determine threshold based on the approach discussed in Chapter 4 section 4.5.1.1.2, the term weight of each of the identified key terms have been calculated using the following formula:

$$q_{ik} = \frac{(\log(f_{ik}) + 1.0) * \log(N_D/n_k)}{\sqrt{\sum_{j=1}^t [(\log(f_{ij}) + 1.0) * \log(N_D/n_j)]^2}}$$

where f_{ik} is the number of times term k occurs in Q_i , N_D is the collection size, and n_k the number of documents with the term k assigned.

Documents have been ranked using the approach described in chapter 4, section 4.5.1.1.1. For the purpose of testing the applicability of this method, the relevant documents (4 in number) of user 087's relevance feedback for December has been used. The following table shows the results of determining threshold based on the approach stated in the previous chapter, section 4.5.1.1.2,

Document	Similarity measure - document / user model (rank of documents)	Utility function
Document1	0.0013462	3
Document2	0.0018953	-3
Document3	0.0021322	3
Document4	0.0028246	-9

Since there are two documents which gave the rank which maximised utility, as per the approach proposed in the preceding chapter section 4.5.1.1.2, the last one will be taken as the threshold, in this case, the first document rank score i.e. 0.0013462 will be taken as a threshold.

For instance, the following table shows the terms and their respective weights as a result of the user's (i.e. user ID 087) relevance feedback in December.

Non-Relevant documents		Relevant documents	
Term	weight	term	weight
Weaning	0.0209790	Cattle	0.0394736
Weight	0.0775862	Simulation	0.0394736
System	0.0139860	Animal	0.0394736
Model	0.0419580	Beef	0.026315

The above term weights will therefore be used to carry out updates in the user profile which, as proposed in the preceding chapter, is constituted by changes in the weights of terms in the user profile. If a term in the user profile is identified as a key term in the non-relevant documents domain, this will reduce its weight in the user profile and vice versa. The threshold will help make a decision so that if the weights of the terms in the user model significantly reduced as a result of the relevance feedback process to be significantly below the threshold the system recommends the elimination of the term from the user profile. However, if the potential key terms identified in the relevant documents are found in the user profile, this

contributes to an increase to their respective weights by a certain amount that should be determined. The threshold will also help to make decisions to add new terms into the user profile if they are identified as potential key term and their weight is above the threshold.

5.4.4 The performance of filtering operations - Recall and precision measures

A few experiments have been performed in examining the filtering operation of 1500 abstracts of technical reports and documents on the basis of the user profiles of 3 users. The test has been done in such a way that the ILRIAlerts system received 375 documents per session of generally 4 sessions and that the results of which have been shown to the users. For each document, each user provided a relevance judgments which was then utilised for computing the recall and precision for both relevant and non-relevant documents. The results obtained show an average improvement in the precision of 12% and an average improvement of the recall of 16%. These tests on the performance of the proposed procedures and methods in terms of recall and precision gives an indication of the performance of the filtering tools created in this form. However, in order to obtain better performance values, it would be important to increase the number of sessions that should be undertaken with the user though there is a difficulty to carry out such tests with widely dispersed users and with users who don't normally offer feedback.

5.5. Discussion

The above results, even if they are based on small number of test cases that can hardly be considered as comprehensive and representative, indicate the applicability of some of the

proposed procedures and methods in the preceding chapter. The various functional components of the feedback handler agents like stop-word removal, identifying potential key words from the relevance feedback to optimise the user profile based on the analysis of the relevant and non-relevant documents show that the approach is promising and performing well.

The experiment on the use of a model free threshold suggested by Roberts and Walker (2000) is found to be a useful approach and applicable for the particular case. However, an extended experiment need to be undertaken with a large set of training data in order to have good result. The approach that is based on the identified threshold, terms weighted both from the relevant and non-relevant documents will be compared to the user model. If terms come from the non-relevant documents and found in the user model, updating will be constituted by changes (decrease) in the weight of terms in the user model until such time that the terms have significantly reduced their weight to be below the threshold. The exclusion of terms from the user model will be constituted in consultation with the user if weights of the terms have significantly been below the threshold set. It has been noticed that terms related to user interests are appearing representing well both the documents that were judged by the user as 'relevant' or 'non-relevant'. If such cases happen quiet frequently, it may create instability in the user model. This may however, happen in the first two or three attempts and that after which the user model stablises unless drastically changes due to shift of user's interests.

The SDI output has been sent to the above active users who more or less provide feedback regularly applying the procedures and methods described in the preceding chapter on the basis of their feedback of December 2000 and three of the users have given their feedback to our

four sessions test cases. As mentioned above, the obtained results show an average improvement of the performance of the precision of 12% and an average improvement of the recall of 16%. Although, this test can't by no means taken as comprehensive and complete, the results show that the procedures and methods proposed are promising to continually update and optimise the user profile and improve filtering.

As part of this testing, some of the users have provided their overall opinion and feedback on the ILRIAlerts system in general and the relevance feedback process which actually strengthen the problems identified in this study in chapter 3 by the existing users of the system. Comments are attached in Annex 1.

CHAPTER VI

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The research described in this thesis has tried to refine the SDI problem domain in information filtering particularly looking at the updating of the user profile based on relevance feedback. The analysis of the relevance feedback has shown that the user profile can be continually updated so as to improve retrieval of more and more of relevant documents and less and less non-relevant documents pertinent to users' area of interests. In this respect, to have an insight on the problem area, a case system, the ILRIAlerts (an SDI service maintained by the ILRI) has been studied.

The evaluation activities carried out on the users profile based on the analysis of the few relevance feedback gathered over a limited period of time supports the claim that the use of various user modeling techniques (in our case one focused on the refinement of the user model based on feedback) can improve the performance of agent-based information filtering. In this line, a test to determine the appropriate threshold has been carried out. Based on relevance feedback analysis, an update to the user model is effected by changing the weights of terms in the user model. The change of weights in the terms in the user model would be monitored by the agent to check against the threshold determined.

The two prototype agents namely the Delivery Agent and the Feedback Handler Agent have been tested to have autonomously established communication and providing response or taking action by sensing their operating environment which showed that the agents have characterised the basic natures of software agents. The feedback handling component of the SDI information filtering system requires a lot of exchange of information among the other components in the system and the user. Therefore, it has been uncovered in this research that agent support to a series of communications required in this sub-component would be a paramount importance. This would be expected to resolve the problems of the database intermediaries problems of chasing up the users to provide feedback on a regular basis, as in the case of the users being reminded to provide the required feedback. The detailed specifications of these two agents have tried to identify the appropriate development approaches or options and algorithms that are based on established methods in agent technology applications and information retrieval. However, the need for reminder (remembrance) agent within the feedback handling agent roles has been found a necessary and proposed to this effect. This is also one of the investigations that this research has come up with. With the aim of experiencing and also demonstrating how the proposed agents can be implemented, a prototype agent has been developed using Reticular Systems Inc.'s AgentBuilder - an integrated toolkit for the construction of intelligent software agents and a C++ programming language to integrate required components for the agent application.

The test cases have been initiated using around 72 documents and 4 active users' feedback record to examine the pattern of users' feedback. The researcher has also used 1,500 abstracts of documents to test the proposed procedures and methods in chapter 4 undertaken in four sessions. Only three of the four active users have provided feedback to the four sessions tests.

The obtained results show an average improvement of the performance of the precision of 12% and an average improvement of the recall of 16%. The small-scale test indicated that the procedures and methods employed in this study would be promising to update the user model and improve filtering.

In summary, therefore, this research has found that interface agent technology can be applied to solve some of the problems of the SDI filtering process based on applying relevance feedback. Particularly, this technology has been found to be useful in developing a client/server side application which handles communication problems between the user and the intermediaries and the system. The use of appropriate term weighting and threshold setting will help build a potential list of 'likes' and 'dislikes' term lists which could help implement learning of users' interests and improve filtering.

Finally, the study has resulted in refining and good understanding of the problem area which has helped to take into account of a practical problem which would attract further investigation and research.

6.2. RECOMMENDATIONS

The application of agent technology in addressing the SDI problem domain in general and improving personalised information filtering has been investigated. The application of agent technology in this area has a paramount importance. It is found out that a lot more research need to be undertaken in order to realise that the systems applying agent-oriented model suggested in this thesis or an updated of it. Therefore, the following areas were identified as needing further research work:

- 6.1.1 A more comprehensive testing of the relevance feedback analysis using the proposed models including the Rocchio's algorithm and with a comprehensive test collection.
- 6.1.2 The use and determination of the threshold should be further researched including determining the proportion of weights to be used to perform updates in the user model which, as proposed, could be in the form of an increase or decrease in the weights of the terms in the user model or those new terms to be added to the user model.
- 6.1.3 In such systems, machine-learning approach has a very important role in terms of reducing involvement of users and intermediaries. Particularly, a machine learning approach to update the user profile with little or no input from user need to be investigated considering the problems identified in this research i.e. the problem that users don't provide feedback on the SDI outputs that they are receiving regularly. Furthermore, the processing algorithms to be adopted to

extract the information content of the incoming documents and the matching algorithms to be explored to assess their potential relevance need also be investigated further and tested with comprehensive data. In terms of machine learning again, some of the observations as part of this research was that the users judge documents as non-relevant if they found out that the document is written in a foreign language that they don't know. Therefore, machine learning based on 'language' would also be an area further investigation and research.

6.1.4 This research precluded the server-side agents completely although console-based communication between agents have been tested. Therefore, further work using possible supporting multiple-agent building toolkit would enhance the full functionality of the proposed agent-oriented models and also can build on works in automatic search re-formulations, filtering techniques, etc.

6.1.5 Agent-based information filtering will have a paramount importance particularly in developing countries where there is limited band-width in terms of accessing information through the World Wide Web and in several other applications such as electronic commerce, etc. Further research in this area would need to uncover the potential of intelligent agents in various application areas so that their use can be looked seriously in a constraining environment to benefit from the global information resources and from the opportunities that technology offers.

REFERENCES

1. Allan, James. (1996) . Incremental Relevance Feedback for Information Filtering.
Zurich: ACM.
2. Ambrosini, Leonardo, Vincenzo, Cirillo and Micarelli, Alessandro (1997). A Hybrid
Architecture for User-Adapted Information Filtering on the World Wide Web.
In Anthony Jameson, Cecile Paris, and Carlo Tasso (eds), User Modelling:
Proceedings of the Sixth International Conference, UM97. Vienna, New York:
Springer Wien New York.
3. Asnicar, Fabio A, Fant, Di Massimo and Tasso, Carlo. (1999). User Model-Based
Information Filtering. Udine: University of Udine.
http://www.dimi.uniud.it/~ift/aiia/html//AI_IA.html
4. Baeza-Yates, Ricardo and Ribeiro-Neto, Berthier (1999). Modern Information
Retrieval. ACM Press, New York; Addison Wesley Longman Ltd, Harlow,
England.
5. Baclace, Paul E. (1991). Personal Information Intake Filtering. presented at the
Bellcore Information Filtering Workshop, November. Available at
<http://www.baclace.net/Resources/ifilter1.html>
6. Buckley, Chris, Salton, Gerard, Allan, James. (199). The Effect of Addis Relevance
Information in Relevance Feedback Environment. Ithaca, NY: Department of
Computer Science, Cornell University.
7. Chambers, Brian D. (1999). Adaptive Bayesian Information Filtering. MSc
Thesis, University of Toronto, 1999. At

http://wwwlib.umi.com/dissertations/preview_all/MQ45945

8. Chen, Hsinchun, Chung, Yi-Ming and Ramsey, Marshall. (1998). A Smart Itsy Bitsy Spider for the Web. Journal of the American Society for Information Science. 49(7): 604-618. New York: John Wiley & Sons, Inc.
9. Chen, Zhixiang, Zhu, Binhai. (1998). Some Formal Analysis of Rocchio's Similarity-Based Relevance Feedback Algorithm. Edinburg, TX: Department of Computer Science, The University of Texas-Pan American.
10. Dawit Yimam. (1998). Applying Interface Agent Technology to Selective Dissemination of Information (SDI) User Profile Management: the Case of ILRI/Alerts. SISA, Addis Ababa University
11. Ferguson, Innes A. and Karakoulas, Grigoris J. Multiagent Learning and Adaptation in an Information Filtering Market.
12. Finin, T., Weber, J., Wiederhoid, G., Genesereth, M., Fritzson, R., McGuire, J., Shapiro, S., McKay, D., Pelavin, R., and Beck, C. (1994). Specification of the KQML Agent-Communication Language plus example agent policies and architectures (DRAFT Report), DARPA Knowledge Sharing Initiative External Interface Workign Group.
13. Gadomski, A. Maria. (1997). Agents and Intelligence.
<http://wwwerg.casaccia.enea.it/ing/tispi/gadomski/gad-ag.html>
14. Gauch, Susan. (1992). Intelligent Information Retrieval: An Introduction.
Journal of the American Society for Information Science. 43(2): 175-182.
15. Garcia-Molina, Hector and Yan, Tak W. (199). Distributed Selective Dissemination of Information. Stanford: Stanford University.

16. Haverkamp, Donna S. and Gauch, Susan (1998). Intelligent Information Agents: Review and Challenges for Distributed Information Sources. Journal of the American Society for Information Science. 59(4):304-311.
17. ILCA. (1991). Information Policy and Procedure Manual: First Draft. Addis Ababa: ILCA, July.
18. ILRI (1998). Strategy and Programme for ILRI Information Resources 1998 - 2010. Addis Ababa: ILRI.
19. Jansen, James. (1996). Using Intelligent Agent to Enhance Search Engine Performance. (http://www.firstmonday.dk/issues/issue2_3/jansen).
20. Kemp, D. A. (1979). Current Awareness Services. London: Clive Bingley.
21. Kendall, E. A. et al. (1995). "A Methodology for Developing Agents Based Systems". In Distributed Artificial Intelligence: Architecture and Modeling, First Australian Workshop on DAI. Canberra, ACT, Australia Nov., 1995 proceedings: 85-99.
22. Kraaij, Wessel and Pohlmann, Renee. Porter's Stemming Algorithm for Dutch.
23. Lancaster, F.W. (1991). Indexing and Abstracting in Theory and Practice. London: The Library Association.
24. Lancaster, F. Wilfred (1968). Information Retrieval Systems. New York: John Wiley & Sons, Inc.
25. Leggate, P. (1975). Computer-based Current Awareness Service. Journal of Documentation. 31 (2): 93-115.
26. Lewis, David D. (1995). Active by Accident: Relevance Feedback in Information Retrieval. Murray Hill, NJ: AT&T Bell Laboratories.
27. Meek, J. (1995). Intelligent agents, Internet information and interface. Australian

28. McCalla, G. and et al . Analogical User Modeling: A Case Study in Individualised Information Filtering.
29. Miller, Damien, (1996). Three Types of Intelligent Agents, their use, and their future: how the next generation of agents will change the Net, leverage, the value of existing content, and become a critical competitive feature.
<http://www.netscapeworld.com/netscapeworld/nw-10-1996/nw-10-agents.html>
30. Nelson, Michael J. and Tague, Jean M. Split Size - Rank Models for the Distribution of Index Terms. Journal of the American Society for Information Science. 36(5): 283-296, 1985. New York: John Wiley & Sons, Inc.
31. Nwana, H S (1996). Software Agents: An Overview. The Knowledge Engineering Review, 11, pp 144 - 205.
32. Nwana, H.S. and Wooldridge, M. (1996). "Software Agent Technologies". BT Technology Review, 11(3) : 205-244.
33. Pao, Miranda Lee. (1989). Concepts of Information Retrieval. Englewood, Colorado: Libraries Unlimited.
34. Oard, Douglas W. (1995). Information Filtering Defined.
http://www.clis.umd.edu/dlrg/filter/filter_definition.html
35. Quiroga, Luz Marina. (1999). Personalised Information Organisation:

- Acquisition and Modeling of Users' Interest Profiles in Information Filtering Systems. PhD Dissertation. Indiana University. Available at <http://wwwlib.umi.com/dissertations/fullcit/9950785>
36. Ram, Ashwin. (1991). Interest-based Information Filtering and Extraction in Natural Language Understanding Systems. Bellcore Workshop High-Performance Information Filtering. Morristown, NJ, November.
 37. Reticular Systems, Inc. (1999). AgentBuilder: An Integrated Toolkit for Constructing Intelligent Software Agents. Revision 1.3. California: Reticular Systems Inc.
 38. Robertson, S. E. and Jones K. Sparck. (1976). Relevance Weighting of Search Terms. Journal of the American Society for Information Science. 27(3):129-146.
 39. Robertson, Stephen and Walker, Stephen. (2000). Threshold Setting in Adaptive Filtering. Journal of Documentation, vol. 56, no. 3, pp. 312-331.
 40. Russel, Stuart and Norvig, Peter. (1995). Artificial Intelligence: A Modern Approach. New Jersey: Prentice-Hall, Inc.
 41. Salton, Gerald and McGill, Michael J. (1983). Introduction to Modern Information Retrieval. New York: McGraw-Hill Book Company
 42. Salton, Gerald and Allan, James . Automatic Text Decomposition and Structuring. Itahaca, NY: Department of Computer Science.
 43. Schweighofer, Erich, Winiwarter, Werner and Merkl, Dieter. Information Filtering; The Computation of Similarities in Large Corpora of Legal Texts. Vienna: Institute of Public International Law.
 44. Sheth, Beerud Dilip. (1994). A Learning Approach to Personalised

Information Filtering. MSc Thesis. Massachusetts Institute of Technology.

Available at <http://citeseer.nj.nec.com/sheth94learning.html>

45. Singhal, Amit and Salton, Gerald . Automatic Text Browsing Using Vector Space Model. Ithaca, NY: Department of Computer Science.
46. Soltysiak, S J and Crabtree, I B (1998). Automatic Learning of User Profiles - Towards the Personalisation of Agent Services. *BT Technology Journal*. 16(3): 110-117.
47. Tedd, Lucy A. and Yeates, Robin. (1998). A Personalised Current Awareness Service for Library and Information Service Staff: an Overview of the NewsAgent for Libraries Project. *Program*, vol. 32, no. 4, pp. 373-390.
48. Torkelson, N.R. (1973). SDI as a control of information overload. *Proceedings of the American Society for Information Science*. 10, 213-2.
49. Vickery, Brian C., ed. (1994). *Fifty Years of Information in Progress: A Journal of Documentation Review*. London: Aslib, 1994.
50. <http://maya.cs.depaul.edu/~mobasher/classes/ds575/software/stoplist>. Jonathan (1993).
51. Wooldridge, M.J., and Jennings, N. R. (Ed.) (1995). *Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, and Languages*. Berlin: Springer-Verlage.

ANNEX I

From: Tim Williams <t.o.williams@cgiar.org> at internet-po
Sent: 13 June 2001 04:00
To: Chekol, Abebe (Ethiopia)
Subject: FW: Urgent need of your response

Here is my response. Usually, the ILRIAlerts come with a short summary of each article. Without this summary it is difficult to say whether or not an article is useful. In the case of the article that you sent to me, there was no summary so I returned a response of "Can't Say". This probably not the response I would give if a summary of the paper is attached.

Overall, the ILRIAlerts system is good. In the past, subscribers could ask for copies of articles and these will be photocopied and mailed to the subscriber. Since this service is no longer being rendered, there is no point in asking for a feedback. A subscriber who is not satisfied with his/her profile will eventually get in touch with the service provider if he/she is still interested in the service. Asking them to provide a feedback all the time is not necessary and is time-consuming. That is why most subscribers fail to respond.

I hope you find these comments useful.

Best of luck with your thesis.

Regards, Tim Williams

Information Services
International Livestock Research Institute
P.O. Box 5689, Addis Ababa, Ethiopia
E-mail: ILRISDI@CGIAR.ORG

=====

User ID: 087
Dr. T.O. Williams
Agricultural Economist
ILRI
c/o IITA, PMB 5320
Ibadan, Nigeria

Profile

=====

(Title / Parallel titles / Title in English / Subject code /
Descriptors /

Proposed descriptors / CABI descriptors ct risk / risks /
uncertainty / probability analysis / stochastic / probability
AND
Title / Parallel titles / Title in English / Subject code / Descriptors /

Proposed descriptors / CABI descriptors ct mixed farm / mixed farming /
mixed
farms / crop w4 livestock / livestock farm / livestock farming
/ livestock farms
/ livestock production / animal production / cattle farming /
goat farming /
sheep farming / buffalo farming / mixed pasture / mixed
pastures / range land /
range lands / rangeland / rangelands)
OR
(Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct agropastoral* /
pastoral* /
traditional farming / traditional system / traditional systems
/ production w2
system / production w2 systems / farming system / farming
systems
AND
Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct model / models /
modelling /
mathematical / simulation/ economics / socioeconomics / impact
w2 analysis)
NOT
Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct weaning* / weight*

=====

Total items output to user this month: 1
User must provide feedback to remain on the service

=====

1. Village chicken production systems in rural Africa. Household food security and gender issues. 1998. Kitalyi, A.J. FAO Animal Production and Health Paper. no. 142. 81p. illus. ISSN 0254-6019. FAO, Rome (Italy). Animal Production and Health Div.. Rome (Italy) : FAO. ISBN 92-5-104160-1.

Key words: CHICKENS; ANIMAL PRODUCTION; HOUSEHOLDS; FOOD SECURITY; MEN; WOMEN; RURAL ENVIRONMENT; VILLAGES; MODELS; RURAL AREAS.

Did you find this item relevant? [Can't say]

I was away from office and thus why my response is belated. Any way all the out of the 9 citations all are very relevant except 3 which I score as may be because is typical breeding paper and number 7 is in the language I can not understand.

I hope that satisfies your question. I am available if need arises for some more information

bye

Fred Mgongo

Information Services
International Livestock Research Institute
P.O. Box 5689, Addis Ababa, Ethiopia
E-mail: ILRISDI@CGIAR.ORG

=====

User ID: 173

Prof. Federick Mgongo

Dept. of Vet. Surgery, Obstetrics & Reprod.

Sokoine Univ. of Agriculture

P.O. Box 3020

Morogoro, Tanzania

Mobile Voice 0811 777756

Private Tel 00255 23 2604547

Office Tel 00255 23 2603511

Email: fmgongo@suanet.ac.tz

Fax: 00255 23 2 604562

=====

Profile

=====

{cattle} / {sheep} / {goats}
OR
Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct (small w2 ruminant
! small ruminant
research) / small w2 ruminants
AND
(Title / Parallel titles / Title in English / Subject code /
Descriptors /
Proposed descriptors / CABI descriptors ct reproduction w2
physiolog* /
reproductive w2 performance* / reproductive w2 physiolog* /
birth weight /
birth weights / puerperium / postpartum w2 period* / partum
w2 period* /
postpartum w2 time* / partum w2 time* / ovarian cyst* /
anoestrou* / anoestru* /
anestrou* / anestru* / ovarian disease* / female genital
disease* / oestrous /
oestrus / estrous / estrus / ovarian* / endocrinolo* / embryo
transfer* /
artificial insemination* / hormone* / progesteron* /
prostaglandi* / oestrogen*
/ pyomet* / endometritis* / dystoci* / reproductive* /
reproduction / genetics /
weight / birth / mortality)

NOT
Title / Parallel titles / Title in English / Subject code /
Descriptors / Proposed descriptors / CABI descriptors ct semen /
production / bulls

=====
Total items output to user this month: 9
User must provide feedback to remain on the service

1. Jinkou youzhi rouniu peitai de zhengpei he fenge hou banpei yizhi shiyan. [Study on the technique of embryo transfer in beef cattle] 1998. Li Shujing; Yu Wenli; Wu Lan. Chinese Journal of Animal Science (China). ((Zhongguo Xumu Zazhi (China))). v. 34(4). p. 30-31. 5 tables; 3 ref. ISSN 0258-7033.

Did you find this item relevant? [Yes]

2. Additive and heterosis effect on milk yield and birth weight from crossbreeding experiments between Holstein and the local breed in Bangladesh. 1995. Hirooka, H. Asian-Australasian Journal of Animal Sciences. v. 8(3). p. 295-300. 6 tables; 27 ref. ISSN 1011-2367.

Did you find this item relevant? [Yes]

3. Laoshan naishanyang zhuyao xingzhuang yichuan chuansu de guce. [Estimation of genetic parameters on Laoshan dairy goat] 1998. Wang Jianmin; Li Fuchang; Wang Zhonghua. Journal of Shandong Agricultural University (China). ((Shandong Nongda Xuebao (China))). v. 29(1). p. 41-45. 2 tables; 9 ref. ISSN 1000-2324.

Did you find this item relevant? [Can't say]

4. Culubaosu he nianling dui mianyang chaoshupainuan xiaoguo de yingxiang. [Effect of follicle-stimulating hormone and the age on superovulation of ewes] 1998. Ding Hong; Mayila; Yang Dongmei; Urumqi Xinjiang. Chinese Journal of Animal Science (China). ((Zhongguo Xumu Zazhi (China))). v. 34(3). p. 18-20. 4 tables; 8 ref. ISSN 0258-7033.

Did you find this item relevant? [Yes]

5. Blood plasma minerals and fertility of dairy cows in central Thailand. 1995. Vijchulata, P. Asian-Australasian Journal of Animal Sciences. v. 8(3). p. 263-265. 2 tables; 10 ref. ISSN 1011-2367.

Did you find this item relevant? [Yes]

6. Follicular dynamics in upgraded indigenous Philippine goat. 1998. Sanchez, G.C.; Ocampo, M.B. In: Recent developments in animal production - 1998: proceedings of the 35th annual convention of the Philippine Society of Animal Science. Sarabia, A.S. (ed.); Lihai, T.S. (ed.) p. 246-250. 4 tables; 9 ref. Philippine Society of Animal Science, College, Laguna.. College, Laguna (Philippines) : PSAS. 35. Annual Convention of the Philippine Society of Animal Science, Manila (Philippines), 15-16 Oct 1998.

Did you find this item relevant? [Yes]

7. Manejo reproductivo de ovinos colombianos de pelo en la región caribe. 1996. Araújo Guerra, A. p. 82-85. Corporación Colombiana de Investigación Agropecuaria, Ibagué.. Ibagué (Colombia) : CCIA. Curso sobre Producción de Ovinos Colombianos de Pelo, Ibagué (Colombia), 29-31 Mar 1995.

Did you find this item relevant? [Can't say]

8. Zhongyao "Yunbao" dui zigongyanbingniu zhongliuhuaisiyinzi-alpha hanliang de yingxiang (chubao) [Effects of the Chinese herbal prescription YUN BAO on uterine tumor necrosis factor-alpha contents of dairy cows with endometriosis] 1999. Li Chengmin; Zhong Xiuhui. Journal of Agricultural University of Hebei (China). ((Hebei Nongye Daxue Xuebao (China))). v. 22(4). p. 108-111. 3 tables; 13 ref. ISSN 1000-1573.

Did you find this item relevant? [Yes]

9. Diagnóstico y alternativas de control al problema de la mortalidad bovina en un área del departamento del Vichada. 1996. Parra Arango, J.L.; Olarte, L.F.; Barrera Méndez, J.E.; Acevedo, L.F. 113p. illus.; 28 ref. Corporación Colombiana de Inverstigacion Agropecuaria, Villavicencio.. Villavicencio (Colombia) : CCIA.

Did you find this item relevant? [Can't say]

Dear Mr Abebe,

I was out of the country till 16th, June, I regret sending the response for your study late. However, I worked on it immediately, hope it will still help you.

Sending you my response in the attachments.

Thanks,

Dr S.M.Das

Information Services
International Livestock Research Institute
P.O. Box 5689, Addis Ababa, Ethiopia
E-mail:ILRISDI@CGIAR.ORG

=====

User ID: 522

Dr. Sachindra M. Das
Livestock Production Research Institute
P.O. Box 202
Mpwapwa, Tanzania

=====

1. Romagnola crossbreeds put more beef on those bones. 1999. Nel, C.
Farmer's Weekly (South Africa). p. 6-10. 11 photos.

Did you find this item relevant? [Yes]

2. The use of breeding values for genetic improvement on a herd basis.
1999. Toit, F. du. SA Guernsey (South Africa). p. 13-15, 25. 1 table.

Did you find this item relevant? [Yes]

3. Angora goat stud breeding - where from and where to now.
[Angorabokstoetteling - gister, vandag en more] 1999. Short, A. Angora
Goat & Mohair Journal (South Africa). v. 41 (2). p. 27-28, 37-38.

Did you find this item relevant? [Yes]

4. A summary of Matopos beef cattle breeding research. 1998. Moyo, S.;
Beffa, M. Farming World (Zimbabwe). ((Kurimi Ukulimi)). v. 24(3). p. 9-11.

Did you find this item relevant? [Yes]

DECLARATION

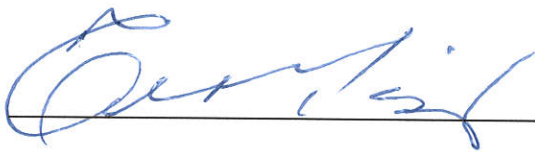
This thesis is my original work, has not been presented for a degree in any other university and that all sources of material used for the thesis have been duly acknowledged.



Abebe Chekol Hunegnaw

June 2001

This thesis has been submitted for examination with our approval as university advisors



Tesfaye Biru



Ermias Abebe