CHAPTER 6

A USER–ADAPTIVE SELF-PROCLAMATIVE MULTI-AGENT BASED RECOMMENDATION SYSTEM DESIGN FOR E-LEARNING DIGITAL LIBRARIES

Chapter 2 explains the problem of user-adaptive retrieval in the various classes of information systems. In this present work a multi-agent based retrieval/recommendation system has been designed to improve the retrieval effectiveness in the federated digital library environment.

In this work, Section 6.1 explains the architecture, component and functionality of the multi-agent system with the design of user-interface agent. Section 6.2 explains the user modeling and user interface design. Section 6.3 presents the ACM CR Classification and Automated Concept-Matrix Formulation method. Section 6.4 presents the method used for concept relativity analysis. Section 6.5 gives the implementation details. Section 6.6 presents the experiment results. Section 6.7 Summarizes this work.

6.1 SELF-PROCLAMATIVE MULTI-AGENT RECOMMENDATION / SEARCH SYSTEM

In this multi-agent system every individual agent is designed to the specific domain (Specific Subject) apart from the agents to be used for the phrase extraction and user-interaction. The domain-specific agent will take care of the identification of documents related to the user requested phrase.
The system is designed to adopt the ACM CR classification schemes. Also a specific domain specific agent is designed for each category at the first level.

A phrase-extraction agent will perform the task of technical phrase extraction. A concept dictionary is attached to the system to provide a reference to the phrase-extraction agents as well as to the other domain-specific agents. The architecture of this multi-agent system is given in Figure 6.1. After extracting the technical phrases from the user-query, it is added with the user-profile and then the concept matrix is reformulated representing the user intention. Before formulating this concept matrix, the system fixes the searching strategy and accordingly it formulates the query for different users. Sometimes if the user wishes to do the rational search then the concept-matrix is framed with the list of one or two technical phrases. This concept-matrix is published in the shared memory system called a blackboard. Every domain-specific agent will use the same for the related document identification. If it finds a list of documents then it will proclaim the same to the blackboard and it will display the same to the user. The domain specific agent will travel to the various servers and identify the document that belongs to the subject-hierarchy. Each server is designed to have a phrase-extraction agent and this will form a concept matrix after the phrase-extraction. A concept-dictionary is stored in a centralized place. After framing the concept matrix for every new query, if a phrase extraction agent finds the new technical phrase, then it will be updated to the centralized dictionary. The arrival of the new query is informed to all servers through a moderator as well as the blackboard. Every domain-specific agent is designed to travel through the different servers to retrieve the user requested documents. A concept relativity table is maintained in all the servers as well as the blackboard and is updated like a concept-dictionary once in every moment. This indicates the movement of concept-matrix to different servers. This concept relativity table simply displays category and server number whether it is related to that or not.
6.1.1 Phrase-Extraction Agent

The phrase extraction agent gets the new search query coming in to the Digital Library from a user-interface agent and then it performs a list of preprocessing steps. These preprocessing steps involve stop-word elimination, phrase comparison and phrase matching. The phrase comparison through the concept dictionary is provided with the system. It also does the phrase matching to eliminate the slight differences and it will record the same in the concept dictionary. After extracting the phrase the system will frame the concept-matrix. Other than the central server there exists a phrase extraction agent in every individual server.

6.1.2 Domain-specific Agents

In this present design there are eleven such agents, which are designed to do the concept matching at the individual first level hierarchy. For every subject at the first top level it is designed with a specific agent. Such an agent takes care of the concept comparison and identification of relevant documents related to the user-query. The dispatcher informs about the arrival of the new query in the blackboard system to the subject-specific (domain-specific) agents. Immediately the domain specific agents are actively involved to get the query from the blackboard and perform the concept relativity analysis to identify the related documents that are stored in the department repository. The concept relativity analysis is performed through the Latent Semantic Analysis. The method for identification of semantic relativity is explained in Section 6.4. After identifying the specific documents, the individual domain-specific agents are proclaims the related documents in the blackboard. Then the same is displayed to the user through the user interface agent.
6.1.3 Concept Dictionary

It provides a list of concepts required to run the system. At the initial stages while starting up the system a list of independent concepts taken from ACM computing review classification index, keywords as well as using the words and phrases of Microsoft on-line computer dictionary are entered into the system. These phrases and words are entered by the user through the user interface agent to provide systems ontology. Later additions of concepts will automatically take-place after the identification of new technical phrases.

6.1.4 User-Interface Agent

User-Interface Agents provide various facilities to learn about the user. These facilities include Login system, Search screen, Concept Dictionary, Document Recommendation window, User-Interaction Window and Help window. This login system allows the system to identify the user for user personalization. It is normally designed with a username and password. The search screen gives the facility for text-input entry screen for user search, additional and related term entry. These additional and related terms entry is the optional one, if the user doesn’t want it then he can skip those entries. Second component is the concept dictionary is as explained the previous section and here the user has the facility to invoke this concept dictionary. Third one is the recommendation window, which recommends different types of new documents related to the user-interested area as soon as the user log onto the system. Also this interface has the option to upload a new document in the servers. If a new document comes into the system then it will be informed through the blackboard. Immediately the category of that document is identified through the classification system. Then the system also recommends that document to the related user. The user interaction system will interact to get more information related to search if the user is of an
interactive type. A special option is provided to do a rational search, which means that the system does not consider the user profile and it will go for searching of its own. At last the help system gives various details about the operations of the system. The complete design of this user interface system is explained in Section 6.2.

UIA – User Interface Agent   PEA – Phrase Extraction Agent
LDB – Local Data Base   LCD – Local Concept Dictionary
MDSSPA – Mobile Domain Specific Self-Proclamative Agent

Figure 6.1 Architecture of Domain Specific Multi-Agent Based Retrieval/Recommendation System
6.1.5 Blackboard

This is a shared memory, which stores and exchanges the query as well as the messages required for different servers. The individual domain-specific agents are permitted to read/write the contents of this shared memory. The phrase-extraction agent is permitted to write the phrases, concept-relativity table entries and new document-hierarchies in this shared memory. Also, after retrieving the relevant document, the domain specific agent will proclaim the same into the blackboard.

6.1.6 Dispatcher

It is a simple component that is used to reduce the message flow increase in the system. The arrival of the new concept in the system needs to be informed to the domain specific agents explicitly. If you permit an individual agent to check the blackboard then the system message flow will go to an unmanageable level. In order to avoid this situation a simple dispatcher is designed. This will watch the arrival of the new query in the blackboard and take care of informing it to all the domain-specific agents.

6.2 USER MODELING AND USER INTERFACE AGENT DESIGN

In the present system design the user-access matrix is used to represent the user’s intention that consists of a set of phrases, which expresses at least a partial intention. This is built in adapting the user by means of different methods. The user adaptation is done through the user profile and the user search history is a part of this user profile. The system is designed to learn the user’s behavior and accordingly behaves in the environment. The method of user classification is presented in Section 6.2.1 and the complete method of
building this user access matrix is explained in the following sub-section 6.2.2. After building the user access matrix the system performs the concept relativity analysis to bring the information that is very much related to the concept. The process of the concept matrix building is explained in Section 6.3 and the process of concept relativity analysis is explained in Section 6.4.

6.2.1 User Classifications

The system is designed to identify the various behaviors of different users to adapt the strategy for recommendation. The various types of users and their required strategies are presented as follows. The first question is about the interest of the user to interact with the agent or not. Sometimes the user may hide the agent and hesitate to interact with the user. An option is given to hide the agent. Second question is about the interest to listen to the agent’s suggestion or not. Suppose the user hides while the agent is warning the user about what will be the impact and if the user chooses to continuously hide the agent in spite of the warning, and then it is clear that has chosen to user assumed to be ignore the warnings and suggestions. Third is about the question whether users are patient enough to give complete information to the agent or not. The user screen has the option to enter the additional/equivalent string to represent the searcher’s intention. If he enters this information it is assumed that he is patient enough to listen to the agent and otherwise his is not. Fourth is the intention of the agent to take its own decision or not. Based on the users understanding, the system decides different strategies for recommendation and searching. The agent’s strategy for information, recommendation and searching is given below. This information is learned over a period of time. But, normally while starting up, the agent has the assumption that it has full freedom to take decisions, interactions, give
recommendations and warnings and the user will present a complete set of information.

1. Agent has to search using the global profiles and using user personal profile collected through the search history. It will not give any recommendations and warnings and does not like to recommend the information to the user. It will do a simple search.

2. Agent can interact with user and so it can get some information while searching, but it will not suggest or recommend any information while searching, because the user is not patient enough to listen to the recommendations and warnings.

3. Agent cannot interact with the user, but will suggest or recommend any information while searching.

4. Agent can interact with the user and give recommendations and warnings but it does not have the freedom to take decisions and the user also does not present the complete information.

5. Agent cannot interact with the user, give recommendations and warnings, does not have the freedom to take decisions, but the user has presented the complete information.

6. Agent can interact with the user, will not give recommendations and warnings, does not have the freedom to take decisions, but the user has presented the complete information.

7. Agent cannot interact with the user, give recommendations and warnings and to accept the complete user presented information, has not freedom to take decisions of its own.

8. Agent can interact with the user, give recommendations and warnings and accept the complete user presented information, has not the freedom to take decisions of its own.
9. Agent has the freedom to interact but cannot take decisions, interact and give recommendations and warnings.
10. Agent has the freedom to take decisions, interact, recommendations and warnings but can not present full information.
11. Agent has the freedom to interact, take decisions and interact but can not give recommendations and warnings.
12. Agent has the full freedom to take decisions, interact and accept recommendations an warnings but does not present full information.
13. Agent has the full freedom to take decisions, get complete information, but is neither interactive nor able to give recommendations and suggestions.
14. Agent has the full freedom to take decisions, get complete information, interact, but does not accept the recommendations and warnings.
15. Agent has the full freedom to take decisions, get complete information, give recommendations and warnings and the agent is not able interact with the user to get his ideas.
16. Agent has the full freedom to take decisions, interactions, give recommendations and warnings and the user is also presenting complete set of information.

According to these users behavior it decides the different types of user’s choice and actions.

### 6.2.2 A Method for User Personal Profile Building

The main components of user profiles are User-Name, User-Id, User-Type, User-Subject Categories and User-Access Matrix. Initially the user enters the User-Subject Categories and later on it will be automatically updated by the system. The user-interested subjects’ hierarchies (User-Subject Categories) are also learned by the system and the same is used for the future
searching and is recorded in the user profile. The method of building the concept matrix will differ in different circumstances. These methods are explained in the following section.

1. In this first method, the user-access matrix is built using the user presented search strings. If the user presents related terms that are also included in the matrix. The user access matrix is also similar to the concept matrix, but is called it as the concept matrix because it represents the user intension. There will be only one user-access matrix used for relativity analysis.

2. In this second case, after presenting the search query its category is also recorded in the history. If the new query is related to that then those related phrases from that query is also added and sent for search. If search history is not available then the global search profile is taken for the first time search.

3. The method of user type prediction is explained in the subsection 6.2.1. This user profile normally represents the various subject interests of the user and the type of the user. The Global subject profile is used to explain the different types of general subject concepts and their related items. Many times the subject that the user refers to is not very much clear. The reason is that the subject matters cannot be simply expressed by one or more keywords or phrases. In such a situation, the system needs to track the user-interest through previous history; otherwise it needs interaction with the user. But, many times the user is not patient enough to interact with the system to answer all the queries the agent asks for. This is attributed to user understanding over a period of time. It basically understands the type of the user and accordingly it takes different types of recommendations or retrieval strategy (Fang Liu 2004). That is the kind of assistance different users want about the
reactions to suggestions, actions, and about the reactions towards interruptions, tolerance of agents errors, until the user is patient enough to give either all or a few related terms. In other words, some may want to make basic assumptions on their own through a global profile or sometimes the user himself will present the required terms and the related terms without user profile. Sometimes the user profile string is needed to be considered for searching. After understanding the various types of users, suitable types of retrieval mechanisms are employed for recommendation and retrieval.

First, the user query is entered into the interface system. Then immediately the system will perform the concept relativity analysis to identify the fourth-level sub-hierarchy concept topic. On this process, the phrases and words in the user-query are first extracted and then it will be related to the various keywords and phrases of all the fourth-level sub-hierarchy categories using Latent Semantic Analysis (method of performing the latent semantic analysis and concept relative analysis is given in Section 6.4). The most related concept hierarchies are identified and then these areas are recorded as the related areas to the user. In case of our present design, the system always gives a chance to do a rational search. The user interface provides a check box to indicate the rational search and the user can indicate it through this option. That is the system also does the search without looking into the user-personal profiles if the user so wishes. An important critical part of the interface design is to always keep track of the interest of the user and then issue the warning if it tries to access the ambiguous terms or tries to travel in a different or new direction by checking the user adaptability. The adaptability of the user is understood through different ways and is explained in the next session. Normally the
general subject hierarchy itself gives sufficient information about the hierarchy and the relativity of the system. It can be taken as the general global profile for the whole system design.

6.3 ACM CR CLASSIFICATION AND AUTOMATED CONCEPT-MATRIX FORMULATION

The complete ACM CR classification tree hierarchy is given in http://www.acm.org/class/1998/ccs98.html [1998 Version]. In this present work we wish to add more number of concepts/phrases at every fourth level sub-hierarchy. Each and every document is represented as a concept and every concept is represented thorough a concept matrix. Concept matrix contains a list of technical phrases. The system will automatically add any number of concepts at this level. The classification and extraction of technical phrases to construct a concept matrix is the very critical task, for which we use the list of ACM proper noun index, Keyword index. Apart from this a list of words and phrases from Microsoft on-line computer dictionary is also used. The phrase extractor agent automatically extracts using these words and phrases, an additional set of words and phrases. The occurrences of all these phrases are taken and then the relativity between the list of index and newly extracted phrase is taken into account in order to include the particular technical phrase in the concept matrix. These phrases are also stored in the concept dictionary and used for the future usage. Normally while extracting the technical phrases from the list of complete phrases extracted from the query, the occurrence of these independent words and phrases are just seen those lists and such selected phrases as termed as the technical phrases.
6.4 METHOD FOR CONCEPT-PATTERN RELATIVITY

Latent Semantic Analysis (LSA) is a theory and method for representing the contextual usage of meaning of phrase relativeness by statistical computations applied to a large corpus of text. Phrase and passage meaning representation derived by LSA have been found to be capable of simulating a variety of human cognitive phenomena. After processing a large sample of machine-readable language, LSA exhibits the phrases, either taken from the original corpus or new, as points in a very high dimensional semantic space.

In this case it is represented as a conceptual matrix and it also permits one to infer about the relation of expected contextual usage of phrases. LSA applies a Singular Value Decomposition (SVD) to the matrix; this is a form of a factor or more properly the mathematical generalization of which factor analysis is a special case. In SVD, a rectangular matrix is decomposed into the product of three other matrices. One component matrix describes the original column entries in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed. The reconstructed two-dimensional matrix that approximates the original matrix and a few highest values are selected to reconstruct the original matrix.

Each document in the particular sub-hierarchy is represented by the rows and the columns represent each phrase with respect to the document. Learning human like knowledge consists in formulating a bivariate frequency table with row I representing the $i^{th}$ phrase and column $j$ representing the $j^{th}$ document (or between any two entities) and $f_{ij}$ the corresponding frequency
measured by the Shannon’s measure of information \( \sum p \log p \). This together with the dimension reduction will constitute the constraint satisfaction for prediction between the user-query and the expected values to make recommendation/retrieval. Actual data pertaining to any two measurable entities (phrases and sentences, text classification in digital libraries), etc. will have to collect. Sets of examples pertaining to each of the two entities can be exhibited in a bi-variate frequency table for determining the relationships between any two examples. Tables can be formulated for comparison and valid retrievals.

SVD is a powerful technique employed for solving a linear system of equations \( AX=B \), in M equations of N unknowns with \( M \geq N \) in order to get unique set of solutions; a set of singular solutions, infinite number of solutions; non trivial solutions or trivial solutions based upon the nature of the coefficient matrix \( A \), whatever maybe the vectors \( X, B \). Concepts of rank, null space, range space of linear algebra are essential in formulating the computer program for any practical problem in conformity with the decomposition of the matrix \( A \)

\[
[A]=[U][W][V^T]
\]

in the usual notation. When more equations than the unknowns are given, relevant solutions can also be obtained by least squares method.

After the reconstruction of the original matrix the correlation between required user-concept matrix and the existing document in the sub-hierarchy is found. If the correlation is high then documents are retrieved and presented to the user. The same process is repeated for all agents. If none of them find good correlation in its sub-hierarchy it is proclaimed that the information is not available in the hierarchy.
The main issue while using the LSA is the size of the matrix could get very high and the system may not be able to process it sometimes. In order to avoid this situation and keep the matrix under control every time only a set of five to ten documents alone are taken out of entire set of the given sequence for relativity analysis.

6.5 SIMULATION EXPERIMENT

The system is simulated using Apache web servers. All these servers are hosted with IBM Tehiti server to run the Aglets (Danny Lange 1998). The subject-specific agent, phrase-extraction agent and user interfaces agent are developed using this aglets. There are four such Apache web servers that are hosted and one of them acts as a central server that runs the moderator, blackboard and central concept dictionary. The system is designed to adopt the ACM CR classification schemes. Also a specific domain specific agent is designed for each category at the first level. Each server stores more than one domain specific aglets and according to the storage of specific set of documents. The entire sample document taken for experimentation is already uploaded in the specific servers. Even through there are eleven such first level categories in the ACM CR Classification scheme, this experiment was conducted with limited set of servers and in this case there are only four domain-specific servers that existing. In this design more than one domain-specific aglet reside in the same server, because every server stores more then domain-documents.

An Agent Transfer Protocol (ATP) is used to communicate with these different agents. Aglet uses a technique called serialization to transmit data on the heap and to migrate the interpretable byte-code. These aglets are supporting message passing and broadcasting. Each aglet is integrated with
the functional components of this architecture. The blackboard system is shown as the explicit component and is implemented by using standard Java serialization. For the domain specific aglets (Agents) initially the user has to specify the training sample document either from the local machine or from the web through user-interface aglet.

Each domain specific aglet is designed to learn the concept-matrix of that specific hierarchy. The training documents are indicated with specific category-hierarchy and this is considered as global subject profile. After the training is over, the query is given to the system through user interface screen and this is preprocessed and then the user access matrix is framed as recorded and passed to blackboard. Then the moderator broadcasts the message to all the domain-specific aglets about the arrival of a new query and every domain-specific aglet gets the same and processes it to retrieve or recommend a set of documents related to the given user query. Normally every user is expected to sign in to the system through proper user-creation method. All the user details are recorded and the user-profile is built as explained in Section 6.2.2. The documents retrieved from the different servers are directly given to the user and no specific filtering is done after collecting it from different domain specific-aglets.

6.6 EXPERIMENT RESULTS AND DISCUSSION

In these experiments, the ACM CR classification system at the fourth-level hierarchy there exists nearly 1120 categories and the present system is trained with few documents under each category. Later while uploading the documents the system is automatically able to classify the documents under a specific category. These documents are compared with the user query as soon as the user presents the query. During this process a specific user profile is
also built over a period of time. Each user has to either sign in to the system or login to the system and during the process of signing in to the system the user has to give the explicit user details to the system. These static profiles get modified over a period of time during the process of searching.

### 6.6.1 Data Sets

In these experiments, during the first phase data sets were collected from different Internet portals. In the first data set collection a set of 6768 papers are collected from ACM and Google Web Directory by submitting nearly two hundred queries on various topics of information technology. For each query at the most 20 to 25 documents are taken for experimentation. Also the metadata was constructed for all these documents. After that these documents were later uploaded in to various domain-specific servers after categorizing in to different ACM categories.

### 6.6.2 Performance Measure

To evaluate this approach different users measure the retrieval/recommendation with 96 queries. In order to evaluate the effectiveness of retrieval/recommendation system it is measured with the well known precision measure used for different queries.

The first set of well known metrics (Precision, Recall) (George Forman 2003) are used and are shown by

\[
\text{Precision} = \frac{\text{Number of retrieved documents that are relevant}}{\text{Total number of document retrieved}} \quad (6.2)
\]
Recall = \frac{\text{Number of retrieved documents that are relevant}}{\text{Total number of relevant documents}} \quad (6.3)

The Second set of metrics is the search assistance (Judit Bar-Ilan 2005) materialized by relevance feedback, related page retrieval and personalization. These search assistance measures are measured through the user satisfaction performance measures. The Doll and Torkzadeh 1988 (Li Xiao 2002) measures the end user satisfaction strengths by various measures such as content, accuracy, format, ease of use, and timeliness. Similarly in this present experiment the user is asked to give five different ranks to estimate user satisfaction feedback after using the system. This is shown in the following Table 6.1

<table>
<thead>
<tr>
<th>Parameters/Strength</th>
<th>Content</th>
<th>Accuracy</th>
<th>Format</th>
<th>Ease of Use</th>
<th>Timeliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High (10)</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>High (8)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium (6)</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Normal (4)</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.6.3 Experiment Results Metrics

Based on the measurement pertaining to precision and recall the relevant documents are retrieved in this retrieval system as well as from the ACM digital library. In case of ACM Digital Library the classification-
hierarchy and the collection are developed specifically for Computational Literature. Similarly, the Multi-Agent System based retrieval/recommendation system also developed for Computational Literatures uses the ACM Classification hierarchy. Therefore it is natural to make a comparison with ACM Digital Library. Compared to retrieval effectiveness and relevancy in terms of precision and recall, the multi-agent system based retrieval indicates much better performance measures than the ACM Digital Library. This is shown in the Figure 6.2.

![Comparison of Precision Vs Recall](image)

**Figure 6.2 Precision and Recall for Multi-agent Retrieval/Recommendation System Vs ACM Digital Library.**

To evaluate this multi-agent based retrieval and recommendation approach a set of 30 different users were chosen to evaluate the system and their satisfaction level are presented in the Figure 6.3. It also compares metrics of the ACM digital library uses satisfaction levels. From this experiment it is established that the retrieval accuracy is reasonably good and also it is observed that over a period of time the system is adaptive in nature.
6.7 CHAPTER SUMMARY

An attempt has been made in this work to design and develop an adaptive multi-agent framework for research literature retrieval using ACM CR classification hierarchy. This work has revealed that the Latent Semantic Analysis along with concept matrix setup enables one to retrieve related concepts effectively even though the user given phrase is not available. The initial system is setup with a set of few predefined words and phrases and later on it acquires whole the set of phrases. It also yields good results in terms of concept relativity. In this work self-proclamation is identified as the new characteristics of the multi-agent system and is established through these experiments. This characteristic of self-proclamation enables the system to retrieve/ recommend documents quickly. The system is tested with ACM CR categories. The same may be extended for other categories like DDC and other type of classification systems.