The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

Intelligent Information Retrieval System

Young Im Cho

Dept. of Computer Science, Univ. of Suwon, San2-1, Wau-ri, Bongdam-eup, Hwaseong, Gyeonggi-do
(Tel : 82-31-229-8214; Fax : 82-31-229-8281)
(ycho@suwon.ac.kr)

***Abstract*: In this paper, we propose a multi agent based information retrieval system in digital library. We propose a new algorithm for each multi agent module in detail. Each module is for personalized information retrieval. In the future, we have to apply to a real situation for proving.**

***Keywords*: Information retrieval, Multi agent system, Agent platform**

I. Introduction

Recent developments of the internet and network technologies evoke the technical change of the data processing from conventional centralized and local processing system to the distributed processing system. Many studies have been actively carried out in a distributed processing environment by using agent systems for efficient network management. Multi agent systems promote the efficiency in solving problems by cooperating among agents. Also, each agent independently manages its own tasks by dividing a whole work into smaller units and assigning them to each agent. There are many application areas in the real world, using the multi agent systems. One of these areas is the digital library system. This is a library developed to replace the conventional library, in order to serve information from databases on the web to users according to the development of computers and the related fields.

University of Michigan Digital Library (UMDL) [1, 2] is one of the most famous agent-based digital library systems. Agents in UMDL autonomously use their resources through negotiation among them. Also, the UMDL is very flexible for updating the library system when a new agent needs to be added. As another famous agent-based digital library, Green Stone Digital Library (GSDL) [3] was developed by New Zealand Digital Library project team. The GSDL provides a novel method for organizing information using the open source digital library [4, 5] and making it useful over the internet. McNab et al. [6] improved the open source digital library system by proposing a flexible protocol for communicating between an interface server and a search engine. Their system is suitable for distributed computing environment. Also, Witten et al. [7] improved the functionalities of finding, collecting, and organizing information in the GSDL for the distributed environment.

Even though most conventional digital library systems provide the convenient life to users for searching and managing information, a hot issue has been paid attention to many researchers. It is that their search results include undesired results because of no information of users' profile. Thus, users have to establish the directories in users' personal computer for developing personalized digital library. In order to solve the problem, the personalized digital library systems have been developed. Bollacker et al. [8] introduced a personalized CiteSeer digital library system which is an automatic paper search system. The system can track and recommend the similar papers whose topics are very relevant, using the content-based relatedness measures [10]. Torres et al. [9] improved the performance of the automatic paper search system by developing the hybrid recommender which is the combination of user-based collaborative filtering method [11] and content-based filtering method [12].

The personalization is done by analyzing the topic of papers. In this case, the papers whose topic is not relevant to the queried keywords request by a user but contents are relevant to them are not recommend. In order to solve the problem, we consider the abstracts in the papers for providing a personalized paper search list according to the user's behavior on the papers and the relevance among keywords in the abstracts.

Also, another problem the multi agent-based digital library systems have is that users themselves should visit all possible search servers one by one. To overcome the problem, we propose a new platform of multi agent digital library system which is mobile search system. Users do not need to visit all possible search servers with the same query by using the new platform. Our system automatically visits to all possible search servers when user requests a query. In this case, the scheduling of visiting the search servers is needed. Also, the negotiation for the results searched from the servers needs to be made for complicated situations such as the duplicated search results from multiple servers. Also, our personalized paper search algorithm builds user's individual relevance network from analyzing the appearance frequencies of keywords in the searched papers. The relevance network is personalized by providing weights to the appearance frequencies of

keywords according to users' behaviors on the searched list, such as "downloading," "opening," and "no-action." Also, we enable interaction among multi agents by developing an artificial negotiation algorithm.

In the experimental section, we demonstrate our method using 100 faculties' search information employed in the University of Suwon. Also, the performance of our method is compared with that of the conventional paper search system by surveying the satisfaction of users for both systems. In addition, we analyze the searching speed of our system to show both the advantage and disadvantage of our mobile-based searching system.

II. Proposed Multi Agent-based Platform for Information Retrieval

A digital library serves a lot of information on-line. The advantages of digital libraries over conventional libraries are user friendly, on-site service, and accessibility. However, in case of not having standardized platform, the search of heterogeneous information from digital libraries may be hard, or impossible in some cases. To solve the problem, we develop a mobile multi agent-based digital library system using DECAF. The 'mobile' means that agents autonomously move a server to a server without user's interference. The mobile multi agent-based digital library system can eliminate unnecessary and duplicate information stored in Internet or DB. Also, existing digital libraries do not have or learn about the user's information. It causes unnecessary or useless information for the user to appear in the searched results. To solve the problem, we propose a mobile multi agent-based personalized digital library system (MAPS). MAPS provide the personalized search list from users' usage history. The overall structure is in Fig.1.

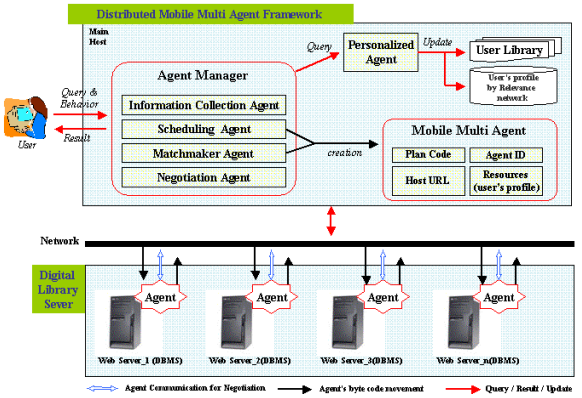

Fig.1 Overall Structure

Once a user requests a query with keywords, Agent Manger (AM) in DMMAF receives the query through the Matchmaker Agent and prepares the user's profile because the search results are obtained by the user's profile. Scheduling Agent determines the order of the agents' visit to the digital library server. The documents are collected from an agent's visit to web servers (DBMS) according to the scheduled order. Digital Library Server (sub host) accepts the agents by defined port, agents approach the DBMS of web servers. During the agent visiting, Negotiation Agent customizes the collection at each web server by eliminating duplicate documents with its previously visited web servers via a network. Each visit is done in company with the Mobile Multi Agent (MMA). The attributes of MMA includes the plan code, agent ID, host URL, and resources (user's profile). The plan code is the agent's task generated by Scheduling Agent. The agent ID is the unique number of each agent, which is generated by Matchmaker Agent. The host URL is the address to return. The resource is the user's profile. Personalization Agent provides the user a personalized search list from the negotiated documents according to the user's profile. Then, Information Collection Agent shows the final searched results to user through the user interface (UI).

DMMAF is a proposed agent framework for distributed environment in this paper. DMMAF has five components such as Agent Manager, Plan Editor, ANS, MMA , Transfer Protocol.

Agent Manager receives a user's query, creates agents, and controls the agents' operations. Matchmaker Agent communicates between Matchmaker Agent and UI. Also, Matchmaker Agent creates agents according to the schedule generated by Scheduling Agent. Scheduling Agent searches the optimal path to visit servers and informs it to the plan editor in DECAF. The plan editor creates the plan codes. Information Collection Agent sends the final searched results to the user through UI. Negotiation Agent removes the duplicated results through KQML communication language among MMAs. Agent Name Server creates the MMA and saves the multi agents' record about creation and deletion information of each MMA. The transfer protocol of the MMA is to encrypt and decrypt the agent byte code, plan code, user profile, and resources. Byte codes which are sent through network are doing the parsing process. MMA visits the host having user keyword and user profile. MMA negotiates with other agents using negotiation algorithm and search the DB.

The proposed negotiation algorithm in Negotiation Agent is explained in detail as seen in Fig. 2.

```
    /* improved negotiation algorithm of MMA */
Switch(one of 5 relations) { /* relation_name(sender, receiver) */
  case 1: Add_R(Ai_Mi, Aj_Mj)
          /* (i≠j), Ai=agent, Mi=method */
       while(Aj_Mj is finished)
          no-operation (Ai-Mi); break;
```

```
 case 2: Compensate_R(Ai_Mi, Aj_Mj)
          Ai_Mi and Aj_Mj are operated in real situation
          by call(inference) function;
         Inference() is from user profiles or user information repository;
         break;
 case 3: Replace_R(Ai_Mi, Aj_Mj)
         Aj_Mj=Ai_Mi;
         break;
 case 4: Contradict_R(Ai_Mi, Aj_Mj)
         Aj_Mj=Aj_Mj;
         break;
 case 5: Activate_R(Ai_Mi, Aj_Mj)
         if wait(Aj_Mj) then awake(Aj_Mj) and restart;
         break;
 default:
}
```

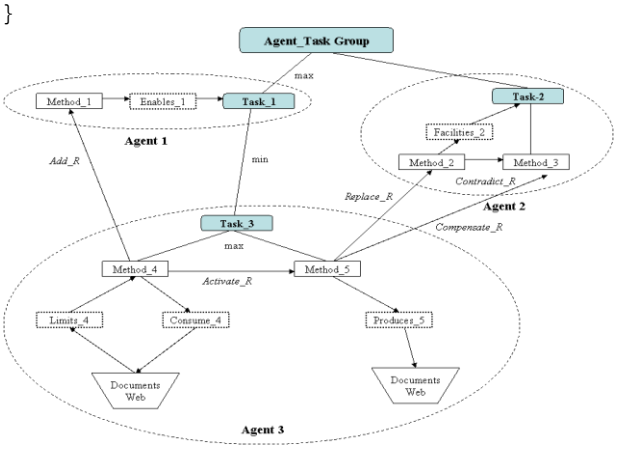

Fig.2 Negotiation method among MMA

Agent_Task Group assigns Task_1, Task_2, and Task_3 to Agent 1, Agent 2, and Agent 3 respectively. The tasks are automatically decomposed into methods, and the agents execute the assigned tasks by the methods. There are five types of relationships among methods. *Add_R* is the add relationship, i.e., adding the result of a method to that of other methods. *Activate_R* is the relationship making the running method keep running continuously. *Compensate_R* is the relationship that compensates the results of methods if needed. *Replace_R* replaces the results of receiving methods with those of sending methods. *Contradict_R* ignores the results of receiving methods. Also, there are lots of relationships between methods and tasks and between methods and resources, such as *Enable*, *Facilitate*, *Produce*, *Consume*, and *Limits*. In the negotiation algorithm, if the agents in the same level take different actions, then *max* operation operates to produce the output of the agents. Otherwise, *min* operation is operated.

Fig. 3 is the flow chart of the negotiation process between two MMAs. Fig. 4 is the schematic of the negotiation mechanism of information exchange between two MMAs by KQML protocol.

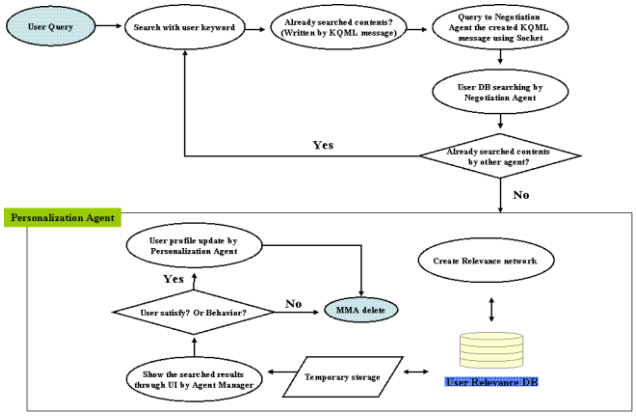

Fig.3 The flowchat of the negotiation process between two MMAs

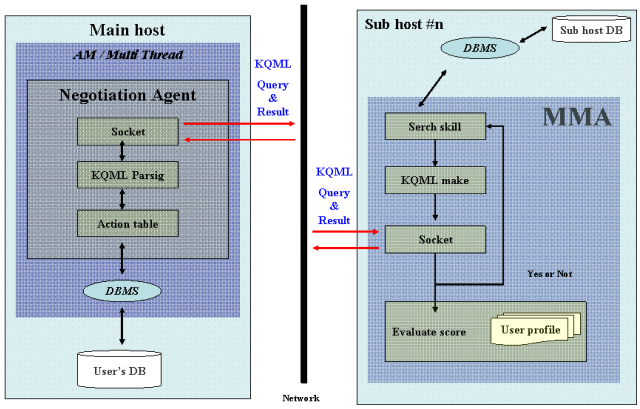

Fig.4 The negotiation mechanism of information exchange between two MMAs by KQML protocol

The information about MMA moving is transferred in byte stream type surrounded by the defined tag of the KQML protocol. The byte stream is parsed according to the defined tag and the clone byte of the byte stream is transferred to the next web server. The detailed algorithm of the moving MMA is the followings.

```
receive( Socket s ) {
  InputStream inputByteStream = new InputStream();
  inputByteStream = new InputStream( s.getByteStream() );
 while( ( tempByte = inputByteStream.read()) != End ) {
   Parsing()
   if( nextHost == null )  {
    try{
      ByteCode[index++] = (byte)tempByte ;
    } catch( ArrayIndexOutOfBoundsException ) {
      System.out.println("Termination of all host visiting");
      Notify to Agent Manager about Job termination;
    } // end of try
    } else if ( Exist sending socket to next host? ){
      nextHostSocket.write( (byte)tempByte );
   }
  } // end of while
} // end of receive()
```

The algorithm of parsing process is the followings.

```
parsing( InputStream in ) {
  while( ( tempByte = in.read() ) != End ){
    if( Start character of Tag? ){
      inputStream mark();
      Read byte until the Tag length;
      Identification the Tag type;
      if( fail the defined TagSearching ?){
        Reset according to marked Stream index;
        Process according to current Tag type;
      } else {
        Branch or activate according to the current Tag and the
identified next Tag;
      }
    } ese if( Ending character of Tag?){
      if( nTag  inTag = false;
      else Process according to current Tag type;     }
  } // end of while
} // end of parsing()
```

III. Conclusion

In this paper, we proposed a platform of a mobile-based personalized paper search system. By using our system, users do not need to visit all possible search servers with the same query by using the new platform. Also, our system provides users' personalized search list by building user's individual relevance network from analyzing the appearance frequencies of keywords in the searched papers.

We have to apply this algorithm to the real system in the future for proving of this algorithm.

REFERENCES

[1] W. P. Birmingham, An Agent-based Architecture for Digital Libraries, D-Lib, July 1995.

[2] W.P. Birmingham, E.H. Durfee, T. Mullen, "The Distributed Agent Architecture of the University of Michigan Digital Library", AAAI Spring Symposium on Information Gathering form Heterogeneou, Distributed Environments, AAAI Press.

[3] http://www.greenstone.org/cgi-bin/library.

[4] C. Logoze and D. Fielding, "Defining Collections in Distributed Digital Libraries", D-Lib Magazine, vol. 4 no. 11, 1998, http://www.dlib.org/dlib/november98/lagoze/11lagoze.html.

[5] C.M. Bowman, P.B. Danzing, U. Manber, and M.F. Schwartz, "Scalable Internet Resouce Discovery: Research Problems and Approaches", Communications of the ACM, vol. 37, no. 8, pp. 98-107, 1994.

[6] R.J. McNab, H. Witten, and S.J. Boddie, "A distributed Digital Library Acchitecture Incoperatiing different index styles", Proceedings of IEEE Advances in Digital Libraries, pp. 36-45, 1998.

[7] H. Witten, R.J. McNab, S.J. Boddie, and D. Bainrige, Greenstone, "A Compreshensive Open-Source Digital Library Software System", Proceedings of Fifth ACM Conference on Digital Libraries, pp. 113-121 2000.

[8] K.D. Bollacker, S. Lawrence, C.L. Giles, "A System for Automatic Personalized Tracking of Scientific Literature on the Web", Proceedings of ACM Conference on Digital Libraries, pp. 105-113, 1999.

[9] R. Torres, S.M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing Digital Libraries with TechLens", ACM/IEEE-CS Joining Conference on Digital Libraries, pp. 228-236, 2004.

[10] K. Bollacker, S. Lawrence, and C.L. Giles," An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications", Proceedings of The Second International Conference on Autonomous Agents, pp. 116-123, 1998.

[11] W.P. Lee and T.H. Yang, "Personalizing Information Appliances: A Multi-agent Framework for TV Program Recommendations", Expert Systems with Applications, vol. 25, no. 3, pp. 331-341, 2003.

[12] T. Kamba, K. Bharat, and M.C. Albers, "An Interactive Personalized Newspaper on the Web", Proceedings of International World Wide Web Conference, pp. 159-170, 1995.

[13] J.R. Graham and K.S. Decker, "Towards Distributed, Environment Centered Agent Framework", Appearing in Intelligent Agents IV, Agent Theories, Architectures, and Languages.

[14] Lesser, Victor; Horling, Bryan; Klassner, Frank; Raja, Anita; Wagner, Thomas; and Zhang, Shelley. "BIG: An Agent for Resource-Bounded Information Gathering and Decision Making", Artificial Intelligence Journal, Special Issue on Internet Information Agents, vol. 118, no. 1-2, Elsevier Science, pp. 197-244. May 2000.

[15] John Graham, Victoria Windley, Daniel McHugh, Foster McGeary, David Cleaver, and Keith Decker, "Tools for Developing and Monitoring Agents in Distributed Multi Agent Systems", Workshop on Agents in Industry at the Fourth International Conference on Autonomous Agents, Barcelona, Spain, June, 2000.

[16] Baeza-Yates, Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1992