Mobile Multi Agent based Intelligent Digital Library System

Young Im Cho* • Myung Sook Kim**

*Dept. of Computer Science, Pyongtaek University
111 Yongi-dong, Pyongtaek, Kyongki-do, KOREA, 450-701
e-mail : yicho@ptuniv.ac.kr
** Graduate School of Education, Computer Education major, Ewha Womans University
11-1 Daehyun-dong, Seodaemun-gu, Seoul, KOREA,120-750
e-mail: gracemskim@ewha.ac.kr

***Abstract*- There are several problems in the searching of database of the existing digital library. First, as the searching method is one dimensional method and distinguish the existence of the searching keyword from the database, the searching result is very simple. Second, the results may contain unnecessary information under condition that not given the priori information on the user. Third, whenever a client connects to the servers, he has to receive the certification and be under the dominant power of influence of network.**

To overcome such problems, we proposed a new platform of mobile multi agents for a personal digital library. For developing a new platform, we combined existing DECAF multi agent platform with Voyager which is a mobile ORB, and also a new negotiation algorithm and a scheduling algorithm are proposed, so that we developed PDS using this new platform.

For the higher relationship among searched documents from mobile servers, an unsupervised neural network is applied. For the user's preference, some modular clients are applied to a neural network. A multi agent platform and a mobile agent platform are combined to develop a new mobile multi agent platform so as to decrease a network burden. Also, a new negotiation algorithm and a scheduling algorithm are activated for the effectiveness of PDLS.

PDLS is more intelligent system capable of establishing database in his computer by learning interests of users. We tried to set up the theoretical structure of the multi mobile agents and develop an algorithm of the modified intelligent negotiation agent for inducing interaction among multi agents. .

***Keywords*. Distributed system, Mobile multi agent platform, Personal digital library**

I. Introduction

Recent developments of the internet and network technologies evoke the technical change of the data processing from a conventional centralized and local processing system to the distributed processing system. The research about this network and the various approaches have been studied in order to efficiently manage mutual operations in such a network environment.

Many studies have been actively carried out in a distributed processing environment by using agent systems for efficient network management. An agent system has the following characteristics: multi agents in the distributed environments promote efficiency by solving one problem through any cooperation. Each agent manages the problem by dividing a common work into the number of agents, or each agent manages it independently, and then they solve the problems by analyzing the results. In addition, it has some advantages such that intelligent agents reflecting the tendency of users make no limitation of movement in a network, and it remarkably decreases the network traffic [1].

There are so many application areas of agents in the real world. One of these areas is a digital library system. The digital library is called an *electronic library* or a *virtual library*. 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.

However, there are several problems in the searching of data of the existing digital libraries. First, as the searching method is one dimensional and distinguishes the existence of the searching keyword from the database, the result is very simple. Secondly, the results may contain unnecessary information under a condition that was not given the prior information about the user. Thirdly, whenever a client connects to the servers, he has to receive the certification and be under the dominant power of the influence of network.

To overcome such problems, I proposed a new platform of mobile multi agents for a personal digital library in this paper. For developing a new platform, I combined the existing DECAF (Distributed Environment Centered Agent Framework) multi agent framework [2] with Voyager which is a mobile ORB (Object Request Broker). Also a new negotiation algorithm and a scheduling algorithm are proposed, so that I developed a PDLS (***P***ersonal ***D***igital ***L***ibrary ***S***ystem) using this new platform. Although the partial studies for a personal digital library have been carried out, there has been none about the integrated and systemized personal digital library.

For the higher relationship among searched documents from mobile servers, an unsupervised neural network is applied. For the user's preference, some modular clients are applied to a neural network. A multi agent platform and a mobile agent platform are combined to develop a new mobile multi agent platform so as to decrease a network burden.

Also, a new negotiation algorithm and a scheduling algorithm are activated for the effectiveness of PDLS.

PDLS is different from the electronic paper service system which is supplied only to members. It is a more intelligent system that is capable of establishing a database in users' computer by learning the interests of those individuals. In this paper, I tried to set up the theoretical structure of the multi mobile agents and develop an algorithm of the modified intelligent negotiation agent for inducing interaction among multi agents.

This paper is composed of five chapters. Multi agents and DECAF framework is explained in chapter 2. PDLS based on a new mobile multi agent platform is explained in chapter 3. The simulation results of PDLS are explained in chapter 4, and finally the conclusions are in chapter 5.

II. Multi Agent System

A. DECAF framework and Voyager

DECAF (Distributed Environment-Centered Agent Framework) is a conventional framework to design a lot of intelligent agents [2]. DECAF is a kind of operating system including agent communication, planning, scheduling, monitoring, coordination, diagnosis, and learning among agents. DECAF makes a socket program by itself, and presents some building blocks which makes messages and communicates between agents. Therefore, users or programmers can produce agents without having some knowledge about API approaches. Also, users or programmers do not need to make the communication codes directly to communicate among agents. DECAF produces a KQML protocol automatically which sends messages and searches other agents and interacts between agents. Agent systems have been developed using various languages and platforms, and they are classified into so many types by purpose. In DECAF, many agents' tasks are divided by both GPGP (Generalized Partial Global Planning) and TAEMS (Task Analysis Environment Modeling and Simulation) algorithms.

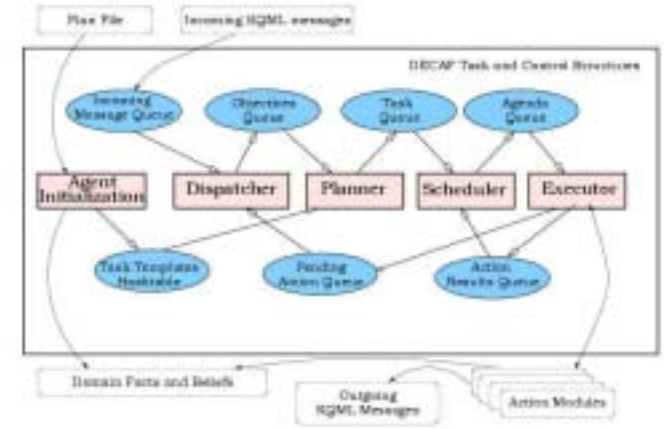


Figure 1. DECAF

GPGP is for improving of PGP which acts as a coordination algorithm of multi agents [3]. The first advantage of GPGP is that it reduces the system overhead which occurs by overlapping interaction among agents. And the second advantage of GPGP is that it is independent from some specific domain areas. Therefore, GPGP can make heterogeneous multi agents system having different functions. User's requirements can be decomposed by GPGP, and be structured by TAEMS [4]. User's requirements can be decomposed by GPGP, and be structured by TAEMS(Task Analysis Environment Modeling and Simulation)[4]. The root task can be decomposed into subtasks, and the subtasks can be decomposed into methods. The leaf node acts as a method which means actually acted elements.

Voyager [5] is a distributed mobile agent's framework for developing agent's applications, whereas DECAF is a non-mobile agent's framework. Voyager is an interactive framework with Java programming. Also, Voyager can activate any Java class in remote sites, and it makes use of network bandwidths effectively.

B. The Concept of Agent based Digital Library

A digital library serves a lot of information on-line [6,7]. The advantages of digital 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, as well as impossible. If it does not have or learn about the user's information, unnecessary or useless information will appear in the searched results from the digital library.

Figure 2 shows the concept of an agent based digital library which is proposed in this paper. This is based on the proposed mobile multi agent framework to search many servers concurrently, using multi agents. Also user's profile can be produce into a database in this system.

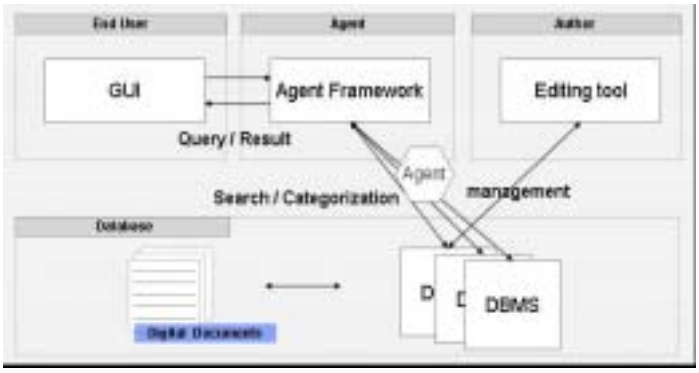


Figure 2. Agent based Digital Library

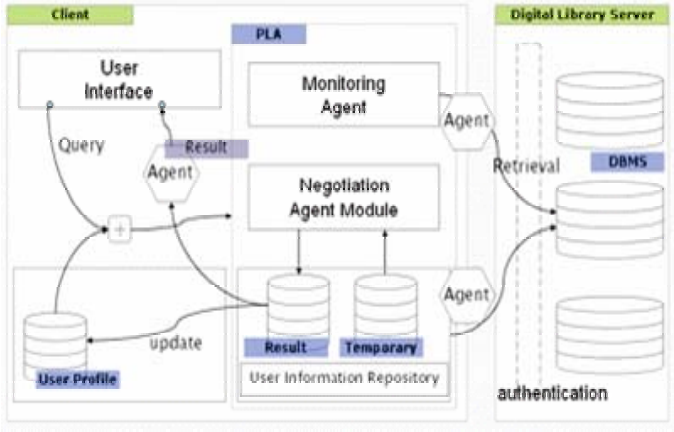
Each agent can access DBMS and search documents according to the user profile. And then each agent categorization of the searched results.

III. Personal Digital Library System based on a New Mobile Agent Platform

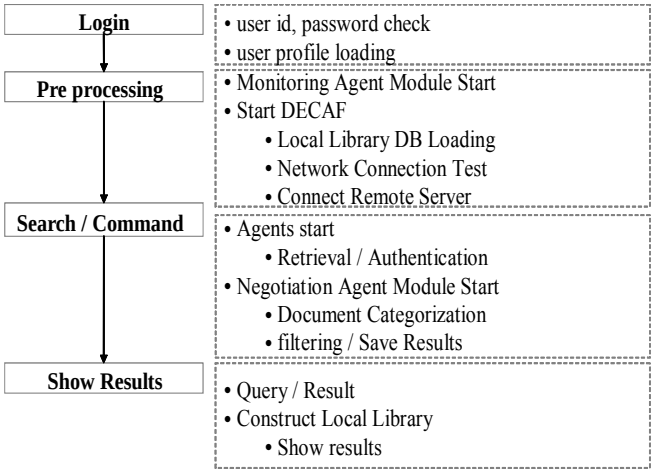
A. System Structure

The proposed system, in Figure 3, is a Personalized Digital Library System (PDLS) based on a new multi mobile agent

platform. The system combines a mobile system and a distributed processing system to make an optimization of behaviors in a distributed environment. To establish a distributed environment, DECAF is used, and to activate a mobile framework, Voyager is used here.



(a) PDLS Structure



(b) Interactions among Detailed Modules in PDLS

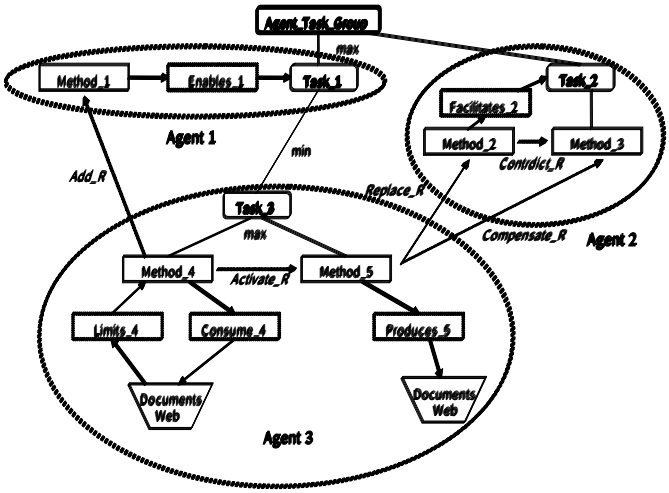
Figure 3. PDLS Structure and Interactions among Modules

The PDLS is composed of two parts: client group and server group. The client group is composed of three modules. First, a user interface module lets users make use and control the library. Second, a user profile control module learns the user's preferences by neural network (SOM [8]), and makes databases accordingly. Third, a PLA (Personal Library Agent) module makes multi agents in real time, and searches information from the library according to the user's profile, and stores the searched results into a database. The interactions among detailed modules in PDLS are explained in Figure 3(b).

B. PLA (Personal Library Agent)

As shown in Figure 2, PLA has two modules and two databases. The monitoring agent module is composed of Voyager and DECAF, and it monitors the agents' movements and controls their executions. When the user's requirements are transferred to the PLA, the monitoring agent module checks whether the servers are available or not. After that, it makes some agents, and passes them to the servers. The searched results are saved in a temporary repository. They are filtered by negotiation agents, and the final results are saved in the result repository.

/* improved negotiation algorithm of multiagents */




```
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:
}
```

Figure 4. Negotiation Algorithm among Multi Agents

In the proposed platform, the relationship among multi agents in negotiation agent module is in Figure 4. Agent_Task Group generates Task_1, Task_2 and Task_3 according to Agent 1, Agent 2, and Agent 3. Tasks are automatically decomposed into methods and do their assigned tasks. Each method has five types of methods' relationships. *Add_R* is to add the results of actions to the results of the other methods. *Activate_R* is to let the running method run continuously, *Compensate_R* is the relationship

that the results among methods need to be compensated. *Replace_R* is to replace the results of receiving methods with the results of sending methods. *Contradict_R* is to disregard 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, Limits* and so on. In the negotiation algorithm, if the agents in the same levels do the different actions, then *max* operation is operated to produce the output of the agents, and if the agents in the lower levels do the different actions, then *min* operation is operated.

C. The Construction of User's Profile

The construction of the initial user's profile is constructed by the user's first input information. According to the user's searched results, PDLS endows the user's keywords to weight values, and updates user's profile information by SOM (self organizing map) network in real time [7].

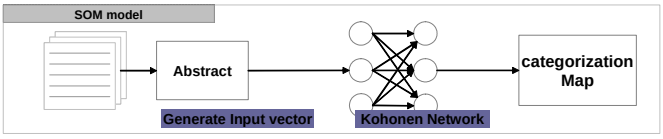


Figure 5. SOM network

SOM is an unsupervised neural network, In this paper, 2-layered SOM network is used here.

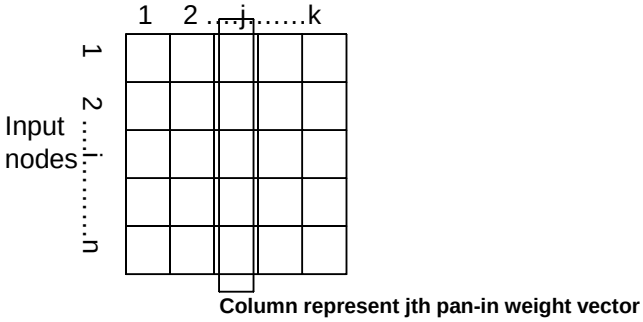


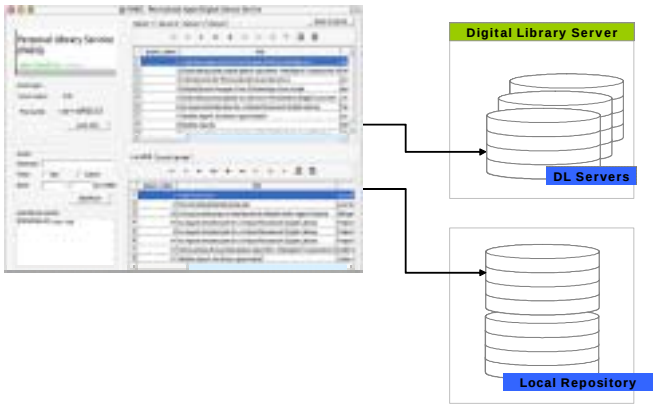
Figure 6. Weight Vector in SOM

In Figure 6, *wij* means a weight vector from node *i* to node *j* in SOM network. In SOM, a weight vector having the nearest Euclidian distance is categorized when searching documents.

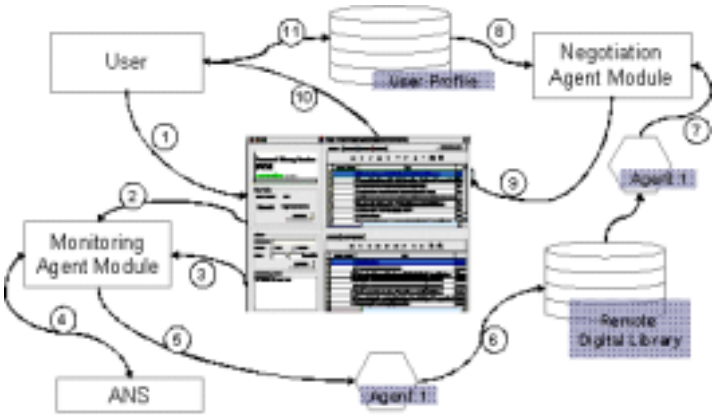
The user interface of PDLS is composed of four windows. The user's window is for entering the user's information and for recalling the user's profile from databases. The monitoring window is for checking agent's activities. The remote window shows the final results, and finally the local window shows the constructed hard disk information of the user's computer by PDLS.

IV. Simulation Results

The user interface is composed of four panes, and each pane is interconnected. The login window and query window is for user login. The user pane is for checking the agents' activation states including monitoring. The remote pane is for representing the information of remote servers and searched results. The local pane is for representing the building states of a local library. The scenario for PDLS simulation is in Figure 7.



(a) User Interface and Module Relations



(b) PDLS Simulation Scenario

Figure 7. PDLS Interface and Simulation Scenario

① User login by user interface
② The monitoring agent(MA) is activated, and MA check the current connected remote digital library. At the same time, a user profile is read from database.
③ The queries according to users are sent to PLA.
④ MA makes a search agent 1and registers it in ANS.
⑤ The search agent 1 is sent to the remote library.
⑥ The agent 1 is activated by receive the parameters from PLA.
⑦ The results from the agent 1 with information such as name and index and abstract of the remote library are sent to PLA. And then they are sent to the negotiation agent(NA) for negotiation. At the same time, they inform to the MA and ANS.
⑧ A is clustered using SOM network according to the received results
⑨ The results by the user profile can be shown in the remote plane.
⑩ user can move to his local library after receiving his necessary results.
⑪ The user profile's update is activated.

Table 1. Keyword based Categorized Results

Category	Agent System	Neural Network	Digital Library
1	framework	learning	virtual
2	mobile	neural	library
3	java	architecture	indexing
4	multi	network	structure
5	personalized	simulation	distributed
6	distributed	layer	agent
7	environment	agent	retrieval
8	neural	artificial	autonomous
9	architecture	associative	neural
10	autonomous	algorithm	multi

Table I shows the categorized simulation results. As a results, there are 3 categorization by PDLS: agent system, neural network, digital library. 10 keywords in each categorization is classified by SOM network. For example, if the words such as framework, mobile, jave are found in any documents, the documents are classified "Agent System" group in PDLS.

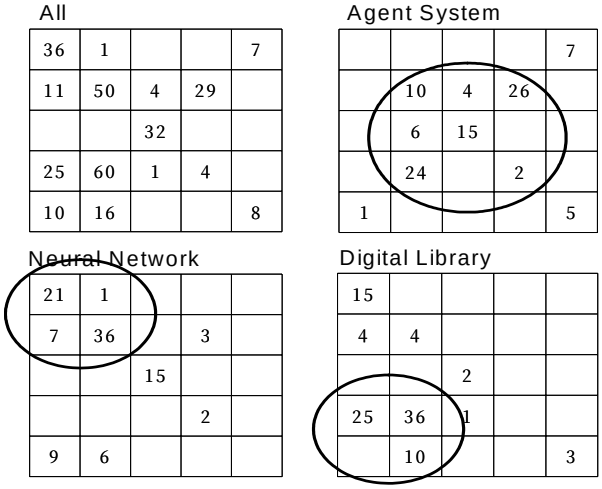
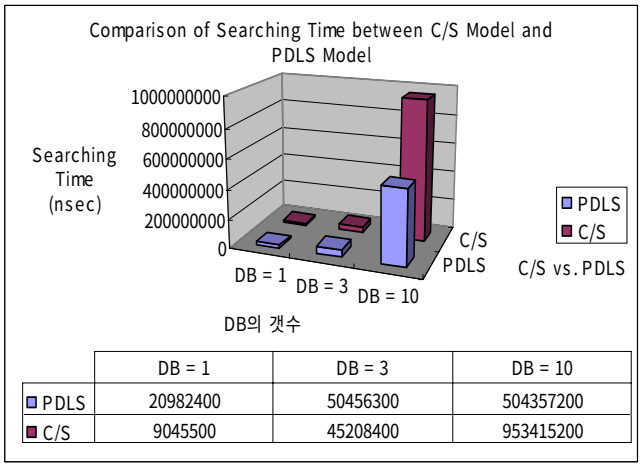


Figure 8. SOM Categorization Results

Figure 8 shows the simulation results according to the random documents in servers. The parameters used in SOM are like the followings. 2 dimensional array for output display and radius 2 for classification are used here. Also, the learning times and learning rate are 500 times and 0.05, respectively. The computer specification is CPU 1.8 GHZ(RAM 1024MB), and window NT based Mysql are used for simulation.

In Figure 8, the numbers in matrix mean the number of the classified documents. If the sum of the numbers in each category, the total will be 100, as we simulated using 100 documents.

The searching times between the proposed PDLS and the traditional client-server model are shown in Figure95. As time passed, PDLS showed a faster search time as well as a much safer search than the client-server model. The result showed that as the numbers of servers were increased, the searching time was decreased in PDLS.



	DB = 1	DB = 3	DB = 10
PDLS	20982400	50456300	504357200
C/S	9045500	45208400	953415200

Figure 9. Simulation Results

V. CONCLUSIONS

In this paper, I proposed a Personal Digital Library System. PDLS is designed based on a new mobile multi agent platform using Voyager and DECAF agent framework. The new platform is a hybrid system of a mobile and a distributed system in order to achieve an optimality in distributed Environments, and to make it operate effectively by the propose of a new negotiation algorithm and a new scheduling algorithm. From the simulation results of PDLS, the performance and the user's satisfaction of this system is higher than any other information search systems as of now. Also, as the numbers of servers and agents are increased, the searched time of PDLS is lowered. And the degree of the user's satisfaction is increased four times than the conventional client-server model. In the future, PDLS needs to be compensated in order to be activated in the real world.

REFERENCES

[1] David C., Colin H., Aaron Kershenbaum, Mobile Agents: Are They a Good Idea?, 1995

[2] John R. Graham, Keith S. Decker, Towards Distributed, Environment Centered Agent Framework. Appearing in *Intelligent Agents IV, Agent Theories, Architectures, and Languages* Springer-Verlag, 2000, Nicholas Jennings, Yves Lesperance, Editors

[3] Keith S. Decker, Victor R. Lessor Generalizing the partial global algorithm. Intelligent Cooperative Information Systems, 1(2), pp.319~346. 1992

[4] Keith S. Decker, Task Environment Centered Simulation, 1996

[5] ObjectSpace, Voyager Core Technology 2.0 User Guide, ObjectSpace, 1998

[6] Sánchez, AGS: Introducing Agents as Services Provided by Digital Libraries, 1997

[7] Jonas Holmstrom, A framework for Personalized Library Services, October 2002

[8] Kohonen, Self Organizing Feature Map, 1995