

# 이동 에이전트를 이용한 교육용 콘텐츠 공유 및 검색 시스템의 설계 및 구현

이철환\* · 한선관\*\*

## 요 약

교육시스템에서 분산된 교육 정보를 검색, 공유하는 새로운 기술로 이동 에이전트가 주목을 받고 있다. 효율적인 검색을 수행하기 위해 많은 양의 콘텐츠를 전송받는 대신 검색을 수행하는 에이전트를 직접 서버에 전달하는 방식을 이용한다. 본 연구에서는 이동 에이전트를 이용하여 교육용 사이트에 분산적으로 존재하는질의응답 게시판과 유즈넷의 자료를 검색, 공유하는 시스템의 모델을 제안한다. 제안된 시스템의 효율성을 살펴보기 위해 기존의 검색 모델과의 비교 실험을 하였으며 교육용 서버로 에이전트가 이동함에 따라 네트워크 트래픽이 감소됨을 볼 수 있었다 또한 이동 에이전트가 검색 위치로 이동 후 게시판과 유즈넷의 자료 변환을 위한 최적 검색 시간이 단축되었다.

## Design and Implementation of Educational Contents Sharing and Retrieval System using Mobile Agent

Chul-Hwan Lee\* · Sun-Gwan Han\*\*

### Abstract

The mobile agent is receiving the attention as new technique to retrieve and to share the distributed contents on web-based educational systems. The retrieval using mobile agent uses the method that delivers the agent to accomplish a search in direct server in substitute for it is transmitted a many contents to accomplish an efficient search. This study proposed the model of retrieval system that shares and searches the distributed educational contents of the bulletin board and newsgroup by using the mobile agent. In order to evaluate the efficiencies of the system that is proposed, we did the comparison test on the search model of existing and proposing system. By a test result, we confirmed network traffics of proposed system are diminished. Moreover we gave proof the fact that the optimum search time of the mobile agent-based system is shortened.

### 1. 서 론

인터넷 상의 웹기반 교육용 사이트에는 교육용 콘텐츠를 포함한 웹 게시판이 수없이 많이 존재한다. 이러한 게시판은 각각의 교육용 서버에서 독립적으로 운영이 되고 있기 때문에 질문과 답변의 중복 문제가 발생하며, 답변 내용 또한 형식이 일정치 않아 비표준화 된 검색 결과를 초래하고 있다. 또한 게시판마다 서로 다른 플랫폼을

이용하는 경우가 대부분이어서 이형의 분산된 게시판의 교육용 콘텐츠를 통합적으로 검색하기에 많은 문제를 지니고 있다.

기존의 연구 방법과 구현된 검색 시스템은 대부분 미리 필터링 된 자료를 이용한다[1][8]. 즉, 각각의 사이트 내에 있는 FAQ의 교육용 콘텐츠를 실시간으로 제공하는 것이 아니라 미리 수집되어 분류되고 정렬된 자료를 이용하는 시스템이다. 따라서 실시간으로 질문과 답변을 공유하는 시스템은 아니라고 할 수 있다. 그러나 동적으로 변화하는 교육용 사이트의 환경을 고려할 때 필터링 되어 검색하는 방식은 효율적인 방법이라 할 수 없다. 이러한 문제를 보완하기 위하여 본 연

\*종신회원: 인천교육대학교 교수

\*\*정회원: 인천교육대학교 시간강사

논문 접수: 2002년 9월 22일, 심사완료: 2002년 10월 25일

구에서는 독립적인 이동 에이전트를 이용하여 실시간으로 서로 분산된 질문과 답변을 공유·검색 가능한 시스템을 제안하였다. 이동 에이전트는 자신을 스스로 이동하는 능력뿐만 아니라 상황에 따른 자기 제어를 할 수 있어 보다 능동적으로 여러 사이트를 이동하며 효율적인 검색을 할 수 있다[2].

본 연구에서 제안한 모델의 효율성을 살펴보기 위하여 기존의 클라이언트-서버 모델의 검색 방법과 이동 검색 에이전트를 이용한 검색 방법을 서로 비교하여 이동 검색 에이전트의 검색 코드 전송에 따른 네트워크 트래픽의 감소 정도를 비교하였다. 또한 뉴스그룹과 웹 게시판에 있는 HTTP와 NNTP형태의 이형의 자료를 데이터베이스로 변환을 위한 정적 고정 에이전트 모델과 최적의 검색경로로 이동한 후에 검색을 하는 이동 에이전트 모델의 검색시간 단축율을 비교하였다.

## 2. 관련 연구

에이전트는 이동성의 유무에 따라 고정 에이전트(Stationary Agent)와 이동 에이전트(Mobile Agent)로 나눌 수 있다. 고정 에이전트는 실행이 되는 시스템에서만 실행이 될 수 있다. 만약 멀티에이전트와 같이 다른 시스템 또는 다른 시스템 내의 에이전트와 통신이 필요한 경우 원격 함수 호출(RPC : Remote Procedure Calling)과 같은 통신 메커니즘을 이용할 수 있다[6]. 반면에 이동 에이전트는 실행을 시작한 시스템에 얽매이지 않고 네트워크를 통해서 자유롭게 이동할 수 있다. 이동 에이전트는 해당 해를 얻기 위해 서버에 질의를 던지거나 자료를 요구하지 않고 에이전트 코드를 직접 이동시킨다. 한번 이동한 에이전트 코드는 해를 구할 때까지 서버와 더 이상 통신이 필요하지 않다. 따라서 이동 에이전트는 네트워크 트래픽의 감소와 이형의 환경 극복, 그리고 비동기적이고 자율적으로 업무를 수행하는 장점을 가진다[1].

컨텐츠 공유 및 검색 시스템에 관한 대표적 선행 연구 사례를 보면 FAQFinder System[3]은 SMART 검색 시스템을 이용하여 데이터베이스에 저장된 FAQ를 자연어로 검색하는 시스템이다. FAQFINDER는 USENET 뉴스그룹의 자료를 여과하여 정보 파일을 구축한다. 사용자의 질문

은 요청 계층(Request Hierarchy)에 의해 질문의 종류에 따라 유형별로 분류가 된다. 사용자의 질문이 들어오면 단어의 상관 관계를 나타낸 WordNet을 이용하여 질문의 의미망을 구성한다. 이를 이용하여 질문의 요청 유형에 맞는 답변을 제시한다[5].

CYLINA(Cyberspace Leveraged Intelligent Agent)는 GTE 연구소에서 개발된 Auto-FAQ시스템이다. 현재 GTE 연구소의 사설망에서 운영 중에 있으며 현재 그 자신과 WWW, 브라우저에 대한 기본적인 질문에 답변을 한다. 데이터베이스의 질문과 답변의 유형은 원본 텍스트, URL, InfoBase내의 다른 자료로 연결되어 있다. CYLINA은 USENET 뉴스그룹이나 사이트의 FAQ를 이용하여 소수의 사용자가 만드는 FAQ가 아닌 다수의 사용자의 노력으로 만들어진 FAQ를 이용하여 자료를 공유한다[4].

## 3. 교육 정보 공유 시스템의 설계

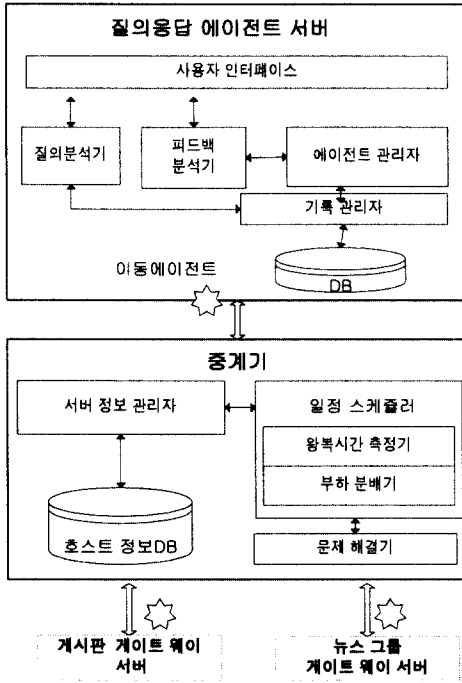
### 3.1 시스템의 구조

본 연구에서 제안한 시스템은 질의응답 에이전트 서버, 중계기, 뉴스그룹 게이트웨이 서버, 웹 게이트웨이 서버, 검색을 수행하기 위해 생성이 되는 이동 검색 에이전트로 구성되어 있다. [그림 1]은 제안된 시스템의 전체 구성도를 나타낸다. 우선 사용자의 질의가 게시판에 게시되면 사용자 인터페이스는 질의 분석기에 전달을 한다. 질의는 키워드 사전과 유사 키워드 사전을 이용하여 키워드를 추출하고 이를 확장하여 분류한다. 분석된 질의는 기록 관리자에 기록이 되어 이동 에이전트 관리자에 전달이 된다. 이동 에이전트 관리자는 이동에이전트를 생성을 시키고 에이전트를 라우팅 시키기 위해서 중계기로 이동을 시킨다.

중계기는 해당 질의의 종류에 따라 최적의 경로로 이동 에이전트들을 관리하고 라우팅 시켜주는 역할을 한다. 일정 스케줄러는 왕복 시간 계산기와 부하 분배기로 구성되어 있다. 서버 정보 관리자는 각 서버의 주소, 이름, 질의 응답의 종류 등을 기록한다.

이동 검색 에이전트는 제안된 시스템의 가장 핵심이 되는 부분으로 원격 서버로 이동 후 지역 데이터베이스를 검색하는 역할을 한다. 검색 수

행시 중계기로 이동하여 중계기의 일정 스케줄러에 의해 지정된 서버로 전송되어 검색 작업을 수행하게 된다. 이동시에 여러 개의 서버로 이동 검색 에이전트를 복제하여 보내는 방법과 하나의 에이전트가 지정된 서버의 목록대로 차례로 방문하는 방법을 이용한다.



[그림 1] 시스템 전체 구성도

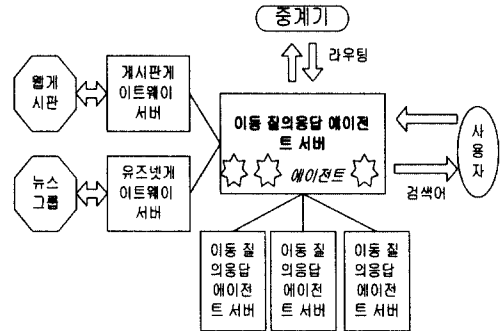
게시판 게이트웨이 서버는 게시판에 저장된 웹 문서를 적재한 뒤 분석하여 원하는 내용만을 추출하여 변환하는 역할을 한다. 웹 게시물에서 다른 게시판의 내용에 연결되는 링크 태그를 검색하여 다른 게시물을 가져오는 자료로 참조를 한다. 파싱 시에 각 웹 게시판의 템플릿을 저장한 정보를 이용하여 각 게시물 내용의 시작과 끝을 나타내는 태그 단위로 파싱을 한 뒤 다시 각 필드의 정보로 다시 파싱을 하는 과정을 거쳐서 내용을 추출하게 된다. 또한 이 서버는 에이전트의 코드이동 기능을 이용하여 게시판의 목록을 이용하여 최적의 검색 위치로 이동하여 검색을 한다.

뉴스그룹 게이트웨이 서버는 NNTP를 이용하여 뉴스그룹 서버에서 기사를 검색하고 입력을 하게 된다. 뉴스그룹 서버들은 서로 연동작용을

통하여 같은 뉴스그룹을 공유하는 경우와 각 서버마다 고유한 뉴스그룹을 가지고 있는 경우가 있다. 이 서버는 이동 뉴스그룹 게이트웨이 에이전트를 생성하여 검색을 수행한다. 고유한 뉴스그룹을 가지고 있는 경우에는 게시판 서버와 같은 검색 이동 위치 변경 정책을 사용하고 같은 뉴스그룹을 공유하는 경우에는 네트워크의 용량과 서버와의 네트워크 거리를 고려하여 선택적으로 이동을 하게 된다.

### 3.2 시스템의 실행 구조

검색을 수행하는 단계는 이동 검색 에이전트를 생성하고 중계기로 검색 서버에 대한 라우팅 경로를 요청하는 단계, 복제 과정을 통해 복제된 에이전트를 검색위치로 이동시키는 단계, 게시판 서버와 뉴스그룹서버가 최적의 검색위치로 이동하는 해서 검색을 수행하는 단계, 검색된 결과를 종합하여 사용자에게 전송하는 단계로 구분할 수 있다.



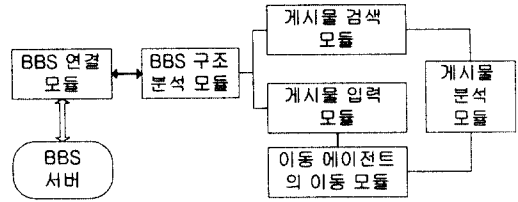
[그림 2] 이동 에이전트를 이용한 공유 검색

첫 번째 단계에서는 사용자의 질의가 입력이 되면 이동 질의응답 에이전트 서버는 질문을 저장하고 해당 질문에 대한 답변을 찾기 위해 이동 에이전트를 생성하고 중계기로 에이전트를 보내 검색 서버에 대한 라우팅 경로를 얻게 된다. 두 번째 단계에서는 이동 질의응답 에이전트 서버가 이동 에이전트를 중계기에서 얻은 라우팅 정책에 따라 복제를 하고, 검색을 위해 각 서버로 전송을 한다. 세 번째 단계에서는 각 서버로 이동한 이동 에이전트가 데이터베이스 내용을 검색한다. 만약 게이트웨이 서버일 경우에는 우선 캐쉬 안의 정보를 검색하여 자료가 있으면 이를 이용을 하고 없으면 에이전트를 생성하여 검색의

최적위치로 이동을 하여 검색을 수행한다. 네 번째 단계에서는 검색을 마친 이동 에이전트가 다시 원래의 위치로 돌아오며 검색된 결과를 종합하여 사용자에게 검색 결과를 보여주게 된다. 시스템의 실행 구조는 [그림 2]와 같다.

### 3.3 게시판 정보 추출

게시판게이트웨이 에이전트가 검색하는 단계는 다음과 같다. 우선 BBS 연결 모듈을 통하여 검색을 원하는 BBS에 접속을 하고 데이터를 수신해 BBS 구조 분석 모듈에 전달한다. BBS의 연결은 일반 웹 로봇의 연결과 다른 추가적인 세션을 유지하고 로그인이 필요한 경우 로그인 스크립트에 의해 로그인을 수행한다. BBS 연결 모듈은 일종의 HTTP 에뮬레이터이며 원활한 연결을 위해 쿠키와 세션을 유지하고 관리한다. 두 번째는 BBS 구조 분석 모듈을 이용하여 게시판의 테이블을 분석을 한다. 게시판의 테이블은 번호, 제목, 작성자, 작성일, 조회 등 필드와 질의와 답변의 두 가지 타입의 레코드로 구분되어진다. 구조 분석 모듈은 필드 분석기와 레코드 분석기를 이용하여 필드와 레코드 단위의 부분 작업을 수행한다. 먼저 필드 분석기를 이용하여 필드별 분석을 수행한 후 해당 필드의 타입을 분석을 하여 필드의 타입을 분석하고 분석된 필드를 기초로 제목 필드를 추출한다. 레코드 분석기는 레코드 분석기를 이용하여 레코드 단위로 추출을 한 뒤 답변 필드의 시작 문자의 차이를 이용하여 구분한다. 세 번째는 게시물 검색 모듈을 이용하여 분석된 구조를 이용하여 연결된 링크를 추출하여 연결된 내용을 검색하고 여과한다. 만약 전체 검색과정을 수행 후에도 답변을 찾지 못했다면 게시물 입력 모듈을 이용하여 질의를 BBS 서버에 입력한다. 마지막 단계는 게시판 게이트웨이 에이전트 이동 모듈을 이용하여 다음 목적지로 이동을 한다. 그림3은 게시판 게이트웨이 에이전트 구조를 나타낸다.



[그림 3] 게시판 게이트웨이 에이전트 구조

뉴스그룹 게이트웨이 에이전트는 뉴스그룹 서버에 있는 기사를 NNTP를 이용하여 추출 및 입력을 하는 에이전트이다. 그 에이전트의 기본 구조는 게시판 게이트웨이 에이전트와 유사하며 역할도 유사하다.

### 3.4 교육 내용 질의 검색

사용자의 질의와 일치하는 답변을 찾기 위해 답변이 있는 질의들 중에 질문의 제목과 내용 부분이 일치하는 질의를 찾는다. 해당 질문과 일치도를 계산하기 위해 벡터 모델을 사용하였다. 이 모델은 사용자의 질의와 검색된 정보를 인덱스 항목의 집합으로 표현한 후 각 인덱스 항목에 가중치를 부여함으로써 이루어진다. 부여된 가중치를 바탕으로 사용자 질의 내용과 검색된 정보의 유사도를 측정하여 그 정보가 사용자 질의와 얼마나 부합되는 지를 판단한다. 또한 이 유사도는 각 정보의 순위를 통한 자료 여과에 사용된다.

사용자 질의  $q$ 의  $i$ 번째 인덱스 항목에 부여된 가중치를  $w_{i,q}$  그리고  $t$ 를 전체 색인어 수라고 하면 질의벡터  $\vec{q}$  는  $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$ 로 정의되며 정보의 벡터  $\vec{d}_j$ 의 벡터는  $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$ 로 표현된다. 사용자의 질의 벡터와 정보 벡터와의 유사도를 구하는 공식은 코사인 계수공식을 사용하였다.

또한 유사도에는 제목과 교육내용에 관한 유사도가 각각 존재하므로 이들을 유사도값의 합을  $sim(d_j, q) = \alpha \times sim_{title}(d'_j, q') + \beta \times sim_{content}(d''_j, q'')$  와 같이 구할 수 있다.

### 3.5 이동 에이전트 스케줄링

이동 에이전트는 사용자가 질의에 대한 시간 제약을 고려하여 이동 경로 정책을 수립한다. 이동 경로 정책은 네트워크 트래픽을 최대한 줄이

는 검색의 경우와 최대한 빠른 시간에 검색의 경우로 구분할 수 있다. 네트워크 트래픽을 최대한 줄이는 방법의 경우는 하나의 이동에이전트를 생성시켜서 지정된 서버로 이동하여 데이터베이스를 검색한다. 반면에 최대한 빠른 시간에 검색을 하는 방법의 경우에는 이동에이전트를 서버의 수만큼 복제하여 각 서버로 전송하여 병렬로 처리하는 방법을 사용한다. 다음은 이동 에이전트가 서버로 이동하는 전송 파라미터를 나타낸다.

$S_i$  서버  
 $t_{hi}$  서버에서 소요된 처리시간  
 $t_{msi}$  서버에서 전송시 걸린 시간  
 $t_{mri}$  서버에서 수신시 걸린 시간  
 $v_{msi}$  서버에서 전송한 데이터의 양  
 $v_{mri}$  서버에서 수신한 데이터의 양  
 $n_0 (=n_{ms0}+n_{mr0})$  서버에서 데이터를 수신한 횟수(송신+수신)  
 $t_{wi}$  서버가 전송 후 응답을 대기 시간  
 $l_i$  서버의 로드

### 3.5.1 기존 C-S시스템 검색 방법

서버-클라이언트 모델을 이용한 검색 시스템의 경우 정보 제공의 위치가 상대적으로 차이가 있다. 이러한 모델은 야후나 엠파스와 같은 일반적인 검색 시스템의 모델에 적용되고 있다. 서버 중심 모델의 경우 출발지 서버  $S_0$ 에서 목표 서버  $S_1$ 으로 데이터를 전송시에 정보 제공 내용 서버에 있으므로 상대적으로 클라이언트 중심 모델에 비해서 목표 서버  $S_1$ 에서 처리되는 시간  $t_{h1}$ 이 많고 클라이언트에서 처리되는 시간  $t_{h0}$ 는 적다.  $S_0$ 에서  $S_1$ 으로 자료 요청을 하고 응답을 받을 때까지 기다리는 시간  $t_{w0}$ 는 순서적으로 요청하는 경우와 동시에 멀티쓰레드를 생성하여 동시에 요청하는 경우가 있다. 만약 순서적으로 요청하는 경우의  $t_{w0}$ 는  $(t_{mr1} + t_{h1} + t_{ms1}) + (t_{mr2} + t_{h2} + t_{ms2} + t_{g2}) + \dots + (t_{mrn} + t_{hn} + t_{msn} + t_{gn})$ 이다.

만약 동시에 요청하는 경우는 서버의 로드  $l_0$ 와 네트워크 트래픽이 급격히 증가한다. 또한 클라이언트는 서버에 요청을 한 후 응답을 기다리는 방식이 동기적이므로 응답을 기다리기까지  $S_0$ 의 자원을 소모하며 기다려야 한다.  $n_0$ 는 목표 서버의 개수  $\times 2 = (n - 1) \times 2$ 이다.

### 3.5.2 이동에이전트를 이용한 검색 방법

기존의 검색 시스템은 검색 요청 후 응답을 기다리는 동기적인 방식을 사용하나 이동 에이전트 검색 방법은 서버로 에이전트가 이동 후에 출발

지와 연결이 필요 없는 비동기적인 방식이다. 이것은 기존 시스템과 비교하여 서버의 로드  $l_0$ 와 네트워크 트래픽을 감소시킬 수 있다. 만약 기존 검색 시스템에서 서버의 최대 한계 로드  $l_{max0}$ 를 고려한 시스템의 작성을 한 경우  $j$ 번째 목표서버로 요청 후에  $l_{max0}$ 에 도달한 경우 응답을 기다리기 위해서는 요청에 대한 응답이 오는 서버가 있어야 한다. 즉, 이동 검색 에이전트를 이용한 방법에서  $t_{gi}$ 는 상대적으로 적다. 그리고 검색을 위해 에이전트 코드를 함께 보내야 하기 때문에  $t_{mri}$ 와  $t_{msi}$ 는 상대적으로 크다. 그러므로  $t_{gi}$ 와  $t_{mri} + t_{msi}$  간에 trade-off가 존재한다. 그러나  $t_{mri} + t_{msi}$ 는  $t_{gi}$ 에 비하여 상대적으로 적다. 이동 검색 에이전트는 검색 코드를 이동시키므로 검색 정책의 변화를 쉽게 적용시킬 수 있다.

### 3.5.3 이동 에이전트의 순차 검색 방법

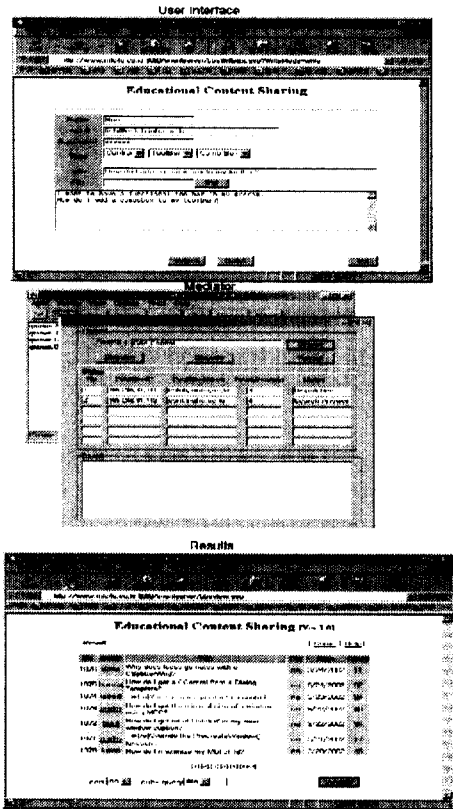
이 방법은 에이전트를 여러개 생성하지 않고 하나의 이동 검색 에이전트가 목표 서버를 차례로 방문하는 방법이다. 기존의 서버 중심 모델이나 클라이언트 중심 모델에 비해서 전체 네트워크 트래픽과 출발지 서버의 로드를 줄일 수 있다. 순서적으로 이동하기 때문에  $t_{w0}$ 는  $(t_{ms0}) + (t_{h1} + t_{ms1}) + \dots + (t_{hn} + t_{msn})$ 이고  $n_0$ 는 2이다. 이 때 각 서버 노드(node)는 각 다른 네트워크 이동거리를 가지고 있다.

## 4. 시스템 구현 및 실험 평가

### 4.1 시스템의 구현

본 연구에서는 일반 PC에 Windows98 10대, Windows2000 server 3대, Linux 3대, SunOS5.6 운영체제의 Unix 1대를 사용하였다. 웹 서버는 MS-IIS와 Apache를 사용하였다. 데이터베이스는 MS-SQL과 MySQL을 사용하였고, DB의 연동을 위해 JDBC를 사용하였다. 이동 에이전트는 IBM에서 제공하는 Java개발환경인 Aglet을 사용하였다. Aglet이란 Agile Applet를 뜻하며 Java로 구현된 이동 에이전트의 의미한다. Aglet에서는 Java-AAPI를 제공하여 Aglet과 동적 환경 사이의 상호작용을 가능하게 하고 수많은 라이브러리를 제공할 수 있게 함으로써 이동 에이전트를 이용하는 네트워크 기반의 어플리케이션 개발을 위한 개발환경을 제공한다[1]. Aglet으로 구현된 이동 에이전트간의 통신은 ATP를 이용한 통신 레

이어에 의해 이루어진다. ATP는 HTTP 프로토콜 위에 구현된 어플리케이션 레벨 인터넷 프로토콜이다. Aglet 서버인 Tahiti는 Aglet의 실행 환경을 제공한다. 모든 이동 에이전트는 이 실행 환경에서만 이동을 할 수 있기 때문에 이동 질의 응답 에이전트 서버, 게시판 게이트웨이 서버, 뉴스 게이트웨이 서버, 중계기 모두 Tahiti 환경 하에서 실행이 된다. [그림 4]는 전체 구현된 시스템의 전체 실행 화면의 관계를 나타낸다.



[그림 4] 이동에이전트 검색 시스템 실행 화면

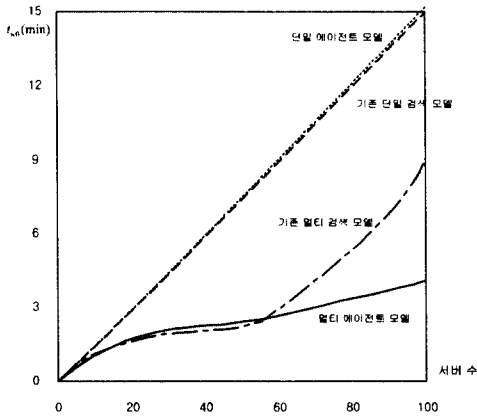
검색 에이전트 모두 같은 것을 사용하였다.

출발지 서버가 전송 후 대기 시간  $t_{w0}$  는 순차적으로 검색을 하는 경우 서버 중심 모델과 에이전트 모두 비슷한 검색 시간을 보였다. 에이전트의 경우는 검색을 위해 에이전트 코드를 전송해야 하기 때문에 상대적으로 시간이 많이 걸렸지만 코드 전송 시간은 서버에서의 검색 시간에 비교하여 상대적으로 작은 시간이기 때문에 전체적인 성능은 비슷하였다. 그러나 서버 중심 모델의 멀티쓰레드를 생성하는 방법과 에이전트의 멀티클로닝 방법의 경우 많은 차이를 보였다. 서버 중심 모델의 멀티쓰레드를 생성하는 방법은 동기적으로 처리되는 방식이기 때문에 서버의 로드가 상대적으로 에이전트에 비하여 많았다. 그래서 서버가 일정한 수 이상의 쓰레드를 생성했을 때 성능이 급격히 감소하는 현상이 발생하였고 에이전트의 멀티 클로닝 방법의 경우 검색 코드 전송 후 연결이 필요 없으므로 상대적으로 성능의 감소가 적었다. 이러한 성능의 급격한 감소는 시스템의 성능과 운영체제에 많은 영향을 받았다.

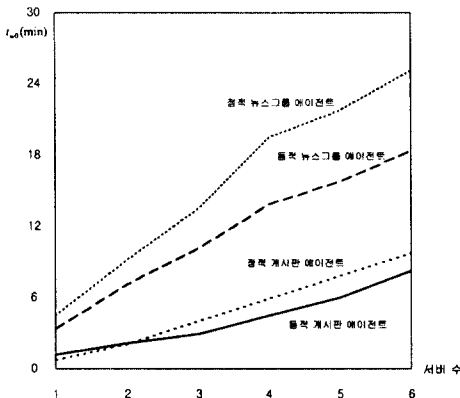
게시판 게이트웨이 에이전트와 뉴스그룹 게이트웨이 에이전트는 이동성 유무에 따른 전송 후 대기 시간  $t_{w0}$  는 많은 차이를 보였다. 뉴스그룹 게이트웨이 에이전트의 경우, 뉴스그룹 기사의 양은 상대적으로 웹 서버의 게시판에 비하여 자료가 적다. 뉴스그룹 게이트웨이 에이전트가 고정적으로 위치하고 있는 경우와 최적의 위치로 이동하는 경우의  $t_{w0}$  는 뉴스 그룹 서버의 수가 적을 때는 고정형이 이동형에 비해 적은  $t_{w0}$  를 보였지만 서버의 수가 증가함에 따라 점진적인 감소를 보였다. 게시판 게이트웨이 에이전트의 경우, 웹 서버의 게시판 자료는 상대적으로 많았다. 그러므로 최적 위치로 이동하는 이동형이 고정형에 비해 훨씬 적은  $t_{w0}$ 를 보였다. 따라서 제안된 시스템의 효율성이 자료가 많을수록 높음을 알 수 있다.

#### 4.2 실험 및 평가

이동 검색 에이전트와 기존 C-S 검색 시스템을 비교하기 위해 각 서버에 Port 번호를 다르게 하여 7개의 Tahiti 서버를 설치하여 총 20대의 서버에 100개의 가상 Tahiti 서버를 가정하였다. 출발지 서버에는 1개의 Tahiti 서버를 설치하였으며, 검색 엔진 모듈은 서버 중심 모델과 이동



[그림 5] 기존 검색 모델과 에이전트 모델의  $tw_0$



[그림 6] 에이전트의 이동 유무에 의한  $tw_0$

## 5. 결론

본 연구는 이질적으로 분산된 교육용 사이트의 콘텐츠 검색과 공유 문제에 대한 해결 방법으로 이동 에이전트를 효율적으로 사용하기 위한 방법을 구체적으로 제시하였다. 기존 검색 시스템의 구조에서 네트워크 트래픽과 검색 시간을 고려하여 검색 정책을 바꿈으로써 네트워크 트래픽의 감소와 서버 과부하의 분산, 그리고 검색 시간의 최적화의 이점을 얻을 수 있었다. 또한 분산된 게시판과 뉴스그룹과 같은 풍부한 교육 시스템의 효율적인 검색을 통하여 콘텐츠의 공유를 최대화할 수 있었다. 이러한 이동 에이전트 모델은 교육 시스템뿐만 아니라 제한된 네트워크 자원이나 서버 자원 등을 효과적으로 관리하고 분산된 자료의 검색에 효과적으로 쓰일 수 있다.

## 참고 문헌

- [1] Danny D. Lange, Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998.
- [2] Edmund H. Durfee and Jeffrey S. Rosenschein., Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples., In Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop, July, 1994.
- [3] Robin D. Burke, Kristian J. Hammond, & Vladimir A. Kulyukin, Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder Systems, Technical Report 97-05, Chicago University, 1997.
- [4] Steven D. Whitehead, Auto-FAQ : an experiment in cyberspace leveraging, Computer Networks and ISDN Systems 28, pp.137-146, 1995.
- [5] Buckley, C., Implementation of the SMART Information Retrieval [sic] System, Technical Report 85-686, Cornell University, 1985.
- [6] Gerhard Weiss, Multiagent Systems, MIT press, London. 2000.
- [7] Sun-Gwan Han, Jae-Bok Park, Jae-Eun Jung, Geun-Sik Jo, "Intelligent Gathering of Contents on Distance Education using Mobile Agent," PROCEEDINGS International Conference on Electronic Commerce' 2000, pp 267-273, Seoul Korea, August. 2000.
- [8] Sun-Gwan Han, Young-GI Kim, 'Cooperative Monitoring System using Mobile Agent', ICCE2000, Taiwan, 2000.
- [9] 한선관, 이철환, '이동에이전트를 이용한 교육 정보 공유 시스템', 한국컴퓨터교육학회 발표논문, 2001

### 이 철 환



1977 인천교육대학교  
(초등교육학사)  
1980 연세대학교 (법학사)  
1988 한국방송대학교 전자계산  
과 (이학사)

1987 연세대학교 (교육학석사)

1993 미 Pittsburgh대 (정보공학석사)

1994 미 Pittsburgh대

(컴퓨터교육/교육공학전공 교육학박사)

1994-현재 인천교육대학교 컴퓨터교육과 교수

관심분야: 초등컴퓨터교육, ICT 교육, 멀티미디어, 웹기반 교육

E-Mail: chlee56@mail.inue.ac.kr

### 한 선 관



1991 인천교육대학교 교육학과  
(교육학학사)

1995 인하대학교  
(전자계산교육학석사)

2002 인하대학교 전자계산공학과  
(공학 박사)

2002 ~ 현재 인천교육대학교 시간강사

관심분야: 인공지능, 지능형 에이전트, ITS  
컴퓨터교육, XML

E-Mail: fsihhan@hanmail.net