인터넷 정보검색 서비스를 위한 다중 에이전트 환경의 설계와 구현

박민식, 최진석, 김영관, 김진혁, 권혁철
부산대학교 전자계산학과

Design and Implementation of Multi-agent Environment for Internet Information Retrieval Service

Min-Sik Park, Jin-Seok Choe, Young-Kwan Kim,
Jin-Hyuk Kim, Hyuk-Chul Kwon
Department of Computer Science, Pusan National University

요 약

최근 인터넷에서 접근할 수 있는 정보의 양이 급속히 증대함에 따라 사용자의 선호도와 목적에 따라 개인화 또는 전문화된 검색기능을 제공하여 기존 인터넷 검색엔진의 약점을 극복할 수 있는 방법으로 인터넷 검색에이전트가 연구되고 있다. 현재 개발하고 있는 인터넷 검색 에이전트는 다양한 목적에서 개발한 인터넷 검색엔진의 검색 결과에 부가가치를 더하는(value-added) 접근 방법을 따른다. 하지만 이 방법은 검색엔진의 재현율이 떨어지면 문제영역 지식이나 사용자의 선호도를 이용해도 정확도가 증가하지 않는다. 따라서 검색엔진의 재현율과 정확도를 동시에 높이면서 효과적인 결과를 얻기 위해서는 기존 검색엔진을 모듈화하여 효율적인 분산 및 다중처리가 가능하도록 함으로써, 문제영역 지식이나 사용자 선호도가 검색결과에 반영되도록 해야한다. 이 논문에서는 부산대학교의 인터넷 검색엔진인 미리내 시스템을 에이전트 기반 시스템으로 변환하고, 분산, 다중 처리 및 실시간 검색이 가능한 검색에이전트 환경을 구축하였다.

1. 서 론

최근 들어 대부분의 문서를 컴퓨터로 작성하며, 통신망을 통해 문서를 배포하고 획득함에 따라 효과적으로 문서를 찾는 기술의 중요성이 매우 커졌다. 더구나 인터넷이 보급됨으로써 전문가 뿐 아니라 일반인도 통신망에 접속하여 정보를 제공하거나 획득하는 것이 일반화되고, 이에 따라 인터넷으로 접근할 수 있는 정보의 양이 기하급수적으로 증가하고 있다. 따라서 역사상 유래 없는 거대한 정보창고이자 정보획득 인프라인 인터넷에서 검색엔진이 가장 성공적인 응용프로그램이 된 것은 당연한 일이다[1]. 하지만 일반인이 기존 인터넷 정보검색엔진으로 문서를 검색하는 데는 근본적인 한계가 있다. 즉, AltaVista, YAHOO, Lycos, 심마니, 정보탐정, Naver 등으로 대표되는 기존 인터넷 검색엔진은 문서의 양이 급격히 증가하고 내용이 자주 바뀌는 인터넷 홈페이지의 현재 상태를 반영할 수 있게 빠른 속도로 색인하여 제공한다는 측면에서 접근했지, 일반 사용자의 선호도에 따라 응용영역(application domain)을 고려하여 검색할 수 있는 기능의 제공이라는 측면은 고려하지 않았다. 이로 인해 기존 인터넷 검색엔진은 많은 문제점을 안고 있다[2절 참고]. 본 논문에서는 이러한 문제점을 해결하기 위해 기존 인터넷 검색엔진과 쉽게 결합할 수 있는 정보검색 에이전트를 설계하고 구현하였다.

논문의 구성은 다음과 같다. 2절에서는 기존 인터넷 검색엔진의 문제점을 지적하고, 정보검색 에이전트의 필요성과 요구사항을 제시하며, 3절에서는 각 에이전트들의 대화를 위한 에이전트간 통신구조와 에이전트 통신언어(ACL)의 설계에 대해 기술한다. 또한 4절에서는 이를 바탕으로 다중에이전트 환경을 구축하며, 마지막으로 5절에서는 결론과 향후과제를 제시한다.

2. 정보검색 에이전트의 필요성과 요구사항

인터넷의 방대함으로 인해 기존 인터넷 검색엔진은 처리속도(speed)와 안정성(robustness)에 초점을 맞추어 개발해 왔다[3]. 이로 인한 기존 인터넷 검색엔진의 문제점은 다음과 같다. ① 기술적 어려움과 처리비용의 문제 때문에 문제영역 정보나 사용자의 선호도를 반영한 정보검색을 하지 못한다. ② 많은 사용자가 동시에 검색을 요청할 때 검색시간과 응답시간이 길어진다. ③ 급증하는 웹 문서를 수용하기 위

해 가장 단순한 방법으로 색인하기 때문에 검색효율이 낮다. ④ 전문 지식이 없는 초보자들은 검색 시스템을 효율적으로 사용하지 못한다. ⑤ 반복적으로 동작하는 문서 수집용 로봇이 Internet traffic을 증가시킨다. ⑥ 프로시듀어한 접근방법으로 인해 분산처리가 어렵다. ⑦ 정보를 전문분야별로 분류하여 저장하지 않으므로 관련이 없는 분야까지 검색하여 검색 속도와 응답 시간을 저해한다. 물론 Yahoo는 분야별로 문서를 분류하여 제공하지만 인간이 분류하므로 전체 인터넷 문서의 일부밖에 분류하지 못한다. ⑧ 검색 결과를 보여주는 인터페이스가 미흡하여 주어진 문서들의 목록에서 실제로 관련된 문서를 찾기가 어렵다.

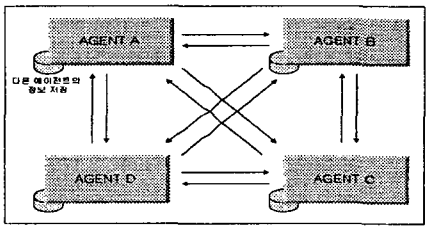
이러한 문제점을 해결하기 위해 기존 인터넷 정보검색엔진에 문제영역이나 사용자의 선호도를 도입한 검색 기법을 사용할 수 있다. 그러나 1993년에 320만 사이트에 불과했던 전 세계의 인터넷 호스트 수가 1998년 1월에는 3,000만개로 약 10배 증가했고, 1999년 1월에는 4,300만 개로 6개월 전보다 1,300만 개가 증가한 자료에서 보듯이[2], 인터넷 자체가 제어할 수 없는 수준에서 확대하고 있기 때문에 일반 검색엔진에 문제영역이나 사용자의 선호도를 도입한 검색 기법을 사용하는 것은 사실상 불가능하다. 최근에는 이로 인한 기존 인터넷 검색엔진의 문제점을 인식하고 사용자의 선호도와 목적에 따라 개인화 또는 전문화한 검색 기능을 제공하여 기존 인터넷 검색엔진의 약점을 극복할 수 있는 방법으로 인터넷 검색에이전트 연구가 이루어지고 있다[4]. 현재 개발하고 있는 검색에이전트는 기존 인터넷 검색엔진이 특정 질의(Query)에 대해 제공하는 홈페이지를 특정 영역 지식(domain specific knowledge)이나 검색 결과에 대한 사용자의 만족도를 바탕으로 학습한 결과를 이용하여 검색효율을 높이는 접근 방법을 따른다. 대표적인 예가 워싱턴 대의 MetaCrawler와 Ahoy!, Shopbot, SavvySearch, SYSKILL & WEBERT, DICA, Grant Learner 등 특정 분야의 검색을 지원하는 인터넷 메타 검색 에이전트다. 이 개념은 현재 data fusion 기술로 구체화하여 개발하고 있다[5]. 하지만 이 방법은 검색엔진의 재현율이 떨어지면 문제영역 지식이나 사용자의 선호도를 이용해도 정확도가 증가하지 않는다. 따라서 기존 검색엔진을 모듈화하여 효과적인 분산 및 다중처리가 가능하도록 함으로써, 문제영역 지식이나 사용자 선호도가 검색 결과에 반영되도록 해야 한다. 이를 위해 정보검색 에이전트가 가져야하는 요구사항은 다음과 같다. ① 검색엔진의 중요기능(예, 문서 탐색 로봇, 색인 시스템, 문서관리 시스템, 검색 시스템, 유사도 계산모듈 등)이 독립된 모듈로 작동할 수 있어야 한다. ② 각 모듈은 네트워크에 연결된 어떤 위치의 시스템에서 수행해도 문제가 없어야 (network transparent)한다. 즉, 각 모듈은 메시지 교환(message passing)과 같은 표준 프로토콜(protocol)에 의해 상호 정보 교환을 하는 agent society 모형에 근간한 방법으로 문제를 풀 수 있어야 한다. ③ 네트워크 환경, 정보의 상태, 사용자의 요구 등에 따라 다수의 같은 모듈(예를 들어 다수 로봇을 보내 문서를 가져오는 것)을 수행해도 전체 시스템의 무결성(integrity)이 유지되어야 한다. ④ 다수의 목적은 같으나 기능이 다른 모듈들이 존재할 수 있어야 한다. 예를 들어 인터넷 홈페이지 추출 로봇과 USENET 문서의 추출 모듈이 동시에 존재할 때, 검색 엔진은 목적에 따라 문서를 두 모듈에서 동시에 받거나, 어느 한 모듈에서 받을 수 있게 선택할 수 있어야 한다. 또한 다수의 색인기가 존재할 때는 목적과 요구사항에 따라 속도가 빠른 색인기나 색인어 추출 효율이 높은 색인기를 선택할 수 있어야 한다. 이와 같이, 정보검색 엔진의 각 모듈을 독립 프로세스로 만들고, 서로 간의 독립(autonomy)성을 극대화하면 각 모듈은 에이전트 역할을 하게 된다. ⑤ 새로운 기술을 도입해도 물리적 또는 프로그램 상으로 다른 모듈에 영향이 없으면서, 기능적으로는 새로 도입한 기술의 효과가 모든 응용시스템에 파급되어야 한다. 예를 들어 PUSH 기술과 정보검색기술을 결합한 클리핑기술을 도입하여도 다른 에이전트 모듈에 영향을 주지 않아야 한다.

본 논문에서는 이와 같이 각 응용시스템이나 구성 모듈을 독립된 에이전트로 만들어 네트워크를 통해 결합함으로써 상호협력으로 문제를 해결할 수 있는 환경을 설계·구현하였다.

3. 에이전트간 통신구조와 통신언어의 설계

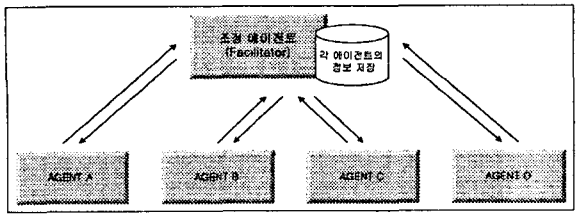
3.1 에이전트간 통신구조

에이전트간 통신구조는 QMW(Queen Mary and Westfield College)대학의 ARCHON[6]과 한국전자통신연구소의 EMAF(Extensible Multi-Agent Framework)[7]의 개념을 통합하여 구현하였다. ARCHON은 모든 에이전트가 다른 에이전트에 대한 정보를 갖고 이를 바탕으로 다른 에이전트에게 자신이 원하는 서비스를 직접 요청하는 방식으로 [그림 1]과 같다.



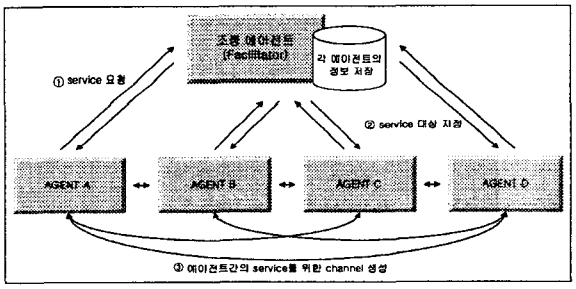
[그림 1] ARCHON 구조

이러한 방식의 단점은 모든 에이전트들이 다른 에이전트 혹은 서비스를 주거나 받아야 하는 에이전트들에 대한 정보를 가지고 있어야 하는 부담과 새로운 에이전트의 추가나 기존 에이전트의 삭제 시 정보들이 모두 변경되어야 한다는 것이다. EMAF는 조정자 역할을 하는 조정 에이전트(Broker, facilitator)를 통해 에이전트들이 서비스를 주고받는 구조이다.



[그림 2] EMAF 구조

이 구조에서는 다른 에이전트에게 서비스를 요청할 때 서비스를 제공할 에이전트에 대한 지정 없이 필요한 서비스만을 요구하고, 조정 에이전트는 요청한 서비스 내용을 분석하여 이를 수행할 에이전트를 선정하고, 해당 에이전트에 요구된 서비스 메시지를 전달한다. 따라서 각 에이전트에 대한 정보는 조정 에이전트만이 가지고 있으면 된다. 에이전트의 추가나 삭제 시 조정에이전트가 가진 데이터베이스만 갱신되면 되므로 ARCHON에 비해 유리하다. 하지만 이러한 구조는 시스템 확장 시 facilitator의 부하가 커져 전체 시스템의 효율을 저해할 우려가 있다. 따라서 ARCHON과 EMAF를 혼합한 [그림 3]과 같은 에이전트간 통신 구조를 설계하였다.

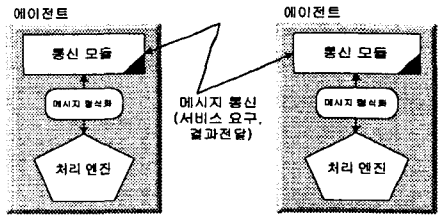


[그림 3] ARCHON과 EMAF의 혼합된 구조

이 구조에서는 facilitator가 하부 에이전트의 정보를 가지며, 서비스 요구 시 정보가 저장된 데이터베이스로부터 서비스가 가능한 에이전트를 찾아 각 에이전트에게 메시지를 보냄으로써 해당 에이전트간에 service를 위한 전용 channel이 만들어진다. 따라서 하부 에이전트들은 각 에이전트들에 대한 정보를 가질 필요가 없으므로, 새로운 에이전트의 추가나 삭제가 용이하다. 또한 facilitator는 해당 서비스를 연결해주는 역할만 할 뿐 서비스 자체에는 관여하지 않으므로, 다른 에이전트를 위한 서비스, 전체 에이전트의 관리, 타 시스템과의 연결 등에 자원을 활용할 수 있으므로 더욱 융통성이 있다.

3.2 에이전트간 통신언어

에이전트끼리 대화한다는 것은 정해진 언어 규약에 따라 메시지를 주고받음을 의미한다. 에이전트는 다른 에이전트에게 서비스를 요청하기 위해 정해진 언어 규약에 따라 요구사항을 미리 메시지 형태로 바꾸고 이를 해당 에이전트에게 전달한다. 다른 에이전트로부터 서비스 요청을 받은 에이전트는 그 메시지를 분석해 내부에서 처리할 수 있는 형태로 변환해서 이를 처리한다. 처리된 결과는 다시 메시지 형태로 바꿔 이를 요청한 에이전트에게 전달한다[그림 4].



[그림 4] 에이전트간 대화 형태

일반적으로 어떤 시스템이든 통신을 위해서는 상호간의 통신규약, 통신언어, 전송규약이 필요하다. 전송규약은 일반적인 물리적 전송 매체를 위한 프로토콜(TCP/IP 등)을 생각하면 된다.

각 에이전트간의 통신을 위한 통신규약과 통신언어에 대한 연구로는 DARPA의 KSE (Knowledge Sharing Effort) 하에서 이뤄지는 ACL(Agent Communication Language)이나 SRI (Stanford Research Institute)의 ICL(Inter-Agent Communication Language) 등이 있다. KQML(Knowledge Query and Manipulation Language)은 에이전트간의 지식 교환을 위한 메시지의 형식이자 상호간의 통신규약이다[8]. 즉 이 말은 KQML이 지식을 표현하는 것이 아니라 표현된 지식을 통신으로 전달하기 위한 형식이라는 것을 말한다. 따라서 KQML은 실제 실어 나르는 지식의 내용과 무관하다. 단지 이들 내용에 대한 태도(질문, 요구, 명령, 단언)를 나타낸다. 이 개념은 사람들간의 의사소통모델인 speech act 이론에서 나온 것으로 KQML에서 태도를 나타내는 기본요소를 performative라고 부른다. performative는 다른 여러 인자와 함께 완전한 KQML 문장을 이루며 또한 상호간의 통신규약을 나타내는 역할을 한다[9].

본 논문에서는 에이전트간 통신규약 및 통신언어를 KQML을 응용하여 구현하였다. 사용한 performative parameter는 :sender(performative의 실제 발신자), :receiver(performative의 실제 수신자), :language(content를 기술하는데 사용한 언어 - 본 설계에서는 미리 정의된 keyword만을 사용함), :ontology(content에 쓰인 어휘를 정의 - 예약된 키워드 집합), :content(메시지 내용) 이다.

또한 에이전트간의 실제 대화를 위해 정의한 performative는 [표 1]과 같다. 이들 performative는 KQML 처리기에 의해 메시지 형태로 encoding되거나 decoding된다. 이러한 KQML 처리기는 facilitator와 일반 에이전트의 통신 모듈에 포함된다. 또한 기존 검색엔진인 미리내 시스템을 에이전트 기반 시스템으로 변환하는 과정에서 서버의 역할이 필요 없는 에이전트를 제외한 다른 에이전트들은 서버와 클라이언트의 역할을 겸하도록 구현하여 이종 기계에서의 분산처리가 가능하다.

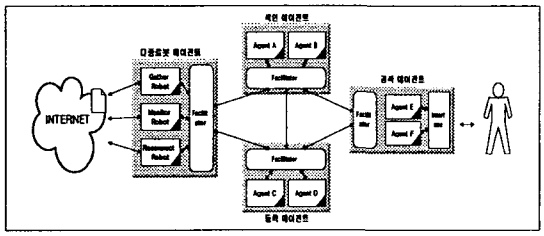
통신 대상	이름	의미
일반 agent → Facilitatot	advertise	agent가 자신의 정보를 알림
	unadvertise	agent의 정보를 삭제 요청
	request	서비스 요청
	accept	facilitator의 서비스 지시 수락
Facilitator → 일반 agent	wait	서비스 요청 agent에게 대기 명령
	kill	특정 agent를 kill
	deny	서비스 불가를 알림
	call	특정 agent에게 서비스 지시
	sleep	특정 agent의 작업중단 지시
일반 agent → 일반 agent	send-data	자료의 전송
	receive-data	자료의 수신
	receive-ok	정상적인 수신 확인

[표 1] Facilitator와 일반 agent를 위한 performative

4. 다중 에이전트 환경 구축

4.1 통합 에이전트 시스템

본 논문에서는 구현된 에이전트간 통신구조와 통신언어를 바탕으로 시스템의 확장을 고려한 통합 에이전트 시스템의 개념을 도입하였다[10]. 이 개념은 위 [그림 3]과 같은 구조가 여러 개 연결된 것으로 보면 된다. 즉 각 작업별로 관련있는 여러 개의 에이전트를 그룹으로 묶고 그룹 내외의 통신을 facilitator가 담당한다. 다음 예는 구축된 통합 에이전트 시스템의 일부이다. 새로운 문서를 가져오는 로봇과 기존 문서의 내용변경 유무를 추적하는 로봇, 이들 에이전트의 중복된 문서처리를 방지하기 위한 데이터베이스 관리 에이전트가 존재한다. 각 로봇들은 단위 작업을 처리하기 위해 데이터베이스를 억세스 해야 하는데, 이를 위한 통신은 facilitator가 담당한다. facilitator는 이 외에도 관련된 에이전트를 관리하는 역할까지 겸한다. 이와 같이 다중로봇 에이전트 시스템은 하나의 그룹이 된다. 또 다른 그룹으로 문서를 색인하여 중요한 정보를 추출하는 색인시스템과 그 정보를 데이터베이스에 저장하는 등록 시스템, 등록된 정보를 검색하는 검색 시스템이 있다. 이들을 통합한 실시간 정보검색 에이전트는 다음과 같은 작업을 처리한다. 다중로봇 에이전트가 가져온 문서는 색인시스템에 의해서 색인되고 그 색인된 결과는 등록시스템에 의해 데이터베이스에 저장되며, 검색 시스템은 사용자의 요구 시 데이터베이스로부터 결과를 얻어 제공한다. 이러한 작업이 실시간으로 이루어지기 위해서는 각 시스템 그룹들 간에 효과적인 통신이 이루어져야 한다. 여기서 각 그룹간의 통신은 역시 facilitator가 담당한다. 이를 그림으로 나타내면 [그림 5]와 같다.

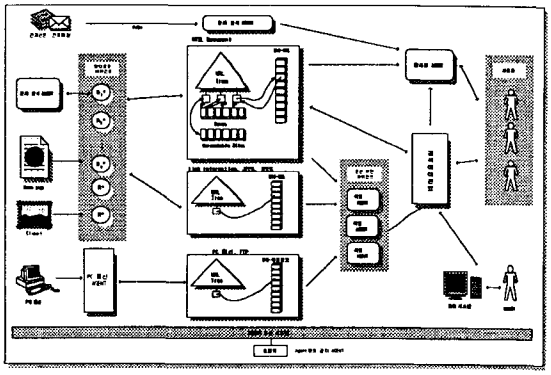


[그림 5] 통합 에이전트 시스템

통합 에이전트 시스템의 개념은 각 역할별로 시스템을 나누어 효과적인 에이전트의 관리는 물론 추가적인 시스템의 확장이나 타 시스템으로의 이식성을 향상시키기 위해서 도입한 것이다. 여기서 각 시스템은 자신에게 주어진 역할만 하며, 필요한 경우 타 시스템에게 서비스를 요청하거나 제공할 수 있다. 따라서 타 시스템과의 통신과 자신이 속한 그룹의 에이전트 관리를 담당하는 facilitator의 역할은 매우 중요하다.

4.2 분산 정보검색 에이전트 시스템

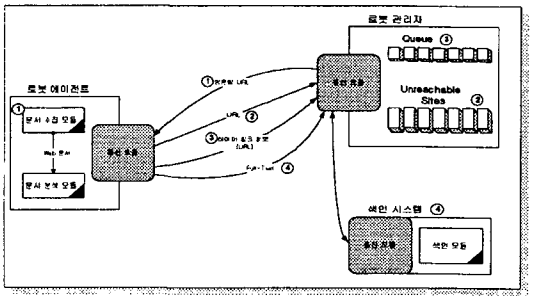
기존 미리내 검색엔진을 정보검색 에이전트로 바꾸면서 분산된 환경에서 동작할 수 있도록 구현하였다. 분산 정보검색 에이전트 시스템의 전체 구조는 다음 [그림 6]과 같다.



<참고> R^n : 등록되지 않은 홈페이지 추적 로봇
R^c : 내용이 변한 홈페이지 추적 로봇
R^a : 접속이 불가능했던 홈페이지 추적 로봇
UID-URL: 검색엔진이 문서에 붙인 이름과 URL의 상호변환테이블

[그림 6] 분산 정보검색 시스템의 구조

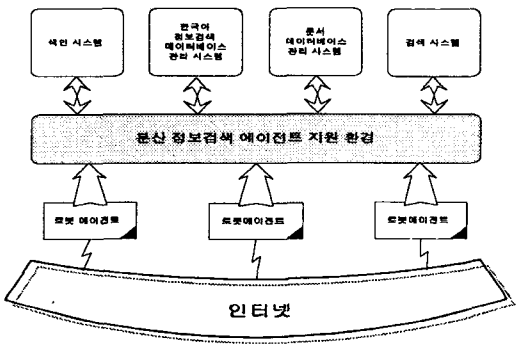
먼저 문서 관리 에이전트는 같은 목적을 가지면서도 다른 기능을 하는 여러 개의 에이전트로 나뉜다. 첫 번째 문서 관리 에이전트는 웹 문서의 내용(제목과 본문)으로 검색하기 위한 에이전트이다. 두 번째 문서 관리 에이전트는 링크 정보를 이용하여 웹 문서를 검색하기 위한 에이전트이다. 세 번째 문서 관리 에이전트는 PC 통신과 FTP 정보검색을 위한 에이전트이다. 이처럼 목적에 따라 문서 관리 에이전트를 만들어 연결하면 되므로 확장성이 뛰어나다.



[그림 7] 새문서를 가져오는 로봇에이전트의 자료 흐름

로봇 에이전트는 새로운 인터넷 홈페이지(등록되지 않은 홈페이지)를 찾는 에이전트[그림 7], 내용이 변한 홈페이지(삭제된 홈페이지도 포함)를 추적하는 에이전트, 접속에 실패한 홈페이지를 추후에 추적하는 에이전트의 3가지로 구성된다. 여기서 새로운 인터넷 홈페이지를 찾는 에이전트는 다수가 될 수 있다. 따라서 다수의 로봇 에이전트가 동시에 상호 협동하여 홈페이지 검색의 효율을 증가시킬 수 있는 환경을 구축하였다. 문서 분석 에이전트는 문서의 특성을 분석하여 문서의 구조와 적용분야에 따라 검색에 필요한 정보를 효과적으로 추출할 수 있도록 구현했다. 웹 문서에서 문서제목, 문서내용, 링크 정보와 기타 문서 검색에 필요한 정보를 추출할 수 있도록 HTML 문서 분석 에이전트를 확장하고, 신문기사와 같은 특수한 목적의 문서에 대한 분석 에이전트도 분리하였다. 색인 에이전트는 분산 환경에서 수행이 가능하며 다른 에이전트에 영향을 주지 않도록 구현하였다. 각 에이전트의 목적과 요구사항에 따라 적당한 색인 서버를 선택할 수 있도록 같은 목적의 여러 다른 기능을 하는 색인 에이전트로 분리하였다, 또한 신문기사의 경우 첫 문장이 주제문일 가능성이 많고, 전자 우편의 경우는 중간에 주제문이 나올 가능성이 많다는 사실을 고려하여 문서의 종류와 특성을 고려한 색인이 가능하도록 구현하였다.

구현된 분산 다중 서버 웹 문서검색시스템의 각 서버 모듈의 구성은 다음 [그림 8]과 같다. 각 서버는 정보검색 시스템의 구성요소를 모두 갖추고 있으며, 다른 서버와의 정보 전송을 위한 분산정보검색에이전트 지원환경을 가지고 있다. 각각의 검색에이전트는 상황에 따라 색인 에이전트나 검색 에이전트만 존재하는 서버도 가능하다.



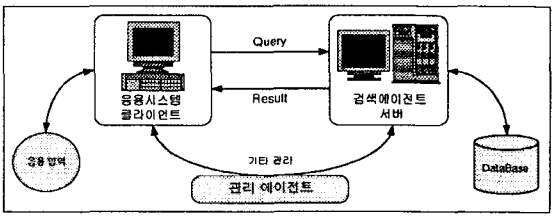
[그림 8] 분산 다중에이전트의 각 서버 모듈 구성

4.3 응용시스템과 검색 에이전트 연동을 위한 인터페이스

Z39.50은 이 기종들간의 상호 운용성을 보장해주며 OSI 참조 모델의 응용계층(Application Layer)에 해당되는 프로토콜이다. Z39.50에 정의된 프로시저와 형식에 의하여, 클라이언트는 서버에게 정보를 요청하고 이 요청에 따라 서버는 클라이언트에게 정보를 제공한다[11].

본 논문에서는 여러 응용시스템과 검색 에이전트 연동을 위한 인터페이스를 Z39.50 프로토콜에 기반하여 구현하였다. 따라서 어떠한 응용시스템이든 정의된 서비스를 이용하여 검색서버와의 통신이 가능하며, 결과를 제공받을 수 있다. 즉 사전에 정의된 통신 API에 맞춰 서비스를 요청하고, 결과를 넘겨준다. 또한 응용시스템은 제공받은 결과를 자체적으로 사용할 수 있는 포맷으로 변환하기 위한 다코딩 모듈을 가진다.

구현된 인터페이스는 다음 [그림 9]와 같이 동작한다.



[그림 9] 인터페이스의 동작 방식

응용 시스템과 검색 에이전트 연동을 위한 인터페이스가 지원하는 기본적인 기능은 다음과 같다.

- 접속 서비스(Connect Service) : 서버와 접속하기 위하여 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라미터를 사용하여 세션에서 지원할 서비스(정보검색, 삭제, 등록 등)를 설정하며, 또한 전송할 메시지 크기 등을 정의한다.
- 질의 서비스(Query Service) : 클라이언트가 서버에 원하는 정보를 질의할 수 있게 하는 기능으로 검색에 필요한 질의 및 다양한 파라미터를 전송하여 검색방식, 응답의 범위와 표현방식, 질의결과처리방식 등을 지정한다.
- 응답처리 서비스(Response Processing Service) : 검색된 결과는 이름이 지정되어 보관되고 보관된 결과를 다음 질의에 의해 재검색할 수 있는 기능이 제공되며, 또한 질의 확장의 기능도 제공된다.
- 결과처리 서비스(Result Processing Service) : 클라이언트의 검색 요구에 대해 데이터베이스를 검색하여 결과를 만들어내며, 결과검색 요구 시에는 이를 클라이언트에게 제공하는 기능을 가진다.
- 결과전송 서비스(Result Transfer Service) : 클라이언트가 요구한 검색결과를 제공한다. 동작환경에 따라 이 결과는 여러 개의 메시지로 나누어 전송할 수 있는 기능도 포함한다.
- 기타 관리 서비스(Other Management Service) : 위의 기본 기능 외에도 서버의 상태, 사용자 접근제어, 문서교환 형식 및 사용 문자의 정의 등 다양한 관리기능을 지원한다.

4.4 미리내 시스템의 실행

다음 [그림 10, 11]은 구현된 미리내 시스템의 사용자 인터페이스이다.

[그림 10] 검색 초기 화면

[그림 11] 검색 결과 화면

5. 결론 및 향후 연구

본 논문에서는 기존 인터넷 정보검색의 문제를 근본적으로 해결하기 위해 정보검색 엔진이나 정보검색 에이전트 기술이 쉽게 결합할 수 있는 구조로 시스템을 설계하고 구현하였다. 이를 위해 부산대학교의 인터넷 검색엔진인 미리내 시스템을 에이전트 기반 시스템으로 변환하고, 분산, 다중 처리 및 실시간 검색이 가능한 검색에이전트 환경을 구축하였다.

향후 연구는 구현된 에이전트 기반 정보검색 시스템을 바탕으로 메타검색 에이전트, PUSH, Clipping 기술을 도입하여 네트웍 환경에 따라 능동적으로 적응하는 한국어 정보검색 에이전트를 개발하는 것이다.

참고문헌

[1] A. E. Howe, D. Dreilinger, "Savvy Search", AI Magazine, Vol 18, No 2, 1997, pp 19~25

[2] Statistics produced by Network Wizards, http://www.nw.com/

[3] O. Etzion : "Moving Up the Information Food Chain : Deploying Softbots on the World Wide Web", AI Magazine, Vol 18, No 2, 1997, pp 11~18

[4] A. Caglayan, C. Harrison, Agent Sourcebook, Wiley Computer Publishing, 1997

[5] R. R. Korfhage, Information Storage and Retrieval, Wiley Computer Publishing, 1997

[6] T. Witting, N.R.Jennings, and E.H.Mamdami, "ARCHON - A Framework for Intelligent Co-operation" IEE-BCS Journal of Intelligent Systems Engineering-Special Issue on Real-time Intelligent Systems in ESPRIT, pp. 168~179, Vol.3, No.3, 1994

[7] 백순철, 최중민, 장명욱, 박상규, 임영환, "이형 분산 환경에서 에이전트들간의 이질성과 분산성을 극복하기 위한 멀티에이전트 기반구조", 정보과학회논문지(C), 제2권, 제1호, pp. 24~37, 1996

[8] R. Patil, R. Files, P. Patel-Schneider, D.McKay, T. Finin, T. Gruber, and R. Neches. The DARPA Know- ledge Sharing Effort: Progress Report. In Principals of Knowledge Representation and Reasonning: Proceed- ings of the Third International Conference, November 1992. Available as http://www.cs.umbc.edu/kqml/papers /KQML-acl.ps

[9] Y. Labrou and T. Finin. A Proposal for a new KQML Specification. TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore Country(UMBC), Feb. 3, 1997, Available as http://www.cs.umbc.edu/kqml/papers/trcs9703.ps

[10] M. R. Genesereth and N. Singh, "A Knowledge Sharing Approach to Software Interoperation", Technical Reports, Logic Group, Computer Science Dep., Stanford University, 1993

[11] The Willow Group, "Using Willow with the Z39.50 Information Retrieval Protocol", Available as http://www.washington.edu/willow/z3950.html