

□ 특집 □

클라이언트/서버 개념을 이용한 상품 정보 수집 에이전트

조 충 호¹ 함 호 상^{**}

◆ 목 차 ◆

- | | |
|-----------------|----------------------------|
| 1 서 론 | 3. Push 개념을 이용한 상품 에이전트 모델 |
| 2 전자거래 디렉토리 서비스 | 4 결 론 |

1. 서 론

최근 국가(정부) 및 기업의 홍보 수단으로써 인터넷의 효율성의 증대와 함께 초고속 정보 통신망과 멀티미디어에 대한 수요의 증가는 80년대 후반부터 국내에 등장하기 시작한 EDI(Electronic Data Interchange)의 개념에 또 다른 변화를 유도하여 EDI의 개념을 점차 EC(Electronic Commerce)로 확대 접목시키는 일대 변혁을 예고하고 있다. 그러나 이러한 인터넷과 같은 네트워크를 기반으로 하는 전자 상거래는 활성화되기 위해 아직도 해결해야 할 많은 문제점을 내포하고 있다. 현재와 같은 초고속 통신망 시대의 분산 환경에서는 다양하고 방대한 상품 정보가 한곳에 집중되어 존재하지 않고 여러 장소에 분산되어 존재하는 양상을 나타내고 있다. 이러한 여건에서 사용자들

각자가 원하는 방대한 상품 정보를 찾아서 그 정보들을 모두 키워드에 의해 분류하고 저장하기란 어려운 일이다. 따라서 이렇게 분산되어 있는 상품 정보들을 모니터링하고 새로운 상품 정보가 발생했을 경우에 이를 EC의 디렉토리 관리기에 알려주고 상품관리 데이터 베이스를 갱신해 주는 상품 정보 검색 및 관리 도구의 필요성이 대두되고 있다. 특히, 인터넷이나 초고속 통신망과 같은 하부 구조를 사용하는 수많은 기업과 거래처 등에서 자신들의 상품 데이터를 외부에 제공하는 서비스를 하는 상황에서는 전자 상거래 서버가 원하는 정보를 손쉽게 찾기는 쉬운 일이 아니다.

전자 상거래 디렉토리 서비스는 인터넷에 분산되어 있는 상품 정보를 수집하고, 이 정보를 가공하여 사용자가 원하는 정보를 손쉽게 찾아 주는 서비스를 제공한다. 상품 정보의 수집은 로봇에 의해서 이루어 질 수 있으나, 디렉토리 서비스의 자연어 처리 능력의 한계와 데이터베이스를 사용하는 쇼핑몰에 대해서는 로봇이 쇼핑몰의 정보를 추출할 수 없는 문제점이 있다. 이러한 정보의

† 정희원 : 고려대학교 전산학과 부교수

** 정희원 : 시스템공학연구소 시스템 통합연구부
전자거래연구실장

분산 문제로 인하여 보다 효율적으로 동작하는 지능형 에이전트의 필요성이 대두되게 되었다.

이렇게 생각하는 객체(objects that think)로서 동작하는 에이전트가 초고속 통신망의 출현과 지식 기반 처리(Knowledge-based processing), Machine Learning, 분산 처리(Distributed processing), 멀티미디어 정보검색 기술 등의 제반 관련 기술의 발달에 따라 실용적인 관점에서 그 구현이 가능할 것으로 보인다[4].

따라서 전자 상거래에서 이용되는 디렉토리 서버를 위한 상품 정보 수집을 위한 지능형 에이전트의 개념을 정립하고, 이 에이전트가 제공할 서비스에 대한 개념 모델과 플랫폼을 개발하는 연구가 필요한 단계라고 하겠다. 즉 수많은 상품 정보 속에서, 원하는 상품을 필터링(filtering)하고 우선 순위를 정하여 찾아 주기 위해서는 검색을 위한 도구가 반드시 필요하다. 본 연구에서는 이러한 분산 환경에서 전자 상거래를 위한 상품 정보 수집과 관리에 적합하다고 생각되는 클라이언트 서버 기반의 지능형 에이전트 협동 구조를 제안 한다.

상품 정보 검색을 위해 기존에 사용하고 있는 방법들은 정보찾는 서버에서 정보 검색 클라이언트에 질의를 보내 결과를 받는 클라이언트 서버 방식으로 이루어지고 있다. 기존의 시스템은 크게 세 가지로 나누어 볼 수 있다. 첫째, 서버중심의 시스템이 있고(Lycos[1], WebCrawler[2]), 두번째로 인덱스 서버 사이의 공조를 통한 시스템(Harvest), 세 번째는 사용자 중심의 에이전트를 이용하는 시스템(WAIBA[5]와 IRA[4], Canot[4])이 있다.

기존의 로봇 서치 방식은 디렉토리 서버에 있는 로봇(스파이더)이 각 쇼핑 물에 돌아다니면서 상품정보가 표시 되어 있는 HTML 화일을 가져와서 그 내용을 파싱하고, 인덱싱한 내용을 사용자에게 보여 주게 되므로 1) 한 줄의 상품정보를

가져오기 위하여 HTML 한 파일을 읽어 들여야 하고, 2) 많은 부분이 그래픽으로 처리 되었기 때문에 정보를 추출하기가 불가능 한 경우가 있고, 3) 화일로 만들어져 있더라도 자연어 처리가 어려워 정확한 상품 정보를 얻기가 힘든 경우가 생긴다 4) 그리고 한번 인덱싱된 상품정보는 원래 쇼핑몰에서 정보가 변경되었거나 삭제되었을 경우에 이를 수정하는 작업이 아주 힘들게 된다. 5) 또한 탐색할 쇼핑몰의 수가 많아지고 상품의 정보가 많아지면 엄청난 트래픽 부하가 유발되어 로봇엔진의 부하에 실제 서비스 사용자의 사용이 제한 받을 수 있다.

따라서 본 연구에서 제안하는 모델은 먼저, PUSH 개념을 이용한 상품 수집 에이전트 모델로서, agent를 각 쇼핑 물에 위치시켜 디렉토리 서버에 있는 collection agent와 상호 협력하여 필요한 정보만 쇼핑 물 에이전트가 디렉토리 서버 에이전트에게 전송할 수 있도록 하였다. 그러므로 불필요한 정보를 네트워크상에서 줄여주고, 변경된 정보의 TRACE가 가능하므로 로봇 에이전트와 협동하여 디렉토리서버의 정보를 관리 할 수 있도록 설계하였다.

2. 전자거래 디렉토리 서비스

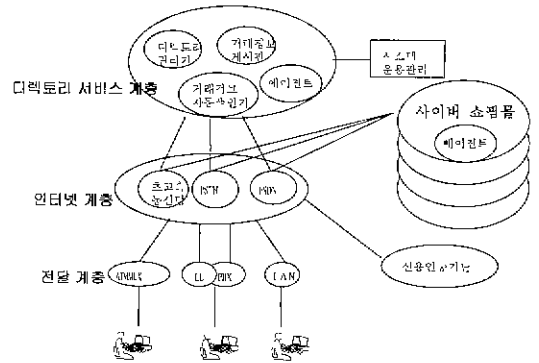
전자 상거래는 초기에는 보안성 때문에 인터넷 보다는 주로 사설 네트워크나 VAN을 통하여 Fax나 E-Forms등을 통한 주문서나 송장을 일괄처리하는 형태로 이루어졌다. 그런데 최근 인터넷 사용의 보편화로 인터넷을 기반으로 하는 개별적인 쇼핑몰들이 국내 (롯데백화점 쇼핑몰: internet.shopping.co.kr, 신세계사이버쇼핑 : www.shinsegae.co.kr, LG 트윈피아 :www.twinpia.com, 현대 신비물: www.shinbimall.com, 데이콤 인터파크: www.interpark.com, 한솔CS클럽: www.hansolcs.co.kr)와

국의 (Worldmall: worldmall.com, Dreamshop: dreamshop.com)에 주류를 이루고 있다. 이들은 인터넷 상에 다양하게 분산되어 있기 때문에 숙달된 사용자들조차도 인터넷에서 이들을 찾기로 그리 쉬운 일이 아니다. 또한 찾은 정보를 다른 쇼핑몰에서 제공하는 정보와 비교하여 구매를 하려고 할 때 이 모든 작업이 사용자의 몫이 된다.

그러나 최근 전자상거래에서 사용자의 이러한 부담을 없애고 관련 상품에 대한 최소한의 정보만가지고도 원하는 상품에 대한 비교 구매, 상품의 정보등을 쉽게 얻을 수 있는 상품 종합정보시스템들이 출현하고 있다. 이들 시스템의 특징은 BarginFinder[6] 처럼 특정한 제품(CD Title)에 대하여 사용자 질의에 대한 가격정보를 각 상점으로로부터 받아 이를 재구성하여 사용자에게 보여주거나, 사용자가 질의를 주면 여러상점들에 질의를 보내 결과를 차례로 출력해주는 형태의 시스템(BarginBot[7])들이 많이 생겨나고 있다.

이러한 시스템들은 다양한 상품에 대한 비교 정보들을 충분히 제공하지 못하고, 사용자의 질의후에 인터넷을 탐색하고 다녀야하므로 처리 속도가 느린 것이 단점이다. 이러한 문제를 해결하는 방안으로 시스템공학 연구소에서는 현재 디렉토리 서비스 시스템을 구현 중이다. 전자 거래 디렉토리 서버는 거래 정보 자동 색인기에 있는 로봇(스파이더)과 에이전트가 분산되어 있는 쇼핑몰들로부터 필요한 상품정보를 가져와서 데이터베이스에 저장하여 보관한다. 서버는 사용자에게 전자거래 디렉토리 서비스에 대한 전반적인 정보를 제공하고 사용자는 원하는 검색방식(주제어 검색, 주제 검색)을 이용하여 원하는 거래 정보를 검색할 수 있는 기능을 제공한다. 디렉토리 서버 시스템은 사용자 검색 기능을 위하여 디렉토리 관리기능과 거래정보 자동 색인기능, 효율적인 상품 검색을 위한 상품 탐색 에이전트의 기

능을 제공할 수 있으며 이들의 처리는 (그림 1)을 전체로 한다.



(그림 1) 전자 상거래 디렉토리 서비스 개념 구조

- 전자 거래 정보 게시판
서비스 이용자에 대한 인터페이스 부분이다. 서비스 이용자의 자연어 질의, 주제별 질의에 대하여 웹문서 및 제품정보 DB를 검색하여 해당 웹문서 또는 제품정보에 대한 URL 정보를 보여 주게 된다. 이들 정보는 디렉토리 관리기, 거래정보 자동색인기에 의해 구축된 정보이다.
- 디렉토리 관리기
전자거래 플랫폼이 가지고 있는 상품정보의 등록,갱신을 관리한다. 상품정보의 등록, 갱신, 삭제, 조회는 공급자에 의해서 이루어진다. 불량공급자 또는 사기업체의 거래정보등록을 방지하기 위하여 판매자 등록정보의 관리도 이루어진다.
- 거래정보 자동색인기
전자거래 플랫폼이 서비스 이용자에게 정보를 제공하기 위하여 정보를 구축하기 위한 기능이다. 이 기능은 인터넷에 존재하는 쇼핑몰 사이트에 접속을 하여 필요한 제품 정보의 데이터를 수집하고 이에 대한 색인기를 만든다.
- 상품 수집 에이전트
웹 로봇이 쇼핑 몰 사이트로부터 정보를 추출하

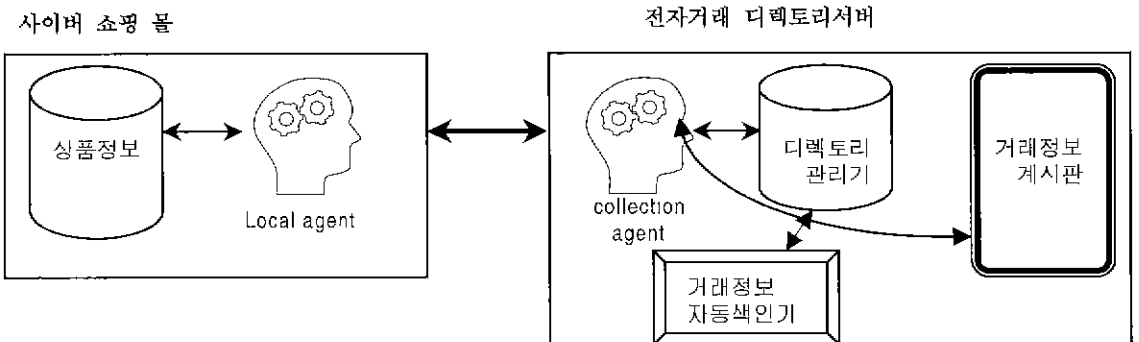
는 것은 네트워크의 부하와 시스템의 로드를 증가시킬 수 있다. 상품 수집 에이전트는 쇼핑몰과 디렉토리 서버에 존재하면서 거래정보 자동색인기가 쇼핑몰 사이트의 정보를 수집하는 일을 도와주는 기능을 하게 된다.

3. Push 개념을 이용한 상품 수집 에이전트 모델

이 장에서는 쇼핑몰 데이터베이스를 주기적으로 검색하면서 변경된 상품정보를 디렉토리서버의 데이터베이스에 저장할 수 있도록 전송해주는 Push 개념을 이용한 상품 수집 에이전트 모델을 제안한다. 이를 위해서 Agent를 각 쇼핑몰에 위치시켜 디렉토리 서버에 있는 collection agent와 상호 협력하여 필요한 정보만 쇼핑몰 에이전트가 디렉토리 서버 에이전트에게 보내고, 보낸 상품정보의 일치성(consistency)을 유지하기 위해서 추가, 수정, 삭제된 상품의 정보를 포함하여 전송한다. 이렇게 함으로써 불필요한 정보를 네트워크 상에서 줄여주고, 변경된 정보의 trace가 가능하게 됨으로서 검색 로봇과 협동하여 디렉토리서버의 정보를 관리 할 수 있도록 하였다.

본 연구에서 구현한 플랫폼에서는 각각의 쇼핑몰이 자체 내에 데이터베이스를 주기적으로 검색하는 에이전트를 포함하고 있고, 마찬가지로 쇼핑몰의 데이터베이스로부터 변경된 상품 정보를 전송받는 디렉토리 서버도 자체내에 에이전트를 포함하고 있다. 그리하여, 각각의 에이전트 간에 정해진 규약에 따라서 서로 통신을 함으로써 정보를 송수신 할 수 있다.

이러한 쇼핑몰의 에이전트와 디렉토리 서버의 에이전트간의 통신에 있어서 먼저, 쇼핑몰의 에이전트는 디렉토리 서버에 접근하기 위한 아이디와 암호를 입력받아서 이를 검증한다. 이는 쇼핑몰의 에이전트가 디렉토리 서버의 에이전트에게 전송해줌으로써 디렉토리 서버쪽에서 쇼핑몰의 에이전트가 합법적인지 불법적인지의 인증이 가능한 구조이다. 따라서, 디렉토리 서버쪽의 에이전트는 아이디와 그에 상응하는 암호를 관리하는 테이블을 유지하고 있어야 한다. 디렉토리 서버는 쇼핑몰의 에이전트로부터 전송받은 아이디와 암호가 적법한지의 여부를 확인하고, 해당 아이디와 암호가 일치하면 쇼핑몰의 에이전트는 주기적으로 검색한 변경된 상품 정보의 결과를 디렉토리 서버의 에이전트로 전송할 수가 있다. 그러나, 해당 아이디와 암호가 일치하지 않으면 쇼핑몰

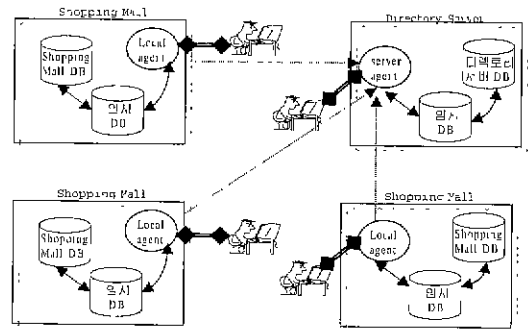


(그림 2) 에이전트간의 관계

의 에이전트는 디렉토리 서버의 에이전트로의 접근 권한이 없으므로 변경된 상품 정보는 물론 어떠한 정보도 전송할 수가 없다. 만약 쇼핑몰의 에이전트로부터 변경된 상품 정보가 정상적으로 디렉토리 서버의 에이전트로 전송되었다면 디렉토리 서버의 에이전트는 쇼핑몰의 에이전트로부터 전송받은 변경된 상품 정보를 자체 내에 가지고 있는 데이터베이스에 저장하고 유지관리한다.

본 모델에서는 기존의 인터넷에서의 정보 전달 체계를 완전히 정반대로 구성하는 개념인 사용자가 정보를 끌어오는 (pull) 형태에서 서비스 이용자가 원하거나 필요로 하는 정보만을 미리 선별하여 알아서 사전에 전송해주는 push 개념을 도입하였다. 그러나 이러한 형태의 모델은 분산된 각 쇼핑몰이 상품 정보 전송을 위한 local agent를 가지고 있어야하므로 디렉토리서버에서 local agent에 대한 소스코드를 다운로드 받아거나 디렉토리서버와 함께 구현중인 에이전트가 내장된 쇼핑몰에서만 가능한 단점이 있다.

본 연구에서 나타나는 에이전트는 쇼핑몰 에이전트와 디렉토리 서버 에이전트로 나눌 수 있으며, 이러한 에이전트간의 협동 모델을 아래의 그림3에 제시하였다. 제시된 바와 같이 하나의 디렉토리 서버에 여러 개의 쇼핑몰이 연결된 형태를 취하고 있으며, 각각의 쇼핑몰과 디렉토리 서버간의 연결은 에이전트 간에 이루어진다. 쇼핑몰과 디렉토리 서버 각각은 자체의 정보들을 저장하기 위한 데이터베이스와 변경된 상품 정보를 임시로 저장하기 위한 임시 데이터베이스를 가지게 된다. 임시 데이터베이스는 전송된 상품정보에 대한 상호 일치성(consistency)을 유지하기 위한 정보를 포함한다.



(그림 3) 에이전트 협동 모델 개념도

3.1 수집된 상품 DB의 Consistency 유지 방법

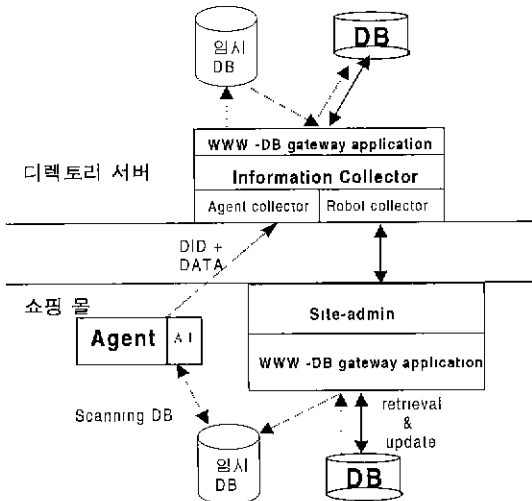
쇼핑몰에 구현되어 있는 상품 DB는 관리자에 의해 지속적으로 상품정보에 대한 추가, 갱신 및 삭제의 프로세스가 일어난다. 이러한 상품 DB를 디렉토리 서비스의 통합 DB로 그 내용을 전달하여 이용하다 보면 시간적인 차이로 인하여 당연히 Consistency문제가 발생한다. 즉, 상품 DB에 새로운 상품을 등록하거나, 가격인하와 같은 상품정보의 변경 그리고 단종 상품의 삭제 등의 정보가 디렉토리 서비스의 통합 DB에 반영되지 못하므로 구매자의 혼란이 예상된다. 이와 같이 쇼핑몰의 상품DB와 디렉토리 서비스의 통합DB의 불일치성에 대한 해결책으로 상품 DB에 프로세스가 동작 할때 마다 통합 DB에 반영시켜주면 되나, 여러 곳의 쇼핑몰로 부터 모든 상품에 대하여 통합 DB에 접근하여 반영하는 것은 디렉토리 서비스 시스템에 큰 부담이 된다.

따라서 쇼핑몰에서 이루어지는 상품 DB에 대한 각종 프로세스들을 주기적으로 추출하여, 추출된 내용을 디렉토리 서비스 DB에 적절히 반영하여, 두개의 DB의 일치성을 유지하여야 한다. 이때, 상품 DB의 변경된 내용을 추출하는 방법은 여러 가지 방법이 있을 수 있으나 본 연구에서는 Tag(삭제, 변경, 추가를 표시)를 포함하는 임시 DB를 만들어 해결하였다. 즉, 기존의 DB에 Tag

필드를 추가한 구조의 임시 DB를 생성하여, 상품 DB에 대한 프로세스를 실행하기 전에 Tag를 포함하여 임시 DB에 기록한 후 상품 DB에 반영하는 방법으로서 상품 DB에 대한 변경이나 삭제 프로세스도 Tag와 함께 임시DB에 기록함으로써 후에 상품 DB에 대한 변경내용에 대하여 추출이 가능하다. 이 방법은 기존에 사용중인 상품DB에 어떠한 부담도 주지 않으면서도 DB Consistency를 확보할 수 있는 방법이 된다.

3.2 에이전트 상품 정보 전송 시나리오

(그림 4)는 에이전트가 WWW-DB gateway 응용 프로그램과는 독립적으로 운용되는 것을 보여주고 있다. 에이전트는 일정한 데이터베이스 탐색 주기를 가지고 데이터베이스의 변동 사항을 체크하여 변동된 데이터를 Agent Interface(A.I.) 에게 전달하게 된다.

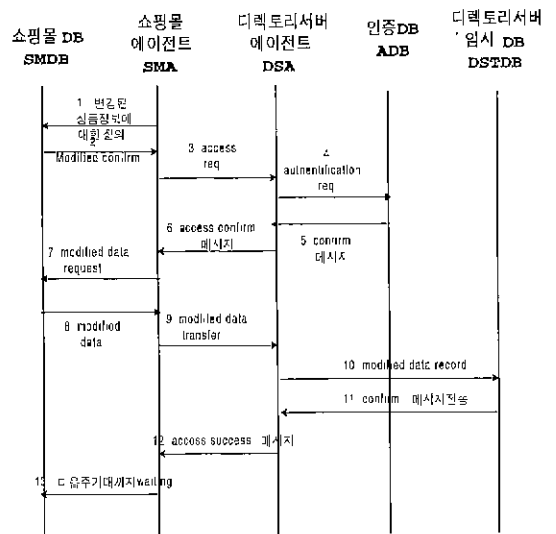


(그림 4) Agent와 웹게이트웨이 응용프로그램과의 관계

위와 같은 디렉토리 서비스를 제공하기 위해서는 쇼핑물에 대한 정보를 수집하는 Information

Collector(IC)는 쇼핑물로부터 정보를 수집해야 하는데, IC가 정보를 수집하는 첫번째 방법은 IC에 존재하는 로봇이 직접 쇼핑물 사이트를 돌아다니면서 정보를 수집하는 방법과 두번째 방법으로는 쇼핑물 사이트에 존재하는 에이전트가 IC에게 정보를 전달하는 방법의 두가지가 있다. 이를 위해서 IC는 robot collector 와 agent interface의 두 가지 요소를 가지고 있어야 한다. 여기서 Robot collector 는 기존의 웹 서치 로봇이 갖는 정보수집 기능을 이용하며, agent interface는 쇼핑물 사이트에 존재하는 agent가 IC에게 정보를 전달하기 위한 인터페이스이다. 또한, 쇼핑물에 존재하는 Agent가 IC에게 데이터를 전달할 때에는 변경된 데이터의 고유한 Document ID를 데이터와 함께 전송해야 한다.

쇼핑물에 존재하는 상품 정보를 디렉토리 서버의 IC에게 전송해주기 위한 상품정보 전송에 대한 시나리오는 정상적으로 에러가 없이 종료되는 경우는 다음 (그림 5)와 같다.



(그림 5) 정상적으로 에러없이 종료되는 경우의 시나리오

- (1) 쇼핑물 에이전트(SMA)가 쇼핑물 데이터베이스 (SMDB) 혹은 쇼핑물 임시 데이터베이스 (SMTDB) 에게 변경된 상품 정보가 있는지를 질의한다.
- (2) SMTDB는 modified confirm 메시지(변경된 상품 정보가 존재한다는 메시지) 를 전송한다.
- (3) SMA는 디렉토리서버 에이전트(DSA) 에게 Access 요구를 한다.
- (4) DSA는 SMA의 식별자를 받아 Authentication DB(ADB) 에게 인증을 요구한다.
- (5) ADB는 DSA에게 confirm 메시지를 보낸다.
- (6) DSA는 SMA에게 access confirm 메시지를 보낸다.
- (7) SMA는 SMDB 혹은 변경된 상품 정보가 추출되어 있는 Shopping Mall Temporary Database (SMTDB) 에게 변경된 상품 정보를 요구한다.
- (8) SMDB 혹은 SMTDB) 변경된 상품 정보를 SMA에게 전달한다.
- (9) SMA는 소켓을 통해 변경된 정보를 DSA에게 전송한다.
- (10) DSA는 Modified Information 을 Directory Server Temporary DB(DSTDB) 에 기록한다.
- (11) DSTDB는 이 정보를 기록한 후 confirm 메시지를 DSA에게 전송한다.
- (12) DSA는 SMA에게 accept success 메시지를 전송한다.
- (13) SMA는 다음 주기 때까지 기다리고 있다가 1번부터 다시 시작한다.

3.3 웹과 상품 데이터베이스 연동

에이전트가 데이터베이스를 검색하기에 앞서서, 디렉토리 서버 에이전트는 전송할 상품에 대하여 몇 가지 정보를 쇼핑물 에이전트에게 보내며, 보내진 정보는 쇼핑물의 데이터베이스에 있는 제품 정보 중 전송할 데이터를 선택하는 수단

이 된다. 따라서 에이전트가 변경된 제품 정보를 디렉토리 서비스에게 전달할 때에는 디렉토리 서비스에서 요구하는 상품 정보만 전달하며, 데이터베이스의 검색 역시 이들 상품 정보만 검색하게 된다. 현재 디렉토리 서비스는 다음의 5가지 정보를 제공한다.

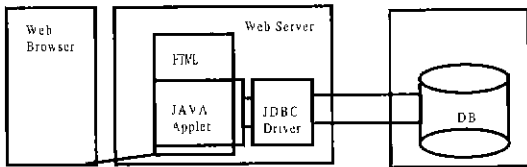
- 상품명
- 그 상품에 대한 주요 사항 및 특징
- 가격,
- 회사
- 모델명
- 상품 정보를 제공하는 쇼핑물 사이트

이들 정보는 에이전트가 최초로 실행될 때 디렉토리 서버 에이전트와 쇼핑물 서버 에이전트 사이에서 협의에 의해서 조정될 수 있다.

쇼핑물의 임시 데이터 베이스에는 디렉토리 서비스가 요구하는 정보 필드 중에서 "상품 정보를 제공하는 쇼핑물 사이트"를 제외한 5가지 정보 필드와 Tag, "제품 내용이 갱신된 시간"의 정보 필드를 포함한다. 그리고 이를 디렉토리 서비스 에이전트에게 전달하게 된다. 상품에 대한 정보들은 다음의 DDL(Data Definition Language)로 정의한다. DDL은 SQL 표준에서 DB 객체, 즉 테이블과 뷰, 스키마, 카탈로그 등에 대한 정의를 담당한다. 데이터 베이스의 접근은 SQL 명령어와 JAVA, 그리고 JDBC driver를 함께 사용하여, 자바에서 바로 데이터베이스를 액세스할 수 있는 형태로 구현하였다. 즉, JDBC 드라이버를 통해서 DB로 직접 접속하여 SQL 질의를 실행하고, 그 결과를 다시 JDBC 드라이버를 통해서 전달받는 이러한 형태로 실행한다.

본 연구에서는 웹 상에서 데이터베이스를 연동하는 방법으로 CGI 이용, 웹서버 API 이용 (MS사의 Internet Information Server가 제공하는 ISAPI와 Netscape의 Enterprise Server가 제공하는

LiveWire Pro : 데이터베이스 벤더에서 자사의 데이터베이스를 전용으로 액세스하기 위한 웹서버와 API 를 제공), Tuxedo, Entera, TopEnd, Encina 등의 미들웨어 벤더에서 제공하는 웹 솔루션으로서, 텍사도의 졸트(Zolt), 엔테라의 맘보(mambo) 등이 있지만 보다 보편적이고 비용 측면에서 유리한 JDBC driver (그림 6)를 사용하였다.



(그림 6) JDBC를 이용한 WWW-DB 연결 모델

자바의 데이터베이스 접속 및 분산 환경을 지원하기 위한 API, 즉 JDBC를 이용하는 방법의 장점으로서는 서버 시스템의 로드를 줄일 수 있고, 데이터베이스와의 커넥션이 지속적으로 유지된다는 점이 있다. 또한 다양한 시스템 구성을 가질 수 있고, 클라이언트 전용 프로그램이 필요없다는 점이 있다. 하지만, 이 방식의 단점으로는 인터넷 구성 후 관리가 힘들다는 문제와 HTTP와 별개의 세션을 갖게 되므로 방화벽이 있을 경우 세션 유지 방안이 고려되어야 한다는 점이 있다. 이러한 JDBC는 자바를 이용하여 데이터베이스로의 접속과 SQL 문장의 실행, 그리고 실행 결과와 얻어진 데이터에 대한 핸들링을 제공하는 방법과 절차에 대한 규약으로 정의 내릴 수 있다. 이러한 JDBC의 간단한 특징은 다음과 같다.

첫째로, JDBC는 통합적인 SQL 데이터베이스를 이용할 수 있는 프레임워크로서 데이터베이스에 비종속적인 자바 애플리케이션을 구현하도록 고안되었다. 즉, JDBC와 연동 가능한 드라이버와 모듈을 번갈아 사용하면서 데이터베이스의 데이터 소스를 액세스할 수 있는데, 이 기술은 DDL

형태의 드라이버와 순수 자바로 만든 드라이버를 모두 지원한다.

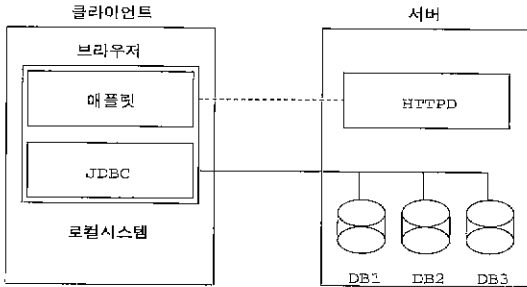
둘째로, JDBC는 SQL CLI(Call Level Interface)를 지원하기 위한 X/Open 정의에 기반을 두고 있다. 이는 기존의 ODBC 보다 상위 레벨의 API를 지원하고, 여러 개의 데이터 소스를 접근할 수 있는 단일 표준화에 바탕을 둔 인터페이스를 제공한다.

그리고, 세번째로 JDBC는 SQL을 사용해 데이터를 액세스할 수 있는 함수 라이브러리를 제공한다. 그리고 JDBC는 SQL 스트링을 DBMS 에게 전달할 수 있는 수단을 제공하고 더불어, 실행될 SQL 스트링에 대한 올바른 표현에 대해 컴파일 단계에서 기본적인 검사가 이뤄진다. 또한, 각각의 데이터베이스에 종속적인 특정 함수에 대해서도 개발자가 이용할 수 있도록 다양한 경로를 제공한다.

이러한 JDBC의 구조는 크게 시스템의 구조에 따라 2-tier 구조와 3-tier 구조의 2가지로 구분되어 있다. 3-tier 방식은 웹서버에 게이트웨이 역할을 담당하는 별도의 서버를 통해 데이터베이스를 액세스하는 구조를 말한다. 이 방식은 먼저 미들웨어를 이용한 경우를 들 수가 있는데, 미들웨어에서는 서버의 역할에 의해 tier가 분리되는 것임에 반해, JDBC는 몇 단계의 서버를 거치느냐에 따라 tier가 결정된다.

본 연구에서 사용한 2-tier 구조는 클라이언트의 웹 브라우저에서 바로 데이터베이스를 액세스하는 방식으로 기존에 개발된 모듈의 사용여부에 따라 가벼운 구조(thin client)와 무거운 구조(heavy client)로 나뉜다. 이 2-tier의 일반적인 구조는 실행시 애플릿이 다운로드되어 서버의 데이터베이스와 직접 연결되는 구조로 보통 하나의 드라이버가 정해진 데이터베이스에 연결하므로 애플릿에서 여러 DBMS를 이용하려 할 경우 각각

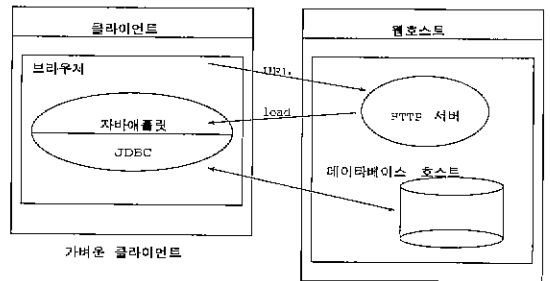
의 DBMS에 해당하는 드라이버를 모두 적재하는 방식으로 다중 데이터 베이스와 연동되는 방식이다. 아래의 (그림 7)은 2-tier를 위한 개념 구조를 나타낸 그림이다.



(그림 7) 데이터베이스 연동을 위한 2-tier 개념 구조

먼저 heavy client는 기존에 개발된 모듈을 기반으로 구성되며, 주로 ODBC나 씨드 파티에서 개발한 전용 프로그램을 이용하는 방식으로서, 클라이언트 시스템에는 미리 필요한 모듈이 인스톨되어 있어야 한다. Thin client 2-tier 구조는 클라이언트 시스템에서 오직 브라우저만을 통해 데이터베이스를 액세스하는 구조이고, 주로 웹에서 순수 JAVA로 개발된 드라이버가 이용된다. 그리고, 이 드라이버는 클라이언트의 JDBC 요청에 대해 특정 데이터베이스 벤더에서 이용하는 네트워크 프로토콜로 바꾸는 작업을 수행한다. 즉, 다시 말해서 기존의 ODBC나 특정 SQL API를 이용하지 않고 바로 네이티브 프로토콜을 사용하기 때문에 빠른 처리 속도를 기대할 수 있다. 이 방식은 일반적으로 하나의 데이터베이스를 이용하는 전용 드라이버와 프로토콜을 이용하는 경우가 많기 때문에 최적화 작업을 거친 후 데이터베이스를 액세스하게 된다. 이점 역시 이 방식의 성능을 향상시키는 중요한 요인이 된다. 또한 이 방식은 클라이언트로 다운로드된 애플릿이 DBMS

에 직접 접속하는 방식으로, 다만 실행 프로그램이 서버에 위치하므로 실행 시 클라이언트로 다운로드되어 수행된다는 점이 다르다. 또한 최적화 구조에 적합한 드라이버와 프로토콜을 이용하므로 대체로 대역폭이 커져서 동시에 다수의 사용자를 지원할 수 있는 장점이 있다. 그러나, 바로 데이터베이스에 접속이 되기 때문에 특수한 예외 상황에 대한 대비책을 마련하기 힘들다는 단점도 있다. 아래의 (그림 8)은 이러한 thin client 2-tier 구조를 나타낸다.



(그림 8) 데이터베이스 연동을 위한 2-tier client 구조

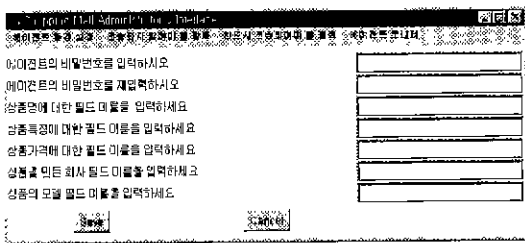
본 연구에서는 2-tier의 가벼운 구조의 성격을 띤 Oracle사에서 제공하는 드라이버를 사용하여 데이터베이스와의 인터페이스를 구현하였다.

3.4 에이전트 인터페이스

쇼핑몰에 존재하는 상품 탐색 보조 에이전트의 구현을 위한 하나의 단계로서, 상품보조 에이전트의 수행주기, 쇼핑몰이 가지고 있는 데이터베이스의 내용중에서 전자거래 자동색인기에 전달되지 말아야 할 데이터 혹은 데이터가 변경된 후 반드시 전달되어야 할 데이터, 디렉토리 서비스가 에이전트 확인을 위한 비밀 번호등의 정보를 쇼핑몰 관리자가 상품보조 에이전트에게 전달할 수 있는 인터페이스가 존재하여야 한다. 관리자가 상품보조 에이전트에게 전달한 정보는 프로

파일(profile)에 저장이 되고, 관리자 인터페이스는 상품보조 에이전트가 처음 시작하기 전에 수행될 수도 있으며, 관리자가 원하는 시기에 실행시킬 수 있다. 또한 관리자 에이전트는 에이전트가 수행하였던 내용을 관리자가 모니터 할 수 있게 한다. 관리자 인터페이스는 최근 동작한 날짜, 디렉토리 서비스에게 전달한 내용, 에이전트 이외의 사람에 의한 쇼핑물 정보 갱신여부의 내용을 모니터할 수 있다.

관리자 인터페이스는 GUI를 제공하여 관리자가 쉽게 상품보조 에이전트에게 정보를 제공할 수 있게 하며 에이전트가 한 일을 모니터할 수 있게 한다. 아래의 그림9는 쇼핑물 관리자가 인터페이스중에서 에이전트의 환경을 설정하는 인터페이스이다. 이 인터페이스에서 쇼핑물 관리자는 디렉토리 서비스가 필요로 하는 상품 정보에 대응하는 데이터베이스의 필드를 지정하며, 또한 디렉토리 서비스와 통신할때 에이전트 인증을 위한 비밀 번호를 설정하게 된다.

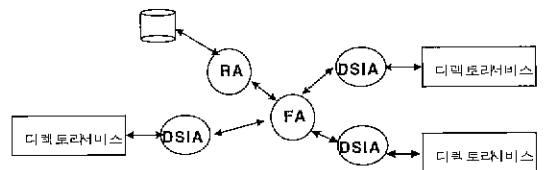


(그림 9) 쇼핑물 관리자 에이전트

3.5 다중 에이전트

현재 구현된 디렉토리 서비스는 여기서 소개하지 않은 다른 검색 기능들도 포함하고 있다. 즉, 단일 상품에 대하여 종류가 아주 많은 경우 (서점,레코드등)에는 BargainFinder처럼 사용자의 질의에 따라서 즉시 인터넷 검색을 수행하여 여러 쇼핑물로부터 얻은 상품정보를 즉시 가공하여

사용자에게 보여주는 기능도 포함하고 있다. 그러나 대부분의 검색 로봇이나 에이전트들이 정보를 미리 가져와서 인덱싱하여 저장하고 있다가, 사용자의 질의에 따라서 전자거래 서비스 로봇이 즉시 자체 데이터베이스를 검색하여 보여주게 되어 있다. 따라서 디렉토리서비스가 사용자 질의에 대한 충분한 정보를 가지고 있지 않는 상황이 생길 수도 있다. 그러나 현재의 전자거래 디렉토리 플랫폼 구조로는 이러한 문제를 해결할 수가 없다. 따라서 이런 경우는 다른 디렉토리 서비스와의 협력을 통하여 사용자의 질의를 처리해야 한다. 이를 위해서 그림 10과 같은 다중 디렉토리 서비스의 구조를 갖게 된다. 이 구조는 RA (Registry Agent), DSIA (Directory Service Interface Agent), FA(Facilitator Agent)를 이용하여 사용자의 질의를 처리하게 된다.



(그림 10) 다중 디렉토리 서비스의 구조

DSIA는 디렉토리 서비스에 대한 인터페이스 에이전트이며, RA는 모든 에이전트에 관한 정보를 관리하며, 각 디렉토리서비스를 구성하는 각각 에이전트의 내용과 능력, 에이전트를 참조할 수 있는 주소를 가지고 있다. FA는 주어진 작업을 수행하기 위해서 다른 에이전트와 협력하여 팀을 이루는데 사용되는 중재자 에이전트로 에이전트간의 중재를 담당한다. 각 에이전트간의 통신은 DARPA의 프로젝트로 진행되어, 현재 표준화 스펙을 제공한 ACL(Agent Communication Language)를 고려하고 있다. ACL은 KQML(Knowledge Query and Manipulation Language)과 KIF(Knowledge

Interchange Format) 등을 이용하여 서로 다른 언어를 쓰더라도 이를 호환시킬 수 있게 하며, 기존의 Query Language와도 호환이 가능하다

4. 결 론

본 논문에서는 분산되어 있는 상품 정보들을 모니터링하고 새로운 상품정보가 발생했을 경우에 이를 전자상거래의 디렉토리 관리기에 알려주고 상품 관리 데이터베이스를 갱신할 수 있는 에이전트 협동 모델을 제시하였다. 디렉토리 서비스의 상품 탐색 보조에이전트 도입은 로봇이 쇼핑물을 돌아다니면서 정보를 획득하는 방법의 문제점인 로봇이 쇼핑물의 데이터를 쉽게 접근할 수 없는 경우와 디렉토리 서비스 시스템의 거래정보 자동 색인 시스템의 자연어 처리 한계문제를 극복할 수 있으며, 실시간으로 쇼핑물에서 변경된 데이터를 디렉토리 서비스가 사용자에게 제공할 수 있다. 또한 상품탐색 보조 에이전트는 디렉토리 서비스 시스템이 쇼핑물로부터 필요로 하는 데이터만을 전송하므로 네트워크의 트래픽을 감소시키며 거래정보 자동색인기의 로봇의 부하를 줄일 수 있을 것이다.

현재까지의 연구는 전자거래 디렉토리서비스의 상품 수집 에이전트의 협동구조를 만들기 위한 기본 구조를 만드는 데에 역점을 두고 이루어져 왔다. 앞으로 본 연구에서는 다음과 같은데에 중점을 두고 연구를 계속해 나가고자 한다.

(1) 현재 에이전트는 테스트 환경에서만 존재하고 있다. 곧 초기의 환경에서는 쇼핑물 관리자나 디렉토리서버 관리자 존재해야하므로, 에이전트가 관리자의 경험을 습득하고 이를 확장하는 과정에는 많은 휴리스틱이 사용될 수 있다. 이러한 휴리스틱은 체계적으로 정리될 필요가 있으며, 적당한 이론으로 체계화되어서

쉽게 이해될 수 있어야 한다. 본 연구에서는 이를 해결하는 방법으로 퍼지/유전자 이론등 인공지능 기술을 접목하여 구현하고자 한다.

(2) 현재의 에이전트 통신 방법은 Socket을 사용한 클라이언트/서버 방식으로 구현되어 있다. 따라서 보다 보편적인 에이전트간 통신을 위해서 ACL(Agent Communication Language)을 사용하여 구현하고자 한다.

참 고 문 헌

- [1] Michael L. Mauldin, John R. R. Leavitt, "Web Agent Related Research at the Center for Machine Translation," <http://fuzine.mt.cs.cmu.edu/mlm/signidr94.html>, Aug, 1994.
- [2] Brian Pinkerton, "Finding What People Want: Experiences with the WebCrawler," Proceedings of the First International World-Wide Web Conference, 1994.
- [3] DeBra, P. and Post, R. , "Information Retrieval in the World-Wide Web: Making Client-based searching feasible," Proceedings of the First International World-Wide Web Conference, 1994.
- [4] Michael Wooldridge and Nicholas R. Jennings, "Intelligent Agents : Theory and Practice ," Knowledge Engineering Reivew, Oct, 1994.
- [5] OSF, "Hypermedia Browsing Technology for Wide Area Information Environments," OSF RI Proposal submitted to BAA 93-42, 1994.
- [6] J.Williams, "Bots and Other Internet Beasities", SamsNet, pp257-263, 1996,"
- [7] B. Anoum, " Agent Technology in Electro Commerce and Information Retrieval on Internet, "Proc. of AusWeb96, 1996
- [8] 한경흠, 한진백, 김종배, 함호상, 조충호, "전

자거래에서 디렉토리 관리를 위한 에이전트 설계”, 한국 정보과학회 학술발표논문집(A), 24권 1호, pp.183-186, 1997

[9] 김병학, 이광형, 조충호, “정보검색을 위한 인텔리전트 웹 에이전트” 정보과학회지, 제14권 제4호, pp12-21, 1996



조 충 호

1981년 고려대학교 공과대학 산업 공학과 (학사)
1983년 고려대학교 대학원 산업 공학과 (석사)
1986년 프랑스 INSA de Lyon 전산학과 (석사)

1989년 프랑스 INSA de Lyon 전산학과 (박사)
1990년-1994년 순천향대학교 전산통계과 조교수
1994년-현재 고려대학교 전산학과 부교수

관심분야 : 지능형 에이전트, 망성능분석, Wireless-ATM



함 호 상

1981년 고려대학교 산업공학과 (학사)
1991년 고려대학교 산업공학과 (석사)
1995년 고려대학교 산업공학과 (박사)
1982년-현재 시스템공학연구소 시스템 통합연구부 전자거래 연구실장

관심분야 : EC/CALS 및 인터넷 보안기술