

추론엔진을 활용한 웹서비스 기반 추천 시스템

김성태

가톨릭 대학교 컴퓨터. 정보공학부
(ariarax@catholic.ac.kr)

박수민

가톨릭 대학교 컴퓨터. 정보공학부
(smtion@catholic.ac.kr)

양정진

가톨릭 대학교 컴퓨터. 정보공학부
(jungjin@catholic.ac.kr)

.....

인터넷의 활용범위는 정보의 검색 및 수집을 넘어서 여러 범위로 확대되고 있고 정보의 양 또한 방대해졌다. 그러나 필요한 정보를 찾기는 더욱 어려워지고 있고, 그에 따라 개인에게 맞는 정보를 제공해주는 시스템이 절실해지고 있다. 본 연구에서는 웹 서비스 기반위에 추론엔진을 사용하여 사용자에게 가장 적합한 상품을 검색하여 추천해주는 추천 시스템의 모델을 제시하고 있다. 현재의 웹 애플리케이션이 사용자에게 필요한 서비스를 제공하는데 비하여 애플리케이션마다 상이한 플랫폼의 구조와 분산된 환경에서 객체간의 통신을 쉽게 하고 통일된 개발을 위해 표준이 필요하게 되었다. 웹 서비스는 프로그램 언어에 독립적이고 상호 운용적 환경을 제공하기 위한 것으로 네트워크를 통해 기술하고 배포하여 실행시킬 수 있는 모듈화된 애플리케이션을 의미한다. 본 논문은 웹 서비스 기반위에 시스템을 구축함으로써 표준 웹 서비스의 실현 가능성을 가늠하고, 추론엔진과 결합하여 사용자의 정보와 변화하는 성향을 토대로 필요한 정보를 예측하여 추천하는 추천시스템 개발에 중점을 둔다.

Key words : 표준 웹 서비스, 추론 엔진, 지식 베이스, 추천 시스템

.....

논문접수일 : 2004년 10월

게재 확정일 : 2004년 12월

교신저자 : 양정진

1. 서론

웹의 급속한 발전으로 인해 인터넷의 활용범위는 정보의 검색 및 수집을 넘어서 쇼핑 및 엔터테인먼트 서비스 등 여러 범위로 확대되고 있고 정보의 양 또한 방대해졌다. 그러나 인터넷의 방대한 정보로 인해 필요한 정보를 찾기는 더욱 어려워지고 있고, 그에 따라 개인에게 맞는 정보를 찾아 제공해주는 시스템이 절실해지고 있다.

본 연구에서는 사용자의 성향을 분석하여 사용자에게 최적의 정보를 제공하는 추천시스템의 모델을 제시하고자 한다. 현재의 웹 애플리케이션이 사용자에게 필요한 서비스를 제공하는데 비하여

애플리케이션마다 상이한 플랫폼의 구조와 분산된 환경에서 객체간의 통신을 쉽게 하고 통일된 개발을 위해 표준이 필요하게 되었다. 웹 서비스는 프로그램 언어에 독립적이고 상호 운용적 환경을 제공하기 위한 것으로 네트워크를 통해 기술하고 배포하여 실행시킬 수 있는 모듈화된 애플리케이션을 의미한다. 사용자의 정보와 변화하는 성향을 토대로 필요한 정보를 예측하여 추천하는 알고리즘에 관한 많은 연구들이 국내외에서 진행되고 있다. 본 논문에서는 현재 표준화를 중심으로 연구되고 있는 웹서비스 기반의 시스템을 구축함으로써 표준 웹서비스의 실현 가능성을 가늠하고, 추론엔진과 결합하여 사용자의 정보와 변

화하는 성향을 토대로 필요한 정보를 예측하여 추천하는 추천시스템 개발에 중점을 둔다.

2. 관련연구

2.1 웹 서비스

웹 서비스는 인터넷을 기반으로 한 오픈 네트워크를 통해 비즈니스 간의 컴퓨터 프로그램을 연합시키기 위한 표준화된 기술을 일컫는다. 웹 서비스는 데이터 표준에 대한 정의를 규정함으로써 데이터 교환의 유연성을 제공하기 위한 것으로 프로그램 언어에 독립적이고 이질적인 시스템이나 프로그램 간의 연결역할을 하여 상호 운용성을 높인다. 클라이언트와 서버는 각기 다른 환경에서 구축될 수 있으며 웹 서비스를 위해 기존 코드를 변경하지 않아도 된다. 또한 웹 서비스는 많은 부분을 오픈 소스 프로젝트를 사용하여 구성되어 있다. 그러므로 벤더에 독립적이고 시스템 구조의 유연성, 사용의 용이성, 통합환경등을 제공하고 자동화된 소프트웨어 통합을 추구하여 상호 운용성을 증대시킨다[1].

2.1.1 웹서비스의 구성요소

웹서비스의 구성요소는 그림 1과 같이 나타낼 수 있다.

■ XML(eXtensible Markup Language)

웹 서비스의 구현은 XML을 기반으로 한다. XML은 메타언어 즉, 태그를 사용하여 문서에 포함된 데이터의 내용을 설명할 수 있으며 이러한 특징을 갖는 XML은 웹을 통해 교환되는 데이터의 표준 언어로써 이질적인 데이터의 상호호환을

보장하기 때문이다. XML은 서로 다른 플랫폼 간에 데이터를 주고받을 수 있게 해주는 언어이며, HTML과 SGML의 필수적인 기능만을 취합하고 복잡하고 어렵거나 비효율적인 부분은 제외함으로써 핵심적인 장점을 그대로 보유하고 있다. 또한 XML의 뛰어난 확장성을 기반으로 데이터베이스를 구축하고 XML 정보를 확인할 수 있는 XML Schema를 응용프로그램 내에 적용함으로써 통합적인 정보를 공유할 수 있다[6][13][14].

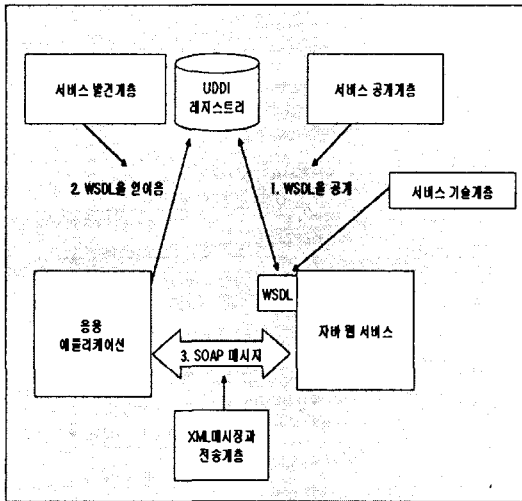
■ UDDI(Universal Discovery Description Interface)

UDDI는 웹 서비스의 디렉토리 서비스를 수행한다. 일종의 온라인 전화번호부와 유사하게 웹 서비스를 찾아주고 홍보해 주는 역할을 수행한다. 그러므로 웹 서비스를 제공하는 서비스 제공자는 웹 서비스 내용과 방법을 UDDI 디렉토리에 등록하고 웹 서비스 요청자들은 외부에서 웹 서비스를 검색하는 데에 UDDI를 이용한다[7][8][9].

■ SOAP(Simple Object Access Protocol)

SOAP는 웹서비스가 인터넷 상에서 어떤 방식으로 통신해야 하는지를 규정하고 있다. SOAP는 분산된 환경에서 정보를 교환하도록 해주는 XML기반의 통신 프로토콜이다. 따라서 다양한 웹 요청자들이 인터넷을 통해 정보를 교환하는 과정에서 필수적인 통신매개의 역할을 담당한다. 인터넷상의 분산 기술인 RPC, CORBA, RMI, DCOM 등은 특정 플랫폼에 종속적인 특성으로 광범위하게 적용되지 못했으나 SOAP은 플랫폼에 독립적인 단순한 XML기반 포맷임으로 웹 서비스에 적합하다.

■ WSDL(Web Service Description Language)



[그림 1] 웹 서비스 구성도

WSDL은 웹 서비스를 정의해 주는 언어이다. 즉, 웹 서비스의 IDL(Interface Definition Language) 버전이며 특정 웹 서비스의 방법과 프로토콜, 데이터 포맷들을 더욱 상세하게 정의하는 일종의 스크립트 언어이다. WSDL은 제공되는 웹 서비스가 어떠한 서비스이며, 어떠한 방식으로 접속하는지를 정의해 주는 역할을 한다. 따라서 개발자가 웹 서비스 프로그램이나 인터페이스 정의 등 웹 서비스를 기술 할 때 WSDL을 이용한다. WSDL은 XML 포맷으로 구성되고 HTTP를 통해서 전달되며 서비스 클라이언트에게 제공되는 서비스가 어떤 인자로 호출되어야 하고 어떤 값을 돌려주는지 알려 주게 된다. 결국 클라이언트는 WSDL로 알게 된 인터페이스 규약에 맞추어 서비스를 호출하고 사용할 수 있게 된다.

2.1.2 웹 서비스 구현

자바 플랫폼에서 웹 서비스를 구현하기 위해 기본적으로 Tomcat, Axis, WSDP가 필요하다.

■ Tomcat

아파치 프로젝트에 의해 진행되고 있는 Tomcat은 서블릿 컨테이너이며, 자바 서블릿 Java Servlet과 자바서버 페이지 JavaServer Pages 기술에 대한 공식적인 참조 구현체이다[2].

■ Axis

Axis는 아파치의 SOAP 엔진이다. SOAP 메시지를 인코딩하고 디코딩한다. 기존에 있던 SOAP과는 속도, 확장성, 안정성 및 여러 면에서 발전되었다.

■ WSDP

자바 플랫폼에서 웹 서비스를 구현하는데 필요한 모든 API와 레퍼런스를 패키지로 구성한 것이 WSDP이다.

2.2 추론엔진

추론엔진은 이미 작성된 지식베이스나 사실들을 룰 기반으로 하여 새로운 사실이나 질의를 추론한다. 추론하는데 도메인의 특성이나 목적에 따라 추론방법이 결정되어진다. 추론방법은 크게 전향추론방법과 후향추론방법으로 분류된다.

2.2.1 전향추론(Forward Chaining)

전향추론은 데이터 중심의 추론방법으로써 기존에 있는 사실과 룰을 토대로 새로운 사실을 추론한다. 추론된 사실은 추론결과로 그치지 않고 지식베이스로 전환되는데, 이는 같은 질의에 대한 해결 시 이미 전환되어진 결과에 대한 재사용성을 제공한다.

전향추론은 구현이 쉬우며 BFS방식의 탐색 추론을 한다. BFS방식을 택하다 보니 추론 과정 중

```

Jess> (deftemplate being (slot name))
Jess> (deftemplate mortal extends being)
Jess> (deftemplate immortal extends being)
Jess> (deftemplate monster extends mortal)
Jess> (deftemplate human extends mortal)
Jess> (deftemplate god extends immortal)
Jess> (defrule list-all-humanoids (being (name ?n)) => (printout t ?n "is a being" crlf))
Jess> (defrule list-all-mortals (mortal (name ?n)) => (printout t ?n "is mortal" crlf))
Jess> (deffacts beings (human (name Bob)) (monster (name Gollum)) (god (name Zeus)))
Jess> (reset)
Jess> (run)
Zeus is a being
Gollum is a being
Gollum is mortal
Bob is mortal
Bob is a being
    
```

[그림 2] JESS 추론 예

```

<Fact-13>
Jess> (defrule flower_boy (name ?x) => (assert (flower_boy ?x)))
TRUE
Jess> (watch facts)
TRUE
Jess> (run)
==> f-14 (flower_boy soomin)
|
Jess> (run)
0
Jess> (assert (name soomin))
FALSE
Jess>
    
```

Clear Window

[그림 3] JESS에서의 재사용성

중복되는 경우가 발생하는데, 중복되는 과정을 필터링하여 추론 속도에 대해서는 어느 정도 보장을 해준다.

전향추론에서는 고정되어진 질의 형태를 가지고 있지 않다. [그림 2]처럼 단순히 룰이 정의되어지고 사실들만을 주어진 후 추론을 시작하면 룰에 적용되어지는 여러 결과들을 산출하고 이 결과들을 지식베이스로 전환하게 된다.

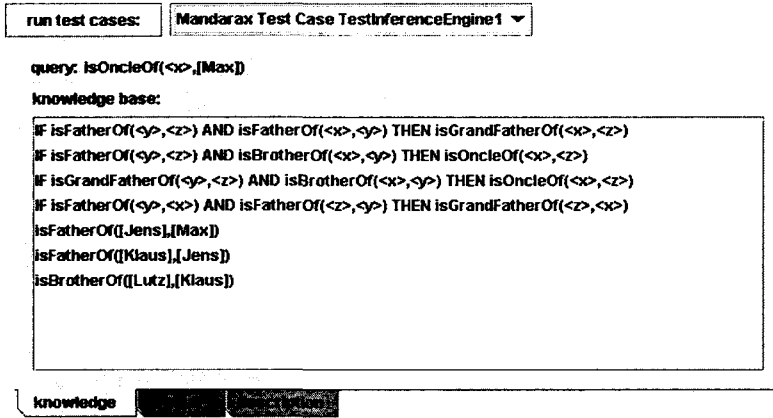
전향추론을 사용하는 대표적인 엔진인 JESS (Java Expert System Shell)로 룰을 정의하고 하나의 사실을 입력으로 주고 나서 추론을 한 결과를 [그림 3]에서 보여주고 있다. 입력된 사실이 새

로운 정보인 경우 추론 후 결과를 지식베이스로 전환한다. 이전에 존재한 사실을 입력하게 되면 결과가 이미 지식베이스로 가지고 있으므로 추론은 실패하게 된다.

지식베이스로 변환된 추론의 결과는 다음 추론시 처리과정을 거치지 않아 빠른 속도로 정보를 보여주게 된다.

2.2.2 후향추론(Backward Chaining)

후향추론은 목표 중심의 추론방법으로써 질의의 입력에 한한 결과 추론을 목적으로 한다. 질의에 해당하는 결론부분만을 관심으로 하며 선행



[그림 4 : Mandarax 추론 예]

조건에 대해선 관심을 가지지 않는다. 단지 룰의 결론만을 찾아 추론하므로 정확한 추론의 결과를 낼 때까지 관계가 있는 없는 모든 가능성들을 탐색하며 추론한다. 위의 [그림 4]는 가족관계를 추론하는 Mandarax 추론[3] 예로써 관계를 추론하는데 사용될 몇 가지 룰들과 사실들을 가지고 있다. Max의 아버지는 Jens이고 Jens의 아버지는 Klaus이며 Lutz는 Klaus의 형제이다. 직관적으로 Max의 할아버지는 Klaus라는 것을 알 수 있지만, 기계가 이러한 관계를 이해하기 위해서는 주어진 관계들을 연결해주는 또 다른 관계에 관한 룰들이 필요하고, 이를 통해서 새로운 사실, 즉 관계를 추론하게 된다.

[그림 4]에서 질의는 isOncleOf(<x>, [Max])로써 Max의 아저씨를 찾는 것이다. 목표는 삼촌을 찾는 것이므로 결론 부분이 삼촌에 관한 모든 룰을 검색하게 된다. 검색은 순차적으로 하게 되고 [그림 4]에서 검색되어 적용되는 룰은 두 개다.

첫 번째 룰에서 첫 번째 조건인 IsFatherOf(<y>, <z>)에서 <z>는 Max로 치환되고 이에 해당하는 사실을 가지고 있으므로 <y>에는 Jens로 치환

된다. 하지만 그 다음 조건인 IsBrotherOf(<x>, <y>)에서 <y>에 Jens가 들어가는 사실을 가지고 있지 않으므로 추론이 실패로 끝나게 되고 두 번째 룰에서는 룰의 조건을 확인할 수 있는 사실들을 모두 가지고 있으므로 질의에 대한 결과가 나오게 된다.

이러한 DFS 방식의 추론은 면적/속도 비용의 최적화를 가지며 전향추론과 같은 지식베이스 확장은 추론 시간의 증가를 가져오므로 질의 결과인 사실을 지식베이스에 추가시키지 않는다.

3. 추천 시스템

3.1 시스템 기능

추천 시스템이란 사용자의 성향을 분석하여 사용자에게 적합한 정보나 상품을 검색하여 추천해주는 시스템이다. 사용자는 이를 통해 도출되어진 결과 중에서 자신이 원하는 것을 선택하게 됨으로써 직접 방대한 정보 중에서 필요한 정보를 찾

아 해매지 않게 된다.

본 논문에서 제시하는 시스템은 표준 웹 서비스를 기반으로 하며 사용자 프로파일을 위한 사용자와 통신을 하고 추천을 위해 추천엔진을 사용한다.

3.2 시스템 구성

시스템 구축을 위해서는 세 가지 요소-클라이언트, 미들웨어, 서버-로 구성되고 표준 웹 서비스 위에서 작동한다. 클라이언트와 미들웨어, 그리고 미들웨어와 서버 사이의 통신은 웹 서비스 표준인 SOAP 프로토콜[19]을 사용한다.

3.2.1 클라이언트

클라이언트는 사용자와 직접 통신하는 부분이다. 추천 시스템이 구동되면 클라이언트는 미들웨어와 통신하여 사용자의 선호도를 분석한 결과를 받아온다. 사용자는 선호도 분석결과 중 추천항목 추천 시 고려해야 할 항목을 변화시킨다. 그 후 검색하고자 하는 검색어를 입력한 후 미들웨어에

요청하게 된다.

3.2.2 미들웨어

미들웨어는 클라이언트와 서버 사이에 위치한다. 미들웨어의 역할은 세 가지로 나누어질 수 있다.

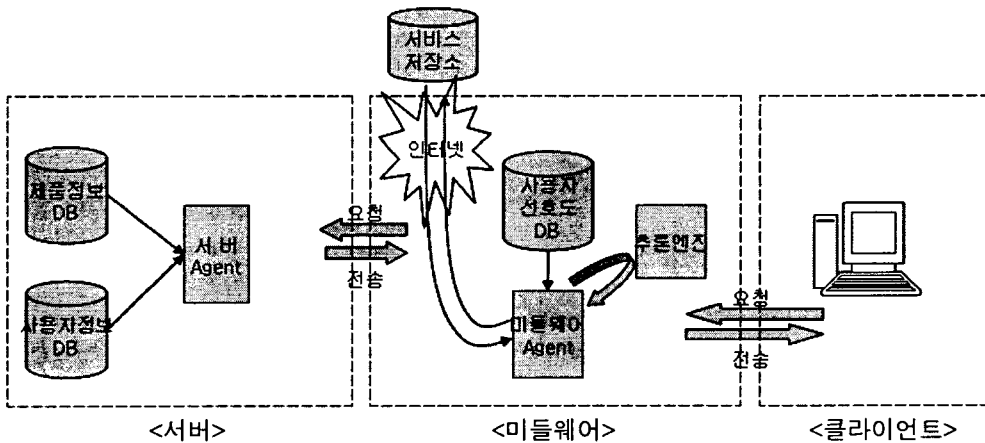
우선 클라이언트로부터 사용자 선호도 정보를 받아들여 이를 누적하여 사용자 성향을 분석하고 UDDI를 통해 사용자에게 필요한 정보를 제공하는 서버를 찾는다.

그리고 서버에 사용자가 보내온 질의어를 보내어 관련된 상품정보를 받아온다. 상품 정보 중에서 사용자 프로파일을 토대로 하여 추천한 후 사용자에게 맞는 상품을 찾아낸다.

마지막으로 찾아진 상품을 다시 서버에 보낸 후 서버로부터 사용자에게 따른 가격정보를 받아온다. 동일한 상품이면 저렴한 상품목록을 찾아 정리하여 클라이언트에게 보낸다.

3.2.3 서버

서버는 데이터베이스와 에이전트로 이루어진



[그림 5] 시스템 구성도

다. 서버는 기본적으로 상품정보 데이터베이스, 고객정보 데이터베이스로 이루어지고 에이전트는 미들웨어에게 검색어를 받는다. 에이전트는 찾고자 하는 상품을 상품정보 데이터베이스에서 검색하고, 고객 정보 데이터베이스에서 고객의 마일리지 및 기타 가격에 관련된 요인을 검색하여 미들웨어에게 전송한다.

3.3 구현

3.3.1 시스템 흐름

1 : 사용자가 추천시스템을 사용하기 위해서는 미들웨어로부터 기존의 사용자의 프로파일 정보를 받아온다.

1.1 : 사용자는 변화가 필요한 항목 부분을 수정하고 찾고자 하는 상품을 검색하기 위한 검색어를 입력한다.

2 : 미들웨어에서는 검색어와 수정된 사용자의 프로파일 정보를 받아 사용자 프로파일 정보를 업데이트시키고 보관한다. 미들웨어는 UDDI를 통해 사용자 요청에 맞는 서비스를 제공하는 서버를 찾는다. 그 후 여러 서버에 각각 검색정보를 보내 결과를 요청한다.

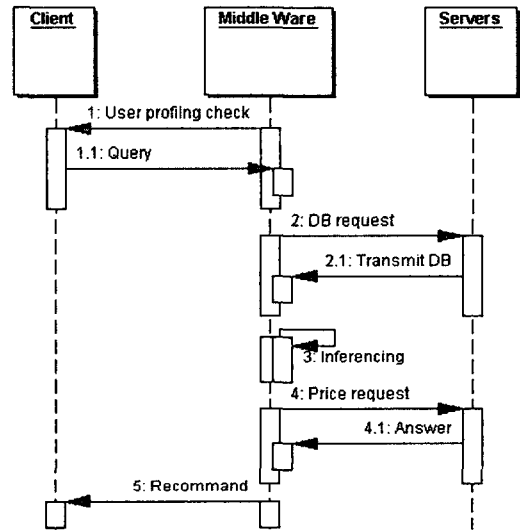
2.1 : 각 서버에 위치한 에이전트는 미들웨어로부터 상품 정보를 받아 상품정보 데이터베이스에서 검색 후 검색결과를 미들웨어에게 보낸다.

3 : 미들웨어는 여러 서버에서 보내온 결과와 사용자 유저파일 분석결과를 토대로 상품을 추천하게 된다.

4 : 미들웨어는 추천된 상품을 각 서버에 보내어 사용자의 마일리지나 행사 등 가격 변동요인에 따른 결과를 요청한다.

4.1 각 서버는 가격 변동요인에 따른 상품가격 미들웨어에게 보낸다.

5 : 미들웨어는 각 서버의 상품가격 중 가장 저렴한 가격을 정리하여 사용자에게 추천하게 된다.



[그림 6] 시스템 흐름도

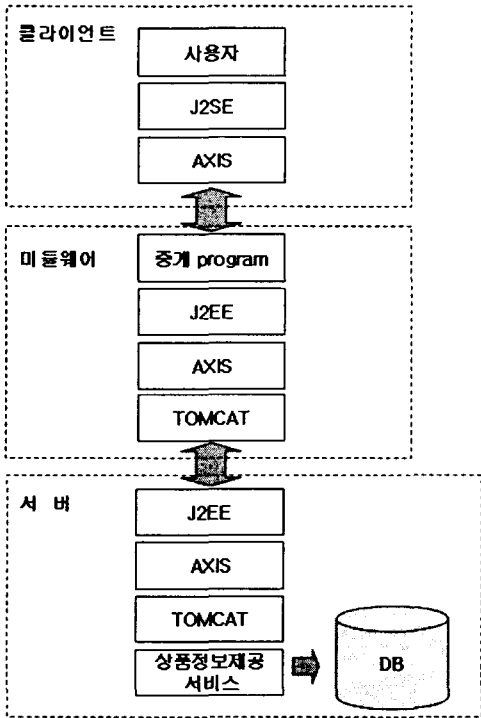
3.3.2 시스템 구현

중고 자동차 판매 사이트를 모델로 하여 서버와 미들웨어 그리고 클라이언트를 구현하였다.

각 구성요소의 구현은 기본적으로 자바 인터페이스를 기반으로 하였다.

웹 서비스를 구현하기 위해서 아파치 프로젝트의 Axis를 사용하였고 이를 설치할 서블릿 컨테이너로 Tomcat이 사용되었다.

이는 서버 프로그램과 미들웨어 프로그램에 동일하게 사용되었다. 또한 서비스를 요청하는 클라이언트 프로그램은 Axis 프레임워크 위에서 동작하게 되고 역시 자바 언어로 구현하였다. [그림 7]은 각 구성요소의 구현도이다[5].



[그림 7] 시스템 구현도

3.3.2.1 서버

웹 서비스를 제공하는 서버를 구축하기 위해 Tomcat, Axis, WSDP, 데이터베이스 및 서비스를 제공하는 프로그램이 필요하다. Tomcat 서블릿 컨테이너에 Axis를 설치하여 SOAP을 사용할 수 있게 된다. 서버 Agent의 역할은 다음과 같다.

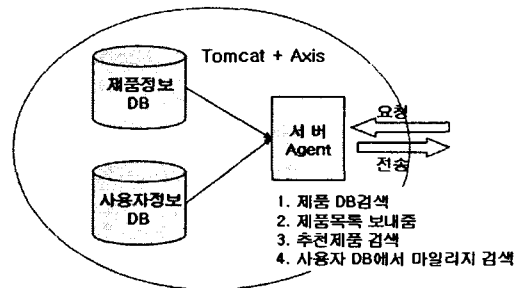
- 사용자 확인 : 일반적으로 사용자 확인을 위해 ID와 패스워드를 통해 사용자를 식별한다.
- 세션식별 : 세션이란 처음 사이트에 진입하여 최종적으로 사이트를 빠져나가는 과정으로 정의된다.
- 요청된 상품 데이터 처리 : 미들웨어로부터 상

품정보와 사용자 정보를 받아 상품 정보 데이터베이스에서 요청된 상품에 대한 정보를 단순 추출하여 제공한다.

- 사용자 정보 처리 : 사용자 데이터베이스에서 사용자에 따른 정보, 즉, 사용자 마일리지 혹은 등급에 따른 할인율을 제공한다.

본 시스템 구현에 있어서 자동차 정보 데이터베이스와 사용자 정보 데이터베이스를 그림 8과 같이 구축하였다. 데이터베이스 레코드로는 제조사, 이름, 연식, 주행거리, 색상, 배기량, 가격, 기어, 지역, 연료가 포함되고 사용자 정보 데이터베이스에는 ID, 이름 및 기타정보 이외에도 마일리지와 고객 등급이 포함된다.

구현된 웹서비스는 UDDI에 등록되어 미들웨어가 UDDI 검색을 통해 서비스를 요청할 수 있게 된다.



[그림 8 : 서버구성]

3.3.2.2 미들웨어

미들웨어는 서버와 클라이언트 사이에 위치하고 다수의 서버와 통신을 하며 클라이언트와 서버 사이의 중개역할을 한다. 미들웨어의 역할은 사용자와 웹 서버 사이를 중개하여 사용자의 선호도를 토대로 사용자에게 가장 적합한 상품을 검색한 후 검색한 웹 사이트 중 고객에게 가장 저

렴한 상품을 구입할 수 있는 사이트를 추천해 주는 것이다.

미들웨어 구현을 위해 사용자 선호도를 저장하는 데이터베이스, 추론엔진 및 추천 Agent로 구성된다.

- 사용자 선호도 분석 : 사용자 선호도 분석을 위해 사용자가 선택하고 실제 구입한 물품의 데이터와 사용자가 추천받기 위한 항목을 변화시킨 데이터를 종합적으로 분석한다.

실제 구현에 있어서는 사용자가 클라이언트를 통해서 자신의 성향을 변화시키도록 하였다. 선호도 항목으로는 제조사, 종류, 연식, 주행거리, 색상, 배기량, 가격, 기어, 지역, 연료로 표 1에서와 같이 이루어져 있다.

<표 1> 사용자 선호도 항목

항 목	선 택 사 항
제 조 사	현대 / 대우 / 기아 / 쌍용 ..
이 름	카렌스 / 코란도...
연 식	2002 / 2003
주행거리	40000km / 50000km
색 상	검정 / 흰색 / 진주...
배 기 량	1500cc / 2000cc..
기 어	자동 / 수동
지 역	서울 / 경기
연 료	가솔린 / 경유 / LPG

미들웨어 구성[그림 9]은 서버와 마찬가지로 표준 웹 서비스 위에 설치되며 웹서비스 구현을 위해서 Tomcat 서블릿 컨테이너 위에 Axis를 설치한다. 미들웨어 에이전트는 사용자 선호도를 분석하고 선호도 정보와 추론엔진을 이용하여 사용자에 적합한 상품을 선택한다. 또한 UDDI를 검색하여 서비스를 제공하는 웹서버를 찾는다. 또한

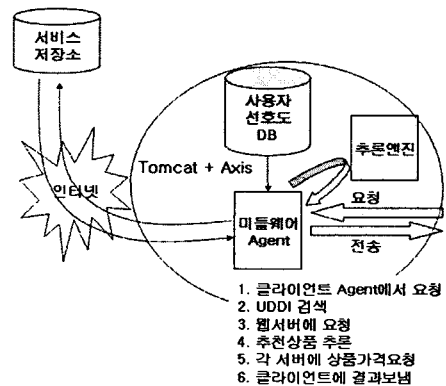
각 서버로부터 받은 상품의 가격을 비교하여 결과를 클라이언트에 전송하게 된다.

3.3.2.3 클라이언트

클라이언트는 사용자와 직접 통신하는 부분으로 사용자 자신이 선호도 정보를 선택적으로 변화시킬 수 있고 찾고자 하는 질의어를 미들웨어에게 보내게 된다. 또한 사용자에게 최종 결과를 출력해주는 역할을 한다.

3.4 추론엔진 & 지식베이스

도메인의 특성과 추론엔진으로 어떠한 입력과 출력이 이루어 지는가에 따라서 쓰이는 추론방법이 달라진다. JESS에서는 추론결과가 지식베이스의 사실로 전환되어 다음 추론 시 추론이 아닌 사실을 보이게 된다. 사용자의 정보는 계속적으로 변화한다. 예로 마일리지 사용으로 인한 마일리지 감소, 마일리지 증가로 인한 회원 등급 상승은 다음 추론 시 다른 결과를 가지게 된다. 후향추론은 추론 결과를 영구적인 데이터로 가지지 않으므로 유동적인 데이터로 부터의 추론하는데 적합하다.



[그림 9] 미들웨어 구성

추천 시스템에 사용될 추론엔진은 Mandarax[3]를 사용하며 Mandarax 엔진이 추가된 서버측은 사용자의 요청이 있을 경우 질의 추론만을 해결한다.

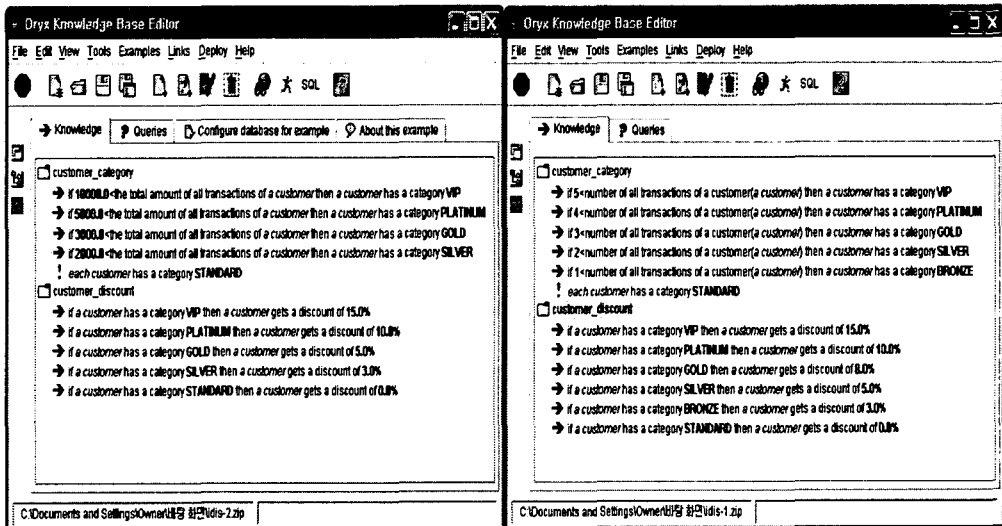
[그림 11]은 추천 시스템 추론엔진에서 사용될 지식베이스를 보여주고 있다. 서버마다 고객을 관리하는 카테고리 구별이 각기 다르게 구성된다. [그림 10]은 하나의 카테고리의 XML정보를 보이고 있다.

[그림 11]의 첫 번째 그림의 고객 카테고리는 'customer_category'에서처럼 VIP, PLATINUM, GOLD, SILVER, STANDARD 5등급으로 나뉜다. 이 서버에서 사용할 카테고리는 사용자의 총 소비액으로 구분한다. 두 번째 그림은 VIP, PLATINUM, GOLD, SILVER, BRONZE, STANDARD 6등급으로 나뉜다. 이 서버에서는 카테고리를 총 트랜잭션의 회수로 구분하게 된다. 각각의 'customer_category'는 고객의 트랜잭션 총 소비액과 소비 회수를 고려해 등급이 결정된다. 고객의 마일리지나

고객 카테고리 정보에 관련하여 할인을 결정하게 되어질 룰은 'customer_discount'에서 보여주고 있다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<knowledgebase class="org.mandarax.reference.AdvancedKnowledgeBase">
  <rule>
    <if connective="and">
      <prereq nat="false">
        <predicate>object:java:predicatzid=f6fc7a:ff43444d1d-7eba</predicate>
        <terms>
          <var name="a customer" type="java.lang.String" />
          <ind type="java.lang.String">string:VIP</ind>
        </terms>
      </prereq>
    </if>
    <then>
      <atom>
        <predicate>object:java:predicatzid=f6fc7a:ff43444d1d-7eb0</predicate>
        <terms>
          <var name="a customer" type="java.lang.String" />
          <ind type="com.jodietrich.oryx.examples.cm.Discount">
            object:java:com.jodietrich.oryx.examples.cm.Discount/id=f6fc7a:ff43444d1d-7eb8</ind>
        </terms>
      </atom>
    </then>
  </properties>
  <property key="@@CREATION_TIME" value="2004. 9. 28. 오후 1:8:21" />
  <property key="@@CREATED_BY" value="Owner" />
</properties>
</rule>
```

[그림 10] 카테고리 XML 문서



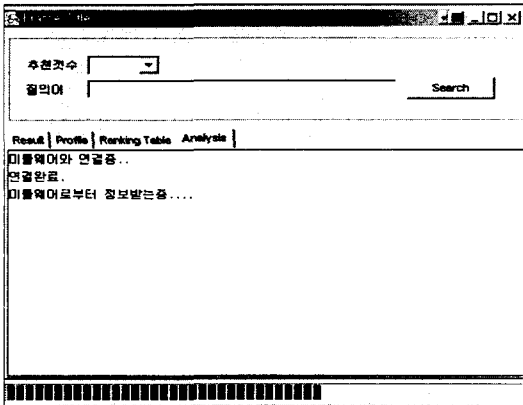
[그림 11] 지식베이스

3.5 결과

3.5.1 클라이언트 - 사용자 질의

사용자는 검색을 위해, 그리고 사용자 프로파일 정보를 업데이트하기 위해 클라이언트 에이전트를 사용한다.

클라이언트 에이전트가 시작되면 먼저 미들웨어에게 사용자 선호도 분석결과를 요청한다. 사용자는 질의어란에 질의어를 입력하고 미들웨어에게서 받은 선호도 변경사항을 확인한 후 미들웨어에게 보낸다. [그림 12]은 클라이언트 에이전트의 인터페이스이다.



[그림 12] 클라이언트 유저 인터페이스

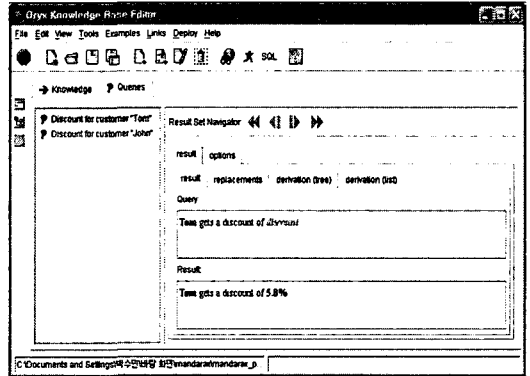
3.5.2 미들웨어 - 추론엔진 - 추천 상품 추론

클라이언트로부터 상품에 대한 정확한 정보나 부분적인 검색어가 미들웨어로 보내지면 미들웨어는 서버 측으로부터 상품 정보를 받아와 유저 프로파일링을 토대로 추천할 상품을 추론한다.

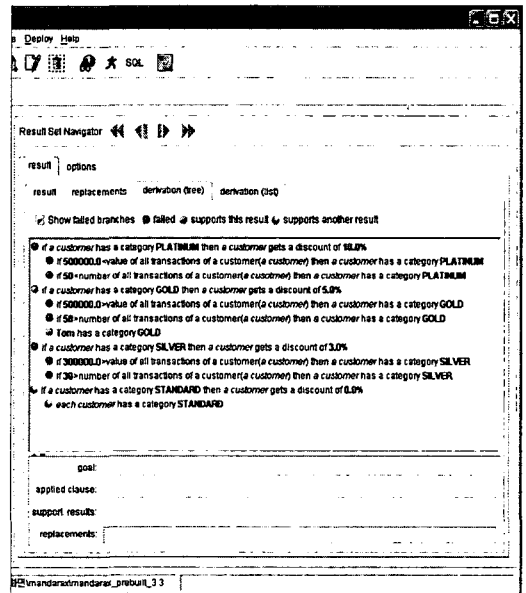
3.5.3 서버 - 추론엔진 - 가격 추론

미들웨어로부터 선정된 상품이 서버 측으로 알

려지면 각 서버들은 자신들의 고객 정보를 토대로 상품 가격을 추론해낸다. 추론을 위한 가상의 사용자 Tom과 John의 할인율을 질의로 보내고 이에 대한 추론 결과를 [그림 13]에서 보여주고 있다.



[그림 13] 질의 - 고객 할인율 추론



[그림 14] 추론 과정 리스트

[그림 14]은 질의를 추론하는 시퀀스 리스트를

보여주고 있다. 추론의 목적은 사용자의 할인을 추천해내는 것이고 사용자 Tom은 GOLD 등급을 사실로 가지고 있다. 추론을 이끌어내는 룰과 그 룰을 입증하는 다른 룰이나 사실들을 서브리스트로 보여주고 있다.

3.5.4 클라이언트 - 결과 출력

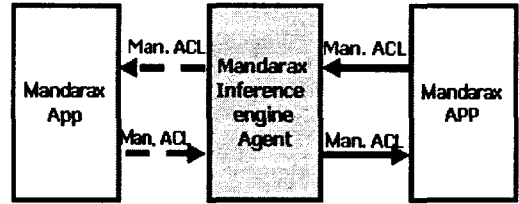
각 서버들로부터 제시된 가격을 미들웨어가 정리하여 최적화된 상품을 클라이언트로 보낸다.

4. 결론

본 논문에서는 표준 웹서비스 기반위에 추론엔진을 결합하여 자동화된 추천시스템을 제안하고 부분적인 구현을 시도하였다. 프로파일로 사용자에게 최적화된 추천뿐만 아니라 가격 추론에서 고객의 카테고리 결정이 복잡해질수록 추론엔진을 사용하는 것이 프로그래밍에 의존하는 것보다 효율적이 된다는 결론을 얻었다. 선행 연구에서 질의 모델[6]에 JESS를 적용하였던 반면, 현행 연구에서는 추천 시스템에 Mandarax를 적용하여 선행추론과 후향추론의 비교와 분석이 이루어졌고 부분적 구현을 통하여 표준 웹 서비스를 기반으로 한 추천시스템을 구현하였다. 이는 표준 웹 서비스 플랫폼 위에 구현하여 확장성이 높고, 사용자 데이터에 따라 사용자 중심의 추천시스템의 기반을 이룬다. 개발자의 입장에서 프로그래밍의 의존도를 줄이고, 유지보수 및 유연성 강화의 초석이 된다.

향후 연구는 첫째는 구현 확장 시 얻어지는 미들웨어의 크기에 따라 그 비중에 따른 서버와 클라이언트간 트래픽 분산방법을 QoS와 연계하여

연구하고 사용자의 데이터에 기계학습적 방법론을 통한 사용자 프로파일 구성 자동화에 초점을 맞추어 진행하고자 한다.



[그림 15] Madarax Agent

둘째로는 [그림 15]에서의 같이 에이전트와 에이전트간의 추론 엔진 활용을 용이하게 하기 위하여 JESS와 Madarax 추론 엔진을 FIPA기반의 ACL(Agent Communication Language)를 통하여 통신가능한 상태로 에이전트화 하도록 진행 중에 있다.

감사의 글

본 연구는 2004년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음

참조문헌

- [1] 김상진, 배경을 (2004) “웹서비스 기반의 SEM 시스템 아키텍처”.
- [2] <http://apache-korea.org/tomcat/index.html>
- [3] <http://mandarax.org>
- [4] <http://herzberg.ca.sandia.gov/jess>
- [5] H.M.Deitel, P.J.Deitel, J.P.Gadzick, K.Lomeli,

- S.E.Santry, S.Zhang. (2003) "Java Web Services for Experienced Programmers", p.108~113.
- [6] 김성태, 도승철, 이준규, 박수민, 양정진 (2004) "위치기반 멀티 에이전트를 활용한 긴급 구조 시스템 모델"
- [7] P.Cauldwell, et al. Professional XML Web Services(Birmingham, UK: Wrox Press, 2001)
- [8] P.Korzeniowski, "Internet Insight: A Little Slice of the UDDI Pie," eWeek 4 February 2002
- [9] Korzeniowski, P. "UDDI : Two Versions Down, One to Go", eWeek 4 February 2002
- [10] UDDI Version 2.0 API Specification," 8 June 2001
- [11] Actaggat, A. "Enabling XML Security," September 2001
- [12] Ballinger, K. and W. Nagy, "The WS-Inspection and UDDI Relationship," November 2001
- [13] Vijayan, J. "Web Services, Internet Collaboration Pose Security Challenges for 2002," Network World 3 January 2002
- [14] D.Singh, "XML and Binary Data,"
- [15] C. Boulton, "VeriSign Bows New XML Specs, Services," 29 November 2000
- [16] M.Andress, "The Road to Secure Web Services," InfoWorld 10 January 2002
- [17] M. Mactaggart, "Enableing XML Security : An Introduction to XML Encryption and XML Signature," September 2001
- [18] R. Yasin, "XML Standard to Keep Web Services Secure," Internet Week 30 July 2001
- [19] "HTTP Authentication : Basic and Digest Access Authentication,"
- [20] M. Milgliore, "IBM, Microsoft and Verisign Release SOAP Security Spec," 11 April 2002
- [21] "UDDI Advances with Web Services Description Language," 29 September 2000

Abstract

Web Service based Recommendation System using Inference Engine

SungTae Kim* · SooMin Park · JungJin Yang

The range of Internet usage is drastically broadened and diversified from information retrieval and collection to many different functions. Contrasting to the increase of Internet use, the efficiency of finding necessary information is decreased. Therefore, the need of information system which provides customized information is emerged. Our research proposes Web Service based recommendation system which employes inference engine to find and recommend the most appropriate products for users. Web applications in present provide useful information for users while they still carry the problem of overcoming different platforms and distributed computing environment. The need of standardized and systematic approach is necessary for easier communication and coherent system development through heterogeneous environments. Web Service is programming language independent and improves interoperability by describing, deploying, and executing modularized applications through network. The paper focuses on developing Web Service based recommendation system which will provide benchmarks of Web Service realization. It is done by integrating inference engine where the dynamics of information and user preferences are taken into account.

Key words : Web Service, Inference Engine, Knowledge Base, Recommendation System

* School of Computer Science and Information Engineering, The Catholic University of Korea