한국지능정보시스템학회논문지 제13권 제4호 2007년 12월(pp.1~16)

웹기반 전문가시스템의 구조 분류

임규건
한양대학교 경영대학 경영학부
(gglim@hanyang.ac.kr)

인터넷의 대중화와 e-비즈니스의 활성화에 따라 향후 인터넷과 유비쿼터스 환경에 대비한 지능형시스템 연구가 가속화 되고 있다. 전문가시스템도 기존의 Stand Alone방식에서 웹기반의 클라이언트-서버방식으로 발전되고 있으며, 다양한 인터넷 환경에서 활용되고 있다. 본 논문에서는 인터넷 웹 환경에서의 전문가시스템 개발환경과 고려사항들을 살펴 보고 웹 기반 전문가시스템들을 유형별로 분류하고, 분석하여 이를 통해 웹 기반 전문가 시스템의 일반적인 유형 모델과 구조를 제시한다. 본 논문에서는 첫 번째 클라이언트와 서버 간의 Load Balancing 측면에 따라 서버중심의 Server Oriented, 클라이언트 중심의 Client Oriented의 두 가지 모델을 구분 하였다. 그리고, 두 번째 지식 및 추론 공유 정도에 따라 지식과 추론에 대한 공유가 전혀 이루어 지지 않는 No Sharing, 서버들 간에 지식 및 추론의 공유가 이루어지는 경우인 Server Sharing, 클라이언트 간의 공유가 이루어 지는 경우인 Client Sharing, 클라이언트, 서버 모두에서 공유가 이루어 지는 경우인 Client-Server Sharing의 네 가지 모델을 분류하였다. 이들의 조합을 통해 총 8가지의 유형을 도출하였다. 또한 각 모델 별로 유형별 구조와 장단점, 인터넷상에서의 지식베이스, Fact Base, 추론엔진 등의 위치 문제와 사용기술, 고려사항, 서비스 유형들을 구분, 분석하였다. 본 논문을 통해서 제시된 프레임워크를 활용하여 새로운 환경에 좀더 능동적으로 대처할 수 있는 효율적인 전문가시스템의 다양한 개발이 기대된다.

논문접수일 : 2007년 02월 게재확정일 : 2007년 10월 교신저자 : 임규건

1. 서론

인터넷의 대중화와 전자상거래의 활성화와 함께 향후 유비쿼터스(ubiquitous) 환경의 도래는 지능형 시스템에 대한 관심을 고조시키고 있다. 기존의 지능형 시스템에 관한 많은 연구와 활용 중에서 가장 현실적으로 활성화된 분야가 전문가 시스템 분야이다. 전문가시스템은 지금까지 생산, 엔지니어링, 비즈니스, 의료, 농업, 통신, 정부, 법률, 교통, 재무, 투자, 관리, 회계/감사, 마케팅, 전자상거래, 인사 등 다양한 분야에 응용되어 왔다(Liebowitz, 1997). 앞으로 전문가시스템의 가치는 인터넷과 유비쿼터스 환경에서 다양한 영역별 응용에 그 진가가 더욱 빛날 것으로 전망된다(Intelligent software Strategies, 1996).

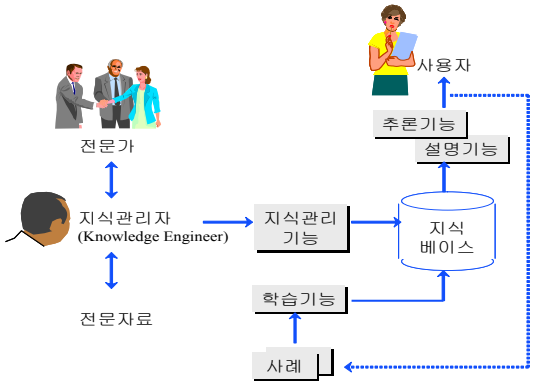
1990년 중반 인터넷의 상용화와 웹의 등장 이후 전문가 시스템도 인터넷 상에서 웹을 기반으로 하여 다양하게 개발, 활용 되어져 왔다. 최근 연구 논문들에서 볼 수 있듯이 교육(Tasi and Tseng, 2002), (Simic and Devedzic, 2003), 질병진단(Lia et al., 2002

; Zetian et al., 2005 ; Duana, 2003), 생산(Song et al., 2006), 농산물 치료(Shaalana et al., 2004), 의료(Huang and Chen, 2007), 마케팅(Li, 2005), 재무(Kim et al., 2005)등 다양한 분야에서 의사결정을 위한 웹기반 시스템이 개발되어 활용되고 있다. 그러나, 웹기반 전문가시스템의 구조에 대해서는 웹기반의 기본적인 구조에 대한 제시들이 있을 뿐 체계적인 분류와 모델의 유형화에 대한 연구는 미비하다. 이에 본 논문에서는 인터넷의 웹 환경에서의 전문가 시스템의 구조에 대한 체계적인 분류체계를 제시하고자 한다.

이를 위해서 제 2장에서 웹 기반 전문가시스템 등장과 일반적인 구조를 살펴 보고, 제 3장에서 웹 기반 전문가시스템을 개발할 때 문제시 되는 여러 가지 사항들을 생각하여 분류 시 고려사항들을 살펴보고, 제 4장에서 이를 고려한 웹 기반 전문가시스템의 일반적인 유형별 모델을 제시한다. 특히 본 연구에서는 전통적인 시스템 구조분류 방식인 Load Balancing 측면과 인터넷 분산환경에서의 지식 공유적 측면을 고려하여 모델을 제시하고자 한다. 또한 각 모델별로 인터넷상에서의 지식베이스, Fact Base, 추론엔진 등의 위치 문제와 장, 단점, 고려사항, 사용기술, 서비스 유형들을 구분, 분석하여 향후 웹기반 전문가 시스템 개발의 프레임워크를 제시하고자 한다. 제 5장에서는 제시된 주요 모델의 특징을 분석하고 본 연구를 위하여 2002년부터 2007년 사이에 Expert Systems with Applications 논문지에 게재된 10개의 주요 웹기반 전문가 시스템들을 본 논문이 제시한 분류 기준으로 분석해 보고 제 6장에서 결론을 요약한다.

2. 웹기반 전문가시스템

전문가시스템의 전형적인 구조는 [그림 1]과 같다. ***지식관리기능***을 통해 지식관리자가 지식을 ***지식베이스***에 저장하게 되고, 사용자의 요구에 따라 ***추론기능***을 통해 논리적 또는 다양한 인공지능 기법을 활용하여 추론을 하고, 이에 대한 응답을 ***설명기능***을 통해 하게 된다. 새로운 지식은 ***학습기능***을 통해 지식베이스에 다시 저장되게 된다(이재규 외, 1998).



[그림 1] 전문가 시스템의 전형적 구조(이재규 외, 1998)

이러한 초기의 전문가시스템은 스탠드얼론(Standalone) 시스템 방식으로 진단, 비즈니스 규칙, 정책 등과 같은 응용분야에서 중요한 정보들을 지식으로 표현하고 저장하여 이용할 수 있는 방법을 제시하였다. 그러나, 이러한 형태의 전문가시스템은 지식을 많은 사용자에게 분배할 수 없었고 단지 접근 가능한 몇몇 사용자들에 의해 이용가능하였다. 이 후 전문가시스템은 이와 같은 지식 분배 문제를 해결할 수 있는 클라이언트 서버(Client Server) 구조로 발전하였다. 그러나 대부분 각 특정 영역의 응용사례별로 특별한 경우에 한정되어 사용되는 구조였고, 네트워크의 취약성 때문에 고비용의 클라이언트의 비중이 큰 팻 클라이언트(Fat Client) 구조가 주로 사용되어 과거의 클라이언트 서버 전문가시스템은 커다란 실효를 거두지 못하였다(Behrouz and Koono, 1996).

이와 같은 상황에서 인터넷과 웹의 등장 및 보급으로 웹 서버에 전문가시스템을 배치하고 웹 브라우져를 가진 사람은 누구나 이용할 수 있는 웹 기반 전문가시스템의 등장은 지능형 시스템의 대중화에 크게 기여 하였다. 웹기반 전문가 시스템은 일반적인 웹의 유연한 접근성과 유용성 등의 장점과 함께, 멀티미디어와 동적 구동 등이 가능하며 정보의 통합뿐 아니라 전문가 시스템의 분배 문제가 간단하다는 장점을 가지고 있다.

이와 같은 웹을 기반으로 한 전문가시스템의 일반적인 모형은 서버에 지식과 추론엔진 등이 위치하고 HTTP프로토콜을 이용하여 브라우져를 통해서 사용자에게 서비스되는 [그림 3]과 같다.

3. 웹 기반 전문가시스템의 분류를 위한 고려 사항

웹기반 전문가시스템의 구현은 일반적인 데이터베이스 시스템을 중심으로 한 웹 기반 클라이언트-서버 구조의 개발과 차이가 있다. 일반적인 데이터베이스 중심 방식은 사용자가 웹 브라우져상에서 웹 서버에 연동되어 있는 데이터베이스에 질의를 요청하고 브라우져는 그 결과를 단순히 화면에 보여준다. 즉, 클라이언트는 아무런 연산이 필요 없고 단지 서비스를 요청하고 그 결과를 보여주는 역할만 담당한다.

그러나, 전문가시스템은 이와 달리 추론과정을 요구하기 때문에 실행 중 사용자와의 계속적인 상호작용을 요구한다. 즉, 서버와 연결이 한번만 연결되고 끊기는 것이 아닌 연결 상태에서 계속적으로 Fact가 교환되고 추론이 일어나야 한다. 따라서, 클라이언트 쪽에서 서버와 연결을 유지할 수 있고, 중간에 발생하는 Fact를 저장하며 필요한 경우 계산을 수행할 수 있는 구조가 필요하다.

또한, 전문가시스템은 추론과정 등에 많은 양의 계산능력을 요구한다. 이와 같은 상황에서 여러 명의 사용자가 동시에 서버에 접속해서 많은 계산을 수행한다면 서버의 부담이 증가하여 서버의 능력이 현저하게 저하될 수 있다. 따라서, 클라이언트도 계산능력을 가질 수 있는 실행 가능한 클라이언트(Client-Executable) 구조가 요구된다. 하지만 클라이언트 프로그램을 다운로드 받아야 하는 부담이 생기므로 이를 고려해야 한다.

전문가시스템의 구성요소 중 지식관리기능 및 학습기능, 설명 기능은 일반적인 전문가시스템과 대동소이하므로 논외로 하고, 이러한 다양한 특성들을 감안하여 웹 기반 전문가시스템을 개발하고자 할 때 고려해야 할 사항들을 정리하면 다음과 같다.

- 서버와 클라이언트 사이의 로드밸런스(Load Balancing)
  - Multi-User로 인한 서버의 부담 분산 문제
  - 네트워크 트래픽의 분산 문제
- 지식베이스(Knowledge Base : KB)의 위치 및 공유 정도
  - 지식베이스의 서버 또는 클라이언트 간의 위치 문제
  - KB의 공유 및 개인별 KB의 유지 문제
- 추론엔진(Inference Engine : IE)의 위치 및 공유 정도
  - 추론엔진의 서버 또는 클라이언트 간의 위치 문제
  - 서버간의 추론 공유 및 클라이언트 간의 추론 공유
- 지식의 공유와 보호
  - 사용자 간의 개인별 지식의 공유 고려

- 정보 보안의 고려
• 추론 Session의 유지
- 추론 과정에서의 Fact 추론 히스토리 유지
• 프로그래밍 기술
- C/C++, Perl, Java, Javascript, Flash, CGI, CORBA, External Helper, MIME, Web Services, XML 등 활용 기술

3.1 서버와 클라이언트 사이의 로드밸런스 (Load Balancing)

전문가시스템을 클라이언트와 서버 구조로 나눌 때 통신 트래픽 측면을 고려해야 한다. 특히 인터넷과 같은 네트워크 환경에서는 데이터 전송률이 네트워크 환경에 따라 예측 불가능한 통신 대역폭(Bandwidth)를 가지기 때문이다. 따라서, 사용자에게 신뢰성있는 응답시간을 보장할 수 있는 클라이언트-서버 구조를 가져야 한다. 이러한 측면에서 서버 중심 또는 Client-Executable 구조를 상황에 따라 적절히 조정하여 효율적인 웹 기반 전문가시스템을 구성해야 한다(Orfali and Harkey, 1997).

우선, 전문가시스템 서버와 사용자 인터페이스간에 통신을 최소화하는 측면에서는 Client-Executable 구조가 적합하다. 즉, 빠른 응답시간을 위해서 사용자 인터페이스는 마우스 조작을 포함하여 모든 계산을 서버가 아닌 사용자 클라이언트에서 이루어 지도록 하는 것이 좋다. 웹 상에서 Java Application을 이용하면 이와 같은 기능을 수행하면서도 플랫폼에 독립적인 클라이언트를 구성할 수 있다. 서버의 계산량을 감소시키고 네트워크 트래픽량을 감소시키기 위해 사용자로부터 받는 입력 값을 미리 JavaScript를 이용하여 클라이언트 쪽에서 체크 가능하게 하거나 Java Applet을 이용하여 클라이언트에 계산 능력을 부여할 수도 있다.

그러나, 사용자 쪽에서는 복잡하고 커다란 클라이언트를 다운로드하고 관리 해야 하는 부담이 있다. 또한, 클라이언트 프로그램을 자주 업데이트 해야 하는 경우 통신으로 인한 부하보다 다운로드로 인한 부하가 더 클 수 있다.

반면에 커다란 지식베이스와 데이터베이스를 이용하는 경우는 서버에 데이터를 집중하고 데이터의 분배를 피하는 서버 중심적인 구조를 취하고 상대적으로 작은 클라이언트를 설계하여 다운로드시 부하를 감소시킬 수 있다.

클라이언트와 서버간의 비중을 결정하는 것 외에도 시스템 특성에 따라 적절한 통신 프로토콜을 쓰는 문제도 함께 고려해야 한다. 사용자 인터페이스와 전문가시스템 서버에 Web Services 또는 CORBA와 같은 공유 객체를 지원하는 객체 통신 프로토콜을 활용할 수도 있다.

3.2 지식베이스의 위치 및 공유 정도

지식베이스의 위치가 서버에 있느냐 클라이언트에 함께 있느냐가 전문가 시스템의 구조 및 데이터 응답시간에 영향을 미친다. 지식베이스가 서버에만 위치한 경우는 클라이언트와는 독립적으로 일관된 지식베이스를 유지할 수 있고 통합된 변경과 제어가 용이하다. 그러나, 클라이언트와 통신하는 게 부담이 될 수 있고 클라이언트 별로 개인별 지식베이스를 운용하기에 부적합하다. 반면에, Java Application 등을 이용하여 클라이언트에 지식베이스를 두는 방안은 중앙 집중적인 제어에는 한계가 있지만 통신에 드는 부하는 많이 감소하며 지식베이스를 개인별로 유지할 수 있고 개인 지식의 보안성도 높일 수 있다.

3.3 추론엔진의 위치 및 공유 정도

전문가시스템에서는 원하는 결과를 얻기 위해

서 추론엔진을 통해 추론을 한다. 이 때, 추론엔진이 서버에 위치하느냐 클라이언트에 위치하느냐도 중요한 이슈이다. 만일 추론엔진이 서버에 위치하는 경우는 중앙집중적인 지식처리가 가능하지만 사용자와의 많은 상호작용으로 인한 부담이 증가하여 통신 부하가 증가된다.

이와 반대로 Java Application 등을 이용하여 클라이언트에 추론엔진을 위치시킬 수도 있다. 또한, Java GUI, Flash 등을 활용하여 서버에 지식를 두고 클라이언트에서 규칙을 다운로드하는 Smart Java Applet(Smartlet)의 형태 등 유연성 있는 추론엔진을 구성할 수도 있다. 그러나, 이 경우는 지식베이스의 양이 크거나 사용자가 많은 지식를 요구하는 경우에는 클라이언트가 지식을 다운로드 받고 유지하는 부담이 클 수 있다(Intelligent Software Strategies, 1996, 1997).

3.4 지식의 공유와 보호

인터넷과 웹의 개방성은 지식을 공유하는 측면에서는 매우 유리하나 반대로 사적인 정보나 중요한 정보를 보호하는데 어려움을 준다. 따라서 지식베이스의 활용 및 추론과정에서의 서버와 클라이언트 간에 지식의 공유와 활용을 고려할 때 지식을 인터넷을 통해 전송 시 암호화된 통신을 이용하여 다른 불법적인 사용자에게 정보를 누출시키지 않도록 하고 사용자 별로 사적인 정보의 이용을 제어할 필요가 있다. 신뢰성 있는 거래처리를 위해서는 기밀성(confidentiality), 무결성(integrity), 인증(authentication), 부인봉쇄(non-repudiation)의 요건이 만족되어야 한다(임규건 외, 2005). 지식 및 데이터는 암호화를 통해서, 사용자에 대한 인증은 ID 또는 인증서 기반으로 처리 가능하다.

3.5 추론 Session의 유지

특히 HTTP는 stateless protocol이기 때문에 추론과정에서 일어나는 Fact들의 히스토리(history)를 유지하기 어렵다. 따라서, 추론과정의 히스토리를 유지하기 위해서는 특별한 조치가 필요하다(송용욱 외, 2003). 다음과 같은 추가적인 방법이 사용될 수 있다.

- HTML의 Hidden Form 활용 : HTML안에 매 추론 과정마다 발생하는 Fact 히스토리를 Hidden Form으로 계속적으로 추가해서 유지하는 방법이다. 이 경우 추론이 길어질 때 마다 HTML의 크기가 커지고 서버와의 통신 부담이 커지게 된다.
- Cookie의 활용 : 추론 과정에서 Cookie를 이용하여 사용자별 Fact 히스토리를 참조할 수 있다. 그러나, 서버측이 각 사용자별로 Fact 히스토리를 유지해야 하므로 서버의 부담이 발생한다.
- 프로그래밍을 통한 세션의 연결 : Java 등 프로그래밍 언어를 통해 지식베이스와의 세션을 계속 유지함으로써 추론과정에 일어나는 모든 추론의 히스토리를 유지할 수 있다.

3.6 프로그래밍 기술

웹 기반 전문가시스템을 가능하게 하기 위해서는 서버와 클라이언트 간의 상호작용을 위해 필요한 프로그래밍 언어와 사용자 인터페이스를 위한 프로그래밍 언어가 필요하다. 상호작용을 위한 CGI용 언어로는 C, C++, Java, ASP, PHP 등을 활용할 수 있다. 클라이언트의 편리한 인터페이스를 위해서는 HTML과 이를 보완해주는 JavaScript, Java, 그래픽 기능이 뛰어난 Flash 등을 활용할

수 있다(Gosling and McGilton, 1995 ; Orfali and Harkey, 1997 ; Srinivas, 1997).

클라이언트와 서버 사이의 구동 방식은 기본적으로 클라이언트와 웹 서버 사이에 HTTP 프로토콜을 이용하여 구현되는데 크게 다음과 같은 네가지 방식이 있다(임규건 외, 1997).

- CGI(Common Gateway Interface)를 이용하는 방법 : 가장 기초적인 방법으로 웹 브라우져로 웹의 Form 기능을 이용하여 전문가시스템 서버에 접근하는 방법으로 간단한 사용자 인터페이스를 제공한다. 그러나, Form 기입과 프로그램 출력으로서의 웹 브라우져 이용 외에는 사용자와의 상호작용이 어렵다.
- External Helper 프로그램을 이용하는 방법 : C나 VC++, Java등을 이용하여 브라우져의 Helper 프로그램으로 구현하여 독립적인 클라이언트 프로그램으로 웹과 연동해서 작동하게 할 수 있다. Plug In형태로도 가능하며 Java Application으로 개발할 수도 있다.
- Javascript 를 이용하는 방법 : Sun Microsystem의 Java 프로그래밍 언어는 전문가시스템 서버와의 인터페이스를 구현하는데 강력한 기초를 제공해 준다. HTML에 JavaApplet 이나 Javascript로 쓰여진 프로그램을 포함시키면 웹 브라우져는 이들 프로그램을 인터넷을 통해 다운로드 받아 지역적으로 수행시킨다(Intelligent Software Strategies, 1997).
- Web Services/CORBA를 이용하는 방법: Web Services또는 CORBA를 이용하여 객체지향의 구현과 지식 공유를 효율적으로 실현시킬 수 있다.

4. 웹 기반 전문가시스템의 유형 분류

위와 같은 고려사항들을 염두에 두고 본 논문에서는 웹 기반 전문가 시스템의 유형을 분류하는 것을 다음과 같은 두 가지의 범주의 문제로 압축하여 분석하였다.

하나는 전통적인 클라이언트-서버 구조에 대한 분류 기준인 서버와 클라이언트 간의 Load Balancing의 문제이다. 즉, 서버 쪽에 추론엔진, KB, FB(F-act Base) 등을 집중적으로 둘 것인가? 아니면 클라이언트 쪽에 이러한 기능들을 둘 것인가의 문제이다. 이에 따라서 개발 시 고려할 사항과 사용될 기술, 그리고 서비스유형이 나누어 질 수 있을 것이다. 이것은 전장에서 기술한 고려 사항 중 지식베이스의 위치 및 공유정도, 추론엔진의 위치 및 공유정도, 추론 세션의 유지, 프로그래밍 기술 등과 연관된 요소이다.

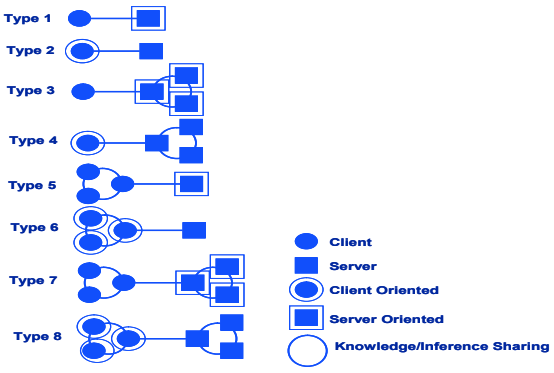
두 번째는 인터넷의 가장 큰 장점인 분산환경에서의 지식과 추론의 공유를 전문가시스템에 어떻게 접목시키느냐에 대한 문제이다. 서버들 간의 지식을 공유시켜 다수의 전문가시스템 간의 상호연동 추론 가능성과 클라이언트 간에도 서로 지식을 교환할 수 있는 환경에 대한 고민이다. 이것은 전장에서 기술한 지식베이스의 위치 및 공유정도, 추론엔진의 위치 및 공유정도, 지식의 공유와 보호, 추론 세션의 유지, 프로그래밍 기술 등과 연관이 있다.

본 논문에서는 이러한 사항들을 고려해서 첫 번째 Load Balancing기준에 의해 Server Oriented, Client Oriented의 두 가지 모델을 구분 하였다. Server Oriented란 많은 부하가 서버쪽에 치우치는 것을 의미하고, Client Oriented란 클라이언트 쪽에 치우치는 것으로 정의한다. 그리고, 두 번째

지식 및 추론 공유에 따라 No Sharing, Server Sharing, Client Sharing, Client-Server Sharing의 네 가지 모델을 분류하였다. No Sharing이란 지식과 추론에 대한 공유가 전혀 이루어 지지 않는 경우, Server Sharing이란 서버들 간에 지식 및 추론의 공유가 이루어지는 경우, Client Sharing이란 클라이언트 간의 공유가 이루어 지는 경우, Client-Server Sharing이란 클라이언트, 서버 모두에서 공유가 이루어 지는 경우를 의미한다. 이를 통해 다음 <표 1>과 [그림 2]같은 8가지의 유형을 도출할 수 있다. 이어서 각 유형별 구조와 장단점, 고려사항, 사용기술, 서비스유형에 대해서 제시하겠다.

<표 1> 웹 기반 전문가시스템의 유형 분류

	No Sharing	Server Sharing	Client Sharing	Client-Server Sharing
Server Oriented	Type 1 Conventional Server Oriented Model	Type 3	Type 5	Type 7
Client Oriented	Type 2 Conventional Client Oriented Model	Type 4	Type 6	Type 8



[그림 2] 웹 기반 전문가시스템의 유형

4.1 Type 1 : Conventional Server Oriented Model

서버 쪽에 추론엔진, KB, FB 등이 위치하여 대부분의 작업을 수행하고 클라이언트 쪽은 단순이 브라우져 기능을 위주로 하는 경우이다. 즉, Thin Client Fat Server형으로 웹을 통한 전문가시스템의 초기 모형으로 일반적이고 대표적인 전문가시스템의 유형이다.

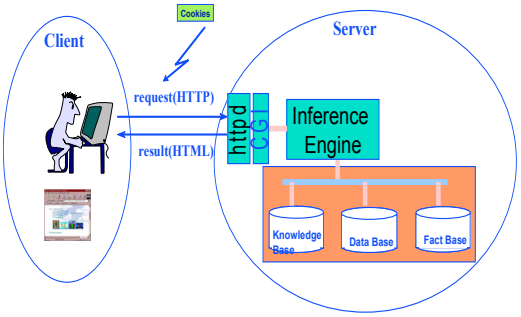
이것은 [그림 3]과 같은 경우로서 서버 쪽의 전문가시스템에서는 HTTP 서비스가 가능하도록 웹 서버의 역할을 해주고, 추론엔진과의 CGI 등을 통한 인터프리터를 만들어 주고, KB, FB등을 효율적으로 유지, 관리 해야 한다. 즉, 모든 부하가 서버 쪽으로 치우치는 경우이다. 클라이언트 쪽에서는 단순히 브라우징 기능을 하는 웹 브라우져를 가지고도 충분히 전문가시스템을 사용할 수 있다.

이러한 경우의 장점은 지식, Fact 등을 서버에서 집중적으로 관리할 수 있으며 개발 부담도 거의 없는 단순 클라이언트 프로그램을 사용함으로 사용자의 부담을 최소화할 수 있다는 것이다. 단점으로는 서버에 과중한 부하가 걸리며 많은 수행들이 서버에서 이루어지므로 상대적으로 사용자의 요구에 대한 처리속도가 늦어 질 수 있다는 것이다.

이 경우 고려해야 할 사항들은 사용량 또는 시간, 정보에 따라 과금을 어떻게 하느냐의 문제와 각 사용자의 히스토리나 개인별 Fact, 지식의 관리, 접속 후 시간이 경과된 후에 다시 접속했을 때 Cookie 등을 이용해 연속성을 유지 시켜주는 문제 등이다.

사용기술로는 기본적인 HTTP 서버와 CGI기술 등으로도 충분하며, 단순하게 기존의 전문가시스템을 CGI를 이용해서 웹과의 인터페이스를 만들어 사용 가능하다.

서비스 영역은 서버에서 주로 많은 작업을 해야 하는 경우, 서버에서 많은 사용자의 정보를 모으고자 하는 경우 등 주로 대중을 대상으로 하는 서비스인 경우와 특별히 정보 보안이 요구되는 전문가시스템의 경우이다.



[그림 3] Type 1 : Conventional Server Oriented Model (웹 기반 전문가시스템의 일반적 모형)

4.2 Type 2 : Conventional Client Oriented Model

클라이언트 쪽에 추론엔진, KB, FB 등의 많은 부분을 위치시켜 주거나 클라이언트 역할에 비중을 두는 경우로서 서버 쪽은 상대적으로 조정 정도의 작업을 하는 경우이다. 즉, Fat Client Thin Server형의 전문가시스템인 경우이다.

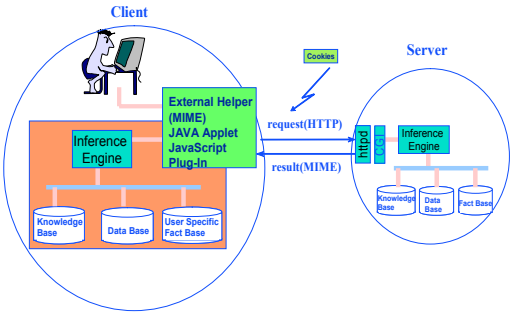
이런 경우에는 [그림 4]와 같이 서버 쪽 전문가시스템에서는 HTTP서비스를 이용하여 전문가시스템의 주 기능을 해주거나 단순한 사용제어 등의 조정을 해주고, 클라이언트 쪽에서는 부라우징 기능 이외에 사용자별 FB를 위주로 KB 또는 추론엔진의 일부를 클라이언트 쪽에 위치하고 관리하는 기능을 지니게 된다.

이러한 유형의 장점은 지식, Fact 등을 클라이언트에서 관리할 수 있으므로 각 사용자가 자신의 기호 또는 취향에 맞는 개인적인 지식을 관리하면서 전문가시스템을 사용할 수 있고, 이를 통해 개인의 사적인 정보의 보호가 가능하며, 서버의 부담을 적게 해주고, 필수적인 최소한의 데이터가 서버와 클라이언트 간에 교환되므로 처리속도를 향상시킬 수 있다. 단점으로는 클라이언트에 부하가 걸릴 수 있다는 것과 구현비용, 그리고 서버 쪽 전문가시스템에서 다양한 고객의 정보를 입수하는데 어려움이 있다는 것이다.

이 경우 고려해야 할 사항은 과연 추론엔진, KB, FB 중에서 어느 것을 어느 정도 클라이언트 부분에 위치 시킬 것인가에 대한 것과 사용자 개인별 지식과 전체 지식과의 연관, 추론시의 전문가시스템 동작 고려 등이다.

사용기술로는 단순한 CGI기술만으로 이러한 시스템을 구현하기는 어려우며 클라이언트에서도 작동 가능한 프로그램이 되어야 하므로 CGI, Java, Flash, MIME, Plug-In 등의 기술을 사용해서 External Helper Program 이나 Java Application 같은 클라이언트에서 동작 가능한 프로그램을 만들어야 한다.

서비스 영역은 클라이언트에서 주로 많은 일을 처리해야 되는 경우, 사용자 개개인의 정보의 보호가 필요한 경우, 클라이언트 정보 만으로 추론이 가능한 경우, 특정 고객을 대상으로 하는 서비스를 개발해야 하는 경우 등이다.



[그림 4] Type 2 : Conventional Client Oriented Model

4.3 Type 3 : Server Oriented Server Sharing Model

서버들이 인터넷상에서 여러 곳에 위치하며 서버간의 통신과 정보교환이 가능한 분산환경 전문가시스템인 경우이다. 즉 각 서버들은 독립된 추론엔진, KB, FB 등을 갖는 전문가시스템들이며 이들은 서로 하나의 객체로서 인식되어 추론이나 KB, FB의 이용을 다른 전문가시스템의 결과를 연계시켜 활용할 수 있는 구조이다. 예를 들어 갑이라는 사람이 주식투자자문을 위해 A라는 전문가시스템을 이용하는 경우 A주식투자자문 전문가시스템은 인터넷상의 B 기업정보 전문가시스템과 C 은행재무정보 전문가시스템 등을 활용하여 적절한 답을 갑에게 내주는 구조이다.

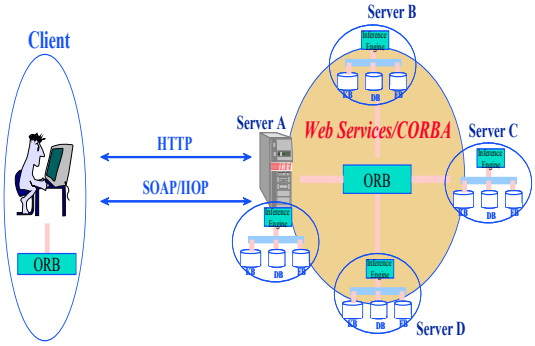
이런 경우에는 [그림 5]와 같이 서버들간에는 분산환경에 적합한 Web Services나 CORBA(Common Object Request Broker Architecture) 등을 이용하여 각 서버들 간의 정보교환이 제대로 이루어질 수 있도록 만들어주고 클라이언트 쪽에는 HTTP나 SOAP(Simple Object Access Protocol) 또는 IIOP(Internet Inter ORB Protocol) 등을 이용해 전문가시스템과 통신을 하게 된다.

이러한 경우의 장점은 분산된 각종 전문가시스템의 지식을 충분히 활용할 수 있다는 것이다. 또한 객체지향기법으로 다른 전문가시스템들을 활용할 수 있으므로 그만큼 새로운 전문가시스템의 구현이 용이해 질 수 있으며, 전체적인 모듈화가 가능하여 효율성을 높일 수 있다. 단점으로는 분산된 서버들을 사용하므로 네트워크 트래픽으로 인한 전체적인 처리속도 문제와 분산환경에 따른 구현의 어려움을 들 수 있다.

이 경우 고려해야 할 사항은 여러 서버들간의 통신 및 과금 등의 역할 정립과 문제해결을 위해 분산환경에서의 질의 최적화(Query Optimization)에 대한 고려이다.

사용기술로는 핵심적으로 웹, Java를 적절히 이용한 Web Services나 CORBA가 사용될 수 있다. 또한 데이터의 교환을 효과적으로 하기 위해서 MIME이나 XML도 필요할 것이다.

서비스 영역은 여러 가지 문제들이 종합적으로 고려되어야 하는 복잡한 서비스와 다른 전문가시스템들을 이용해서 새로운 부가서비스를 하고자 할 경우 등이라고 볼 수 있다. 이러한 모델은 미래 인터넷과 e-비즈니스 환경에서 새로운 서비스를 창출할 수 있는 상당히 유용한 모델이라고 하겠다.



[그림 5] Type 3 : Server Oriented Server Sharing Model

4.4 Type 4 : Client Oriented Server Sharing Model

클라이언트 쪽에 추론엔진, KB, FB 등의 많은 부분을 위치시켜 주는 Fat Client 형이지만, 서버들간의 추론 및 지식 등을 공유하는 형태이다. 이 형태는 Type 2에서 서버들 간의 공유를 지원하는 경우로서 서버 쪽의 지식 및 추론 공유는 Web Services 등으로 연결할 수 있다. 주요 특징들은 Type 2와 Type 3의 특징들을 따른다.

4.5 Type 5 : Server Oriented Client Sharing Model

서버 쪽에 추론엔진, KB, FB 등이 위치하여 대부분의 작업을 수행하고 클라이언트 쪽은 단순이 브라우져 기능을 위주로 하는 Type 1의 변형이다. 즉, Type 1의 특징을 가지고 있으면서, 클라이언트 간에 지식의 공유가 가능한 형태라고 할 수 있다. 주요 특징들은 Type 1과 Type 6의 특징들을 따른다.

4.6 Type 6 : Client Oriented Client Sharing Model

인터넷상에서 여러 곳에 위치한 클라이언트들 간에도 서로가 정보를 교환할 수 있는 구조의 전문가시스템인 경우이다. 새로운 개념의 모델로서 각각의 클라이언트들이 지니고 있는 개인별 FB 등을 클라이언트 간에 서로 이용 가능하도록 하는 모델이다. 예를 들면 인터넷상에서 10명이 주식투자동호인 클럽을 결성했다고 할 때, 각각의 투자자는 개인의 전문가시스템 용 클라이언트 프로그램을 가지고 개인의 투자기록이나 선호도에 의한 FB와 전문가시스템 서버를 이용해서 투자자문을 받고, 10명의 동호인클럽인 끼리는 서로의 클라이언트 프로그램을 이용해서 다른 동호인의 전문지식과 정보를 활용할 수 있는 경우이다.

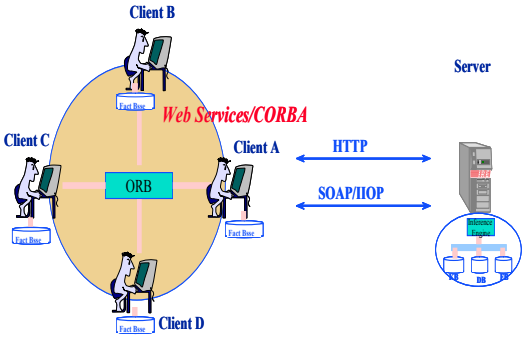
이런 경우에는 [그림 6]과 같이 Web Services나 CORBA 등을 이용하여 각 클라이언트들 간의 정보교환이 제대로 이루어 질 수 있도록 만들어주고 서버 쪽은 일반적인 HTTP나 SOAP, IIOP 등을 이용해 전문가시스템과 통신을 하게 된다.

이러한 경우의 장점은 클라이언트끼리 지식과 정보를 서로 공유할 수 있으므로 클라이언트 간의 CUG(Closed User Group)의 효과가 있다는 것이다. 단점으로는 여러 클라이언트 간의 통신을 처리해 주어야 하므로 서로 간의 정보를 주고 받을 수 있는 구조 구현의 어려움을 들 수 있다.

이 경우 고려해야 할 사항은 클라이언트 간의 직접적인 정보전달이 어려울 수 있으므로 매개 서버를 통해서 구현할 것인가 직접 클라이언트간의 통신을 허락할 것인가 등 클라이언트 간과 서버와의 통신상의 관계정립 등을 들 수 있다.

사용기술로는 클라이언트용 Helper Program을 이용해서 클라이언트 간의 통신하는 것과 웹, Java를 적절히 이용한 Web Services나 CORBA가 사용될 수 있다. 근래의 P2P(Peer-to-Peer) 기술도 활용할 수 있으며, 데이터의 교환을 효과적으로 하기 위해서 MIME, XML도 필요할 것이다.

서비스 영역은 각 부서별 이용, 전문동호인 클럽 등 사용자를 그룹핑할 수 있는 서비스의 경우와 사용자들의 개인정보의 연계 활용이 많은 비중을 차지하는 서비스 등이다. 이러한 모델 또한 다양한 환경에서 사용자들의 정보교류가 요구되는 유비쿼터스 시대에 적합한 모델이라고 볼 수 있다.



[그림 6] Type 6 : Client Oriented Client Sharing Model

4.7 Type 7 : Server Oriented Client-Server Sharing Model

클라이언트와 서버 쪽 모두가 분산 및 공유 환

경을 제공하는 경우로서, 추론엔진, KB, FB 등이 서버에 주로 위치하여 서로 정보를 공유하며 대부분의 작업을 수행하고 클라이언트 쪽은 브라우징 기능을 하면서 지식을 공유하는 형태이다. 즉, Type 3의 특징을 가지고 있으면서, 서버간 뿐만 아니라 클라이언트 간에도 지식의 공유가 가능한 형태라고 할 수 있다. 주요 특징들은 Type 3을 중심으로 Type 6의 특징들을 띈다.

4.8 Type 8 : Client Oriented Client-Server Sharing Model

클라이언트와 서버 쪽 모두가 분산 및 공유 환경을 제공하는 다른 경우로서, 추론엔진, KB, FB 등이 클라이언트에 주로 위치하여 서로 정보를 공유하며 대부분의 작업을 수행하고 서버쪽도 지식을 서로 참조하여 동작하는 형태이다. 즉, Type 6의 특징을 가지고 있으면서, 클라이언트 간과 서버간에도 지식의 공유가 가능한 형태라고 할 수 있다. 주요 특징들은 Type 6을 중심으로 Type 3의 특징들을 띈다.

5. 주요 모델에 대한 분석

5.1 웹 기반 전문가시스템의 주요 모델 비교

위에서 살펴 본 8가지 모델 중에 주요 모델에 대해서 장점, 단점, 고려사항, 사용기술, 추론엔진의 위치, KB의 위치, FB의 위치, 서비스영역과 그 적용 분야 예를 아래 <표 2>에 정리 하였다. 이렇듯 웹 기반 전문가시스템을 개발하고자 할 경우에는 전략적으로 그 서비스 영역에 따라서 각 모델별로 서로 다른 고려사항들과 다른 개발방법을 사용할 수 있을 것이다.

5.2 주요 웹 기반 전문가시스템 사례

본 연구를 위해서 2002년~2007년 사이의 Expert Systems with Applications 논문지에 게재되었던 주요 웹 기반 전문가 시스템 10개를 살펴 보았다. 이들을 분석한 표는 <표 3>과 같다. 역시 일반적인 구조인 Type 1의 Conventional Server Oriented Model이 가장 많았다. 특징적인 것들로 Multi Agent Diagnosis(Shaalana et al, 2004)의 농작물 질병 치료시스템은 클라이언트 중심이면서 서버간의 지식의 공유가 가능한 Type 4의 Client Oriented Server Sharing Model이었고, WebStra(Shuliang Li, 2005)의 마케팅전략 계획 지원시스템은 서버 중심이면서 클라이언트 간에 Fact를 공유하는 Type 5의 Server Oriented Client-Sharing Model이었으며, T-VET(Duana, 2003)의 어류질병 진단의 서브시스템은 서버중심이면서 클라이언트 간과 서버간에 지식의 공유가 가능한 Type 7의 Server Oriented-Client-Server Sharing Model이었다. 즉 기본적인 웹 전문가시스템의 구조가 아직도 많이 활용되고 있지만 차츰 지식 및 추론을 분산하고 공유하여 처리하는 시스템들도 개발되고 있음을 알 수 있었다.

6. 결론

지금까지 인터넷 웹 환경에서의 전문가시스템 개발환경과 고려사항들을 살펴 보고 웹 기반 전문가시스템들을 유형별로 분류하고, 분석하여 이를 통해 8가지 웹 기반 전문가시스템의 일반적인 유형 모델과 구조를 제시하였다. 인터넷상의 웹기반에서는 추론엔진, KB, FB의 위치와 클라이언트 프로그램의 형태에 따라서 Load Balancing 차원에서 서버 쪽에 많은 비중이 있는 Server Oriented

<표 2> 웹 기반 전문가시스템의 주요 모델 비교

	Type 1 : Conventional Server Oriented Model	Type 2 : Conventional Client Oriented Model	Type 3 : Server Oriented Server Sharing Model	Type 6 : Client Oriented Client Sharing Model
장점	Fact, 지식 등을 서버에서 집중 관리 사용자 부담 최소화 요금 부과 등을 한곳에서 처리가능	사용자용 Fact, 지식 등을 클라이언트에서 관리 개인의 사적 정보 보호 서버 부담 최소화 개인정보의 유출 방지 처리 속도	분산된 각종 전문가시스템의 정보를 십분 활용 객체지향기법으로 새로운 전문가시스템의 구현이 용이 전체적인 모듈화가 가능하여 효율성을 높일 수 있음	CUG의 효과 새로운 형태의 전문가시스템 구현 가능
단점	서버의 로드 과중 사용자 요구에 대한 처리 속도	클라이언트의 부하 과중 중요정보의 서버관리측면에서의 입수 어려움 구현 비용	네트워크 트래픽으로 인한 전체적인 처리속도 구현의 어려움	구현의 어려움
고려사항	사용자 과금 사용자 연속성 유지	어느 정도를 클라이언트 부분에 위치 시킬 것인가? 개인별 지식과 전체공유지식과의 추론시의 고려문제	서버들간의 통신 및 과금 문제 질의 최적화(Query Optimization)	클라이언트간의 통신과 서버와의 통신과의 관계정립
사용기술	HTTP Server, CGI, PHP, ASP, C/C++, Perl	HTTP Server, CGI, PHP, ASP, Java, MIME, Plug In, C/C++, Perl, Flash	HTTP Server, CGI, PHP, ASP, Web Services, CORBA, Java, MIME, XML, C/C++, Perl	HTTP Server, CGI, PHP, ASP, Web Services, CORBA, Java, MIME, XML, C/C++, Perl, Flash, P2P
추론엔진의 위치	서버 쪽에 치중	클라이언트 쪽에 위치 할 수 있음	각 서버들에 독립된 추론엔진이 있음	각 클라이언트에 위치할 수 있음
KB의 위치	서버 쪽에 치중	클라이언트 쪽에 사용자별 KB를 위치할 수 있음	각 서버들에 독립된 KB가 있음	각 클라이언트에 사용자별 KB를 위치할 수 있음
FB의 위치	서버 쪽에 치중	클라이언트 쪽에 사용자용 FB를 둠	각 서버들에 독립된 FB가 있음	각 클라이언트에 사용자별 FB가 위치
서비스영역	서버에서 주로 많은 일을 처리해야 되는 경우 서버에서 정보를 집중적으로 모으고자 할 경우 전문가시스템 서버의 지식에 대한 보안이 요구되는 경우 대중을 대상으로 서비스를 하고자 할 경우	클라이언트에서 주로 많은 일을 처리해야 되는 경우 사용자 개인의 사적 정보의 유출을 막아야 하는 경우 클라이언트 정보 만으로 추론이 가능한 경우 특정 고객을 대상으로 하는 서비스	여러 가지 문제들(여러 서버에 지식이 분산된)이 종합적으로 고려되어야 하는 복잡한 서비스 다른 전문가시스템들을 이용해서 새로운 부가서비스를 하고자 할 경우	사용자를 그룹핑할수 있는 서비스의 경우 여러 사용자들의 개인정보의 활용이 많은 비중을 차지하는 서비스
적용 가능 예	기상정보, 문헌정보, 포털 서비스	개인의 재무관리, 개인기호(의식주, 레저)	주식투자, 국가안보, 물가예측	전문동호인, 주식투자가, 외환딜러

Model과 클라이언트 쪽에 많은 비중을 두는 Client Oriented Model로 나눌 수 있으며, 지식 및 추론 공유 정도에 따라 지식과 추론에 대한 공유가 전혀 이루어 지지 않는 No Sharing, 서버들 간에 지식 및 추론의 공유가 이루어지는 경우인 Server Sharing, 클라이언트 간의 공유가 이루어 지는 경

<표 3> 주요 웹 기반 전문가시스템 사례

제목	년도	목적	추론 방식	추론엔진 위치	KB의 위치	FB의 위치	사용 기술	모델 유형	모델유형 판단
CAL-ES (Tsai and Tseng, 2002)	2002	학생의 교육전략 방향 제공	Rule 기반(forward chaining)	서버	서버	서버	XML, Ontology, ASP	Type 1 : Conventional Server Oriented Model	클라이언트에서도 데이터처리하나 서버 지향적 모델의 특성 가짐
Fish-Expert (Lia et al., 2002)	2002	어류질병 진단	Rule 기반	서버	서버	서버	Java script, ASP	Type 1 : Conventional Server Oriented Model	전형적인 서버 지향적 모델의 특성 가짐
Code Tutor (Simic and Devedzic, 2003)	2003	웹 상에서의 교육	Rule 기반 (CLIPS)	서버	서버	서버	CLIPS, XML, JAVA, GUI, database, JESS	Type 1 : Conventional Server Oriented Model	전형적인 서버 지향적 모델의 특성 가짐
T-VET (Duana et al., 2003)	2003	어류질병 진단의 서브시스템	Rule 기반	서버	서버	서버, 클라이언트	ASP	Type 7 : Server Oriented Client-Server Sharing Model	서버측에 전문가시스템들이 연동되며, 클라이언트들은 자신들의 정보를 공유함
Multi Agent Diagnosis (Shaalana et al., 2004)	2004	농작물 질병 치료	Rule 기반	서버	서버	클라이언트	Java, KQML	Type 4 : Client Oriented Server Sharing Model	데이터를 클라이언트가 처리하고 서버의 에이전트들이 각 분야의 진단을 맡음
Pig-vet (Zetian, 2005)	2005	돼지질병 진단	Rule 기반	서버	서버	서버	ASP, ADO	Type 1 : Conventional Server Oriented Model	전형적인 서버 지향적 모델의 특성 가짐
WebStra (Shuliang Li, 2005)	2005	마케팅전략 계획 지원시스템	Rule 기반	서버	서버	서버, 클라이언트	HTTP Server	Type 5 : Server Oriented Client Sharing Model	서버지향적이며 클라이언트들간의 정보의 공유 가능
WeBIS (Kim et al., 2005)	2005	재무 컨설팅 시스템	Rule 기반 (Hyperlink)	클라이언트, 서버	서버	클라이언트	Hyperlink, CGI, HTTP Server, JSP,ASP,MIME	Type 2 : Conventional Client Oriented Model	클라이언트 측에서 추론이 실행됨
HYDES (Song et al., 2006)	2006	수력터빈 결함진단 시스템	Rule 기반	서버	서버	서버	JSP, JAVABean, GUI	Type 1 : Conventional Server Oriented Model	전형적인 서버 지향적 모델적 특성 가짐
CDMS (Huang and chen, 2007)	2007	사람의 소화기 진단, 치료 시스템	Rule 기반 (JESS)	서버	서버	서버	Java, XML, Ontology	Type 1 : Conventional Server Oriented Model	전형적인 서버 지향적 모델의 특성 가짐

우인 Client Sharing, 클라이언트, 서버 모두에서 공유가 이루어 지는 경우인 Client-Server Sharing의 네 가지로 분류할 수 있다. 본 논문에서는 이들의 조합을 통해 총 8가지의 유형을 도출하였다. 또한 각 모델 별로 유형별 구조와 장단점, 인터넷 상에서의 지식베이스, Fact Base, 추론엔진 등의 위치 문제와 사용기술, 고려사항, 서비스 유형들을 구분, 분석하였다. 그리고, 본 프레임워크를 바탕으로 2002년~2007년 사이의 Expert Systems with Applications 논문지에 게재되었던 주요 웹 기반 전문가 시스템 10개를 살펴 보았다. 이를 통해서 Type1 형태의 기본적인 웹 전문가시스템의 구조가 아직도 많이 활용되고 있지만 차츰 지식 및 추론을 분산하고 공유하여 처리하는 시스템들도 개발되고 있음을 좀 더 구조화된 시각에서 파악할 수 있었다.

본 논문을 통해서 제시된 프레임워크는 향후 새로운 환경에 좀더 능동적으로 대처할 수 있는 효율적인 다양한 전문가시스템 개발에 기준이 되기를 기대한다. 향후 연구과제는 이러한 각 웹 기반전문가시스템 유형별 효율성에 관한 연구와 각 모델별로 고려사항들에 대한 합리적인 해결방안의 연구와 관련된 실증연구 등으로 사료된다. 또한 KB 지식의 표현방법(규칙, frame, Constraint 등)과 추론방법(RBS, NN, CBS, CSP 등)(이재규 외, 1998)의 차이에 의한 시스템 유형모델의 차이 가능성도 향후 연구 과제이다. 앞으로 지능화된 유비쿼터스 미래사회에서 참여와 공유의 Web2.0 정신(Hammersley, 2006)에 입각한 Type 7, Type 8과 같은 지식과 추론의 공유가 자유롭게 활용 가능한 전문가시스템의 개발이 기대된다.

참고문헌

[1] 송용욱, 김우주, 홍준석, "지식분석도를 이용한 지식기반 웹 사이트 자동 생성 도구의 개발", 경영정보학연구, Vol.13, No.1(2003).

[2] 이재규 외, 전문가시스템 원리와 개발, 법영사(1998).

[3] 임규건 외, "E-비즈니스 경영", 이프레스(2005).

[4] 임규건, 강주영, 이재규, "Web 기반 전문가 시스템의 구조 분석", 한국전문가시스템학회 추계학술대회(1997), 63~73.

[5] Behrouz H.Far and Zenya Koono, "EX-W-Pert System : A Web-Based Distributed Expert System for Groupware Design," Expert System With Applications, Vol.11, No.4(1996), 475~480.

[6] Cjang-Jiun Tsai, S.S. Tseng, "Building a CAL Expert System based upon Two-phase Knowledge Acquisition", Expert Systems with Applications, Vol.22(2002), 235~248.

[7] Daoliang Lia, Zetian Fua, Yanqing Duanb, "Fish-Expert : a web-based expert system for fish disease diagnosis", Expert Systems with Applications, Vol.23(2002), 311~320.

[8] Fu Zetian, Xu Feng, Zhou Yun, Zhang Xiao-Shuan, "Pig-vet : a web-based expert system for pig disease diagnosis", Expert Systems with Applications, Vol.29(2005), 93~103

[9] Goran Simic, Vladan Devedzic, "Building an intelligent system using modern Internet technologies", Expert Systems with Applications, Vol.25(2003), 231~246.

[10] Guangxiong Song, Yongyong, Fulei Chu, Yujiong Gu, " HYDES : A Web-based hydro turbine fault diagnosis system", Expert Systems with Applications Vol.34(2006), 764~772.

[11] Hammersley, B., "Web2.0 Report", Tim O'reilly & Associates Inc, (2006).

[12] "Java Expert Systems Tools", Intelligent Software Strategies, Summer, (1997).

[13] Kankanahalli Srinivas, "Java and Beyond : Executable Content", IEEE Micro, June, (1997).

[14] Khaled Shaalana,*, Mona El-Badryb, Ahmed Rafeac, "A multi-agent approach for diagnostic expert systems via the internet", Expert Systems with Applications, Vol.27 (2004), 1~10.

[15] Mu-Jung Huang, Mu-Yen Chen, "Integrated design of the intelligent web-based Chinese medical Diagnostic System (CMDS). Systematic development for digestive health", Expert Systems with Applications, Vol.32(2007), 658~673.

[16] "Selectica : Java-based Configuration for Internet and Electronic Commerce Applications", Intelligent Software Strategies, October, (1996).

[17] Jay Liebowitz, "Worldwide Perspectives and Trends in Expert Systems", AAAI, summer(1997), 115~119.

[18] J.Gosling and H.McGilton, "The Java Language Environment : A White Paper", tech. report, Sun Mirosystems(1995), http://java.sun.com/.

[19] Robert Orfali and Dan Harkey, Client/Server Programming with Java and CORBA, John Wiley & Sons, New York, 1997.

[20] Shuliang Li, "A Web-enabled hybrid approach to strategic marketing planning: Group Delphi + Web-based expert system", Expert Systems with Applications 29 (2005), 393~400.

[21] "The intelligent Software development tools market : part1", Intelligent Software Strategies, Feb, Vol. XII, No.2(1996).

[22] "The intelligent Software development tools market : part2", Intelligent Software Strategies, March. Vol. XII, No.3(1996).

[23] Wooju Kim, Yong U. Song, June S. Hong, "Web enabled expert systems using hyperlink-based inference", Expert Systems with Applications, Vol.28(2005), 79~91

[24] Yanqing Duana, Zetian Fub, Daoliang Lib, "Toward developing and using Web-based tele-diagnosis in aquaculture", Expert Systems with Applications, Vol.25(2003), 247~254

Abstract

Classification of the Architectures of Web based Expert Systems

Gyoo Gun Lim*

According to the expansion of the Internet use and the utilization of e-business, there are an increasing number of studies of intelligent-based systems for the preparation of ubiquitous environment. In addition, expert systems have been developed from Stand Alone types to web-based Client-Server types, which are now used in various Internet environments. In this paper, we investigated the environment of development for web-based expert systems, we classified and analyzed them according to type, and suggested general typical models of web-based expert systems and their architectures. We classified the web-based expert systems with two perspectives. First, we classified them into the Server Oriented model and Client Oriented model based on the Load Balancing aspect between client and server. Second, based on the degree of knowledge and inference-sharing, we classified them into the No Sharing model, Server Sharing model, Client Sharing model and Client-Server Sharing model. By combining them we derived eight types of web-based expert systems. We also analyzed the location problems of Knowledge Bases, Fact Bases, and Inference Engines on the Internet, and analyzed the pros & cons, the technologies, the considerations, and the service types for each model. With the framework proposed from this study, we can develop more efficient expert systems in future environments.

Key Words : Web-Based Expert System, Expert System, Model, Architecture, Framework, Classification

* School of Business, Hanyang University