

Web 기반 전문가시스템의 구조 분석 (Architectures of Web based Expert Systems)

임규건, 강주영, 이재규

한국과학기술원 테크노경영대학원

E-mail: gglim@iis.kaist.ac.kr, jykgang@msd.kaist.ac.kr, jklee@msd.kaist.ac.kr

요 약

인터넷의 대중화와 전자상거래의 실현을 눈앞에 둔 상황에서 기존의 AI 기술들의 돌파구로서 인터넷환경에서의 적용과 적용이 가속화되고 있다. 전문가시스템 분야도 기존의 Stand Alone 방식에서 탈피 Clie-Ser-Ver 방식으로 발전되고 있으며, 인터넷 환경으로의 적응을 해 나가고 있는 상황이다. 본 논문에서는 인터넷의 Web 환경에서의 전문가시스템 개발현황들을 살펴 보고 Web 기반에서는 어떠한 방법으로 전문가시스템을 설계, 개발 할 것인가에 대한 방향제시를 위하여, Web 기반 전문가시스템들을 유형별로 분류하고, 분석하여 이를 통해 Web 기반 전문가시스템의 일반적인 유형별 구조를 제시하고 모델을 제시하고자 한다. 또한 각 모델별로 인터넷상에서의 Knowledge Base, Fact Base, Inference Engine 등의 위치 문제와 사용기술, 서비스 유형들을 구분, 분석하고자 한다. 이러한 작업을 통해서 새로운 환경에 좀더 대처할 수 있는 효율적인 전문가시스템 개발의 틀을 마련하고자 한다.

1. 들어가는 말

ARPA 프로젝트에서 시작한 인터넷은 연구중심의 활용에서 이제 전세계 160여개국 2천만대이상의 호스트 컴퓨터가 연결된 거대 네트워크로 발전했으며 그로 인해서 인터넷의 대중화와 상업화가 가속화 되고 있다. 최근 들어서는 인터넷상에서의 전자상거래에 대한 관심이 고조되고 있는 추세이며, 이로 인해서 기존의 시스템들은 인터넷환경에서 다시 한번 자리매김을 해나가고 있는 실정이다.

전문가시스템은 지금까지 Engineering-manufacturing, Business, Medicine, Environment-Energy, Agriculture, Telecommunications, Government, Law, Transportation, Finance, Production Management, General Management, Accounting-auditing, Marketing-sales, Electronic Commerce, International Business, Human Resource

Management 등 여러가지 분야에 응용되어 왔다[1]. 앞으로 전문가시스템의 가치는 실제 Domain Specific 한 응용에서 그 진가가 더욱 빛 날 것으로 전망되고 있는데 인터넷과 전자상거래의 활성화로 인해서 전문가시스템의 개발 연구 환경도 잇따라 변화하고 있는 추세이다[2][3].

이에 본 논문에서는 인터넷 환경 특히, 월드 와이드 웹(WWW)을 이용한 전문가시스템의 구조와 발전 방향에 대해서 살펴보고자 한다. 이를 위해서 본론에서 Web 기반 전문가시스템의 일반적인 구조와 장단점을 살펴 보고, 현재 개발된 Web 기반 전문가시스템들을 조사하고 이들의 현황 및 각 구조별, 유형별, 사용기술별 등으로 분류 및 분석한다. 이를 통해 Web 기반 전문가시스템을 구축하기 위해 제기되는 여러가지 설계 문제들을 살펴보고, 전문가시스템의 응용과 Web의 특징을 고려한 Web 기반 전문가시스템의 일반적인 유형별 구조를 제시하고 Load Balancing 측면에서의 Server Oriented/ Client Oriented 구조와 인터넷 분산환경에서의 Distributed Server/ Distributed Client 모델을 제시하고자 한다. 또한 각 모델별로 인터넷상에서의 Knowledge Base, Fact Base, Inference Engine 등의 위치 문제와 장, 단점, 고려사항, 사용기술, 서비스 유형들을 구분, 분석하고자 한다. 결론에서는 이러한 모델링 효과와 향후전망을 하고자 한다.

2. Web 기반 전문가시스템의 개요

2.1 Web 기반 전문가시스템의 등장

초기의 전문가시스템은 Standalone 시스템 방식으로 Diagnostics, Corporate Policy, Business Rules 와 같은 응용분야에서 값진 정보들을 지식으로 표현하고 저장하여 이용할 수 있는 방법을 제시하였다. 그러나, 이러한 전문가시스템 지식은 많은 사용자에게 분배할 수 없었기 때문에 단지 접근 가능한 몇몇 사용자들에 의해 이용가능 하였다. 이와 같은 지식 분배 문제를 해결하기 위해, Client-

Server 구조를 채택하였으나 대부분 각 응용별로 특정하였고 Client의 비중이 큰 Fat-Client의 구조로 인해 비용이 많이 들었기 때문에 과거의 Client-Server 전문가시스템은 커다란 실효를 거두지 못했다[4].

이와 같은 상황에서 Internet과 WWW의 등장 및 보급으로 Web Server에 전문가시스템을 배치하고 인터넷을 통해 접근 가능하며 플랫폼에 독립적인 Web Browser를 가진 사람은 누구나 이용할 수 있는 구조를 통해 전문가시스템을 이용할 수 있는 기회를 더욱 확장시켰다.

2.2 Web 기반 전문가시스템의 개발 방안

Web 기반 전문가시스템은 기본적으로 Web Server와 Http 프로토콜을 이용하여 구현된다. 이와 같은 기본적인 Web 기술을 이용하여 전문가시스템을 개발할 때, 크게 다음과 같은 세가지 방안으로 나눌 수 있다.

- CGI를 이용하는 방법 : Form 기반 Front End를 가지고 Database와 Knowledge Server에 접근하는 방법으로 간단한 사용자 인터페이스를 제공한다. 그러나, Form Filling과 프로그램 출력으로서의 Web Browser 이용 외에는 사용자와의 상호작용을 허용하지 않는다.
- Java를 이용하는 방법 : Sun Microsystem의 Java 프로그래밍 언어는 Knowledge Server와의 인터페이스를 구현하는데 강력한 기초를 제공해 준다. 개발자가 HTML에 Java로 쓰여진 프로그램을 포함시키면 Web Browser는 이들 프로그램을 인터넷을 통해 다운로드 받아 지역적으로 수행시킨다. 이때, Java 프로그램은 Knowledge Server와 지속적인 연결을 유지시켜주기 때문에 사용자는 Server와 계속해서 상호작용을 할 수 있다[16].
- External Helper 프로그램을 이용하는 방법: C나 VC++, Java 등을 이용하여 Browser의 Helper 프로그램으로 독립적인 Client 프로그램이 Web과 연동해서 작동하게 할 수 있다. Plug In 형태로도 가능하다.

이와 같은 Web을 기반으로 한 전문가시스템의 일반적인 모형은 본문 4.1의 [그림 1]과 같다.

2.3 Web 기반 전문가시스템의 장단점

위에서 제시한 Web 기반 전문가시스템의 장단점을 분석하면 다음의 [표 1]과 같다. 보느바와 같이 주로 단점 보다는 장점이 많음을

알 수 있다[9].

[표 1] Web 기반 전문가시스템의 장단점

장점	<input type="checkbox"/> 여러명의 사용자가 동시에 사용가능하다. <input type="checkbox"/> 24시간 가용하다. <input type="checkbox"/> 유연성을 제공한다. <input type="checkbox"/> 사용하기 쉽고 편리한 Browser를 사용한다. <input type="checkbox"/> HTML과 같은 Common Language로 인터페이스 제작이 용이하다. <input type="checkbox"/> 멀티미디어를 이용한 다양한 사용자 화면의 구성이 가능하다. <input type="checkbox"/> 데이터 수준에서 사용자가 응용 프로그램에 접근 가능하다. <input type="checkbox"/> 동적이다. <input type="checkbox"/> 주식과 같은 실시간 데이터를 보여줄 수 있다. <input type="checkbox"/> 공용 전문가시스템이 가능하다. <input type="checkbox"/> 전문가시스템 분배 문제가 간단해진다. <input type="checkbox"/> 플랫폼에 독립적으로 다양한 플랫폼 지원이 가능하다.
단점	<input type="checkbox"/> 신뢰성 있고 빠른 네트워크의 연결이 어렵다. <input type="checkbox"/> 인터넷과 Web의 개방성으로 사적인 정보에 대한 보안 문제가 있다. <input type="checkbox"/> 인터넷 상의 요금 체제의 미정립으로 인한 요금 부과 문제가 있다.

2.4 Web 기반 전문가시스템의 현황

현재까지 개발되어진 Web 환경에서의 전문가시스템들을 응용 유형별로 분류하여 응용 및 이용기관, 개발 기술, 특징등을 표로 나타내면 아래와 같다.

- Intelligent E-Mail Interpretation and Classification

정기적인 사용자들에게 보낸 메일들이 되돌아오는 경우 이를 해결하고, 메일로 들어오는 질문들을 판별하여 자동적으로 응답을 보내주는 Email 전문가시스템이다.

응용 및 이용기관	개발기술	특징
EZReader Chase/Manhattan Bank/BrightWare	Rule-based parsing expertsystem, CBR	Email 해석과 라우팅을 위한 embedded rule-base와 CBR 시스템으로 Web으로 부터 온 고객과 고용인의 email을 전에 받았던 것들과 유사성을 통해 분류하여 적절한 응답과 충고를 자동적으로 보내준다.
Internet Financial Service/Mail White House/ MIT AI Lab	Intelligent email reader	마케팅과 판매에 도움이 되기 위해 Web을 통해 자유로운 형식의 email을 받아서 처리한다.

- Smart Advisor for Marketing Products and Services, Training

Web Browser를 통해 고객과 고용인에게 좀 더 빠르고 효율적인 advisory system을 제공해준다.

응용 및 이용기관	개발기술	특징
Internet Loan Advisor	KBS loan advisor	사용자에게 대화식의

Security Technologies/Security First Network Bank	First		Web 을 통해 그들이 자신의 상황에 따라 가장 적절한 loan 을 선택하여 안전한 거래가 이루어질 수 있도록 제공해준다.
Internet Site Guide		Prolog-based WebServer 와 Development Tool	webmaster 에게 자신에 맞는 intelligent web site 를 개발하도록 충고를 제공한다. 이 때 사용자의 질문에 따라 article, demo, newsgroup, tool, 다른 사이트들을 포함해서 어떻게 Web 을 구축할 수 있는지 충고를 제공한다.
Military Training System		Rule-based Development/Server Tool	사용자의 모델별로 데이터베이스를 통해 운영되며 Web 을 통해 모델별로 훈련 양식이 제공된다.
NASA Personnel Security Processing Expert System NASA		Rule-based Expert System with Web Interface	Web 을 이용하여 시공무원이나 국가 보안 기관에 근무하는 사람들에게 적절한 수준의 인사 정보 제공 수준을 결정 및 제공한다.
Personalized Web Service Open Sesame	Entertainment	Neural Network/ KBS agent technology	Web 을 방문한 고객의 선호도를 learning 하여 그에 따라 책, 비디오, 음악에 관한 정보를 제공한다.

- Online Configuration Systems
Constraint Programming 기술을 이용하여 Online Configuration System 을 제공한다.

응용 및 이용기관	개발 기술	특징
PC Buying and Help System Monoral Inc.	KBS Web product configurator 와 help desk	홈 PC 를 구입하고 구성하는 고객을 위한 self 서비스 비즈니스 응용 프로그램으로 고객의 문제점을 해결하기 위해 knowledge base 를 이용한다.

- Help desk for Technical Support.
가장 많이 쓰이는 Web-based Knowledge Base 이다.

응용 및 이용기관	개발 기술	특징
Customer Support Borderbund Software	Web-based help desk with Inference's CBR and CasePoint WebServer Tools	고객이 자유로운 형식으로 자신의 시스템에 발생한 문제를 적어서 거기에 대한 도움말을 얻을 수 있다.
FAQ Finder University of Chicago	CBR with natural language interface	CBR engine 위에 natural language interface 툴을 제공하여 사용자의 수천개의 FAQ 리스트에서 자신이 원하는 질문에 가장 가까운 걸 찾아 주도록 제공한다.
Netscape Help System Netscape	Intellisystem 의 SmartSite 를 이용하여 사용자의 질문에 대해 WebServer 에 배치	Web Browser 을 이용하는 온라인 고객에 대해 기술적인 도움을 help system 을 통해 제공한다.

	되어 있는 intelligent 하고 자동적으로 연결되는 hmt 파일을 제공 하는 기술	
UltraNet Self Service Support	KBS Web Help desk	UltraNet 의 고객에게 공통의 문제들을 스스로 해결할 수 있도록 Web 을 통해 정보를 제공한다.

3. Web 기반 전문가시스템의 분석

Web 기반 전문가시스템의 구현은 기존의 Database 시스템을 모델로한 Web 기반 Client-Server 구조로 구현하는 것과 차이가 있다. 그 이유는 다음과 같다.

- Database 시스템은 사용자가 Web Browser 상에서 질의문을 통해 Web Server 에 연동되어 있는 Database 에 요청하고 질의결과를 다시 Web Browser 를 통해 화면에 보여준다. 즉, Client 는 아무런 연산이 필요없고 단지 서비스를 요청하고 그 결과를 보여주는 역할만 담당한다. 그러나, 전문가시스템은 이와 달리 추론과정을 요구하기 때문에 중간 중간 사용자와의 지속적인 상호작용을 요구한다. 즉, Server 와 연결이 한번만 연결되고 끊기는 것이 아닌 계속적으로 연결 상태에서 Fact 가 교환되고 추론이 일어나는 것이다. 따라서, Client 쪽에서 Server 와 연결을 유지할 수 있고, 중간에 발생하는 Fact 를 저장하며 필요한 경우 계산 능력을 수행할 수 있어야 한다.
- 전문가시스템은 많은 양의 계산능력을 요구한다. 이와 같은 상황에서 여러 명의 사용자가 동시에 Server 에 접속해서 많은 계산을 수행한다면 Server 의 부담이 증가하여 Server 의 능력이 현저하게 저하될 수 있다. 따라서, Client 도 계산능력을 가질 수 있는 Client-Executable 구조가 요구된다. 그러나, 이러한 상황에서는 Client 프로그램을 다운로드 받아야 하는 부담이 생기므로 이를 고려해야 한다.

그렇다면 이러한 특성을 감안한 Web 기반 전문가시스템을 개발하고자 할 때 발생하는 문제점과 고려해야 할 사항을 나타내면 다음과 같다.

- User Interface 는 단순 HTML 문서로 만들어 줄 것인가 아니면 VC++

아니 Java 등으로 만든 External Helper 프로그램으로 할 것인가?

- Multi-User 로 인한 Server 의 부담을 어떻게 분산하는가?
- Network Traffic 은 어떻게 맞출 것인가?
- Server 과 Client 사이의 Load Balancing 은 어떻게 할 것인가?
- KB 의 위치는 어떻게 할 것인가?
- KB 는 공용으로 할 것인가, 사유로 할 것인가?
- FB 의 위치는 어떻게 할 것인가?
- 사용자끼리의 관련성은 어떻게 둘 것인가?
- Inference Engine 의 위치는 어떻게 할 것인가?
- 어떠한 Programming(C/C++, Perl, JAVA, CGI, CORBA,MIME)기술을 사용할 것인가?
- 보안에 대한 문제는 어떻게 할 것인가?

본 논문에서는 위에서 제시한 여러가지 문제점들과 현재 개발되고 있는 Web 기반 전문가시스템들의 동향을 토대로 Web 기반 전문가시스템을 체계적으로 분석하는 틀을 마련해 보고자 한다. 아래에 설계 및 구현시 고려되어야 할 사항들을 카테고리 별로 분류 및 설명하겠다.

3.1 Client-Server 구조

통합된 전문가시스템을 통신 채널을 가진 Client 와 Server 구조로 나눌 때 통신 속도 측면을 고려해야 한다. 왜냐하면, 인터넷과 같은 네트워크 환경에서는 데이터 전송율이 매우 넓은 범위에서 변하고 유한하고 예측 불가능한 통신 Bandwidth 를 가지기 때문이다. 따라서, 사용자에게 허용가능한 응답 시간을 제공하고 중요한 응용의 경우 신뢰성있는 응답 시간을 보장할 수 있는 Client-Server 구조를 가져야 한다. 이러한 측면에서 Server-Oriented 구조와 Client-Executable 구조 두가지를 통해 어떻게 위의 문제점을 극복하면서 효율적인 Client-Server Web 전문가시스템을 구성할 지를 생각해 봐야한다[8].

우선, Knowledge Server 와 사용자 인터페이스간에 통신을 최소화하는 측면에서는 Client-Executable 구조가 적합하다. 즉, 빠른 응답시간을 위해서 사용자 인터페이스는 마우스 조작과 같은 계산 집중한 연산을 수행해야 하거나 모든 계산은 Server 가 아닌 사용자 Front End 에서 이루어 지도록 해야 한다. Web 상에서 Java 프로그램을 이용하면 이와 같은 기능을 수행하면서도 플랫폼에 독립적인

Client 를 구성할 수 있다.

그러나, 사용자쪽에서는 복잡하고 커다란 Front End 를 다운로드 해야 하는데 속도가 느린 네트워크 연결에서는 사용하기 힘들다. 또한, Front End 를 자주 업데이트 해야 하는 경우 통신으로 인한 부하보다 다운로드로 인한 부하가 더 크게 된다. 또한 커다란 Knowledge Base 와 Database 를 이용하는 경우는 Server 에 데이터를 집중하고 데이터의 분배를 피하는 Server-Oriented 구조를 취함으로써 개발자는 상대적으로 작은 Front End 를 설계하여 다운로드시 부하를 감소시킬 수 있다.

Client 와 Server 간의 비중을 결정하는 것 외에도 적당한 통신 프로토콜을 쓰는 문제도 고려하여 시스템 특성에 따라 일반적인 프로토콜에 그 응용에 맞는 특성을 부여해야 한다. 그리고, 사용자 인터페이스와 Knowledge Server 가 만일 객체 지향이라면, 공유 객체를 지원하는 CORBA 와 같은 객체 통신 모델이 표준 프로토콜로 이용될 수 있을 것이다.

3.2 Web 기반 Knowledge Base 구성 및 위치

Web 기반 KBS(Knowledge Based System)는 Web 사이트 방문객과 상호작용하는데 적합하다. 왜냐하면 Server 의 Knowledge Base 를 Client 는 다이얼로그를 통해 접속하고 필요한 feedback 을 주어 자신에게 적합한 추천을 제공받을 수 있기 때문이다. Web 을 이용하여 Knowledge Based System 을 구성할 때 특성은 다음과 같다[9].

- Product/Service 지식은 Knowledge Base 에 있는 if-then rule 과 fact 로서 표현될 수 있다.
- System/User 다이얼로그는 사용자 입력에 기반하여 계속적으로 세분화될 수 있다. 즉, 사용자의 관심에 적합한 질문만 요청된다.
- 정보를 모으고 나타내는 forms, documents, graphics 는 동적으로 생성된다.
- 사용자의 질문에 대한 대답은 인터넷이나 회사 인트라넷상에서 유용한 다른 정보와의 직접적인 연결을 포함할 수 있다.
- 플랫폼에 독립적이다.
- 느리고 안전하지 않아서 실시간 데이터와 같은 경우 적합하지 않을 수 있다.

위와 같은 특성을 고려할 때 Knowledge Base 의 위치가 Server 에 있느냐 Client 에 함께 들어 있느냐가 데이터 응답시간에 미치는 영향을 결정한다. 첫째는 Knowledge Base 가 Server 에만 위치한 경우를 볼 수 있는데, Client 와는 독립적으로 일관된 Knowledge Base

을 유지할 수 있고, 변경과 제어가 쉽다. 그러나, Client와 통신하는데 드는 부담이 크고, Client에 따른 사적인 Knowledge Base을 운용하기에 부적합하다. 또한, 모든 Client가 Knowledge Base을 공유하므로, 보안 문제나 개인 정보 보호문제가 발생할 수 있다. 둘째로, Java 등을 이용하여 Client에 Knowledge Base을 두는 방안으로 통신에 드는 부하가 많이 감소한다. 또한, Knowledge Base를 개인 별로 유지할 수 있기 때문에, 보안 문제를 어느 정도 해결할 수 있다. 그러나, 중앙 집중적인 제어는 어려워진다.

3.3 Inference Engine의 위치

전문가시스템에서 원하는 결과를 얻기 위해 Inference Engine을 통해 추론을 한다. 이때, Inference Engine이 Server에 위치하느냐 Client에 위치하느냐에 따라 Client-Server 구조로 인해 생기는 부하를 감소시킬 수 있다. 만일 Inference Engine이 Server에 위치하는 경우는 사용자와의 많은 상호작용으로 인한 부담이 증가하여 통신 부하가 증가된다. 이와 반대로 Java를 이용하여 Client가 이용하는 Knowledge Base와 함께 Client에 Inference Engine을 위치시키는 경우를 고려할 수 있다. 자바를 이용하면 Java GUI를 만들기 위한 Interface Tool과 결합되어 매우 compact, portable한 Inference Engine을 제공한다. Java를 이용한 Inference Engine은 그 Engine이 Server에서 수행될지 다운로드 받을지 개발자에게 결정하는 걸 허용할 수 있고 다양한 구성을 가능하게 한다. 예를 들어 몇몇 Rule Base는 Server에 두고, Client processing 위해 사용자의 데스크탑에 다운로드되는 Smart Java Applet(Smartlet)의 형태로 Rule들이 Java Applet에 배치되는 것도 가능하다. 그러나, 이 경우는 Knowledge Base의 양이 크거나, 사용자가 많은 Knowledge를 요구하는 경우는 Client가 다운로드 받고 유지하는 부담이 크다[16][17].

3.3 Fact Base History의 유지

Http는 stateless protocol이기 때문에 추론 과정에서 일어나는 Fact들의 History를 유지하기 어렵다. 따라서, Fact Base History를 유지하기 위해서는 다음과 추가적인 방법이 요구된다.

- Hidden Form in HTML : HTML안에서 매 추론 과정마다 발생하는 Fact Base History를 Hidden Form으로 계속해서 추가해서 유지하는 방법이다. 즉, 추론이 길어질 때 마다 HTML의 크기가 커지고 서버와의 통신 부담이 커지게 된다.

- Cookie : 추론 중간에 발생하는 Fact Base History를 cookie를 이용하여 저장해 두는 방법이다. 그러나, cookie는 일반 text파일의 형태로서 그 내용이 외부로 유출되는 보안 문제가 발생할 수 있고 fact의 내용이 커지게 되면 용량 부족 문제가 발생한다.
- Java applet : Java를 통해 Knowledge Base와의 세션을 계속 유지함으로써 추론과정에 일어나는 모든 Fact를 Java의 데이터 구조를 통해 Client의 메모리에 유지함으로써 빠른 통신 속도를 보장하며 효율적인 추론을 할 수 있다.

3.4 Knowledge Base의 Protection과 Sharing

인터넷과 Web의 개방성은 Knowledge Base를 공유하는 측면에서는 매우 유리하나 반대로 사적인 정보나 중요한 정보를 보호하는데 어려움을 준다. 따라서 사용자가 Knowledge Base를 인터넷을 통해 사용할 때 암호화된 통신을 이용하여 다른 불법적인 사용자에게 정보를 누출시키지 않도록 하고, 또한 사용자 암호를 부여하여 사용자 별로 사적인 정보의 이용을 제어한다. 또한, Java를 이용하여 악용 연산을 미연에 방지하는 보안 함수를 이용하여 이러한 문제점을 해결할 수 있다. 그러나, 결국 사적인 정보가 공용 Server에 존재하게 되면 사용자는 자신의 정보를 Server 측이 안전하게 보호하고 있을 것이라는 신뢰를 하고 있다는 가정하에 시스템을 사용해야 한다[9].

3.5 Server의 계산과 Network Traffic의 감소

Server의 계산량을 감소시키고 네트워크 통신량을 감소시키기 위해 사용자로부터 받는 입력값을 미리 JavaScript를 이용하여 Client 쪽에서 체크 가능하게 한다. 또한, Java Applet을 이용하여 Client에 계산 능력을 부여하고 Java Smartlet을 이용하여 Server가 Update될 때 자동적으로 Client도 Update될 수 있도록 다운로드 받는다.

3.6 개발 시 선택되는 언어(C, Java, JavaScript 등)

Web 기반 전문가시스템을 가능하게 하기 위해서는 CGI를 구현할 수 있는 언어들과 HTML, Java, JavaScript가 필요하다. CGI를 위한 언어의 선택시 속도를 먼저 고려하여 C나 C++이 적합할 것이다. 그러나, 다양한 플랫폼을 가지고 있는 Client쪽 프로그램을 위해서는 HTML, HTML을 보완해주는 JavaScript, 그리고 아래의 특성을 갖는 Java가 적합하다

[6][8][10].

- 네트워크를 통해 실행되는 32 비트 Multithreaded language 이다.
- Write-once/Deploy-anywhere : Java 응용에게 JVM(Java Virtual Machine)을 통한 platform-independent 제공하여 Client는 단지 Java가 지원되는 Browser와 네트워크 연결만 가지고 있으면 된다.
- Sound와 Graphics를 위해 Multimedia 지원을 제공하는 Object-Oriented language 이다.
- 사용자 입력에 따라 전송 중에 변화 가능한 동적인 GUI에 대한 좋은 선택을 할 수 있도록 한다.
- Server과 Client 사이에 사용자의 데스크탑에 작은 Java Applet을 다운로드받게 함으로써 분산 응용 처리 능력이 제공된다.
- 소프트웨어가 즉시 분배되고 갱신될 수 있다.
- JDBC(Java Database Connectivity)라는 새롭고 더 강력한 데이터 접근 기법이 있어서, Java front-end를 가지고 고용자, 고객은 Database, DSS, Electronic Commerce Sales를 단지 네트워크 연결과 Web Browser를 통해 어디서든지 이용가능하다.
- 안전하고 신뢰성있는 응용프로그램을 작성할 수 있도록 보장한다.
- 언어자체의 구현적 측면에서 버그가 없는 안정성을 제공한다.
- 속도, 응답시간, 메모리 요구량과 같은 성능적인 측면이 우수하다.

4. Web 기반 전문가시스템의 4 가지 모델

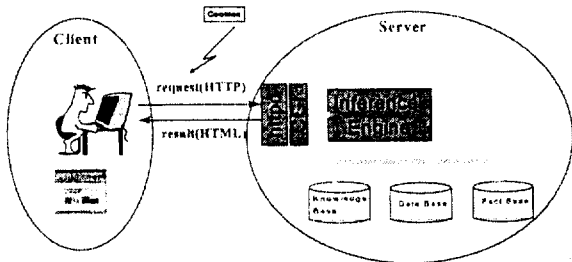
위와 같이 Web 기반 전문가시스템을 개발할 때 고려해야 할 사항들을 종합적으로 살펴 보면 크게 두가지의 범주의 문제로 압축할 수 있다. 하나는 Server와 Client간의 Load Balancing의 문제이다. 즉, Server쪽에 Inference Engine, KB, FB 등을 집중적으로 둘 것인가? 아니면 Client쪽에 이러한 기능들을 어느정도 둘 것인가의 문제이다. 이에 따라서 개발시 고려할 사항과 사용될 기술, 그리고 서비스유형이 나누어 질 수 있다. 두번째는 인터넷의 가장 큰 장점인 분산환경을 어떻게 접목시키느냐에 따른 구분이다. Server들을 분산시켜 여러개의 전문가시스템이 연동되어서 문제의 해답을 풀어주는 구조와 Client간에도 서로 정보를 교환할 수 있는 환경으로 만들어 주느냐에 따른 구분이다. 우리는 이 논문에서

이러한 사항들을 고려해서 4 가지의 Web 기반 전문가시스템의 일반모델을 디자인하였다. 첫 번째 기준에 의해 Server Oriented, Client Oriented의 두가지 모델을, 두번째에 의해 Distributed Server, Distributed Client의 두가지 모델을, 합해서 4 가지 모델을 제시하였다 아래에 각 모델별 구조와 장단점, 고려사항, 사용기술, 서비스유형에 대해서 제시하겠다.

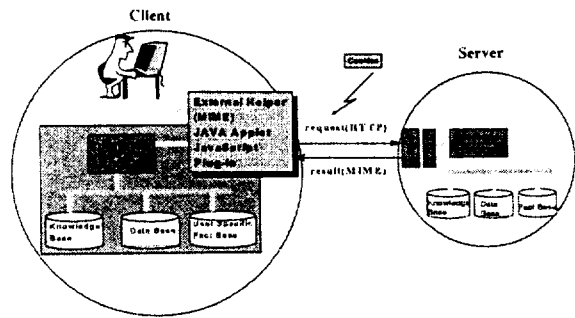
4.1 Server Oriented Model

Server 쪽에 Inference Engine, KB, FB 등이 위치하여 대부분의 작업을 수행하고 단순히 Client 쪽은 Browser 기능을 위주로하는 경우이다. 즉, Thin Client-Fat Server 형의 전문가시스템인 경우이다.

이런 경우에는 [그림 1]과 같은 경우로서 Server 쪽의 전문가시스템에서는 Http 서비스를 가능하도록 Web Server의 역할을 해주고, Inference Engine 과의 CGI 등을 통한 인터프리터를 만들어 주고, KB, FB 등을 효율적으로 유지, 관리 해야 한다. 즉, 모든 로드가 Server 쪽으로 치우치는 경우이다. 이 모델은 Web을 통한 전문가시스템의 일반적인 유형이라고 볼 수 있다. Client 쪽에서는 기존의 Netscape나 Explorer 같은 단순히 Browsing 기능을 하는 Web Browser를 가지고도 충분히 전문가시스템을 사용할 수 있다. 이러한 경우의 장점은 Knowledge, Fact 등을 Server에서 집중적으로 관리할 수 있으며 개발 부담도 작고 단순 Client 프로그램을 사용함으로 사용자의 부담을 최소화할 수 있다는 것이다. 단점으로는 Server에 과중한 로드가 걸리며 많은 수행들이 Server에서 이루어지므로 상대적으로 사용자의 요구에 대한 처리속도가 늦어 질 수 있다는 것이다. 이 경우 고려해야할 사항은 사용량 또는 시간, 정보에 따라 과금을 어떻게 하느냐의 문제와 각 사용자의 History나 개인용 Fact, Knowledge의 관리, 접속후 시간이 경과된 후에 다시 접속했을때 Cookie 등을 이용해 연속성을 유지 시켜주는 문제들이 고려되어 설계되어야 할 것이다. 사용기술로는 기본적인 Http Server와 CGI 기술 등으로도 충분하며, 쉽게는 기존의 전문가시스템을 CGI를 이용해서 Web과의 인터페이스를 만들어 사용해도 된다. 서비스 영역은 Server에서 주로 많은 작업을 해야하는 경우, Server에서 많은 사용자의 정보를 모으고자 하는 경우 등 주로 대중을 대상으로하는 서비스인 경우와 특별히 보안이 요구되는 전문가시스템의 경우이다.



[그림 1] Server Oriented Model (Web 기반 전문가시스템의 일반적 모형)



[그림 2] Client Oriented Model

4.2 Client Oriented Model

Client 쪽에 Inference Engine, KB, FB 등의 일부분 또는 전부를 위치시켜 주거나 역할의 일정부분을 할당해 주어 Client 쪽에서 많은 부분을 처리하고 Server 쪽은 상대적으로 Coordination 정도의 일을 하는 경우이다. 즉, Fat Client-Thin Server 형의 전문가시스템인 경우이다.

이런 경우에는 [그림 2]와 같이 Server 쪽 전문가시스템에서는 Http 서비스를 이용하여 전문가시스템의 주 기능을 해주거나 단순한 사용 제어 등의 coordination 을 해주고, Client 쪽에서는 Browsing 기능 이외에 주로 사용자용 Fact Base 를 위주로 KB 또는 Inference Engine 의 일부를 Client 쪽에 위치하고 관리하는 기능을 지녀야 한다.

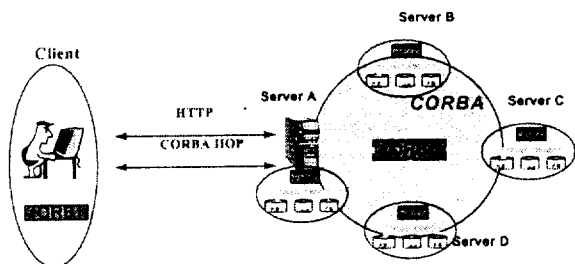
이러한 경우의 장점은 Knowledge, Fact 등을 Client 에서 관리할 수 있으므로 각 사용자가 자신의 기호 또는 취향에 맞는 개인적인 지식을 관리하면서 전문가시스템을 사용할 수 있고, 이를 통해 개인의 사적인 정보의 보호가 가능하며, Server 의 부담을 작게 해주고, 필수적인 데이터 등의 최소한의 데이터가 Server 과 Client 간에 교환되므로 처리속도를 향상시킬 수 있다. 단점으로는 Client 에 로드가 걸릴 수 있다는 것과 구현비용이 든다는 것, 그리고 Server 쪽 전문가시스템에서 다양한 고객의 정보를 입수하는데 어려움이 있다는 것이다. 이 경우 고려해야 할 사항은 과연 Inference Engine, KB, FB 중에서는 어느것을 어느정도 Client 부분에 위치 시킬 것인가에 대한 것과 사용자 개인 지식과 전체 지식과의 Inference 시의 전문가시스템 동작시의 고려문제 등이다. 사용기술로는 단순한 CGI 기술만으로는 이러한 시스템을 구현하기에는 부족하며 Client 에서도 작동 가능한 프로그램이 되어야 하므로 CGI, JAVA, MIME, Plug-In 등의 기술을 사용해서 External Helper Programr 같은 Client 에서 동작 가능한 프로그램을 만들어야 한다. 서비스 영역은 Client 에서 주로 많은 일을 처리해야 되는 경우, 사용자 개개인의 정보의 보호가 필요한 경우, Client 정보 만으로 Inference 가 가능한 경우, 특정 고객을 대상으로 하는 서비스를 개발해야 하는 경우이다.

4.3 Distributed Server Model

Server 들이 인터넷상에서 여러곳에 위치하며 Server 간의 통신과 정보교환이 가능한 분산환경 전문가시스템인 경우이다. 즉 각 Server 들은 독립된 Inference Engine, KB, FB 등을 갖는 전문가시스템들이며 이들은 서로 하나의 객체로서 인식되어 Inference 나 KB, FB 의 이용을 다른 전문가시스템의 결과를 활용할 수 있는 구조이다. 예를 들어 갑이라는 사람이 주식투자자문을 위해 A 라는 전문가시스템을 이용하는 경우 A 주식투자자문 전문가시스템은 인터넷상의 B 기업정보 전문가시스템과 C 은행재무정보 전문가시스템 등을 활용하여 적절한 답을 갑에게 내주는 구조이다.

이런 경우에는 [그림 3]과 같이 Server 들 간에는 분산환경에 적합한 CORBA(Common Object Request Broker Architecture)등을 이용하여 각 Server 들간의 정보교환이 제대로 이루어 질수 있도록 만들어주고 Client 쪽에는 Http 나 CORBA 의 IIOP(Internet Inter-ORB Protocol) 등을 이용해 전문가시스템과 통신을 하게 된다. 이러한 경우의 장점은 분산된 각종 전문가시스템의 정보를 십분 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 있다는 것이다. 단점으로는 분산된 Server 들을 사용하므로 네트워크 트래픽으로 인한 전체적인 처리속도문제와 분산환경에 따른 구현의 어려움을 들 수 있다. 이 경우 고려해야 할 사항은 여러 Server 들간의 과금 문제 등을 어떻게 해결해야 할 것인가와 문제해결을 위해 분산환경에서의 Query Optimization 도 고려해 볼 사항이다. 사용기술로는 핵심적으로 Web, JAVA 를 적절히 이용한 CORBA 가 사용될 수 있다. 또한 데이터의 교환을 효과적으로 하기위해서 MIME 도 필요할 것이다. 서비스 영역은 여러가지 문제들이 종합적으로 고려되어야 하는 복잡한 서비스와 다른 전문가시스템들을 이용해서 새로운 부가서비스를 하

고자 할 경우 등이라고 볼 수 있다. 이러한 모델은 다가오는 인터넷과 전자상거래 환경에서 상당히 유용한 모델이라고 하겠다.



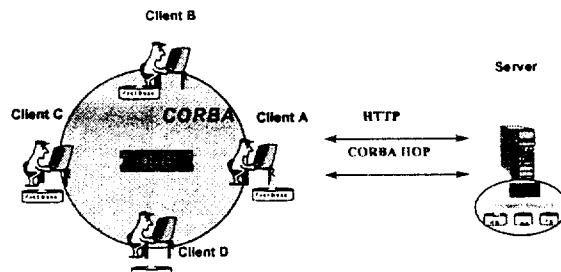
[그림 3] Distributed Server Model

4.4 Distributed Client Model

인터넷상에서 여러 곳에 위치한 Client 들 간에도 서로가 정보를 교환할 수 있는 구조의 전문가시스템인 경우이다. 새로운 개념의 모델로서 각각의 Client 들이 지니고 있는 개인용 Fact Base 등을 Client 간에 서로 이용 가능하도록 하는 모델이다. 예를 들면 인터넷상에서 10 명이 주식투자동호인 클럽을 결성했다고 할때, 각각의 투자자는 개인의 전문가시스템 용 Client 프로그램을 가지고 개인의 투자 기록이나 선호도에 의한 Fact Base와 전문가시스템 Server 을 이용해서 투자자문을 받고, 10 명의 동호인클럽인 끼리는 서로의 Client 프로그램을 이용해서 다른 동호인의 전문지식과 정보를 활용할 수 있는 경우이다.

이런 경우에는 [그림 4]과 같이 Client 간에는 CORBA 등을 이용하여 각 Client 들간의 정보교환이 제대로 이루어 질 수 있도록 만들어주고 Server 쪽은 일반적인 Http 나 CORBA 의 IOP 등을 이용해 전문가시스템과 통신을 하게 된다. 이러한 경우의 장점은 Client 끼리 지식과 정보를 서로 사용할 수 있으므로 Client 간의 CUG(Closed User Group)의 효과가 있다는 것이다. 단점으로는 여러 Client 간의 통신을 처리해 주어야 하므로 서로간의 정보를 주고 받을 수 있는 구조로 만드는 구현의

어려움을 들 수 있다. 이 경우 고려해야 할 사항은 Client 간의 직접적인 정보전달이 어려우므로 매개 Server 을 통해서 어떻게 구현할 것인가 등의 Client 간의 통신과 Server 과의 통신과의 관계정립 등을 들 수 있겠다. 사용기술로는 Client 용 Helper 프로그램을 이용해서 Client 간의 통신하는 것과 Web, JAVA 를 적절히 이용한 CORBA 가 사용될 수 있다. 또한 데이터의 교환을 효과적으로 하기 위해서 MIME 도 필요할 것이다. 서비스 영역은 각부 서별이용, 전문동호인클럽등 사용자를 그룹핑할 수 있는 서비스의 경우와 사용자들의 개인 정보의 활용이 많은 비중을 차지하는 서비스 등이다. 이러한 모델 또한 다양한 사용자와 환경을 지니고 가상사회가 만들어지는 인터넷 시대에 적합한 모델이라고 볼 수 있다.



[그림 4] Distributed Client Model

5. 4 가지 모델의 분석 및 혼합형태 분류

위에서 살펴 본 4 가지 모델에 대해서 장점, 단점, 고려사항, 사용기술, Inference Engine 의 위치, Knowledge Base 의 위치, Fact Base 의 위치, 서비스영역과 그 예를 아래 [표 2]에 보였다. 이렇듯 인터넷 Web 기반 전문가시스템을 개발하고자 할 경우에는 그 서비스 영역에 따라서 각 모델별로 서로 다른 고려사항들과 다른 개발방법을 사용해야 함을 알 수 있다.

[표 2] Web 기반 전문가시스템의 4 가지 모델 비교

	Server-Oriented	Client-Oriented	Distributed Server	Distributed Client
장점	-Fact Knowledge 등을 Server 에서 집중 관리. -사용자 부담 최소화 -요금 부과 등을 한 곳에서 처리가능하다.	-사용자용 Fact, Knowledge 등을 Client 에서 관리. -개인의 사적정보 보호 -Server 부담 최소화 -개인정보의 유출 방지 -처리속도	-분산된 각종 전문가시스템의 정보를 심분 활용 -객체지향기법으로 새로운 전문가시스템의 구현이 용이 -전체적인 모듈화가 가능하여 효율성을 높일 수 있다	-CUG 의 효과 -새로운 형태의 전문가시스템 구현 가능
단점	-Server 의 로드과중 -사용자 요구에 대	-Client 의 로드과중 중요정보의 Server	-네트워크 트래픽으로 인한 전체적인 처	-구현의 어려움

	한 처리속도	관리측면에서의 입수 어려움 -구현 비용	리속도 -구현의 어려움	
고려사항	-사용자 과금 -사용자 연속성 유지	-어느 정도를 Client 부분에 위치 시킬 것인가? -개인 지식과 전체 지식과의 Inference 시의 고려문제	-Server 들간의 과금 문제 -Query Optimization	-Client 간의 통신과 Server 과의 통신과의 관계정립
사용기술	Http Server, CGI, C/C++, Perl	Http Server, CGI, JAVA, MIME, Plug-In, C/C++, Perl	Http Server, CGI, CORBA, JAVA, MIME, C/C++, Perl	Http Server, CGI, CORBA, JAVA, MIME, C/C++, Perl
Inference Engine 의 위치	Server 쪽에 치중	Client 쪽에도 위치할 수 있음	각 Server 들에 독립된 Inference Engine 이 있음	각 Client 쪽에도 위치할 수 있음
Knowledge Base 의 위치	Server 쪽에 치중	Client 쪽에도 사용자용 KB 를 위치할 수 있음	각 Server 들에 독립된 KB 이 있음	각 Client 에도 사용자용 KB 를 위치할 수 있음
Fact Base 의 위치	Server 쪽에 치중	Client 쪽에는 사용자용 Fact Base 를 둠	각 Server 들에 독립된 FB 이 있음	각 Client 에 사용자용 Fact Base 들이 위치
서비스영역	-Server 에서 주로 많은 일을 처리해야 되는 경우 -Server 에서 Information 을 집중적으로 모으고자 할 경우 -전문가시스템 Server 의 지식의 Security 측면의 보안이 요구되는 경우 -대중을 대상으로 서비스를 하고자 할 경우	-Client 주로 많은 일을 처리해야 되는 경우 사용자 개인의 사적 정보의 유출을 막아야 하는 경우 -Client 정보 만으로 Inference 가 가능한 경우 -특정 고객을 대상으로 하는 서비스	-여러가지 문제들이 종합적으로 고려되어야 하는 복잡한 서비스 -다른 전문가시스템들을 이용해서 새로운 부가서비스를 하고자 할 경우	-사용자를 그룹핑할 수 있는 서비스의 경우 -사용자들의 개인정보의 활용이 많은 비중을 차지하는 서비스
예	기상정보, 문헌정보	개인의 재무관리, 개인기호(의식주, 레저)	주식투자, 국가안보, 물가예측,	전문동호인, 주식투자가, 외환딜러

이러한 모델들은 실제상황에서는 서로 조합된 형태로서 나타날 수 있는데 이러한 조합에 의해 생각할 수 있는 혼합형태들을 [표 3] 과 [그림 5]에 나타내었다. 본 논문에서는 이러한 각 형태들에 대한 사항들은 위의 4가지 모델에서 설명한 사항들을 토대로 생각하면 될 것이므로 자세한 분석은 생략하기로 하고 각각의 Type 에 따른 특징을 간략하게 설명하면 아래와 같다.

● **Type0 (Server Oriented - Client Oriented)**
Server 와 Client 를 서로 1 대 1 로 보았을 때 균형있게 만들어진 경우이다. 이상적인 경우

라고 할 수 있겠다

● **Type1 (Server Oriented - Distributed Server)**

Server 은 분산되어 있고 각각의 Server 이 독립적인 Inference Engine, KB, FB 를 가지고 있으며, Client 는 단순한 Browsing 기능을 가지고 있는 경우이다.

● **Type2 (Server Oriented - Distributed Client)**

이러한 형태는 존재하지 않는다. Server 쪽에 Inference Engine, KB, FB 등이 집중적으로 위치하고 Client 들이 분산되어 Inference Engine, KB, FB 를 가진다는 것은 불가능 하다. 왜냐하면 Client 들이 분산되어 작동한다는 것은 Client

안에 Inference Engine, KB, FB 등을 가지고 있다는 말이고 이는 Client가 단순 Browsing 기능만을 한다는 Server Oriented 개념에 위배된다.

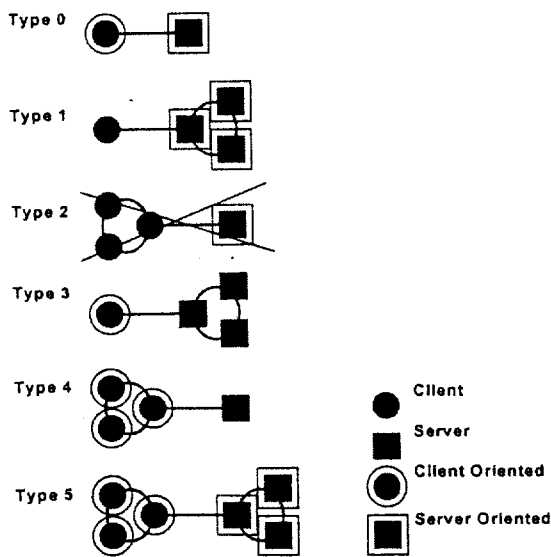
- **Type3 (Client Oriented - Distributed Server)**
Server 들은 분산되어 있고 Client 쪽에 Inference Engine, KB, FB 등을 처리하는 기능이 중점이 되는 경우이다.

- **Type4 (Client Oriented - Distributed Client)**
Distributed Client 들의 전형적인 형태이다. 즉, Client 쪽에 Inference Engine, KB, FB 등을 처리하는 JAVA 나 그 외 VC++ 같은 언어로 만들어진 External Helper 프로그램에 의해 Client 간의 정보전달도 가능하게 되 있는 경우이다.

- **Type5 (Distributed Server - Distributed Client)**
Server 도 분산되었고 Client 도 분산되어 있는 균형있게 분산되어 있는 형태이다. 다양한 사용자와 다양한 전문가시스템이 존재하는 인터넷 환경에서의 가장 이상적인 형태라 하겠다.

[표 3] 혼합형태에 따른 분류

	Server Oriented	Client Oriented	Distributed Server	Distributed Client
Server Oriented		Type0 Balanced	Type1	
Client Oriented			Type3	Type4
Distributed Server				Type5 Distributed Balanced
Distributed Client				



[그림 5] 혼합형태에 따른 분류 모형

6. 결론 및 향후전망

지금까지 인터넷의 Web 환경에서의 전문가시스템 개발현황들을 살펴 보고 Web 기반 전문가시스템들을 유형별로 분류하고, 분석하여 일반적인 모델 4 가지를 제시하였다. 인터넷상의 Web 기반에서는 Inference Engine, Knowledge Base, Fact Base 의 위치와 Client 프로그램의 형태에 따라서 Load Balancing 차원에서 Server 쪽에 많은 비중이 있는 Server Oriented Model 과 Client 쪽에 많은 비중을 두는 Client Oriented Model 로 나눌 수 있으며, 분산환경의 측면에서는 Server 들이 분산되어 있는 경우인 Distributed Server Model 과 Client 들이 서로 분산환경을 사용할 수 있는 Distributed Client Model 로 나누어 볼 수 있다. 본 논문에서는 이렇게 나누어진 각 모델별로 특징 및 장단점, 환경, 사용기술과 서비스 유형들을 분류하고, 또한 실제 인터넷 환경에서는 위에서 제시한 4 가지 모델들이 조합된 형태의 시스템들이 사용될 것이므로 이를 Type0 부터 Type5 까지 분류하고 그 특징들을 설명해 놓았다.

본 논문을 통해서 이러한 모델들을 만들고 분석함으로써 향후 인터넷 Web 기반 전문가시스템 개발에 기준점을 제시했다고 하겠다. 향후 연구과제는 이러한 Web 기반전문가시스템을 효율적으로 구현할 수 있는 상세한 기준에 대한 연구와 본문에서 각 모델별로 제시한 여러가지 고려사항들에 대한 합리적인 해결방안의 연구, Agent 를 이용하는 경우에 대한 고려, 그리고 각 모델의 조합형태로 나타나는 Type 별로의 실제 활용에 대한 연구도 필요하리라 생각된다. 또한 본 논문을 통해서 새로이 제시하는 인터넷 환경에서의 Distributed Client 개념에 의한 새로운 형태의 전문가시스템의 개발을 위한 연구도 앞으로의 과제이다. 앞으로 본문에서 제시한 Type5 형태의 많은 전문가시스템들의 출현이 기대된다.

[참고문헌]

- [1] Jay Liebowitz, "Worldwide Perspectives and Trends in Expert Systems", AAAI, summer 1997, p115-119
- [2] "The intelligent Software development tools market: part1", Intelligent Software Strategies, Feb. 1996, vol XII no.2 ,
- [3] "The intelligent Software development tools market: part2", Intelligent Software Strategies, March. 1996, vol XII no.3
- [4] 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, pp.475-480, 1996

- [5] O.Etzioni and D.S.Weld, "Intelligent Agents on the Internet: Fact, Fiction, and Forecast," *IEEE Expert*, Vol.10, No.4, Aug.1995, pp.44-49
- [6] J.Gosling and H.McGilton, "The Java Language Environment: A White Paper," tech. report, Sun Microsystems, 1995, <http://java.sun.com/>
- [7] T.J.Mowbray and R.Zahavi, *The Essential CORBA : System Intergration Using Distributed Objects*, John Wiley & Sons, New York, 1995.
- [8] Robert Orfali and Dan Harkey, *Client/Server Programming with Java and CORBA*, John Wiley & Sons, New York, 1997.
- [9] Henrik Eriksson, Linkoping University, "Expert Systems as Knowledge Servers", *IEEE Expert*, June 1996
- [10] Kankanahalli Srinivas, "Java and Beyond : Executable Content", *IEEE Micro*, June 1997
- [11] Daniel E. O'Leary, " The Internet, Intranets, and the AI Renaissance", *IEEE Computer*, January 1997.
- [12] Eric Evans and Daniel Rogers, "Using Java Applets and CORBA for Multi-User Distributed Applicationsm," *IEEE Internet Computing*, May-June 1997
- [13] " Expert Systems and the Web", *Intelligent Software Strategies*, April 1996
- [14] "Intelligent Internet Systems: Tools and Applications - Part I", *Intelligent Software Strategies*, December 1996
- [15] "Intelligent Internet Systems: Tools and Applications - Part II", *Intelligent Software Strategies*, Winter 1997
- [16] " Java Expert Systems Tools", *Intelligent Software Strategies*, Summer 1997
- [17] " Selectica : Java-based Configuration for Internet and Electronic Commerce Applications", *Intelligent Software Strategies*, October 1996
- [18] " Expert Systems and the Web", *Intelligent Software Strategies*, April 1996