

# 웹 기반 전문가시스템의 자동생성체계

송용욱\* · 이재규\*\*

## Automatic Generation of Web-based Expert Systems

Yong Uk Song\* and Jae Kyu Lee\*\*

### 요 약

본 논문은 웹 기반 전문가시스템의 구현 구조들을 분석, 장단점을 비교하고, 웹 서버의 부하를 절감할 수 있는 HTML을 이용한 역방향 추론기관의 구현 방안을 제시한다. 웹 환경 하에서 전문가 시스템을 구현하는 방안으로는 CGI를 이용하는 방법, 웹 서버에 추론기관을 내장하는 방법, 외부 뷰어를 이용하는 방법, Java Applet을 이용하는 방법, 그리고 HTML을 이용하는 방법 등이 있다. 이들은 시스템 개발, 시험, 확장, 이식, 유지보수의 용이성, 대규모 서비스 가능성 등에서 각기 장단점을 갖고 있다. 특히, HTML을 이용하여 역방향 추론을 구현하면 위의 장점들을 상대적으로 많이 누릴 수 있다. 따라서, 이 논문에서는 역방향 추론에서 진위형, OAV형 및 수치형 변수를 HTML과 JavaScript를 이용하여 표현하는 방안을 설명한 후 이를 바탕으로 HTML 방식의 전문가시스템 설계 방법론을 제시한다. 또한, 기존 전문가시스템을 웹 기반으로 변환하기 위하여 기존 방식의 규칙베이스를 지식분석도로 바꾼 후 HTML 기반 전문가시스템을 생성하는 방법론도 설명한다.

### Abstract

This paper analyzes the approaches of Web-based expert systems by comparing their pros and cons, and proposes a methodology of implementing the Web-based backward inference engines with reduced burden to Web servers. There are several alternatives to implement expert systems under the WWW environment: CGI, Web servers embedding inference engines, external viewers, Java Applets, and HTML. Each of the alternatives have advantages and disadvantages of each own in terms of development and deployment, testing, scalability, portability, maintenance, and mass service. Especially, inference engines implemented using HTML possess relatively large number of advantages compared with those implemented using other techniques. This paper explains the methodology to present rules and variables for backward inference by HTML and JavaScript, and suggests a framework for design and development of HTML-based Expert Systems. A methodology to convert a traditional rule base to an Experts Diagram and then generate a new HTML-based Expert System from the Experts Diagram is also addressed.

Keywords: 역방향 추론, 웹, 전문가시스템, 지식분석도, HTML, JavaScript

\* 국립경상대학교 경영학부

\*\* 한국과학기술원 테크노경영대학원

## 1. 서 론

웹(WWW)을 이용한 정보서비스의 폭발적인 증가와 함께 이를 전문가시스템에 응용한 사례들이 속속 발표되고 있다. 현재까지 웹 환경에서 개발되어진 전문가시스템의 응용분야는 지능형 전자 메일 해석 및 분류(Intelligent E-Mail Interpretation and Classification), 제품 및 서비스에 대한 지능형 마케팅 자문 및 교육 시스템(Smart Advisor for Marketing Products & Services and Training), 온라인 구성 시스템(Online Configuration Systems), 기술지원 안내 시스템(Help Desk for Technical Support) 등이다(Intelligent Software Strategies, 1997). 이처럼 전문가시스템에 웹을 응용하려 하는 이유는 기존의 독립형(stand-alone) 방식에서 탈피하여 클라이언트/서버(Client/Server) 환경으로 이행함으로써 다중 사용자를 지원하고, 지식을 공유하며, 시스템의 유지보수를 쉽게 하고자 하는 데에서 찾을 수 있다. 그러나, 웹이 하이퍼 미디어(hyper-media)의 기능에 의해 편리한 사용자 인터페이스를 제공하고 시스템의 개발 및 유지보수를 쉽게 하고는 있으나, 웹의 무연결성(connectionless), 무상태성(stateless) 특성(Fielding, et al., 1998)은 사용자와의 여러 번에 걸친 대화를 필요로 하는 전문가시스템을 구현하는 데에는 장애가 되고 있다.

따라서, 본 논문에서는 웹 환경에서 전문가시스템을 구현하기 위한 가능한 구조들을 분석하고 그 장단점을 비교한 후, 역방향 추론에 알맞은 HTML(Raggett, et al., 1998) 기반 전문가시스템 구축 프레임워크를 제시하고자 한다. 이 프레임워크는 HTML 방식의 전문가시스템의 설계방법론 및 기존 전문가시스템을 웹 기반으로 변환하기 위하여 기존 방식의 규칙베이스를 지식분석

도록 바꾼 후 HTML 기반 전문가시스템을 생성하는 방법론을 포함한다. 2장에서는 웹 기반 전문가시스템의 구조들을 비교하며, 3장에서는 HTML과 의사결정나무 간의 관계를 살펴봄으로써 웹의 HTML기반 전문가시스템의 구현가능성을 타진한 후, 4장에서 HTML 기반 전문가시스템의 구현 방안을 살펴본다. 이를 바탕으로 5장에서는 웹 기반 역방향 추론시스템의 구축 프레임워크를 제시한다. 그리고, 6장에서 웹 기반 역방향 추론시스템의 유지보수 방안을 설명한 후 7장에서 HTML 기반 전문가시스템의 장단점 비교하면서 결론을 맺도록 하겠다.

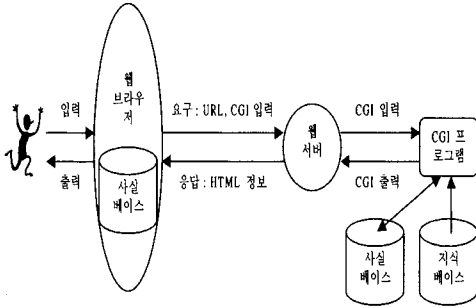
## 2. 웹 기반 전문가시스템의 구조 비교

김평철(김평철, 1996)은 웹과 데이터베이스의 연동 구조를 분류하고 장단점을 비교한 바 있다. 본 장에서는 이것을 확장, 적용하여 웹 기반 전문가시스템 구조들을 분류하고 장단점을 비교해 본다. 웹 환경에서의 전문가시스템은 추론기관이 서버(Server) 측에 있는 경우와 클라이언트(Client) 측에 있는 경우로 대별된다.

### (1) 추론기관이 서버 측에 있는 경우

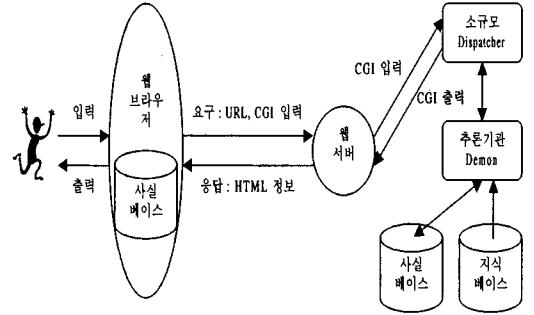
추론기관이 서버 측에 있는 경우는 다시 세 가지로 나누어 볼 수 있다. 첫째는 CGI를 이용하는 경우이고, 둘째는 웹 서버 자체가 추론기관을 내장하는 경우이고, 셋째는 이 논문의 주제인 HTML 만을 이용하는 경우이다.

CGI(Dwight, Erwin, and Niles, 1997)를 이용하는 경우 [그림 1]는 HTML의 FORM 태그(Tag)와 CGI 규약을 이용하여 사용자의 입력을 받아 처



(그림 1) 추론기관을 CGI로 구현한 경우

리하는 것이다. 이 방법은 기존의 웹 서버, 웹 브라우저, URL, HTTP, HTML, CGI 등의 기술을 변경 없이 사용할 수 있으며, 시스템의 개발, 시험이 용이하며 추후의 시스템 확장과 유지보수가 쉽다는 장점이 있다. 이 방식에서 문제가 되는 것은 웹의 무연결성, 무상태성 특성 때문에 그 사용자와의 계속되는 대화의 내용을 기억하기 위한 방안이 따로 마련되어야 한다는 것이다. 현재, 이를 위한 방안으로 대표적인 것은 HTML 태그 중의 하나인 FORM 태그의 HIDDEN 컨트롤을<sup>1)</sup> 이용하는 방법, 쿠키(Cookie)<sup>2)</sup>를 이용하는 방법, 데이터베이스에 사용자별로 대화내용을 저장하여 이용하는 방법 등이 있다. 또한, CGI로 구현된 전문가시스템이 추론기관과 지식베이스를 내장하였을 경우 그 프로세스의 크기가 크



(그림 2) 추론기관을 CGI로 구현한 경우 (대규모 서비스 용)

므로 다수의 사용자가 동시에 서비스를 요구할 경우 시스템 자원의 부족과 성능저하를 초래하며, 특히 대규모 서비스의 경우 치명적일 수 있다. 이를 극복하기 위해서 추론기관을 CGI 프로그램이 아닌 디몬(Demon)<sup>3)</sup>으로 바꾸는 방법을 사용할 수 있다(그림 2). 이 방식에서는 웹 서버의 CGI에 의해 소규모의 디스패처(Dispatcher)<sup>4)</sup> 프로세스가 구동되고, 이 디스패처 프로세스는 클라이언트로서 서버인 추론기관에 서비스를 요구하여 그 결과를 웹 서버를 통하여 사용자에게 보낸다. 이 경우 CGI 프로그램인 디스패처는 소규모 프로세스이므로 대규모 서비스에 의한 성능저하가 없게 된다. 그러나, 이 방식을 구현하기 위해서는 디스패처와 서버 추론기관 간에 추가적인 통신규약이 마련되어야 하고, 또, 추론기관의 서비스 일정을 계획하고 자원을 분배해주는 추가적

1) Hidden 컨트롤은 HTML FORM 태그 내의 텍스트, 버튼, 체크박스 등 다른 컨트롤들과 달리 웹 브라우저 화면에 표시되지는 않지만, 다른 컨트롤들처럼 그 값은 Submit 시에 웹 서버에 전달된다. 이 기능을 이용하여 웹 서버는 웹 브라우저로 하여금 특정 정보를 저장하였다가 자신에게 제시하게 할 수 있다.

2) Cookie란 웹 브라우저가 어떤 웹 서버로부터 HTML 정보를 받아들일 때 그 웹 서버의 요구에 의하여 별도로 받아서 저장하고 있는 정보이다. 저장된 Cookie는 그 웹 서버로부터 다시 HTML 정보를 받아와야 할 때, 웹 브라우저가 HTML 요구정보와 함께 웹 서버에 제시하도록 되어 있다.

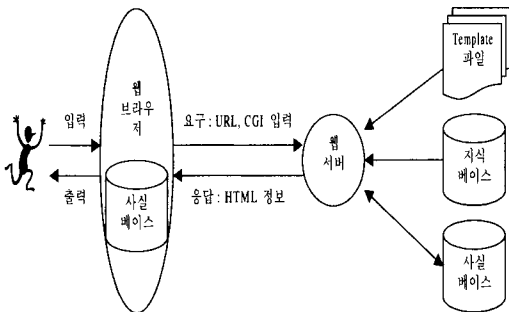
3) Demon이란 서버 컴퓨터에서 클라이언트의 서비스 요구를 기다리면서 대기하고 있는 서버 프로그램을 말한다. Demon은 자신의 부하상태에 따라 클라이언트의 서비스 요구를 제어할 수 있으므로 시스템이 극단적인 자원 부족 상황으로 빠지지 않도록 조절할 수 있다.

4) Dispatcher는 Demon의 클라이언트를 말한다. Dispatcher는 Demon과의 대화를 위한 간단한 프로그램만으로 이루어져 있기 때문에 수행 파일의 크기가 매우 작으므로 수 많은 Dispatcher가 동시에 수행되어도 시스템에 대한 부담이 크지 않다.

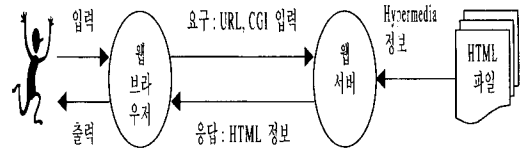
인 방법론과 프로그램이 필요하다.

웹 서버가 추론기관을 내장하는 경우 [그림 3]는 두 가지 방법으로 구현이 가능하다. 하나는 추론기관을 내장한 전문가시스템 전용 웹 서버를 개발하는 것이고, 둘째는 기존의 웹 서버에서 지원하는 확장 API(Application Program Interface; 예를 들어 Netscape의 NSAPI 등)를 이용하여 웹 서버에 추론기관을 추가하는 것이다. 이 방식은 추론기관이 웹 서버에 내장되어 있기 때문에 개발, 시험, 시스템 확장, 유지보수 등이 용이하지 않고, 또 현재까지는 추론기관을 웹 서버에 내장하여 사용하는 방법에 대한 연구나 개발시도는 이루어진 바가 없다. 특히, 전용 웹 서버의 경우는 특정 웹 서버에 종속되기 때문에 차후에 확장하거나 이식하고자 할 때 신축성이 떨어지게 된다.

HTML만을 이용하는 경우 [그림 4]는 웹의 기본 기술인 HTML만을 사용하기 때문에 초보자도 개발, 시험, 확장, 유지보수 등이 매우 용이하다. 또한, CGI를 사용할 때 나타나는 대규모 서비스 시의 시스템 부하 등의 문제도 전혀 없다. 다만, 이 방식이 문제가 되는 것은 HTML만을 사용하여 얼마나 일반적인 전문가 시스템을 구현할 수 있는가 하는 것이다. 본 논문의 3장 이후에서는 HTML만을 이용한 역방향 추론기관



(그림 3) 웹 서버가 추론기관을 내장한 경우



(그림 4) HTML기반 전문가시스템

의 구현에 대해 설명할 것이다.

(2) 추론기관이 클라이언트 측에 있는 경우

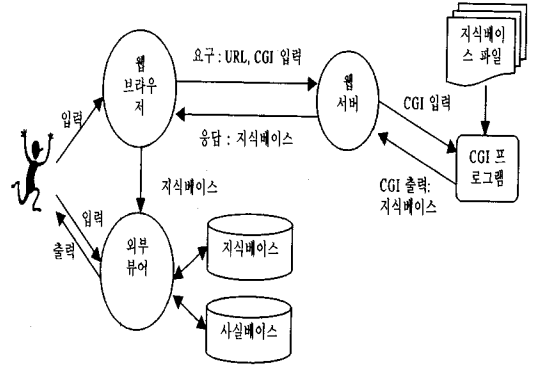
추론기관이 클라이언트 측에 있는 경우는 다시 두 가지로 나누어 볼 수 있다. 첫째는 외부 뷰어(External Viewer)를 이용하는 경우이고, 둘째는 웹 브라우저에 추론기관을 포함시키는 경우이다.

웹 브라우저는 HTTP를 이용하여 웹 서버로부터 받은 메시지의 Content-type 헤더(Header)<sup>5)</sup> 값이 자체적으로 처리할 수 없는 것일 경우 외부 응용프로그램을 구동할 수 있도록 지원하고 있다. 이 외부 응용프로그램을 외부 뷰어라고 부른다. 이 방식을 사용할 경우 추론기관은 외부 뷰어가 갖고 있고, 웹 서버로부터는 지식베이스가 전달되면 된다. 또, 추론 도중 추가 지식이 필요하면 외부 뷰어가 지식서버와 별도의 네트워크로 연결하여 필요한 지식을 가져올 수도 있다. 이 방식의 경우 [그림 5] 추론기관은 전적

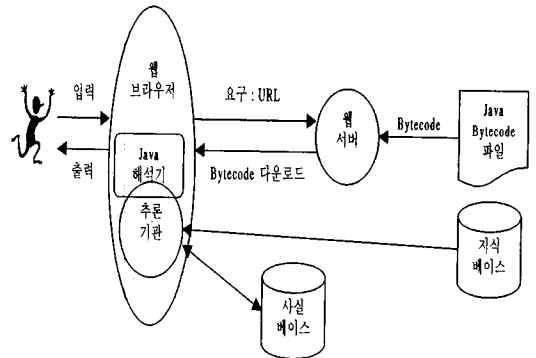
5) HTTP의 응답 전문은 머리 부분(Header)과 내용(Content)으로 이루어져 있다. 머리 부분에는 응답 전문의 형식, 길이, 에러 상황 등 각종 정보가 표시되는데, Content-type에서는 내용(Content)의 형태 HTML, 그림, 오디오, 워드 프로세서 파일 등을 표시해 준다. 내용이 HTML이나 GIF, JPG 등의 그림 파일일 경우는 웹 브라우저가 자체적으로 처리하여 화면에 표시해 주지만 기타 형태일 경우에는 웹 브라우저가 자체 처리를 못하므로 미리 설정된 외부 응용프로그램을 수행시키게 된다.

으로 사용자의 컴퓨터에서 하나의 응용프로그램으로서 수행되기 때문에 서버와 네트워크에 추가 부담이 없으며, 따라서, 서버측에 추론기관을 둔 경우보다 빠르게 추론 결과를 얻을 수 있다. 또한, 웹의 무연결성, 무상태성 특성에서 오는 문제들도 나타나지 않게 된다. 그러나, 이 방식의 문제점은 사용자가 웹 브라우저 이외에 외부 뷰어라는 별도의 소프트웨어의 기능을 익혀야 하며, 또 외부 뷰어를 사용하기 위하여 사전에 이것을 다운로드를 받아서 설치하여야 한다. 특히, 외부 뷰어는 외부 뷰어가 설치되어 수행될 시스템 플랫폼에 맞추어 여러 가지 버전이 준비되어야 한다. 또, 외부 뷰어 프로그램이 변경되면 모든 사용자는 변경된 소프트웨어를 다시 다운로드 받아 설치하여야 하므로 시스템 확장과 유지보수가 어렵게 된다.

웹 브라우저에 추론기관을 포함시키는 것은 웹 서버에 추론기관을 내장하는 것처럼 웹 브라우저에 추론기관을 프로그래밍하여 넣는 것을 의미하지는 않는다. 이것은 Java Applet(Lemay and Cadenhead, 1998)을 이용하는 것을 말한다. 즉, 웹 브라우저에 이미 Java 바이트코드(Bytecodes)<sup>6)</sup>를 실행하는 인터프리터가 있으므로, 추론기관을 Java Applet으로 구현한 후 웹 서버에서 추론기관의 Java 바이트코드와 지식베이스를 브라우저에 보내면 웹 브라우저의 Java 인터프리터가 이것을 수행하도록 하는 것이다. [그림 6]이 방식



(그림 5) 외부 뷰어를 사용한 전문가시스템 구조



(그림 6) Java Applet으로 구현한 전문가시스템 구조

은 외부 뷰어에서 나타나는 시스템 종속성, 설치 등의 모든 문제를 해결한다. 따라서, 추론기관의 개발, 시험, 확장, 유지보수도 용이하다. 동시에, 외부 뷰어의 장점을 그대로 향유하게 된다. 이 방식의 단점은 웹 서버로부터 Java 바이트코드를 전달 받는 시간이 길다는 것이다. 또한, HTML보다는 Java 언어를 이용한 고도의 프로그래밍 능력을 필요로 하기 때문에 초보자가 쉽게 접근할 수 없다는 것도 상대적인 단점으로 지적할 수 있다.

지금까지 설명한 각 방식들의 장단점을 비교하면 [표 1]과 같다.

6) Java 언어의 특징은 컴파일된 결과 파일을 어떤 컴퓨터에서도 새로 컴파일할 필요 없이 바로 수행시킬 수 있다는 데 있다. 그러나, 컴퓨터 별로 기계어가 다르기 때문에, 다른 프로그래밍 언어에서처럼 각 컴퓨터에서 바로 수행될 수 있는 이진(binary) 기계어가 아니라, Java용 인터프리터가 효율적으로 해석하여 수행시킬 수 있는 이진 코드로 컴파일 된다. 이것을 보통의 수행파일과 구분하기 위하여 Bytecodes라고 부른다.

(표 1) 웹 기반 전문가시스템 구조 비교

	의사결정기관				클라이언트측 추론기관		
	CGI		웹 서버		HTML	외부 뷰어	Java Applet
	내장식	외장식	전문	API			
개발용이성	▽	▽	×	×	○	▽	▽
시험용이성	○	○	×	×	○	○	○
확장성	▽	○	×	×	○	×	○
이식성	▽	▽	×	×	○	×	○
유지보수	○	○	×	×	○	▽	○
대규모서비스	×	○	○	○	○	○	○
상태 저장 문제	×	×	×	×	○	○	○
시스템 단순성	○	×	○	○	○	○	○
설치 문제	○	○	○	○	○	×	○
수행준비속도	○	○	○	○	○	○	×
수행속도	×	×	×	×	○	▽	○
규칙 표현성	○	○	○	○	×	○	○

(범례) ○ - 좋음 ▽ - 보통 × - 나쁨

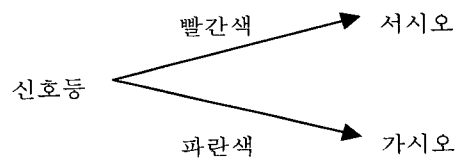
현재 일반적으로 많이 사용되는 방식들은 CGI 방식, HTML 방식, Java Applet 방식 등이다. Java Applet을 이용한 개발에 비해 상대적으로 개발이 간편하고 초기 다운로드 시간이 없기 때문에 현재와 같이 느린 인터넷 속도 하에서 상대적 우위성을 갖는 CGI 방식이 그 중에서도 지금까지 널리 사용되어 왔다(EXSYS). 그러나, 네트워크 속도가 개선되어 초기 다운로드 시간이 충분히 짧아지게 되면 대부분의 전문가시스템은 Java Applet 방식으로 이행할 것으로 보인다. CGI나 Java Applet에 대한 이러한 깊은 관심과는 대조적으로 HTML 방식에 대한 관심은 상대적으로 매우 적었던 것으로 보인다. 그 이유는 [표 1]의 마지막 줄에 보여지듯이 규칙의 표현성이 제한되기 때문에 HTML 만으로는 일반적인 전문가시스템을 구현하는 것이 매우 어렵기 때문인 것으로 생각된다. 그러나, HTML의 하이퍼링크(Hyperlink) 기능이 의사결정나무의 구현 및 일반적인 역방향 추론의 구현을 가능하게 할 수 있다는 점에 착안하면, 적어도 역방향 추론

을 이용한 진단, 분류 등의 분석문제에 관한 한 HTML 방식의 전문가시스템은 충분히 구현 가능하며, 그 시스템은 [표 1]에서와 같은 장점을 향유하게 된다. 따라서, 다음 3장부터는 HTML을 이용한 역방향 추론의 구현에 대해 설명한다.

### 3. HTML과 의사결정나무

HTML의 하이퍼링크를 이용하면 의사결정나무를 구현할 수 있다. 예를 들어 [그림 7]과 같은 의사결정나무(Decision Tree)가 있다고 하자.

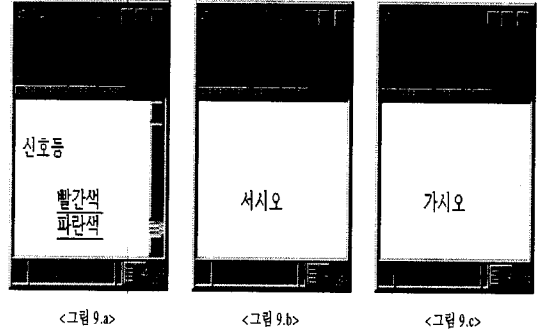
이때 [그림 8]과 같은 세 개의 HTML 파일로 [그림 9]에서와 같은 웹 사이트를 구성하면, 위



(그림 7) 의사결정나무 예제

의 의사결정나무를 웹 상에서 구현할 수 있다. [그림 9.a]에서 하이퍼텍스트(Hypertext) 빨간색은 [그림 9.b]에 하이퍼링크 되어 있고, 하이퍼텍스트 파란색은 [그림 9.c]에 하이퍼링크 되어 있다.

또한, 의사결정문제를 구현한 HTML 파일들이 반드시 나무(Tree) 구조일 필요는 없으며, 그래프(Graph)의 구조를 갖는 것도 가능하다. 이 특성을 이용하면 하이퍼링크 된 HTML 파일들의 나열을 통하여 역방향 추론을 구현할 수 있다. 이것은 4장에서 설명한다.



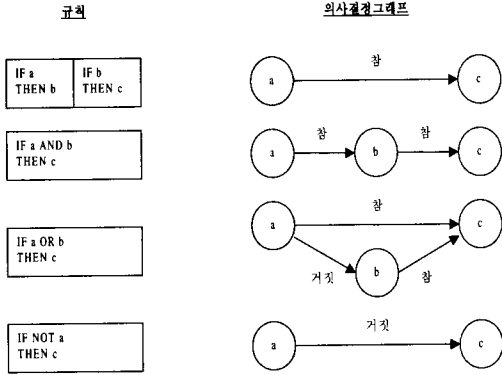
(그림 9) WWW상에서의 의사결정나무 구현

#### 4. HTML 기반 역방향 추론

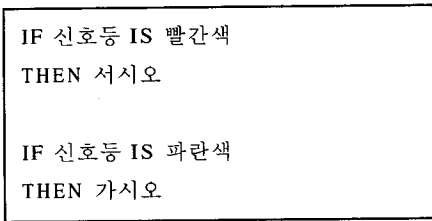
역방향 추론기관용 지식베이스의 규칙들을 연쇄(Chaining) 구조의 관점에서 살펴보면 그들이 의사결정그래프 구조를 갖고 있음을 알 수 있다. [그림 10]에 나타난 바와 같이 AND, OR, NOT 등에 의해 연결된 다수개의 조건들을 조건절에 가진 규칙들은 의사결정그래프로 변환할 수 있다. 따라서, 앞의 3장의 논의에 따라 하이퍼링크된 HTML 파일들의 나열만으로도 역방향 추론을 구현할 수 있다. 예를 들어 [그림 7]의 의사결정나무는 [그림 11]과 같은 규칙들로 표현될 것이고 이것은 [그림 8]과 같이 HTML 파일에 의해 구현될 수 있다. 이때, 의사결정그래프의 각 노드가 각각 하나의 HTML 파일에 해당함에 유의하기 바란다. 그리고, 이를 표시하기 위하여 [그림 10]의 의사결정그래프에서 각 노드를 순서도의 화면도형으로 표현하였다. 역방향 추론을 구현하기 위해서는 사실(Fact)의 입력, 하위목표(Subgoal)에 의한 연쇄, 추론결과의 출력 등이 지원되어야 한다. 하위목표에 의한 연쇄는 바로 위에서 이야기한 바와 같이 하이퍼링크에 의해 구현할 수 있으며, 추론결과의 출력은 웹 브라우저가 디스플레이할 HTML

파일 : a.html
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Decision Tree &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;BR&gt; 신호등 &lt;BR&gt;&lt;BR&gt; &lt;BLOCKQUOTE&gt; &lt;A HREF="b.html"&gt; 빨간색 &lt;/A&gt; &lt;BR&gt; &lt;A HREF="c.html"&gt; 파란색 &lt;/A&gt; &lt;/BLOCKQUOTE&gt; &lt;/BODY&gt; &lt;/HTML&gt;</pre>
파일 : b.html
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Decision Tree &lt;/TITLE&gt;&lt;/HEAD&gt; &lt;BODY&gt; &lt;BR&gt;&lt;BR&gt;&lt;BR&gt; &lt;BLOCKQUOTE&gt; 서시오 &lt;/BLOCKQUOTE&gt; &lt;/BODY&gt; &lt;/HTML&gt;</pre>
파일 : c.html
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Decision Tree &lt;/TITLE&gt;&lt;/HEAD&gt; &lt;BODY&gt; &lt;BR&gt;&lt;BR&gt;&lt;BR&gt; &lt;BLOCKQUOTE&gt; 가시오 &lt;/BLOCKQUOTE&gt; &lt;/BODY&gt; &lt;/HTML&gt;</pre>

(그림 8) 의사결정나무를 위한 HTML 파일



(그림 10) 의사결정그래프로의 규칙 변환



(그림 11) 규칙베이스 예제

파일을 만들면 된다. 따라서, 남은 문제는 사실의 입력을 지원하는 방안이다. 역방향 추론에서 사실 값은 변수에 저장된다. 역방향 추론에서 변수의 형태는 진위형(Fact Type), OAV(Object-Attribute-Value Type)형, 수치형(Numeric Type)의 세 가지가 있다. 여기서, 진위형과 OAV형 변수는 하이퍼텍스트의 나열에 의해 쉽게 해결된다. 예를 들어 [그림 11]의 규칙베이스에서 신호등 변수는 빨간색 또는 파란색을 값으로 갖는 OAV형 변수이며 이것은 [그림 9.a]처럼 구현된다. 진위형 변수의 경우도 "참"과 "거짓"을 나열하면 된다. 예를 들어 [그림 13]의 첫번째 화면은 다음과 같이 진위형 변수가 나타난 규칙들의 조건절을 구현한 것이다.

IF 동물이 털이 있습니까? IS TRUE

THEN ...

IF 동물이 털이 있습니까? IS FALSE

THEN ...

수치형은 값을 입력 받은 후 계산하여야 하기 때문에 하이퍼텍스트의 단순 나열로는 해결되지 않는다. 한가지 방안은 수치형 변수 처리용 CGI를 구현하는 것이나, 이것보다 더 간단하고 효과적인 방법은 JavaScript(Danesh, 1996), VBScript (Jerke et al., 1997) 등과 같은 클라이언트 측 스크립트 언어(Client-side Script Language)를 이용하는 것이다. 예를 들어 다음과 같은 규칙이 있다고 하자.

IF total\_income >= 0.2 \* threshold  
THEN pay\_tax

IF total\_income < 0.2 \* threshold  
THEN do\_not\_pay\_tax

위 두 규칙에서 total\_income과 threshold 변수는 수치형 변수이다. 이 두 규칙은 [그림 12]와 같이 JavaScript를 사용하여 HTML 파일에 의해 구현될 수 있다. [그림 12]에서 보듯이 규칙의 조건절에 나타나는 모든 변수들의 값은 FORM 태그에서 입력 받으며, 입력된 변수를 포함한 수식의 평가 및 이에 따른 하이퍼링크는 SCRIPT

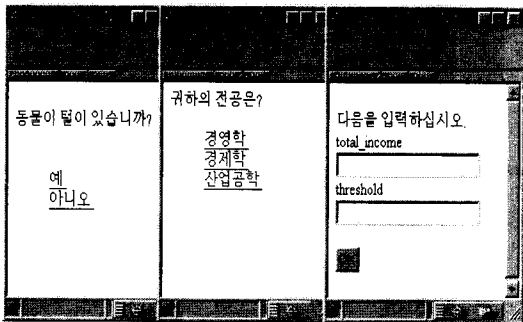
7) Script 언어란 컴파일러가 아닌 해석기(Interpreter)에 의해 해석되면서 바로 수행되는 프로그래밍 언어를 말한다. 웹 환경에서는 웹 서버 측에서 해석되어 수행되는 스크립트 언어와 웹 브라우저 측에서 해석되어 수행되는 스크립트 언어가 있다. JavaScript와 VBScript는 웹 브라우저 내의 해석기에 의해 수행되어 웹 브라우저에 결과를 표시하는 대표적인 클라이언트 측 스크립트 언어이다.



```

<HTML>
<HEAD>
<TITLE> Expression Type </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function verifyValue(form)
{
if (form.total_income.value >= 0.2 * form.threshold.value)
location="http://iis.kaist.ac.kr/tax.html"
else
location="http://iis.kaist.ac.kr/notax.html"
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="exprform">
다음을 입력하십시오. <BR>
total_income <INPUT TYPE="text" NAME="total_income"> <BR>
threshold <INPUT TYPE="text" NAME="threshold"> <BR>
<INPUT TYPE="button" VALUE="OK" onClick="verifyValue(this.form)">
</FORM>
</BODY>
</HTML>
    
```

(그림 12) JavaScript를 이용한 수치형 변수의 구현



(그림 13) 변수 형태별 질문 화면

태그 내에서 이루어진다. JavaScript 자체가 하나의 프로그래밍 언어이므로 조건절의 수식이 임의의 모든 수식이 될 수 있다는 것이 이 방식의

또 다른 장점이다. 변수의 형태별로 질문화면을 구성한 예가 [그림 13]의 세 번째 화면에 나타나 있다.

JavaScript를 이용한 수치형 변수 처리에서 한 가지 지적할 수 있는 기능상의 한계는 한 규칙의 조건절에서 사용한 변수를 그 규칙의 결론절에서 다시 이용할 수 없다는 점이다. 이 말은 한 규칙의 조건절에서 입력 받은 변수의 값이 연쇄된 다른 조건절로 전달될 수 없다는 뜻이기도 하다. 다음 예를 살펴보자.

```

IF total_income < 0.2 * threshold
THEN DISPLAY "TOTAL INCOME must be larger than" + 0.2 * threshold
    
```

위 예의 조건절에서 total\_income과 threshold 변수의 값을 입력받아 계산, 비교한 후 사용자에게 오류 메시지를 디스플레이하고 있다. 오류 메시지에 threshold 변수값이 사용되고 있음에 유의 하자. 이 경우 조건절과 결론절이 별도의 HTML 파일로 처리되고 HTML이나 JavaScript로는 threshold 변수의 값을 이 두 파일 간에 전달해 주는 방법이 없기 때문에 위의 예와 같은 규칙은 구현이 불가능하다. 다만, CGI를 이용하면 가능하나 이것은 또 다시 CGI의 단점을 발생시키게 된다.

## 5. 웹 기반 역방향 추론 시스템 구축 프레임워크

본 장에서는 4장에서 설명한 HTML 기반 역방향 추론기관의 구현 방법론을 바탕으로 주어진 규칙베이스로부터 이를 웹 기반 역방향 추론 시스템으로 구축하는 프레임워크를 제시한다. 추

론 시스템의 구축 과정은 아래와 같이 요약할 수 있다.

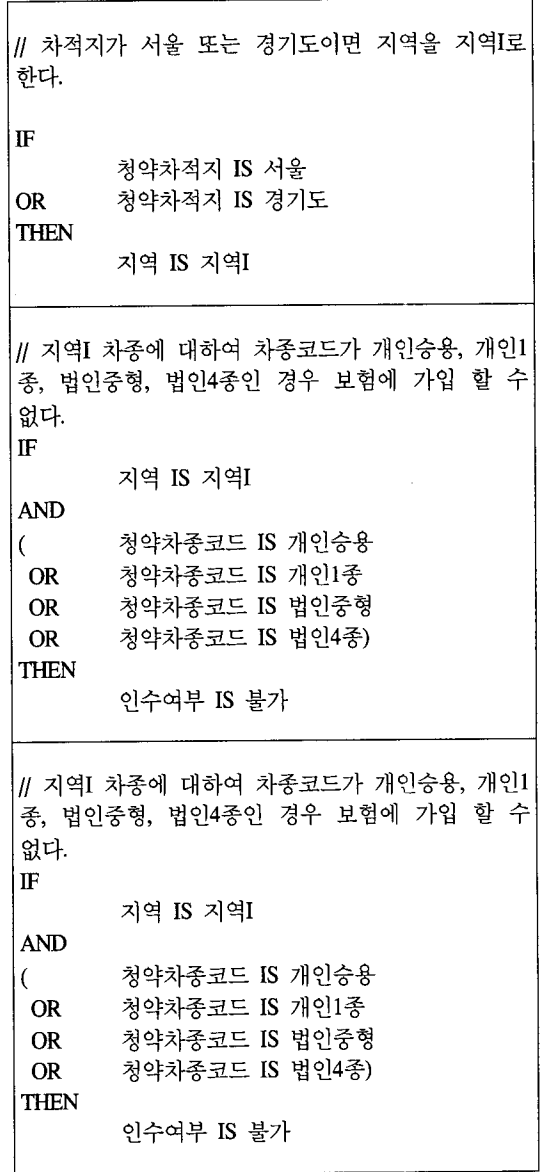
- (단계 1) 규칙베이스를 마련한다.
- (단계 2) 규칙베이스를 의사결정그래프로 변환한다.
- (단계 3) 의사결정그래프의 각 노드를 HTML로 변환한다.
- (단계 4) 변환된 HTML 파일을 웹 서버에 설치한다.
- (단계 5) 추론 서비스를 제공한다.

각 단계를 설명하면 다음과 같다. 단, 6장에서 설명할 지식분석도를 이용하여 지식을 획득한 경우에는 지식분석도와 의사결정그래프가 동일하기 때문에 (단계 3)부터 시작할 수 있음에 유의하기 바란다.

(1) 규칙베이스를 마련한다.

이 단계에서는 웹 기반 역방향 추론 시스템에서 사용한 규칙베이스를 마련한다.

이하의 설명에서는 자동차 보험회사의 보험청약 인수지침의 일부를 규칙으로 표현한 규칙베이스를 예로 사용한다. 이 규칙베이스는 [그림 14]에 나타나 있다. [그림 14]의 첫 번째 규칙은 보험청약차량의 차적지가 서울 또는 경기도이면 지역을 지역I으로 본다는 지침을 표현하고 있다. 또, 두 번째 규칙은 지역I의 보험청약차량의 차종이 개인승용, 개인1종, 법인중형, 법인4종 중의 하나이면 보험청약을 받아주면 안 된다는 인수지침을 표현하고 있다. 그리고, 세 번째 규칙은 지역I의 보험청약자의 나이가 26세 미만이고 청약차량의 차종이 기타업무용이면 보험청약을 받아주면 안 된다는 인수지침을 표현하고 있다.



(그림 14) 보험청약 인수지침 규칙베이스

(2) 규칙베이스를 의사결정그래프로 변환한다.

이 단계에서는 제 (1) 단계에서 획득한 규칙베이스를 의사결정그래프로 변환한다. 그 세부 과정은 다음과 같다.

(2.1) 규칙베이스에 나타난 변수들을 취합하여 정리한다.

규칙의 조건절과 결론절에 나타난 상수 이외의 변수들을 조건절 및 결론절 별로 모아서 정리한다.

우리의 예에서 조건절에 나타난 변수는 청약차적지, 지역, 청약차종코드, 및 청약자나이이다. 그리고, 결론절에 나타난 변수는 지역 및 인수여부이다.

(2.2) 취합한 변수를 목적별로 분류한다.

변수는 목적별로 질문변수, 중간결론변수, 결론변수로 나누어 볼 수 있다. 질문변수는 시스템 사용자에게 그 값을 질문해야 하는 변수이다. 질문변수는 조건절에만 나타난다. 우리의 예에서 질문변수는 청약차적지, 청약차종코드, 청약자나이이다. 중간결론변수는 추론 연쇄의 목적으로 사용되는 변수이다. 중간결론변수는 조건절과 결론절 모두에 나타난다. 우리의 예에서 중간결론변수는 지역이다. 결론변수는 최종 결론값을 저장하는 변수이다. 결론변수는 결론절에만 나타난다. 우리의 예에서 결론변수는 인수여부이다.

(2.3) 질문변수를 형태별로 분류한다.

질문변수를 진위형, OAV형, 수치형 변수로 분류한다. 변수의 형태는 질문의 형태를 결정한다. 진위형 변수는 예/아니오 또는 참/거짓의 형태로 그 값을 질문하고, OAV형 변수는 선택을 요구하는 형태로 그 값을 질문하고, 수치형 변수는 입력을 요구하는 형태로 그 값을 질문 한다.

우리의 예에서, 청약차적지와 청약차종코드는

OAV형 변수이고 청약자나이는 수치형 변수이다.

(2.4) 질문변수를 의사결정그래프의 각 노드로 표시한다.

질문변수는 의사결정그래프의 각 노드가 되고, 질문변수의 가능한 값들은 해당 노드로부터의 호(arc)가 된다. 진위형 질문변수의 가능한 값은 예 또는 아니오 (또는 참, 거짓)이고, OAV형 질문변수의 가능한 값은 규칙베이스에서 그 변수에 할당되는 모든 값들이 된다. 수치형 질문변수의 경우에는 가능한 값이 무한 개이고, 또한, 변수의 값이 다음 노드로 전달될 수 없으므로 같은 수치형 질문변수가 나타난 조건문들을 취합한 후 그 조건문에 나타난 모든 수치형 변수를 모아서 한 노드로 표현한 다음, 앞의 각 조건문들을 이 노드로부터 나오는 호로 한다.

우리의 예에서, 청약차적지, 청약차종코드 및 청약자나이가 의사결정그래프의 각 노드가 된다. 그리고, 청약차적지 노드로부터 나오는 호는 서울과 경기도가 되고, 청약차종코드 노드로부터 나오는 호는 개인승용, 개인1종, 법인중형, 법인4종, 기타업무용이 된다. 또, 청약자나이로부터 나오는 호는 청약자나이 < 26이 된다.

(2.5) 결론변수의 각 값들을 의사결정그래프의 각 단말노드로 표시한다.

결론변수의 각 값들은 의사결정그래프의 결론노드로서 단말노드가 된다. 우리의 예에서 인수여부 IS 불가와 인수여부 IS 가가 단말노드가 된다.

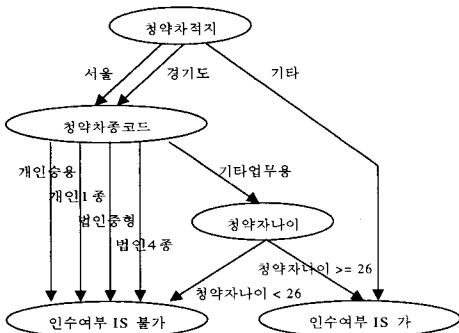
중간결론변수는 의사결정그래프의 노드가 되지는 못하며 다음 단계에서 각 노드를 연결할 때 연결 정보를 제공하는 역할만을 한다.

(2.6) 의사결정그래프의 각 노드를 연결한다.

앞 단계까지 나열한 노드들의 규칙베이스의 추론 연쇄에 맞추어 연결하면 의사결정그래프가 완성된다. 이 연결을 위한 정형화된 규칙은 따로 없으며, 의사결정그래프를 그리는 지식공학자(Knowledge Engineer)가 영역지식(Domain Knowledge)을 바탕으로 알맞은 연결관계를 찾아 표시하여야 한다. 다만, 이때 사용할 수 있는 몇 가지 지침을 제시하면 다음과 같다.

- i) AND, OR 등으로 표현된 조건문의 노드 표현은 [그림 10]의 변환 규칙을 이용한다.
- ii) 중간결론 변수로 연결된 두 규칙 중 앞의 규칙에 나타난 질문변수의 노드가 의사결정그래프에서 상위 노드가 된다.
- iii) ii)의 지침에 어긋나지 않는 한, 규칙에 나타난 빈도가 높은 질문변수의 노드가 의사결정그래프에서 상위 노드가 된다.

우리의 규칙베이스 예제를 위 단계를 거쳐 의사결정그래프로 표현한 것이 [그림 15]에 나타나 있다. 예제 규칙베이스가 인수지침의 일부에 불과하기 때문에 각 변수의 값이 다른 경우에 대한 부분은 표시되지 않았음에 유의하기 바란다.



(그림 15) 예제 규칙베이스의 의사결정 그래프 표현

(3) 의사결정그래프의 각 노드를 HTML로 변환한다.

이 단계에서는 (2) 단계에서 획득한 의사결정그래프를 HTML 파일들로 변환한다. 그 세부 과정은 다음과 같다.

(3.1) 의사결정그래프의 각 노드에 HTML 파일 이름을 부여한다.

의사결정그래프의 각 노드는 HTML 파일로 변환될 것이다. 따라서, 각 노드에 HTML 파일 이름을 먼저 부여한다. 우리의 예에서는 청약차적지, 청약차종코드, 청약자나이, 인수여부 IS 불가 노드에 각각 청약차적지.html, 청약차종코드.html, 청약자나이.html, 인수불가.html, 인수가.html이라고 파일 이름을 부여한다.

(3.2) 각 노드별로 HTML 파일을 생성한다.

의사결정그래프의 각 노드별로 부여된 이름을 파일이름으로 갖는 HTML 파일을 생성한다. HTML 파일은 제 4장에서 설명한대로 각 노드를 이루

```

<HTML>
<HEAD> <TITLE> XX화재 자동차보험 인수지침 </TITLE> </HEAD>
<BODY>
<BR> 청약차종코드 <BR><BR>
<BLOCKQUOTE>
<A HREF="인수불가.html"> 개인승용 </A> <BR>
<A HREF="인수불가.html"> 개인1종 </A> <BR>
<A HREF="인수불가.html"> 법인중형 </A> <BR>
<A HREF="인수불가.html"> 법인4종 </A> <BR>
<A HREF="청약자나이.html"> 기타업무용 </A> <BR>
</BLOCKQUOTE>
</BODY> </HTML>
    
```

(그림 16) 청약차종코드.html 파일의 내용

는 질문변수의 형태에 따라 HTML과 JavaScript를 사용하여 생성한다. 예를 들어 청약차종코드.html 파일의 내용은 [그림 16]과 같다.

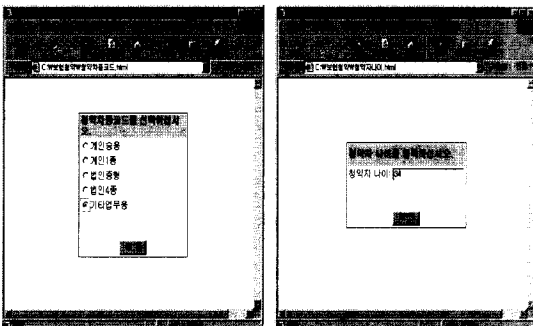
- (4) 변환된 HTML 파일을 웹 서버에 설치한다.

제 (3) 단계에서 만들어진 HTML 파일들을 웹 서버의 해당 디렉토리에 복사한다. 우리의 예에서는 청약차적지.html, 청약차종코드.html, 청약차나이.html, 인수불가.html을 복사하여 설치하면 된다.

- (5) 추론 서비스를 제공한다.

서비스의 제공은 웹 사이트의 주소를 알려줌으로써 시작된다. 우리의 예에서 웹 서버가 설치된 호스트의 이름이 www.gsnu.ac.kr이라고 한다면 추론 시스템의 홈페이지 주소는 http://www.gsnu.ac.kr/청약차적지.html이 된다.

이상의 5단계를 걸쳐 하나의 웹 기반 추론 시스템이 완성되었다. 위와 같이 완성된 추론 시스템 화면의 예가 <그림 17>에 나타나 있다.



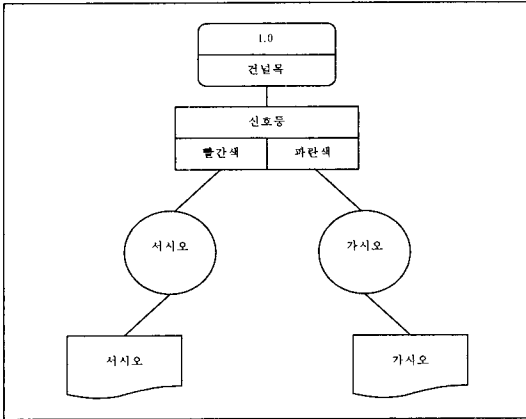
(그림 17) 보험청약 인수 결정 시스템의 예제 질문 화면

## 6. 지식분석도

HTML과 JavaScript를 이용하여 구현된 역방향 추론 웹 사이트의 문제점으로 지적할 수 있는 것이 웹 사이트 관리의 문제이다. 이러한 웹 사이트는 <그림 17>에서 보듯이 수많은 HTML 파일 및 그림 파일들로 이루어져 있을 것이고, 따라서 이들을 수작업으로 관리, 변경하는 것은 규칙의 수가 많아지고 또 규칙이 복잡해질수록 더욱 더 어려워질 것이다. 일반 웹 사이트의 경우에도 웹 사이트 저작 및 관리용 도구들이 많이 개발되어 상품화되고 있는 것을 보더라도, 웹 사이트의 생성 및 관리는 상당히 손이 가는 작업이다. 따라서, 역방향 추론 웹 사이트의 경우에도 그러한 저작 및 관리 도구가 필요할 것이다.

J. K. Lee 등은 전문가의 지식을 지식분석도라는 그래프로 표현 한 후 이 그래프로부터 추론 기관용 규칙을 생성하는 방안을 제시한 바 있다. (Lee, Lee, and Choi, 1990; 이재규 외, 1996A; 이재규 외, 1996B) 이 지식분석도는 본 논문이 제시하고 있는 HTML 기반 역방향 추론 웹 사이트를 생성하고 관리할 때 유용하게 사용할 수 있다. 예를 들어 <그림 11>의 규칙을 지식분석도로 표시하면 <그림 18>와 같다. <그림 18>에서 근 노드(Topic Node)와 결론노드(Conclusion Node)를 제외한 중간노드(Branch Node) 1개와 단말노드(Display Node) 2개가 각각 하나의 HTML 파일에 대응되며, 이것은 각각 <그림 8>의 a.html, b.html, c.html과 같다. 따라서, 지식분석도의 각 노드를 HTML 파일로 변환해 주는 시스템은 쉽게 구현 가능하다.

원래 지식분석도는 현장 전문가로부터의 지식 습득을 그래프에 의해 지원하기 위하여 개발된



[그림 18] 지식분석도 예

것이다. 규칙이라는 개념이 불분명한 현장 전문가로 하여금 자신의 지식을 그래프로 표현하도록 함으로써 지식 습득을 용이하게 한 것이었다. 그리고, 지식분석도를 시스템으로 구현한 LIFT는 지식분석도로 표현한 지식을 SKI2라는 역방향 추론기관용 규칙베이스로 자동 변환해주는 기능을 갖고 있다. 이 개념의 연장선 상에서 보면, 지식분석도로 표현된 지식으로부터 웹 기반 역방향 추론을 위한 HTML 파일들의 자동 생성 역시 가능하다. 지식분석도를 지원하는 시스템이 개발된다면 그래프 지원에 의한 현장 전문가로부터의 지식의 습득이 용이해짐과 동시에 지식이 변경되었을 때 지식분석도를 변경한 후 지식베이스를 이루는 HTML 파일들을 자동 생성하면 되므로 지식베이스의 유지보수가 쉬어지게 된다. 또한, 이것은 동시에 추론용 웹 사이트의 생성 및 관리가 용이해진다는 의미이기도 하다.

## 7. 결 론

제 2장에서 지적한 바와 같이 HTML 기반 역방향 추론은 개발, 시험, 확장, 이식 및 유지보

수가 용이하며, 시스템의 부하 없이 대규모 서비스가 가능하다. 또한 사용자와의 대화에 의해 입력한 상태 정보를 유지하도록 추가적인 처리를 할 필요가 없으며, 다수 사용자 지원을 위한 시스템 일정계획 등의 부가 프로그램을 개발할 필요도 없다. 또 외부 뷰어 방식에서처럼 시스템을 추가로 설치할 필요가 없으며 사용자는 웹 브라우저만 갖고 있으면 된다. 또한 웹 서버는 해당 HTML 파일의 내용을 읽어서 추가적인 처리 없이 바로 웹 브라우저로 보내면 되므로 처리속도도 전혀 문제가 되지 않는다.

또한 HTML 기반 역방향 추론은 추론 진행 중 취소(Undo) 기능의 구현이 매우 용이하다. CGI 방식에서는 취소가 일어난 경우 상태정보를 전 상태로 되돌리기 위하여 해당 사실(Fact)을 찾아서 없애주는 추가적인 작업이 필요하다. 이것은 독립형 추론기관이나 외부 뷰어방식, Java Applet 방식에서도 마찬가지이다. 그러나, HTML 방식에서는 단순히 직전 HTML 파일로 돌아가기만 하면 되며 이것은 취소 개수에 제한 없이 반복될 수 있다.

선택적 설명 기능의 경우는 미리 설명을 하는 HTML 파일을 만들어 두면 된다. 일반적으로 설명파일의 크기가 작게는 1KB로부터 크게는 수십 KB이내가 될 것임을 고려하면, 이러한 파일들이 보조기억장치를 차지하는 공간은 미미하다고 볼 수 있다. Hypothetical Reasoning의 경우도 마찬가지이다. What-if의 모든 경우에 대해 결과를 설명하는 HTML 파일을 미리 만들어 두면 된다.

그러나, 앞에서 지적한 바와 같이 지식의 표현성, 변수의 전달에서의 제한점이 이 방식의 가장 큰 단점으로 지적된다. 또, 웹 사이트의 생성 및 관리, 설명기능, Hypothetical Reasoning, 불확실성 하에서의 추론기능의 구현 등의 경우는 지식 관리자가 일일이 수작업으로 HTML 파일

을 만들고 관리하기가 어렵기 때문에 지식분석 도를 이용한 저작 및 관리 도구 개발이 추가로 필요하다.

이 논문에서는 웹 기반 전문가시스템의 구현 구조들을 알아보고 장단점을 비교한 후, 역방향 추론에 알맞은 HTML 기반 전문가시스템 구축 프레임워크를 제안하였다. 이 프레임워크 내에서 HTML 방식의 전문가시스템의 설계방법론을 제시하였으며, 또한 기존 전문가시스템을 웹 기반으로 변환하기 위하여 기존 방식의 규칙베이스를 지식분석도록 바꾼 후 HTML 기반 전문가시스템을 생성하는 방법론도 설명하였다. 현재는 CGI를 이용하는 방식과 Java Applet을 이용하는 방식이 개발, 시험, 확장, 이식, 유지보수의 용이성 등의 장점 때문에 널리 사용되고 있음을 알 수 있었다. 이에 반하여 HTML을 이용하는 방식은 그 일반성의 부족 때문에 현재 그다지 각광을 받지 못하고 있다. 그러나, 역방향 추론에 초점을 맞추어 보면 HTML은 충분히 사용 가능하며, HTML을 이용하여 구현된 시스템은 CGI나 Java Applet 보다 더 큰 장점을 갖는다. 따라서, 적어도 역방향 추론 분야에서는 HTML 방식이 앞으로 널리 사용될 수 있으리라고 생각한다. 앞으로 남은 연구과제는 규칙의 일반성을 확보하기 위한 방안과 불확실성의 표현 방안 등이며, 또한 지식분석도 등을 이용한 추론용 웹사이트 생성 및 관리 도구의 개발도 필요하다.

### 참 고 문 헌

- [1] 김평철, "웹을 위한 데이터베이스 통로의 분류체계," WWW 96-1(1996), 50-66.
- [2] 이재규, 송용욱, 권순범, 김우주, 김민용, UNIX를 이용한 전문가시스템의 개발, 법영사, 1996A.
- [3] 이재규, 최형림, 김현수, 서민수, 주석진, 지원철, 전문가시스템 원리와 개발, 법영사, 1996B.
- [4] 이재규 외, 자동차보험 선별인수 전문가시스템의 개발, 동부화재, 1995.11.
- [5] 임규건, 강주영, 이재규, "Web 기반 전문가시스템의 구조 분석," 한국전문가시스템학회 '97 추계학술대회, pp. 63-73.
- [6] Danesh, Arman, *Teach Yourself JavaScript in a Week*, Sams.net Publishing, 1996.
- [7] Dwight, Jeffry, Michael Erwin, and Robert Niles, *Special Edition Using CGI*, Second Edition, QUE, 1997.
- [8] EXSYS, Inc., *Moving an EXSYS Application to the EXSYS Web Runtime Engine (WREN)*.
- [9] Eriksson, Henrik, "Expert Systems as Knowledge Servers," *IEEE Expert*, June 1996, pp. 14-19.
- [10] Far, Behrouz H. and Zenya Koono, "Ex-W-Pert System: A Web-Based Distributed Expert System for Groupware Design," *Expert Systems With Applications*, Vol. 11, No. 4(1996), pp. 475-480.
- [11] Fielding, R., J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol-HTTP/1.1*, INTERNET-DRAFT <draft-ietf-http-v11-spec-rev-05>, Internet Engineering Task Force, 11 September 1998.
- [12] Giarratano, Joseph and Gary Riley, *Expert Systems: Principles and Programming*, 2<sup>nd</sup> Edition, PWS Publishing Company, 1994.
- [13] Jerke, N., M. Hatmaker, J. Anderson, *VBScript Interactive Course*, Waite Group Press, 1997.

- [14] Lee, J. K., I. K. Lee, and H. R. Choi, "Automatic rule generation by the transformation of Expert's Diagram: LIFT," *Int. J. Man-Machine Studies*, 32(1990), 275-292.
- [15] Lemay, Laura and Rogers Cadenhead, *Teach Yourself JAVA 1.2 in 21 Days*, SAMS Publishing, 1998.
- [16] O'Leary, Daniel E., "The Internet, Intranets, and the AI Renaissance," *IEEE Computer*, January 1997, pp. 71-78.
- [17] Raggett, Dave, Arnaud Le Hors, and Ian Jacobs(ed.), *HTML 4.0 Specification REC-html40-19980424*, W3C, 24 April 1998.
- [18] "Intelligent Internet Systems: Tools and Applications Part I," *Intelligent Software Strategies*, December 1996.
- [19] "Intelligent Internet Systems: Tools and Applications Part II," *Intelligent Software Strategies*, Winter 1997.
- [20] "Java Expert Systems Tools," *Intelligent Software Strategies*, Summer 1997.
- [21] "SELECTICA: Java-based Configuration for Internet and Electronic Commerce Applications," *Intelligent Software Strategies*, October 1996.
- [22] Amzi! URL→<http://www.amzi.com/>
- [23] Applied Logic Systems URL→<http://www.als.com/>
- [24] Attar Software URL→<http://www.attar.com/>
- [25] Bowne Internet URL→<http://www.bowneinternet.com/>
- [26] Brightware URL→<http://www.brightware.com/>
- [27] Gensym URL→<http://www.gensym.com/>
- [28] ILOG Inc. URL→<http://www.ilog.com/>
- [29] Inference URL→<http://www.inference.com/>
- [30] IntelliSystems URL→<http://www.intellisystems.com/>
- [31] JESS URL→<http://herzberg1.ca.sandia.gov/jess/>
- [32] MultiLogic URL→<http://www.multilogic.com/>
- [33] Neuron Data URL→<http://www.neurondata.com/>
- [34] RadNet URL→<http://www.radnet.com/>
- [35] Selectica URL→<http://www.selectica.com/>
- [36] The Molloy Group URL→<http://www.molloy.com/>