소화기 질환을 위한 인터넷 기반 전문가 시스템

하상호[†] · 유일대[††] · 천인국[†††] · 박상흠[††††] · 김선주[†††††]

요 약

국외에서는 수많은 종류의 의료 전문가 시스템이 개발되어 의사의 진단 및 처방에 효과적으로 활용되고 있으나, 국내에서 인터넷 기반의 의료 전문가 시스템의 개발 사례는 매우 미흡한 편이다. 본 논문에서는 인터넷을 통해서 손쉽게 사용 가능한 의료 진단 전문가 시스템을 개발한다. 이 시스템은 Java 기술과 Java와 연동 가능한 전문가 시스템 쉘 Jess를 사용하여 개발된다. 따라서 개발된 시스템은 특정 컴퓨팅 플랫폼에 독립되어 사용 가능하다는 특징을 갖는다. 또한, 시스템의 지식베이스를 구성하는 사실들과 규칙들을 통합하지 않고 구분하여 구성함으로써 지식베이스의 수정과 확장이 용이하다는 특징을 갖는다.

The Internet based Expert System for Gastroenteritis

Sangho Ha[†], Ildae You[††], In-Gook Chun[†††], Sang-Heum Park[††††] and Sun-Joo Kim[†††††]

ABSTRACT

A lot of expert systems have been developed in the area of medical diagnosis and prescription, but there has been relatively little effort to develop medical expert systems using domestic technologies. In this paper, a medical expert system for gastroenteritis is developed, which can be easily used on the Internet. This system is implemented using Java technologies and Jess, which is a Java expert system shell. Therefore, the system can be used independently from a specific platform on the Internet. In addition, facts and rules of a knowledge base in the system are represented separately, making the system updatable and expandable with ease.

Key words: 전문가 시스템, 의료 진단, 지식베이스, Jess, Java.

1. 서 론

최초의 전문가 시스템 DENDRAL[1]이 1965년에 미국 스탠퍼드 대학의 파이젠바움(E. Feigenbaum)에 의해서 개발된 이후로, 지식의 각 분야에서 전문가 시스템이 활발하게 개발되어 사용되고 있다. 특히, 의료 분야에서 의사의 진단과 처방을 보조하기 위한 의료 전문가 시스템이 활발하게 개발되어 오고 있다. 의료 전문가 시스템에서 구축된 의료 지식은 해부학, 생리학, 병리 생리학, 약학, 유행 병학 등과 같이 여러 영역으로 구분되며, 각 영역에서 의료 지식을 사용하여 진단(diagnosis), 예후(prognosis)와 같은 일반적인 추론 작업, 처방에 대한 결정, 의료 검진 계획에 대한 결정 등을 효과적으로 보조하고 있다.

외국의 경우 의료 분야에서 헤아리기 어려울 정도로 상당히 많고 다양한 의료 전문가 시스템이 개발되어 활용되고 있다. 1988년만해도 의료 전문가 시스템의 개발 사례는 1500여가지에 이르며, 현재에는 그 숫자가 수만 혹은 수십만 가지에 이를 것으로 추정된다. 또한, 인터넷 발달에 힘입어 최근에는 웹 상에서 사용 가능한 전문가 시스템[2-4]들이 개발되고 있으

본 연구과제는 2000학년도 순천향대학교 산업기술연구소 학술연구조성비 교육개혁 연구과제로 지원을 받아 수행하였음.
접수일 : 2002년 9월 30일, 완료일 : 2003년 4월 7일
[†] 순천향대학교 정보기술공학부 부교수
[††] 순천향대학교 정보기술공학부
[†††] 순천향대학교 교수
[††††] 순천향대학교 천안병원 소화기내과 부교수
[†††††] 순천향대학교 천안병원 소화기내과 교수

나, 국내에서는 그 개발이 매우 미흡한 편이다.

본 논문에서는 인터넷을 통해서 사용 가능한, 소화기 질환을 위한 의료 진단 전문가 시스템을 개발한다. 먼저, 소화기질환 전문의 임상교수로부터 전문 지식을 획득하여 소화기 질환 지식베이스를 구축하고, 이 지식베이스와 연동하여 사용자에게 의료 정보 서비스를 인터넷을 통해서 제공할 수 있는 소화기 질환 의료 진단 전문가 시스템을 개발한다. [3,4]의 경우에 시스템이 Prolog 혹은 마이크로소프트사의 Visual Basic으로 개발되어 있어서, 시스템이 특정 플랫폼에서만 동작 가능하다. 따라서 시스템이 다기종인 인터넷 환경에서 그 사용이 제한된다는 문제점을 갖는다.

논문에서는 인터넷 환경에서 특정 시스템의 플랫폼에 독립되어 사용될 수 있는 전문가 시스템 개발을 고려한다. 이를 위해서, 전문가 시스템을 J2EE[5] 플랫폼 상에서 개발한다. 또한, Java와 연동 가능한 전문가 시스템 쉘 Jess[6]를 사용하여 지식베이스를 기술하고, Jess에서 제공하고 있는 추론 엔진 Rete[7]를 사용한다. Jess는 규칙-기반 전문가 시스템의 개발 도구인 CLIPS[10]의 Java 버전이다. 또한, 시스템 제어 모듈과 사용자 인터페이스는 Java Servlet[8]과 JSP[9]를 사용하여 각각 구현한다. Java는 언어의 특징상 특정 기계 구조에 중립적이며[11], 따라서 Java로 작성된 소프트웨어는 특정 시스템에 독립되어 동작될 수 있다.

시스템의 또 다른 특징으로, 친숙한 사용자 인터페이스, 처방에 대한 설명 기능 제공, 신뢰성, 확장성 등이 고려된다. 특히, 지식베이스를 구성하는 사실들과 규칙들이 통합되지 않고 별도의 파일로 구성함으로써 지식베이스의 수정과 확장이 용이하다.

논문의 순서는 다음과 같다. 2장에서는 전문가 시스템의 지식베이스를 설명하고, 3장에서는 시스템의 구현 사항에 대해서 설명하고, 4장에서는 시스템의 실행 예제를 보여준다. 마지막으로, 5장에서는 결론을 언급한다.

2. 소화기 질환 지식베이스

전문가 시스템에서 지식베이스는 사실과 규칙들로 구성된다. 여기서는 소화기 질환 지식베이스의 전체 모델을 기술하는 결정 트리를 제시하고, 이 결정 트리로부터 사실과 규칙을 설계한다. 결정 트리는 소화기 질환에 관한 지식을 표현하고, 이 지식에 대한 추론 방법을 표현하는 기법이다.

2.1 결정 트리

그림 1은 소화기 질환 지식베이스를 표현하는 결정 트리(decision tree)의 일부를 보여준다. 그림 1의 트리는 각 부모 노드가 2개의 자식 노드를 갖는 이진 트리이고, 트리의 루트 노드와 중간 노드에 대해서는 질문이 연관되어 있고, 잎 노드에 대해서는 진단 결과인 병명이 연관되어 있음을 알 수 있다. 사용자에 대한 질문 위치는 항상 루트 노드부터 시작된다. 현재 노드에 연관된 질문에 대해서 사용자가 Yes를 답변하면, 다음 위치는 왼쪽 자식 노드가 되고, No를 답변하면 다음 위치는 오른쪽 노드가 된다. 현재의 노드가 잎 노드이면 그 노드에 연관된 병명으로 진단이 내려지고, 중간 노드이면 다시 그 노드에 연관된 질문이 제시된다. 이와 같은 방식으로 루트 노드부터 잎 노드에 이르기까지 사용자는 질문에 대해서 적절히 답변한다. 가령, 다음의 질문과 그 질문에 대하여 다음과 같이 답변한다면,

질문	답변
황달이 있으신가요?	No
열이 있으신가요?	Yes
숨쉬기가 곤란하십니까?	No

사용자는 "담도석이나 간농양"의 진단을 받게 된다. 즉, 황달이 없고, 열이 있고, 숨쉬기가 곤란하지 않으면 담도석이나 간농양의 증세가 의심된다는 것이다. 이와 같은 진단 과정은 결정 트리 상의 굵은 선으로 표시된 경로를 통해서 확인할 수 있다.

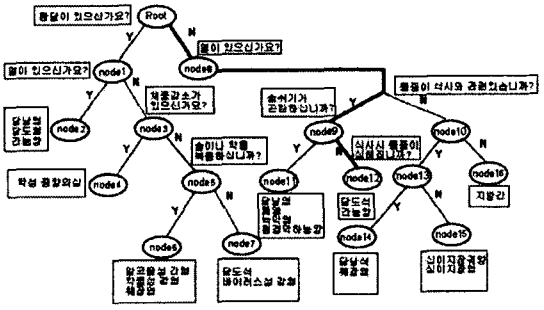

그림 1. 소화기 질환 지식베이스의 결정 트리 일부

그림 1의 결정 트리로부터 사실과 규칙을 생성하기 위해서, 트리의 노드 구조를 그림 2와 같이 정의한다.

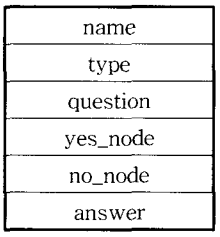

그림 2. 결정 트리 노드의 구조

다음은 노드의 각 구성 요소에 대해서 설명한다. 각 구성 요소를 슬롯(slot)이라 한다.

name : 노드의 이름을 나타낸다.
type : 노드의 타입을 나타낸다. 노드의 타입은 decision이거나 answer일 수 있다. decision 타입 노드는 질문을 갖는 노드로 루트나 중간 노드가 이 타입의 노드이다. answer 타입 노드는 진단을 갖는 노드로 잎 노드가 이 타입의 노드이다.
question : 노드가 decision 타입일 때, 질문 내용을 포함한다.
yes_node : question의 질문에 대해서 사용자가 yes로 답변했을 때, 이동할 노드의 이름을 나타낸다. 이 항목은 node의 타입이 decision인 경우에 의미가 있다.
no_node : question의 질문에 대해서 사용자가 no로 답변했을 때, 이동할 노드의 이름을 나타낸다. 이 항목은 node의 타입이 decision인 경우에 의미가 있다.
answer : 진단 결과인 병명을 포함한다. 이 항목은 node의 타입이 answer인 경우에 의미가 있다.

2.2 사실

그림 3은 그림 1의 결정 트리로부터 생성한 소화기 질환 지식베이스의 사실(facts)을 보여준다. 결정 트리의 각 노드에 대해서 한 개의 사실이 생성된다. 따라서 그림 1의 결정 트리에 대해서 17개의 사실이 생성된다. 이러한 사실들은 F1부터 F17까지의 자신의 식별자를 가지며, 그림 1의 노드를 그림 2의 노드 구조에 따라서 Jess로 기술하는 것으로 표현된다.

사실은 노드의 타입에 따라서 두 가지 유형으로 구분된다: 노드의 타입이 decision인 경우와 answer인 경우. 노드의 타입이 decision인 경우에 question, yes-node, no-node가 의미가 있으며, answer는 의미가 없다. 노드의 타입이 answer인 경우에는 question, yes-node, no-node가 의미가 없으며, answer만이 의미가 있다. 각 경우에 의미가 없는 요소는 nil로 표현한다. 노드의 타입이 decision인 경우 yes-node와 no-node의 슬롯에는 사용자의 각 답변, yes/no에 따라서 이동할 노드를 각각 나타내고, 이때 answer 슬롯은 nil의 값을 갖는다. 그러나 노드 타입이 answer인 경우에 yes-node와 no-node의 슬롯은 의미가 없으므로 nil의 값을 갖고, answer 슬롯은 적절한 병명을 갖는다. 그림 1에서 루트 노드와 중간 노드는 타입이 question이고, 잎 노드는 타입이 answer임을 알 수 있다.

사실 F2에서, 노드의 타입은 decision이고, 해당 질문은 "열이 있으십니까?"이다. 이러한 질문에 대한 사용자의 Yes 답변에 대해서는 node2로 이동하고, No 답변에 대해서는 node3로 이동한다. 사실 F3에서는 노드의 타입이 answer이므로, yes-node와 no-node는 의미가 없으며, answer에는 진단 결과에 대한 병명이 포함된다.

2.3 규칙

규칙(rules)은 어떤 결정을 위해서 사실을 이용하는 법칙이다. 여기서는 그림 3의 사실들을 이용해서 사용자의 질문으로부터 답변을 끌어내는데 사용되는 규칙들을 기술한다. 먼저, Jess에서 규칙을 기술하는 방법에 대해서 간략히 설명한다. Jess에서 규칙은 두 부분으로 구성되고, 이들은 "=>"의 기호로 분리된다. 규칙의 첫 번째 부분은 지식베이스에 포함된 사실들을 매칭하는데 사용되는 패턴들로 구성되고, 두 번째 부분은 함수 호출들로 구성되는 행동을 나타낸다. 첫 번째 부분은 "=>"의 왼쪽에 위치하기 때문에 규칙의 LHS(left-hand-sides)라 부르고, 두 번째 부분은 오른쪽에 위치하기 때문에 규칙의 RHS (right-hand-sides)라 부른다. LHS는 규칙의 if 부분을 나타내고, RHS는 then 부분을 나타낸다. if 부분이 만족되면(이 부분을 표현하는 패턴이 매칭되면), then 부분을 표현하는 행동들이 실행된다.

그림 4는 그림 1의 결정 트리를 설명하는데 필요

```
F1 : (node (name root) (type decision) (question "황달이 있으십니까?") (yes-node node1)
       (no-node node8) (answer nil))
F2 : (node (name node1) (type decision) (question "열이 있으십니까?") (yes-node node2)
       (no-node node3) (answer nil))
F3 : (node (name node2) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 담낭염 담도염 간농양 "))
F4 : (node (name node3) (type decision) (question "최근에 체중감소가 있으십니까?")
       (yes-node node4)(no-node node5) (answer nil))
F5 : (node (name node4) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 악성종양의심 "))
F6 : (node (name node5) (type decision) (question "최근에 술을 많이 드시거나 약물을
         복용한적이 있습니까?") (yes-node node6) (no-node node7) (answer nil))
F7 : (node (name node6) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 알코올성감염 약물성감염 췌장염 "))
F8 : (node (name node7) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 담도석 바이러스성감염 "))
F9 : (node (name node8) (type decision) (question "열이 있으십니까?") (yes-node node9)
       (no-node node10) (answer nil))
F10 : (node (name node9) (type decision) (question "숨쉬기가 곤란한가요?")
         (yes-node node11) (no-node node12) (answer nil))
F11 : (node (name node11) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 담낭염 폐염 늑막염 횡경막 하농양 "))
F12 : (node (name node12) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 담도석 간농양 "))
F13 : (node (name node10) (type decision) (question "통증이 식사와관련이 있습니까?")
       (yes-node node13)(no-node node110) (answer nil))
F14 : (node (name node13) (type decision) (question "식사시 통증이 더 심해집니까?")
       (yes-node node14)(no-node node15) (answer nil))
F15 : (node (name node14) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 담낭석 췌장염 "))
F16 : (node (name node15) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 십이지장궤양 십이지장염 "))
F17 : (node (name node110) (type answer) (question nil) (yes-node nil) (no-node nil)
       (answer " 지방간"))
```

그림 3. 소화기 질환 지식베이스의 사실 일부

한 규칙 R1~R5를 보여준다. 여기서는 시스템과 사용자간의 인터페이스가 콘솔을 통해서 이루어지는 것을 가정한다. 웹 브라우저를 통한 시스템과 사용자와의 인터페이스는 3장에서 기술된다. 실행 초기에 R1의 규칙만이 실행 가능하며, 이 규칙의 실행 결과로 gast.dat의 파일로부터 사실들을 읽어들인다. gast.dat는 그림 1의 소화기 질환의 결정 트리로부터 구성된 사실들로 구성된다. 사실과 규칙이 각각 별도의 파일로 구성되었음을 유의하라.

R3의 규칙에서, LHS상의 "<-"의 기호는 패턴 바인딩(pattern binding) 연산자를 나타낸다. 이 연산자는 왼쪽 상에 표현된 패턴에 일치하는 사실을 오른쪽 상의 변수에 바인딩시킨다. ?name은 어떠한 값에도 매칭될 수 있는 패턴을 나타낸다. R3이 실행 조건과 실행 결과는 다음과 같다:

실행조건: (current-node ?name)으로부터 식별되는 현재 노드의 타입이 decision이고, (answer yes)의 사실이 존재한다. ?node가 (current-node ?name)의 사실을 가리키고, ?answer가 (answer yes)의 사실을 가리킨다.

실행결과: ?node와 ?answer가 가리키는 사실들을 사실 리스트로부터 제거하고, (current-node, ?yes-branch)의 사실을 사실 리스트에 삽입한다. ?yes-branch에는 현재 노드의 yes-node 슬롯에 저

```
(deftemplate node
    (slot name) (slot type) (slot question)
    (slot yes-node) (slot no-node) (slot answer))

R1 : (defrule initialize
    (not (node (name root)))
    =>
    (load-facts "gast.dat")
    (assert (current-node root)))

R2 : (defrule ask-question
    ?node <- (current-node ?name)
    (node (name ?name)(type decision)(question ?question))
    (not (answer ?))
    =>
    (printout t ?question "yes or no")

    (assert (answer (read))))
R3 : (defrule proceed-yes
    ?node <- (current-node ?name)
    (node (name ?name)(type decision)(yes-node ?yes-branch))
    ?answer <- (answer yes)
    =>
    (retract ?node ?answer)
    (assert (current-node ?yes-branch)))

R4 : (defrule proceed-no
    ?node <- (current-node ?name)
    (node (name ?name)(type decision)(no-node ?no-branch))
    ?answer <- (answer no)
    =>
    (retract ?node ?answer)
    (assert (current-node ?no-branch)))

R5 : (defrule answer-print
    ?node <- (current-node ?name)
    (node (name ?      name) (type answer) (answer ?value))
    (not (answer ?))
    =>
    (printout t ?value crlf)
```

그림 4. 소화기 질환 지식베이스의 규칙들

장된 값 즉, 노드 이름을 나타낸다.

3. 구 현

여기서는 2장에서 설계한 소화기 질환 지식베이스와 연동을 갖는 전문가 시스템 구현 사항에 대해서 기술한다. 그림 5는 시스템의 전체 구조를 보여준다. 시스템은 크게 client 측과 server 측으로 구분된다. client 측은 웹 브라우저로 구성되며, 시스템 측은 시스템 제어 모듈, Jess 엔진, 지식베이스, I/O 인터페이스로 구성된다. 시스템 제어 모듈은 사용자와 Jess

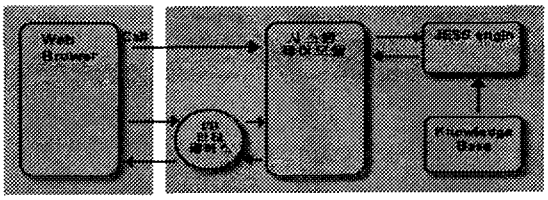

그림 5. 시스템의 전체 구조도

엔진과의 인터페이스를 위한 모든 제어를 담당하며, Java Servlet을 사용하여 구현된다. I/O 인터페이스는 시스템과 사용자와의 인터페이스를 담당하며, JSP를 사용하여 구현된다. Jess 엔진은 Jess에서 제공되는 추론 엔진이며, 지식베이스는 소화기 질환 지식베이스로 2장에서 구축한 사실과 규칙들로 구축된다.

Client가 웹 브라우저 상에서 시스템을 호출하면, 시스템 제어 모듈인 Servlet이 실행된다. 이 모듈은 I/O 인터페이스를 통해서 사용자에게 질문을 제시하고, 사용자의 응답을 대기한다. 사용자가 응답을 하면, 시스템 제어 모듈은 이 응답에 기초하여 Jess 엔진과의 인터페이스를 가지면서 규칙을 실행시킨다. 규칙의 실행 결과는 다시 시스템 모듈과 I/O 인터페이스를 순서대로 거치면서 사용자에게 전달된다. I/O 인터페이스는 사용자에게 보여지는 웹브라우저의 화면을 구성하고, Jess 엔진으로부터 출력되어 시스템 제어 모듈을 통해서 전달된 내용을 보여주는 역할을 수행한다.

Jess는 다음 함수를 통해서 시스템 제어 모듈과 Jess 엔진간에 값을 주고받을 수 있는 방법을 제공한다: store와 fetch. store는 (store name value)의 형식을 가지며, name의 기억공간 이름에 value의 값을 저장한다. 값 value의 타입은 Jess에서는 jess.Value 타입의 객체이고, Java에서는 임의의 객체일 수 있다. fetch는 (fetch name)의 형식을 가지며, name의 이름을 갖는 기억공간에 저장된 값을 반환한다. 이러한 두 함수를 사용하여 Jess와 시스템 제어 모듈간에 데이터를 주고받을 수 있다. 그림 6은 store와 fetch 함수를 사용한 Jess 엔진과 시스템 제어 모듈간의 인터페이스를 보여준다.

이제, 그림 4의 소화기 질환 지식베이스의 규칙 중에서 규칙의 실행 결과를 콘솔이 아닌 시스템 제어 모듈에 전달할 수 있도록 R2와 R5를 다음과 같이 각각 수정한다.

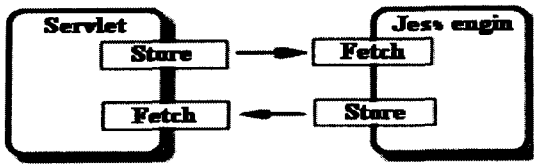

그림 6. Servlet 모듈과 Jess간의 인터페이스

```
R2 : (defrule ask-question
        ?node <- (current-node ?name)
        (node (name ?name)(type
        decision)(question ?question))
        (not (answer ?))
        =>
        (store "output" ?question)
        (store "type" "question"))

R5 : (defrule answer-print
        ?node <- (current-node ?name)
        (node (name ?name) (type answer)
        (answer ?value))
        (not (answer ?))
        =>
        (store "output" ?value)
        (store "type" "answer"))
```

4. 실행 예제

지금까지 소화기 질환의 지식베이스와 이 지식베이스를 갖는 전문가 시스템의 구현에 관해서 논의하였다. 여기서는 특정 예제를 통해 구현한 전문가 시스템의 실행 예제를 보인다. 실행 예제에서 사용된 결정 트리는 그림 1과 같다. 이 결정 트리상에서 사용자가 표 1의 단계로 응답했을 때 전문가 시스템의 실행 과정을 단계별로 보여준다. 표 1의 사용자 질문 시퀀스는 그림 1에서 굵은 선으로 표시된 트리 경로와 일치하는 것을 알 수 있다.

표 1. 실행 예제: 사용자의 질문 시퀀스

단계	질문사항	사용자 응답
1	아픈부위를 선택해주세요	우상복부 선택
2	황달이 있으십니까?	No 선택
3	열이 있으십니까?	Yes 선택
4	숨쉬기가 곤란한가요?	No 선택
5	병명 출력: 담도석 간농양	

다음은 표 1의 사용자 질문 단계별로 시스템이 실행되는 과정을 보여준다. 그림 7은 전문가 시스템을 처음 실행시켰을 때 나타나는 초기 화면이다. 사용자는 그래픽으로 구현된 복부 이미지로부터 아픈 부위를 선택할 수 있다.

사용자가 복부 이미지에서 우상 복부를 선택하면, 우상복부에 대한 결정 트리가 지식베이스로부터 불려지고, 그 결정 트리 상에서 경로는 그림 1의 루트 노드에 위치한다. 이 노드의 타입은 질문 노드이고, 명세된 질문은 "황달이 있으신가요?"이다. 따라서 그림 8의 화면이 제시된다. 이 화면은 질문 화면으로 사용자에게 질문을 제시하는데 사용된다. 그림 8에서 볼 수 있듯이, 질문 화면은 2개의 박스와 4개의 버튼으로 구성되어 있음을 알 수 있다: 질문 박스, 히스토리 박스, YES 버튼, NO 버튼, BACK 버튼, RESET 버튼. 표 2는 질문 화면의 각 구성 요소에 대해서 설명한다.

그림 8에서 사용자가 No 버튼을 누르면, 결정 트리 상의 현재 위치는 그림 1의 node8이다. 이 노드의

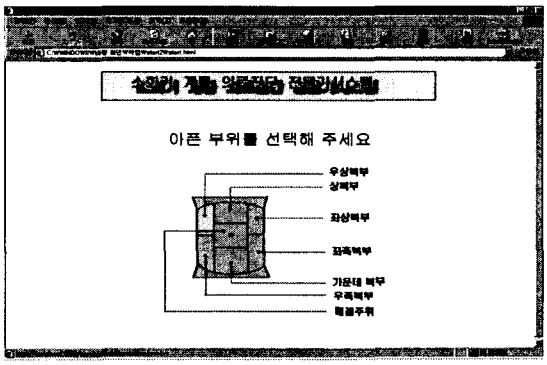

그림 7. 초기 화면

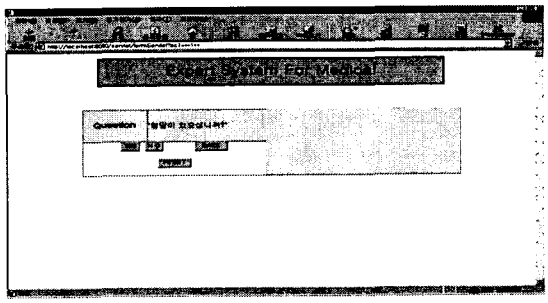

그림 8. 질문 화면(1)

표 2. 질문 화면의 구성 요소

구성요소	설 명
질문 박스	질문 사항이 출력된다.
히스토리 박스	질문 박스 우측에 위치하며, 사용자가 지금까지 질문한 내역이 (질문/답변)의 리스트 형태로 출력된다.
YES 버튼	질문 사항에 대해서 사용자가 yes로 반응하고자 할 때, 이 버튼을 누른다.
NO 버튼	질문 사항에 대해서 사용자가 No로 반응하고자 할 때, 이 버튼을 누른다.
BACK 버튼	현재 질문 사항에 대한 반응을 취소할 때 사용한다. 이 버튼이 눌러지면 바로 직전의 화면 상태로 복귀한다.
RESET 버튼	전문가 시스템을 처음부터 시작시킬 때 사용한다. 이 버튼이 눌러지면, 그림 13의 초기 화면이 나타난다.

타입은 질문 노드이고, 명세된 질문은 "열이 있으신가요?"이다. 따라서 그림 9의 질문 화면이 제시된다. 이 화면의 질문 박스에 node8의 질문이 제시되는 것을 알 수 있다. 그리고 히스토리 박스에는 그림 8에서의 질문과 사용자의 답변에 제시된다.

그림 9에서 사용자가 Yes 버튼을 누르면, 결정 트리 상의 현재 위치는 그림 1의 node9이다. 이 노드의 타입은 질문 노드이고, 명세된 질문은 "숨쉬기가 곤란합니까?"이다. 따라서 그림 10의 질문 박스에 이 질문이 제시된다. 그리고 히스토리 박스에는 지금까지의 질문과 사용자 답변의 리스트가 출력된다.

그림 10에서 사용자가 No 버튼을 누르면, 결정 트리 상의 현재 위치는 그림 1의 node12이다. 이 노드의 타입은 답변 노드이고, 명세된 병명은 "담도석 간농양"이다. 이러한 병명이 그림 11의 화면에 디스플레이되는데, 이 화면은 지금까지 보아 온 질문 화면과

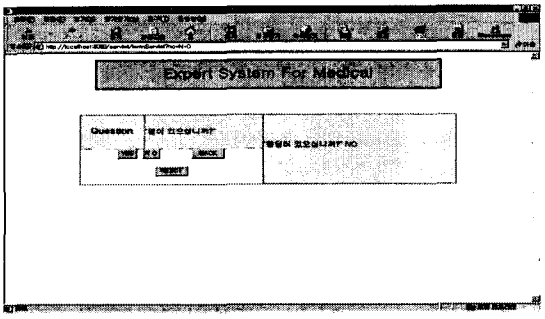

그림 9. 질문 화면(2)

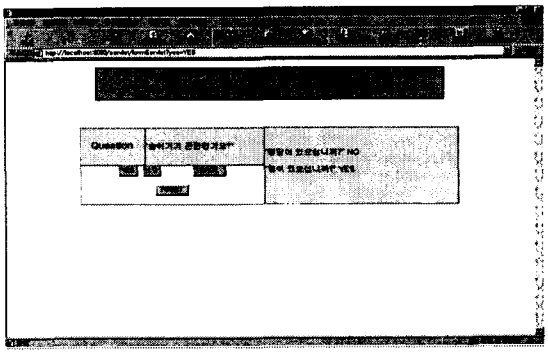

그림 10. 질문 화면(3)

는 다른 답변 화면이다. 답변 화면이 3개의 박스와 한 개의 버튼으로 구성되어 있음을 알 수 있다: 답변 박스, 히스토리 박스, 설명 박스, RESET 버튼. 답변 박스에는 지금까지 사용자가 질문에 대한 응답 결과로서 병명이 출력되는 곳이며, 설명 박스는 답변 박스에 출력된 병명에 대한 설명이 출력되는 곳이다. 히스토리 박스와 RESET 버튼의 용도는 표 2와 동일하다.

그림 11의 화면의 답변 박스에 병명이 출력되고 히스토리 박스에는 지금까지의 질문과 사용자 답변에 대한 리스트가 순서대로 출력되고, 설명 박스에는 이 병명을 설명하는 사항이 출력되는 것을 볼 수 있다. 이때 사용자는 시스템의 초기 화면으로 돌아가기 위해서 RESET 버튼을 누를 수 있다.

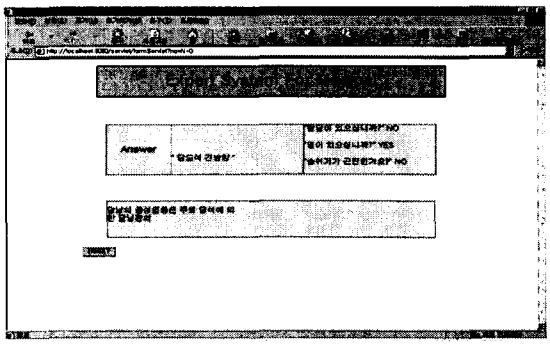

그림 11. 답변 화면

5. 결 론

본 논문에서는 소화기질환 전문의 임상교수로부터 전문 지식을 획득하여 소화기질환 지식베이스를 구축하고, 이 지식베이스와 연동하여 사용자에게 의

료 정보서비스를 제공할 수 있는 소화기 질환 의료 전문가 시스템을 전문가 시스템 쉘인 Jess를 사용하여 개발하였다. 이 시스템은 Java 기술로 개발되었고, 따라서 플랫폼에 독립되어 어떠한 시스템 환경에서도 사용될 수 있다. 개발된 시스템은 인터넷 상에서 사용 가능하고, 친숙한 사용자 인터페이스, 처방에 대한 설명 기능 제공, 신뢰성, 확장성 등의 특징을 가지며, 특히 지식베이스를 구성하는 사실들과 규칙들이 통합되지 않고 별도의 파일로 구성함으로써 지식베이스의 수정과 확장을 용이하게 하는 특징을 갖는다.

전문가 시스템이 잘 활용되려면 신뢰성을 지녀야 하는데, 이를 위해서는 사용자의 질문에 정확한 답변을 할 수 있는 것이 중요하다. 정확한 답변은 정확한 지식에서 오는 것이고, 이러한 지식이 지식베이스로 표현된다. 지식베이스는 분야의 전문가가 작성해야 하는데, 보통 전문가는 전문가 시스템을 잘 다룰줄 모른다. 따라서 시스템 전문가의 해당 분야 지식 전문가로부터 지식을 가져와서 지식베이스를 구축하게 되는데, 이러한 지식 전달 과정에서 어느 정도의 정확성 손실이 초래될 수 있다. 따라서 전문적인 지식을 갖는 전문가가 지식베이스를 직접 손쉽게 구축할 수 있게 하고, 이렇게 구축된 지식베이스를 갖는 전문가 시스템을 구축할 수 있는 전문가 시스템 도구의 개발이 필요하다. 앞으로, 지식의 전문가가 지식베이스를 손쉽게 구축할 수 있는 도구를 개발할 계획이다.

참 고 문 헌

[1] http://sungkyul.edu/~mchoo/planner/ai/intro.htm

[2] Xudong W. Yu Ph.D., Jerry Weinberg Ph.D., Mei C. Huang, M.D. Ph.D. and Dan K. Anderson, M.D., "HDA-An Internet-Enabled System for Healthcare Management", 2000 IEEE International Conference on Systems, Man and Cybernetics - Vol.3, 1836-1841, 2000.

[3] Alison Cawsey, Floriana Grasso, and Ray Jones, "A Conversational Model for Health Promotion on the World Wide Web", Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, 1999.

[4] Wayne Goodridge, Hadrian Peter, Akin Abayomi, "The Case-Based Neural Network Model and Its Use in Medical Expert Systems", Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, 1999.

[5] Michael Yawn, J2EE and JAX, reading, PHPTR, 2003.

[6] Friedman-Hill, E., JESS: The Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/

[7] Charles L. Forgy, "Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem", Artificial Intelligence 19 (1982), 17-37.

[8] Marth Hall, Core Servlets and JavaServer Pages, Reading, PHPTR, 2000.

[9] Duane K. Fields and Mark A. Kolb, Web Development with Java Server Pages, Readng, Manning, 2000.

[10] Culbert C., Riley G., Donnell B., CLIPS Reference Manual, Volume 1-3, Johnson Space Center NASA, 1993.

[11] Cay S. Horstmann and Gary Cornell, core Java 2, Reading, PTR PH. 1999.

하 상 호

1988년 서울대학교 계산통계학과(학사)

1991년 서울대학교 계산통계학과(석사)

1995년 서울대학교 전산학과(박사)

1995년~1996년 한국전자통신연구원 박사후 연구원

1996년~1997년 MIT LCS 박사후 연구원

1997년~현재 순천향대학교 정보기술공학부 부교수

관심분야 : 프로그래밍언어, e-commerce, ubiquitous computing.

유 일 대

2001년 2월 순천향대학교 정보기술공학부(학사)
2003년 2월 순천향대학교 정보기술공학부 전산학 전공(석사)

관심분야 : 전문가 시스템, 웹 프로그래밍

천 인 국

1983년 서울대학교 전자공학과(공학사)
1985년 한국과학기술원 전자공학과(공학석사)
1993년 한국과학기술원 전자공학과(공학박사)
1983년~1988년 삼성전자 종합연구소
1993년~현재 순천향대학교 교수
관심분야 : 영상처리, 컴퓨터 비젼, 인터넷 응용 기술

박 상 홈

1985년 순천향대학교 의학과 (의학사)
1985년~1989년 순천향대학교 부속병원 인턴 및 내과전공의 수료
1991년~2001년 순천향대학교 천안병원 소화기내과 부교수
2001년 10월~2003년 2월 미국 Indiana University Medical Center 연수
2003년~현재 순천향대학교 천안병원 소화기내과 부교수
관심분야 : 췌장 및 담도 질환

김 선 주

1976년 한양대학교 의학과 (의학사)
1976년~1981년 한양대학교 부속병원 인턴 및 내과전공의 수료
1992년 순천향대학교 의학과 (의학박사)
1984년~현재 순천향대학교 천안병원 소화기내과 교수
전공분야 : 위장관 질환

교 신 저 자

하 상 호 336-745 충남 아산시 신창면 읍내리 산 53-1 순천향대학교 정보기술공학부