

# 무선 기기를 위한 전문가 시스템의 설계 및 구현

유일대\*\*· 하상호\*  
순천향대학교 정보기술공학부  
\*\*youyou2@sch.ac.kr, \*hsh@sch.ac.kr

## Design and Implementation An Expert System for Mobile Devices

Il-Dae You\*\*, Sangho Ha\*  
Dept. of Informatin Technology SoonChunHyang University

### 요 약

현재 모바일 기기가 대중화되고 이를 위한 다양한 콘텐츠들이 개발되고 있다. 하지만 모바일 기기에서 사용 가능한 전문가 시스템은 그 유용함에도 불구하고 아직 개발이 미약한 것이 현실이다. 논문에서는 모바일 환경, 특히 무선 폰상에서 사용가능한 전문가 시스템을 개발한다. 이 시스템은 서버-클라이언트 구조로 개발되는데, 이들간의 인터페이스는 XML문서를 통해서 이루어진다. 마지막으로, 이 시스템의 실행 예를 소화기 질환의 의료지식베이스에 대해서 보여준다.

### 1. 서론

현재 모바일 기기가 대중화되면서 다양한 콘텐츠들이 제공되고 있다. 금융거래, 상품거래, 결제 등을 지원하는 모바일 커머스를 비롯해 교통정보 제공, 쇼핑, 뱅킹, 주식, 쿠폰등 다양한 콘텐츠들이 쏟아져 나오고 있다. 하지만 모바일 기기에서 사용하기 위한 전문가 시스템의 개발은 그 장점에도 불구하고 미비하다. PDA기기에서 사용하기 위한 전문가 시스템의 개발이 이루어지고 있지만 아직 문제점이 많다. 우리는 이러한 현실에 관심을 갖고 모바일 기기에서 사용하기 위한 전문가 시스템을 개발할 것이다.

모바일 기기에서 사용할 수 있는 전문가 시스템은 많은 장점을 갖는다. 모바일 기기는 휴대가 용이하기 때문에 때와 장소를 가리지 않고 전문가 시스템을 사용할 수가 있다. 일반적인 전문가 시스템을 사용할 수 없는 상황이라면 이점은 급할 때 많은 도움이 된다. 그리고 모바일 기기는 사용법이 간단하기 때문에 복잡한 사용법을 숙지하고 있지 않아도 누구나 쉽게 사용할 수가 있다. 일반인들도 쉽게 사용할 수가 있다. 또 모바일에서 사용할 수 있는 전문가 시스템

은 전문가의 작업을 언제 어디서나 훌륭히 보조할 수 있다. 이러한 점들은 모바일 기기를 위한 전문가 시스템의 개발이 매우 큰 의미를 갖는 것임을 말해준다.

전문가 시스템은 서버와 클라이언트로 나뉜다. 클라이언트에서 사용자의 응답을 받으면 그 응답은 서버로 전달되어 지식베이스에 접근한다. 지식베이스로부터 다음 추론결과를 가져오는데 이 추론결과는 XML 코드에 적재되어 클라이언트로 보내진다. 클라이언트는 XML코드를 파싱하여 정보를 가져온다. 정보는 사용자에게 보여지고 다시 클라이언트는 사용자의 응답을 기다린다. 그리고 위의 동작을 반복하여 결과값을 얻어낸다.

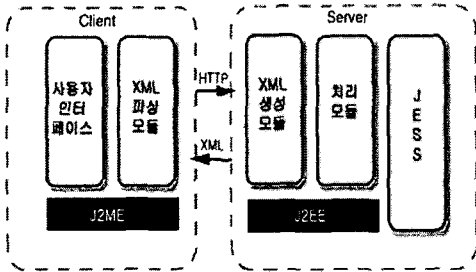
전문가 시스템내의 지식베이스의 효율적인 관리를 위하여 JESS[1]를 사용한다. 그리고 서버-클라이언트의 구조를 사용하여 모바일 기기의 시스템 부하를 줄일 것이다.

2장에서는 시스템의 전체적인 구조에 대해 설명하고 3장에서는 서버 모듈에 대해 그리고 4장에서는 클라이언트 모듈에 대해 설명할 것이다. 5장에서 실행의 예제를 보이고 6장에서 결론을 논하고 마칠 것이다.

\*본 연구는 정보통신부의 대학 IT연구센터 육성 지원 사업에 의해 수행된 것임

## 2. 시스템 구조

그림 1은 모바일 폰(Mobile Phone)에서 사용할 수 있는 전문가 시스템의 전체적인 구조이다.



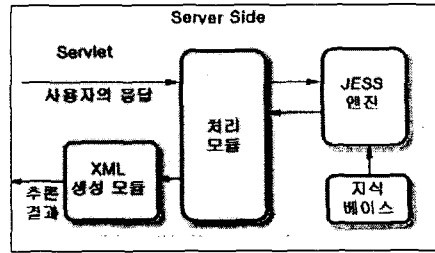
<그림 1> 시스템의 전체 구조

먼저 Client를 통해 사용자의 응답을 받는다. 이 응답은 전문가 시스템의 질문에 대한 YES/NO가 될 것이다. 이 정보는 Server의 처리모듈로 보내져 지식베이스에 접근한다. 지식베이스의 접근은 JESS를 사용한다. 지식베이스에서 가져오는 추론 결과는 XML 생성 모듈로 전달되고 XML생성 모듈에서 정보를 XML코드로 변환해 Client 즉 모바일 폰으로 다시 전달한다. Client의 XML Parsing모듈에서 XML코드를 파싱한 뒤 추론 결과를 사용자 인터페이스를 통해 사용자에게 보여준다.

Client에서 Server로 정보를 보낼 때는 HTTP 커넥션을 사용하고, Server에서 Client로 정보를 보낼 때는 XML코드를 사용한다. Server에서 Client로 보내는 정보는 복잡하기 때문에 XML은 이런 형태의 데이터 교환에 매우 적합하다. Client는 J2ME[2]환경에서 작성한 MIDlet[3]으로 구현하고 Server는 J2EE[4]환경에서 작성한 Servlet[5]으로 구현할 것이다.

## 3. 서버 모듈

그림 2는 서버모듈의 구조를 보여준다. 서버 모듈은 XML생성 모듈, 지식베이스 접근 모듈, JESS엔진, 그리고 지식베이스로 구성되어 있다. Client로부터 사용자의 응답이 전송되면 지식베이스 접근 모듈은 JESS를 사용해 지식베이스에 접근한다. 가져온 추론 결과는 XML생성 모듈에서 XML코드로 변환한 뒤 Client모듈로 전송한다. 지식베이스에는 전문가의 정보를 데이터화하여 저장하고 있다.



<그림 2> 서버 모듈의 구조

### 3.1 처리 모듈

Client가 사용자의 응답을 받으면 Server의 처리 모듈로 전송된다. 처리 모듈은 JESS엔진을 이용해 지식베이스에 접근한다. 지식베이스에는 전문가의 지식을 체계화 하여 저장하고 있다. 지식베이스로부터 가져온 추론 결과는 사용자의 응답이 YES인지 또는 NO인지에 따라 달라진다. 이 추론 결과는 XML생성 모듈로 전달된다.

### 3.2 XML생성 모듈

XML생성 모듈에서는 클라이언트에 보낼 정보를 XML코드로 변환한다. 사용자에게 보낼 정보는 크게 QnA와 History 두가지로 나뉜다. QnA는 사용자에게 할 질문이나 결과값을 포함하고 History는 사용자에게 제시한 질문들과 이러한 질문들에 대한 사용자의 응답의 결과들로 구성된다. 그림 5는 이러한 두가지 정보를 표현하기 위한 XML의 DTD를 보여준다.

```
<!ELEMENT ExpertSystem (QnA, History)>
<!ELEMENT QnA (Question, Answer)>
<!ELEMENT History (HistoryQnA)*>
<!ELEMENT HistoryQnA (Question,Answer)>
<!ELEMENT Question (#PCDATA)>
<!ELEMENT Answer (#PCDATA)>
<!ELEMENT Question (#PCDATA)>
<!ELEMENT Answer (#PCDATA)>
```

<그림 3>클라이언트와 서버간의 인터페이스를위한 XML DTD

Question tag는 사용자에게 하게될 질문, Answer tag는 사용자에게 보여줄 결과값을 가지고 있다. HistoryQnA tag는 History정보를 뜻하며 HistoryQnA tag내의 Question tag는 사용자에게 제시한 질문, Answer tag는 질문에 대한 사용자의 응답을 뜻한다. HistoryQnA tag는 여러번 반복이 가능하다. 위의 DTD를 통해 작성된 XML코드를 Client로 보낸다.

3.2 JESS 엔진

처리 모듈에서 지식베이스에 접근하기 위해 JESS 엔진을 사용한다. JESS 엔진은 Java와 연동 가능한 전문가시스템 셸(Java Expert System Shell)로써 CLIPS[7] 라는 C로 작성한 전문가 시스템 셸을 Java로 다시 작성한 것이다. JESS엔진을 이용하면 효율적으로 지식베이스를 만드는 것이 가능하고 지식베이스에 접근하여 사용자가 원하는 추론 결과를 가져올 수 있다.

본 시스템에서는 지식베이스에 접근하기 위해 JESS엔진을 사용한다. 지식베이스에 접근하여 추론 결과를 가져오는데 추론결과는 사용자에게 대한 질문 또는 결과 값이다.

3.3 지식 베이스

그림 4는 소화기 질환을 진단하는데 사용하는 지식베이스를 보여준다. 이 지식베이스는 Facts(사실)와 Rule(규칙)들로 구성되는데 그림 5와 그림 6은 각각 Fact와 Rule의 일부를 보여준다.

그림 5는 Fact내의 한 노드이다. Fact는 노드들의 집합으로 이루어져 있다. 노드는 name, type, question, answer, yes-node, no-node의 슬롯으로 이루어진다. name에는 노드의 이름을 명시하고 type에는 현재 노드가 질문 노드 인지 결과값 노드 인지를 정한다. 그리고 question과 answer에는 질문과 결과값이 저장되어 있고 만약에 값이 들어있지 않다면 nil이라는 값이 들어간다. yes-node와 no-node에는 사용자의 응답이 YES 또는 NO일때 이동할 노드의 이름이 들어간다.

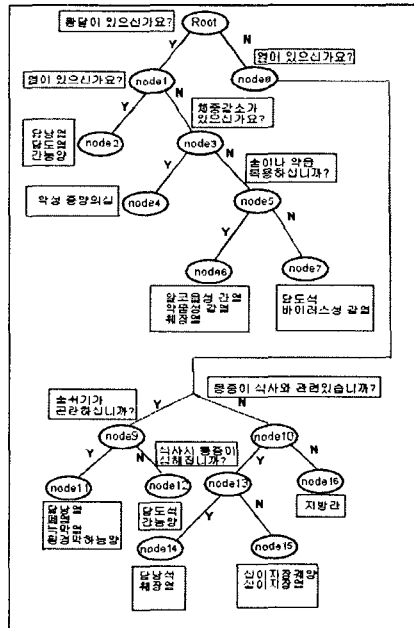
그림 6은 Rule의 일부를 보여주고 있다. Rule을 실행하면 먼저 initialize 규칙에서 load-facts명령어를 이용해 해당하는 fact를 읽어들인다. 다음에 ask-question 규칙에서 현재 노드의 질문을 출력하고 입력을 기다린다. 사용자가 yes를 입력할 경우 proceed- yes 규칙을 실행하고 해당하는 노드로 이동한다. 사용자가 no를 입력했다면 proceed-no 규칙을 적용해 해당하는 노드로 이동한다. 만약 answer노드로 이동했다면 answer-print 규칙을 실행해 결과값을 출력한다. 반대로 question노드로 이동하면 질문을 출력하고 위의 Rule을 다시 반복한다.

```
(node (name root)(type decision)(question
"기침가래 호흡곤란이 있습니까?")(yes-node
node2)(no-node node3)(answer nil))
```

<그림5>fact의 예

```
(defrule initialize
(not (node (name root)))
=>
(load-facts "fact의 이름")
(assert (current-node root)))
(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))))
(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)))
(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)))
(defrule answer-print
?node <- (current-node ?name)
(node (name ?name) (type answer) (answer ?value))
(not (answer ?))
=>
(printout t ?value crlf))
```

<그림 6>Rule의 일부

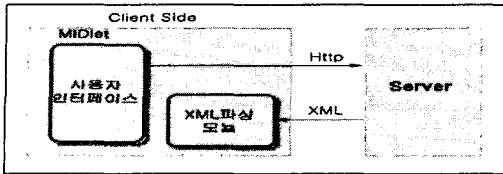


<그림 4>예제 지식베이스 결정트리

4 클라이언트

다음 그림 7은 클라이언트 모듈의 전체 구조이다. Server에서 보내온 XML코드는 Client의 XML파싱 모듈에서 파싱한다. 그리고 사용자 인터페이스에서 추

론결과를 보여주거나 사용자의 응답을 받는다.



<그림 7>클라이언트 모듈

#### 4.1 XML 파싱 모듈

Server로부터 받은 XML코드를 파싱 한다. 파서는 TinyXML[7]파서를 사용한다. 파싱을 통해 질문이나 결과값, 그리고 지금까지의 질문과 사용자의 응답 두 가지 정보를 얻게되는데 이 정보를 두개의 Displayable[4]에 적재한다. Displayable은 디스플레이에 나타날 수 있는 MIDlet내의 객체로써 여러개의 생성이 가능하다.

#### 4.2 사용자 인터페이스

사용자가 선택한 Displayable객체를 화면에 보인다. default로 질문이나 결과값을 가지고있는 Displayable을 보여주고 사용자가 History 버튼을 선택하면 지금까지의 질문과 사용자의 응답을 가지고있는 Displayable을 보여준다. 그리고 YES또는 NO라는 메뉴를 만들어 사용자의 응답을 받는다.

### 5. 실험 예

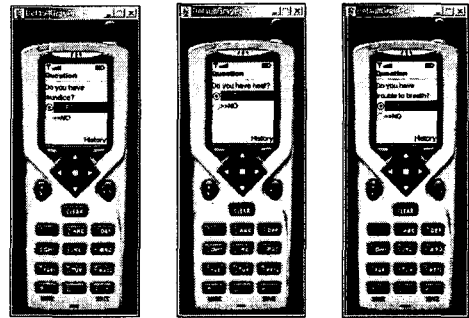
다음 (a)~(e)는 모바일상에서 전문가 시스템을 사용한 예제이다. 그림 4의 결정트리를 예제로 사용했다. 사용자가 전문가 시스템을 시작하면 서버는 JESS를 이용해 지식베이스에 접근한다. 그리고 “황달이 있으신가요?”라는 질문을 가져온다. 이것을 XML코드로 변경해 Client로 보낸다. Client에서는 XML코드를 파싱하고 정보를 Displayable 객체에 적재해 화면에 보인다. (a)는 모바일 폰에 출력된 화면이다. 그림 4의 지식베이스 트리에서는 Root노드에 해당한다.

사용자가 NO 라고 답하면 “열이 있으신가요?”라는 새로운 질문을 화면에 보여준다. (b)는 모바일 폰에 출력된 화면이다. 그림 4의 지식베이스 트리에서는 node8에 해당한다.

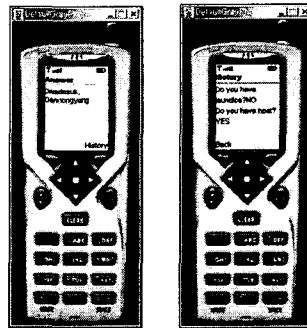
사용자가 YES 라고 답하면 “숨쉬기가 곤란하십니까?”라는 새로운 질문을 화면에 보여준다. (c)는 모바일 폰에 출력된 화면이다. 그림 4의 지식베이스 트리에서는 node9에 해당한다.

사용자가 NO라고 답하면 “담도석 간농양”라는 추론결과가 화면에 출력된다. (d)는 추론결과가 모바일

폰에 출력된 화면이다. 그림 4의 지식베이스 트리에서는 (d)에 해당한다. (e)는 History 메뉴를 선택했을 때의 화면이다. 지금까지의 질문과 사용자의 응답이 출력되는 것을 볼 수 있다.



(a) (b) (c)



(d) (e)

### 6. 결론

본 논문에서는 Mobile상에 서버-클라이언트 모델을 사용한 전문가 시스템을 연구하고 이를 구현하였다. 본 연구는 Mobile Phone만 있으면 어디서나 전문가 시스템을 사용할 수 있는 장점을 제공할 것이다. 향후 과제는 Client상의 사용자 인터페이스를 개선하여 사용자에게 더 편리하게 사용할 수 있도록 하며 다양한 분야의 전문가 시스템을 개발하는 것이다.

#### 참고 문헌

- [1] <http://herzberg.ca.sandia.gov/jess/>
- [2] Feng & Zhu, 'Wireless Java Programming with J2ME', SAMS, 2001
- [3] <http://java.sun.com/j2ee/>
- [4] <http://jakarta.apache.org/>
- [5] Joseph Giarratano, Gary Riley "Expert System Principles and Programming Second Edition", PWS Publishing Company, 1993
- [6] <http://www.ghg.net/clips/CLIPS.html>
- [7] <http://gibaradunn.srac.org/tiny/index.shtml>