

吳 吉 祿  
韓國電子技術研究所  
컴퓨터 연구부장/工博

# Expert System 개발

“  
일반적으로  
전문가 시스템은  
Identification  
Conceptualization  
Formalization  
Implementation  
Testing 과정으로  
구성된다.  
”

## 1. 전문가 시스템의 개관

1960년대 중반부터 A.I 분야는 전문가 시스템 개발에 있어서 비교적 성공을 거두었다. 이 분야는 어떤 전문적인 분야에 대하여 고성능의 프로그램을 작성하는 것으로, Domain에 관계없는 문제 해결 이론보다 인간의 전문기술에 기초가 되는 지식을 크게 중요시하고 있다. 최근 이 분야에 대한 새로운 이론, 기술 및 Tool들이 등장하여 지식 공학의 기초를 이루고 있다.

전문가 시스템의 기본이 되는 전문 지식은 크게 정해진 분야에 대한 지식, 정해진 문제 이해에 관한 지식 및 문제를 푸는 기술에 관한 지식으로 구성되며, 정해진 분야에 대한 지식은 Public한 부분과 Private한 부분으로 나누어진다. Public한 지식은 그 분야에 대한 사실, 정리, 이론 등이며, Private한 지식은 Heuristic한 것에서 나오는 규칙과 같은 것에 크게 의존한다. 인간은 이와 같은 Heuristic한 것에 의존하여 불확실한 데이터에서도 효과적으로 문제를 해결한다. 이러한 지식을 설명하고 재생산하는 것이 전문가 시스템 건설의 주된 일이다.

이 분야에 대한 연구자들은 Formal 추론 방법보다 지식 그 자체를 더 강조하고 있다. 그 이유는 첫째로 일상 생활의 어려운 문제들은 너무나 복잡하여 표현하기가 어렵고 분석하기도 어려워서 정확한 Algorithm이 있을 수 없기 때문에, 이러한 것들은 문제를 잘 표현하고 적당한 지식을 선별하여 적용함으로써 해결할 수 있다고 보기 때문이며, 둘째로 인간은 지식 때문에 탁월한 기술을 보인다고 생각하기 때문이다. 세째로 지식은 불충분한 것이지만 Private한 지식에 의해 지식은 세련되고 풍부해진다고 보기 때문이다. 즉 전문적인 능력은 전문적인 지식에 달려 있다. 왜냐하면 지식은 중요한 문제를 푸는데 기본이 되는 요소이기 때문이다.

전문가 시스템은 데이터 Processing 시스템과는 물론이고 다른 A.I 분야와도 다르다. A.I 분야는 Symbolic 표현, Symbolic 추론 그리고 Heuristic Search 등의 특징을 가지고 있어 데이터 Processing 시스템과는 다르다. 또 전문가 시스템은 여러 이유 때문에 광범위한 A.I 분야와는 다르다. 그것은 전문가 시스템이 일반적이 아닌 주어진 특수한 분야에 대한 문제 해결 방법을 사용하며, 자신의 추론을 위해 자신의 지식을 사용한다는 것과, 그 대상은 주로 해석, 진단, Debugging, Design, Planning, Monitoring, 수리, 교육, 제어 등이다.

### 2. 전문가 시스템 역사

1977년 IJCAI에 발표된 Feigenbaum의 논문에서 그는 전문가 시스템의 능력은 그것이 사용하는 특별한 Formalism이나 추론 방법에서가 아니라 그것이 가진 지식에서부터 나온다고 하였다. 이 말은 A.I 연구에 큰 변화를 주었다. 초기에는 몇가지의 추론 방법과 막강한 힘을 가진 컴퓨터의 도움으로부터 인간을 능가하는 능력을 기대했었기 때문이다.

Stanford 대학에서 시작한 (1968) DENDRAL Project는 이제 16년째 접어들고 있는데, 이 Project는 DENDRAL과 META-DENDRAL이라는 두 시스템을 만들었다. DENDRAL은 어떤 물질의 구조를 알아내기 위해 Spectrum 사진과 핵 자기 공명을 분석한다. 이것은 문제를 풀어가는데 Generate & Test라는 방법을 쓴다. 여기서 Generator는 부분적인 분자의 구조를 합성적으로 생성함으로써 데이터의 조건을 만족하는 모든 가능한 구조를 만들어낸다. 여기서 이 구조들 중에서 불가능한 구조를 제거하여 Exponential Search를 피하고 원하는 구조를 알아낸다.

MIT에서 개발된 (1971) MACSYMA는 Symbol로서 미적분을 처리하는 전문가 시스템이다. 이 시스템은 많은 수학자와 물리학자들에 의해 사용되어지고 있다. 이것은 응용 수학의 전문가로부터 얻은 수백 가지의 Rule을 이용한다. 이 Rule은 어떤 수식을 동등한 식으로 변환하는 방법을 나타내며, 이 Rule에 의해 어떤 식이 쉽게 풀릴 수 있는 간단한 식으로 변환되어 문제가 해결된다.

EXPERT는 (1978) 전문가 시스템을 만들기 위한 언어로서 원래 녹내장 치료 및 진단을 위한 전문가 시스템인 CASNET에서 유래되었다. 이 EXPERT는 주로 안과학, 내분비학, 류마티스학에 관한 진찰 모델을 만드는 데 사용되었다.

Carnegie Mellon 대학의 CADUCEUS는 Stanford 대학의 MYCIN과 함께 의학 진단용 전문가 시스템이다. CADUCEUS는 내과 분야의 질병과 그 증상 사이의 관계를 나타내는 Semantic Networks를 이용하는데, 1982년 현재 이 시스템은 10만개의 이 관계를 가지고 있다. 이 시스템은 또 여러 병을 구분하는데 필요한 다른 어떤 방법도 사용한다. MYCIN은 약 400개의 IF-THEN Rule을 사용하여 외부 데이터로부터 원하는 답을 찾는다. MYCIN의 문제 해결 방법을 이용한 다른 시스템은 TEIRESIAS와 PROSPECTOR 등이 있으며, MYCIN의 Domain Independent Version인 EMYCIN이 있다. EMYCIN에는 MYCIN의 Domain 지식을 제외하고 모든 것이 있는데 이것은 진단용 응용에 많이 사용된다. 이를 이용한 시스템으로는 PUFF와 ROSIE가 있다.

Carnegie Mellon 대학의 PGS는 (1973) 인간의 인식을 Modeling하고 연구하기 위한 Production 시스템 언어로서 OPS Series의 개발과 R1 전문가 시스템의 개발을 유도했다.

HEARSAY II 시스템은 Carnegie Mellon 대

학에서 개발된 Speech Understanding 시스템이다. 이 시스템은 1000여개의 단어로써 연결된 문장을 이해한다. Speech Understanding task는 가장 어려운 문제중의 하나이지만 많은 연구자들은 HEARSAY II 시스템의 기본 Idea는 앞으로의 전문가 시스템에 큰 역할을 하리라고 믿고 있다.

### 3. 전문가 시스템의 종류

대부분의 지식 공학 응용들은 아래와 같은 시스템들이다. 해석 시스템은 관측 데이터로부터의 자료를 받아서 상황을 묘사하는 시스템으로, 주로 Speech 이해, Image 분석, 화학 구조 분석, Signal 해석 등이다. 해석 시스템은 관측 데이터를 설명하고 이 데이터로부터 시스템의 상태나 상황을 설명한다.

Prediction 시스템은 주어진 환경으로부터 예견되는 환경을 추리한다. 이것의 보기로는 날씨 예보, 교통 예보, 인구 예보 등이다.

진단 시스템은 어떤 관측 데이터로부터 시스템의 Malfunction을 추리한다. 이런 종류의 시스템에는 의학, Electronic, Mechanical 및 소프트웨어 진단 등이 있다. 이 시스템은 주로 관측된 오동작에 관한 데이터와 그 원인과의 관계를 중요시하여 표현한다. 이에 대한 방법으로는 관측 결과와 그에 연관된 진단법에 대한 Table을 사용하기도 하고 또 다른 방법으로는 관측으로부터 관계된 진단을 찾아내기 위해 시스템 Design에 관한 지식과 그 시스템을 실제 Design 하거나 구현할 때 생기는 결점들에 대한 지식을 사용하는 방법이 있다.

Design 시스템은 어떤 Design 조건을 만족하는 시스템을 Design한다. 이런 종류의 시스템에는 Circuit Layout, 건물 Design 등이 있다.

계획 시스템은 행동을 Design하는 시스템으

로서 어떤 기능을 행하는 행동에 대하여 전문적인 지식을 가지고 있다. 이러한 시스템에는 자동프로그래밍 및 로봇, 프로젝트, 통신 등에 관한 계획 시스템이 있다. 이 계획 시스템은 계획된 행동의 현상을 추론하기 위해 행위자의 행동에 관한 모델을 이용한다.

감시 시스템은 어떤 시스템의 예상 결과를 좋지 못하게 하는 행동을 찾아내는 시스템이다. 이러한 감시 시스템은 핵 발전소 관리, 비행기 공중 감시 등에서 볼 수 있다.

Debugging 시스템은 어떤 시스템의 오동작에 대한 치료 방법을 제공한다. 이러한 시스템은 오동작의 치료 방법을 구하기 위하여 Planning, Design, Prediction 능력들을 사용한다. 이 분야의 응용으로는 컴퓨터 프로그램의 Debugging을 위하여 지식 베이스와 텍스트 에디터를 사용한 시스템이 있을 수 있다.

교육 시스템은 학생들의 행동을 진단하고 고쳐주는 시스템이다. 이 시스템은 진단과 Debugging을 위한 부분 시스템들로 구성되어 있다. 이러한 시스템은 먼저 학생들의 행동을 판단하는 가상적인 지식을 만들어 놓고, 학생들의 지식에 대한 취약 부분을 진단하여 적당한 치료법을 찾아낸 후, 그 치료법을 학생들에게 전달하기 위하여 전반적인 대화를 가진다.

마지막으로 제어 시스템은 시스템의 전반적인 행동을 관리하는 시스템이다. 이러한 일을 하기 위하여 제어 시스템은 시스템의 현재 상태를 계속적으로 관찰하고 미래를 예견하며 관련 문제를 진단하여 그의 치료 방법을 제시하며 그 결과를 관찰한다.

### 4. 전문가 시스템의 구성 요소

그림 1은 전문가 시스템의 이상적인 표현이다. 모든 전문가 시스템이 이 그림에 나타난 모든 요

## 컴퓨터 技術開發 강좌

소를 가지는 것은 아니지만 전문가 시스템은 보통 이들 요소의 몇 가지는 가지고 있다.

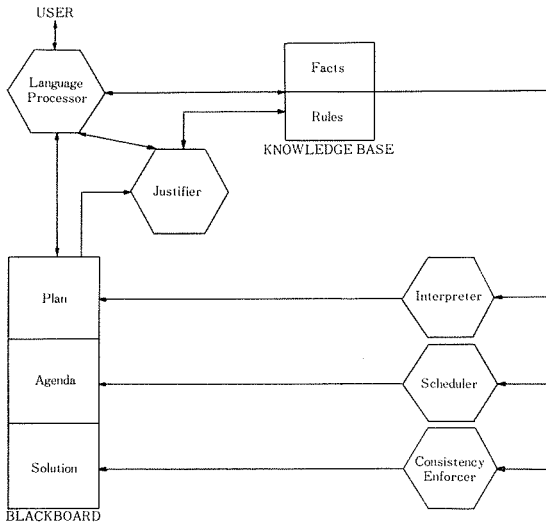


그림 1 전문가 시스템의 해부도

이상적인 전문가 시스템은 사용자와 전문가 시스템과의 대화를 위하여 언어 처리기(Language Processor)를 가지고 있다. 여기서 Blackboard는 중간 결과를 저장하고 있으며, Knowledge Base는 어떤 사실(fact)과 문제 해결 규칙(Rule)을 가지고 있으며, Interpreter는 이 규칙을 적용하고, Scheduler는 규칙 처리 순서를 조정하며, Consistency Enforcer는 새로운 Data가 입력되었을 때 이전 결과를 조정하며, Justifier는 시스템의 동작 이유를 합리적으로 설명한다. 사용자와 전문가 시스템과의 Interface는 보통 제한된 자연어를 사용하는데, 실존하는 전문가 시스템은 보통 Interlisp로 작성된 자연어 Parser를 쓴다.

Blackboard는 전문가 시스템이 동작되는 도중에 발생하는 중간 과정이 저장되는 곳이다. 거의 대부분의 전문가 시스템은 중간 과정의 표현

법을 가지고 있다. 앞의 그림에서와 같이 Blackboard는 세가지로 나누어진다. 여기서 Plan 부분은 문제를 풀기 위한 전반적인 방법을 가지고 있는데 이것에는 현재 Plan, Goal, Problem State, Context 등이 포함된다. 예를 들면 Plan은 먼저 입력 데이터를 분석하고, 기본적인 가설을 설정하고, 이 가설들을 세밀히 분석 정돈하여 하나의 가장 적절한 가설을 만들며, 여기서 최종 해답을 구한다. Agenda 부분은 수행을 기다리는 Action들을 저장한 곳이다. 이것은 일반적으로 지식 베이스의 Rule에 해당된다. Solution 부분은 시스템이 만들어낸 가설이나 결정들이 저장되는 곳이다. 여기서 최종 해답이 나온다.

Scheduler는 Agenda의 제어를 담당하는 곳으로 다음 수행될 Action을 결정하는 곳이다. 이 Scheduler는 상당한 지식을 가지고 있어서 원하는 답을 얻기 위해 각 Agenda Item에 대해서 적절한 우선 순위를 지정한다.

Interpreter는 주어진 Agenda Item을 대응되는 지식 베이스의 Rule을 적용시키므로써 수행한다. 일반적으로 Interpreter는 Rule의 조건 부분을 검사하여 이 조건의 변수들을 Binding시키고, 그 결과를 Blackboard에 저장한다. 보통 Interpreter는 Lisp에 의해 작성된다.

Consistency Enforcer는 계속 나오는 해답들의 서로 모순됨을 지적한다. 이것은 어떤 해답이 가장적인 진단일 때 새로운 데이터가 도달하면 그를 이용하여 일어날 만한 새로운 답을 만든다. 대부분의 시스템은 어떤 결정의 진실성 정도를 결정하기 위하여 수치적인 조정 방법을 이용한다. 이 방법을 이용하여 적절한 결론은 선택되고 모순되는 것은 제외된다.

Justifier는 시스템이 사용자에게 행한 행동의 이유를 설명한다. 사용자가 왜 이러한 결론이 일어났는지 혹은 왜 이러한 결론은 일어나지 않는

지를 물으면 Justifier가 이에 대한 대답을 한다. 이러한 일을 하기 위하여 Justifier는 Blackboard의 Solution을 추적한다.

마지막으로 지식 베이스는 Rule과 Fact 및 현

재 문제에 필요한 정보를 저장하고 있다. 여기서 지식 베이스의 Rule은 Procedural Interpretation을 위한 것이고 Fact는 단지 능동적인 역할을 한다.

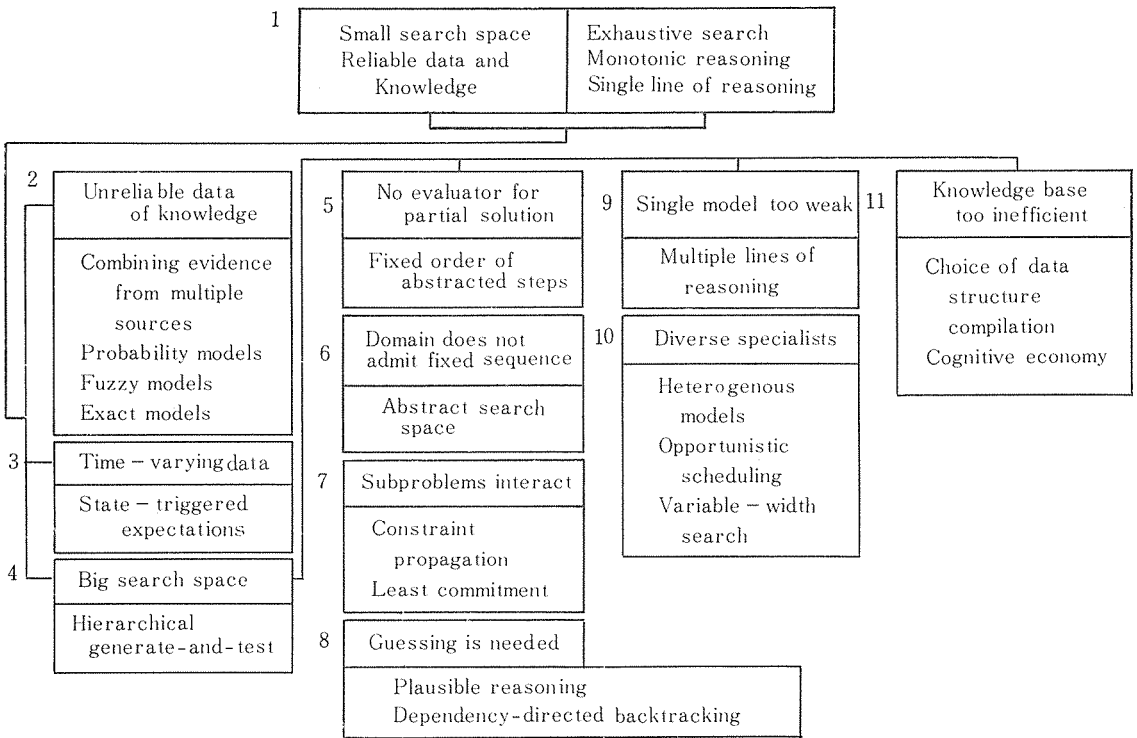


그림 2 점진적으로 복잡해지는 전문가 시스템의 여러 종류와 그 구조

### 5. 전문가 시스템의 구조

전문가 시스템의 구조는 문제의 특징에 따라 서 여러가지 Class로 나누어지며 이들 각각에 대해 적합한 처리 방법을 선택하여야 한다. 그림 2는 이들 문제들의 Class와 그에 대응되는 Design 방법을 그림으로 잘 나타내고 있다.

앞의 그림에서 Box 1은 가장 간단한 구조의 시스템이며 여기서 Box 2, 3, 4로 가면서 점점 복잡한 시스템이 된다. Box 2는 Unreliable

한 Data를 처리하는 시스템이며, Box 3은 Time Varing Data를, Box 4는 큰 Search Space를 가지는 시스템이다. 그리고 Box 4에서 제시된 큰 Search Space를 가지는 시스템에 사용되는 Hierarchical Generate & Test 방법에 추가되어 3개의 다른 종류의 구조를 나열하고 있다.

### 6. 전문가 시스템의 구성 방법

실제 전문가로부터 지식을 가지고 와서 기계

에 프로그램 형태로 변환시키는 과정을 Knowledge Acquisition이라고 한다. 이러한 변환은 전문가 시스템 개발 과정에서 가장 중요한 부분이다. 실제 전문가의 지식을 Formalize하고 밝히는 일은 Knowledge Engineer가 할 일이다. Knowledge Engineer는 문제를 정확하게 Define 하여야 하고 관계된 문제의 기본 개념을 알아야 하며 그들로부터 Rule을 발견하여야 한다. 일반적으로 전문가 시스템을 만들어 나가는 과정은 다섯 과정으로 나눌 수 있으며 그 과정은 Identification, Conceptualization, Formalization, Implementation, Testing이다.

Identification 과정은 Knowledge Engineer와 실제 전문가가 같이 일을 한다. 이 과정에서 이들은 문제를 정확히 밝혀서 그 범위를 결정하며, 개발 과정에 필요한 참가자와 필요한 자원을 결정하여, 전문가 시스템 건축의 목적을 정의한다. Conceptualization 과정에서는 Knowledge Engineer와 실제 전문가가 주어진 Domain의 문제 해결 과정에 필요한 기본 개념, 정보의 흐름에 관한 특징들을 분석한다. 또 이들은 문제에 관련된 Subtask, Strategy, 제한점 등을 지적한다. Formalization 과정에서는 전문가 시스템 건축 Tool을 이용하여 문제 해결에 필요한 기본개념을 표현한다. 여기서 Knowledge Engineer는 사용 언어를 결정하여야 한다. Implementation 과정에서 Knowledge Engineer는 Formalize된 Knowledge를 재조직하여 문제의 정보 흐름 특징

과 일치되도록 한다. 이 과정에서의 결과는 문제와 관련된 Rule과 Control 구조를 포함한 전문가 시스템의 시제품이다. 마지막으로 Test 과정은 시제품 전문가 시스템의 성능을 평가하고 주어진 문제 Domain에서 우수한 시스템이 되도록 고쳐나가는 과정이다.

## 7. 전문가 시스템 건축을 위한 Tool

표 1은 Knowledge Engineering에 관계된 일을 하기 위해서 필요한 Tool들이다. 이 Tool들은 OPS를 제외하고는 모두 Interlisp로 작성되어 있다. 이중 ROSIE는 전문가 시스템의 시제품을 만들기 위한 우수한 Tool이며, EMYCIN, KS300과 KAS도 진단 시스템을 위한 일반적인 Tool이다. KAS는 EMYCIN과 비슷한 기능을 가지고 있지만 EMYCIN 만큼 많이 사용되지는 않고 있으며, KS300은 EMYCIN에 기반을 둔 산업용 시스템이다. OPS는 Production System Interpreter로는 가장 고성능이며, AGE는 시스템을 만들 때 여러 구조로 다양하게 확장할 수 있는 기능을 제공한다. 그리고 Interlisp는 이 분야에서 가장 많이 쓰이는 언어이다.

전문가 시스템을 구성하는데 있어서 Tool을 선택하는 것은 대단히 중요하다. 적절한 Tool을 잘 선택하면 개발 기간과 성능에 큰 도움을 준다. 표 2는 전문가 시스템의 건설에 필요한 Tool을 선택하는데 고려될 사항이다.

(표 1)

Tool	Developer	Feature
ROSIE	Rand Corporation	Rule-based, Procedure-oriented, General
OPS 5	Carnegie Mellon University	Production system, General
EMYCIN	Stanford University	Rule-based Diagnostic and explanation
KAS	Stanford research institute	Rule-based, Diagnostic and explanation
AGE	Stanford University	LISP-based, Build various PS architecture
Interlisp	Xerox Corporation	Lisp program environment

〈표 2〉

Generality	Pick a tool with only the generality necessary to solve the problem
Selection	Let the problem characteristics determine the tool selected
Speed	When time is critical, choose a tool with built-in explanation / selection facilities
Testing	Test the tool early on by building a very small prototype system

## 8. 연구개발 계획

한국전자기술연구소는 이를 위해 4개년 계획으로 다음과 같이 추진하고 있다.

연 도	연 구 개 발 목 표	연구비(천원)
1 차 년도 (1984년)	<ul style="list-style-type: none"> <li>• A.I를 위한 기본 이론연구                             <ul style="list-style-type: none"> <li>- Knowledge Representation 및 Problem Solving에 관한 연구</li> <li>- A.I를 위한 환경조성</li> </ul> </li> <li>• A.I 언어 연구                             <ul style="list-style-type: none"> <li>- PROLOG, LISP 언어 연구</li> <li>- PROLOG, LISP Porting</li> <li>- 한글 LISP, 한글 PROLOG 개발</li> </ul> </li> <li>• A.I를 위한 Tool sets up</li> </ul>	65,000
2 차 년도 (1985년)	<ul style="list-style-type: none"> <li>• 한글 Natural Language 연구개발</li> <li>• A.I 응용시스템 연구                             <ul style="list-style-type: none"> <li>- Expert System</li> <li>- Computer Vision</li> <li>- Natural Language Processing</li> </ul> </li> </ul>	165,000
3 차 년도 (1986년)	<ul style="list-style-type: none"> <li>• Expert system의 시제품개발</li> <li>• 한글을 위한 Natural Language Processing의 Front-End개발</li> <li>• Computer Vision의 Prototype개발 System</li> </ul>	275,000
4 차 년도 (1987년)	<ul style="list-style-type: none"> <li>• Commercial Expert System 개발</li> <li>• 한글 자연어 처리 시스템 개발</li> <li>• Commercial Computer Vision System 개발</li> </ul>	368,000

## 9. 결 론

아직 많은 정보가 없고 전문연구인력이 부족하며 연구기자재를 거의 보유하지 않고 있기 때문에 이제부터 연구개발 환경을 마련하여야 한다는 어려움이 따르며 이보다 더욱 어려운 것은 대상이 되는 Expert System을 선정하여 해당

하는 분야에서 저희들과 같이 공감하고 같이 연구에 참여해 줄 Partner를 구하기가 대단히 어렵다. 혹 이 글을 보고 연구원 5명 정도와 그것에 필요한 연구비를 지원할 업체가 있다면 전자기술연구소는 본 사업을 국가 주도 연구사업에서 기업주도 연구사업으로 방향전환할 수도 있다.