

# 지능처리 기술

林 榮 煥  
(正 會 員)  
韓國電子通信研究所

## I. 서 론

인공지능이란 컴퓨터를 이용한 문제해결에 경험적인 지식을 어떻게 잘 이용할 수 있을까 하는 방법을 연구하는 분야라고 할 수 있다. 즉, 인공지능에서 가장 중요한 핵심적인 요소는 지식을 이용한다는 점이다.

컴퓨터가 지능(intelligence)을 가지고 있다는 것은 무엇인가? 우선 실시간(realtime)으로 수행하여 예상치 못한 상황의 입력에도 적용할 수 있고, 주위 환경에서 새로운 지식을 배워 시간이 지나감에 따라 발전이 있어야 한다는 것이다. 그러므로 인공지능 기술이 잘 적용될 수 있는 분야는 기존의 시스템에 지능(intelligence)을 부여하는 분야이다. 또 다른 인공지능의 특징은 문제의 성격상 인간만이 해결할 수 있는 분야, 혹은 문제를 해결할 수 있는 알고리즘(algorithm)이 존재한다 할지라도 기존의 기계로는 기억장치의 부족이나 수행시간이 너무 오래 소요되기 때문에 해결할 수 없는 분야를 해결해 보려는 것이다. 이를 위하여 인간의 경험적인 지식을 이용하려는 것이고 최선의 해결책을 찾을 수 없는 분야라 할지라도 차선책도 중요한 것이니까 받아들일 수 있는 대안이라도 찾으려는 시도이다.

이러한 인공지능 기술은 전산학이 존재하기 이전부터 연구되어 왔지만 한때는 절망적인 시기도 있었다. 최근 인공지능 기술을 이용한 시스템이 실제생활에 활용할 수 있게 됨에 따라 인공지능에 대한 연구가 세계 여러 나라에서 경쟁적으로 이루어지고 있지만 우리나라에서는 초보적인 단계에 있다. 일본의 5세대 컴퓨터 사업에 영향을 받아 국내에서도 인공지능에 대한 인식은 많이 하고 있지만 그에 대한 부작용으로 인공지능이면 모든 것을 해결해 줄 수 있는 것처럼 오해하고 있는 것도 사실이다. 한편으로는 기대한 만큼 새롭고 환상적인 제품이 나오지 않기 때문에 실망하고 있는

것도 사실이다.

여기서는 지능처리 기술이 무엇이며 그것이 어떻게 이용되고 있는지 살펴보았다.

인공지능 전 분야는 그림 1에서 처럼 계층적으로 구분해 볼 수 있다. 인공지능 기본기술은 지식을 컴퓨터에 표현하고 처리하는 기술에 관한 것이고, 인공지능 언어 및 도구는 기본기술을 응용시스템에 도입할 때 필요한 소프트웨어 도구 및 하드웨어 도구에 대한 것이다. 그리고 응용분야로는 최근에 상품화되어 나온 제품과 연구소나 학교에서 수행하고 있는 과제 중심으로 전개했다.

## II. 인공지능 기본 기술

인공지능은 경험적인 지식을 이용하는 학문이다. 여기 기본기술에는 지식을 이용하여 문제를 해결하기 위한 표현방법 및 추론과 계획수립(planning) 그리고 추

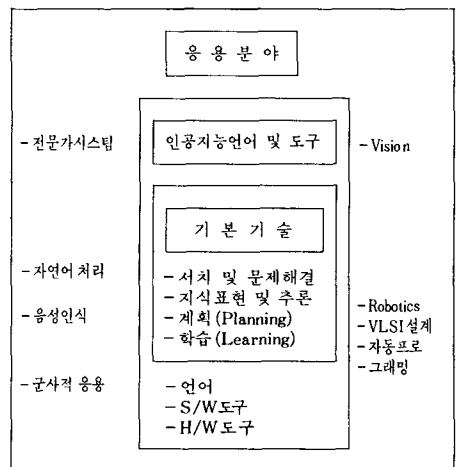


그림 1. 인공지능 기술개념도

위환경과 경험에서 배우는 학습(learning) 기술에 대하여 살펴보았다.

### 1. 서치(search) 및 문제해결(problem solving) 기술

인공지능 프로그램은 주어진 문제를 해결하기 위하여 여러가지 접근방식이 고안되었는데 다음 3 가지 방법이 대표적인 것이다.

- 상태공간방법(state-space approach)
- 문제축소방법(problem reduction approach)
- 정리증명방법(theorem proving approach)

이러한 문제해결 시스템은 기본적으로 문제표현(problem description), 연산자(operator) 집합, 그리고 제어전략(control strategy) 등 세가지 기본요소로 구성되어 있다. 인공지능이 실생활에 적용됨에 따라 위의 세가지 방법은 생성시스템(production system)이라는 정형화된 시스템으로 발전되었다.

#### (1) 상태공간 방법(state-space approach)

이 방법은 문제를 상태(state)로 표현하고 연산자는 문제가 나타낼 수 있는 모든 경우에 대하여 적용될 수 있는 상태를 변환시키는 연산자이다. 문제를 해결한다는 것은 초기상태(initial state)에서 적용할 수 있는 모든 연산자를 적용시켜 새로운 상태로 변환해 가면서 목적상태(goal state)를 찾는 것이다. 이 과정에서 제어 전략 즉 서치방법은 모든 경우의 수를 전부다 발생시키지 않고도 목적상태를 효과적으로 찾으려 하는 것으로 A\* 알고리즘이 대표적인 것이다. 이 방법은 순방향 추론 방법에 해당하는 것으로 문제 해결 과정을 일반적인 그래프(graph)로 모델화 하여 연구되어 왔다.

#### (2) 문제축소(problem reduction approach)

문제축소 방법은 주어진 문제를 해결하기 보다 쉬운 부분제(subproblem)로 축소시키는 과정을 반복하여 문제해결이 쉽거나 혹은 이미 해결된 문제인 원시적 문제(primitive problem)의 집합으로 축소시켜 해결하는 방식이다. 문제축소 방식은 AND/OR 그래프로 일반화되어 제어전략을 연구하는데 대표적인 것으로 A\* 알고리즘이 있다. 이 방식의 특수한 경우로 게임트리를 구성할 수 있는데, 이것은 어떤 게임에 있어서 가장 그럴듯 한 다음 수를 찾는 데 이용된다.

대표적인 서치방법으로는 미니맥스(mini-max) 서치와 알파베타( $\alpha$ - $\beta$ ) 서치방법 등이 있다. 문제 축소 방식은 원칙적으로 역방향 추론에 해당된다.

#### (3) 정리증명 방법(theorem proving approach)

이 방식은 문제를 해결하는 것은 정리를 증명하는

것과 마찬가지로의 관점에서 출발한다. 즉 문제의 정의나 공리, 그리고 가정 및 증명하고자 하는 정리를 논리적인 공식으로 나타내고, 정리가 문제의 논리적인 결과(logical consequence)인지 증명하는 방식이다. 대표적인 것으로 술어논리를 이용한 표현방식과 분해방법을 이용한 추론방법이 잘 개발되어 있다.

### 2. 지식표현 및 추론방법(knowledge representation and inference mechanism)

인공지능에 대한 중요성이 점차 인식되고 문제표현 방식이 문제해결의 효과에 중요한 영향을 미친다는 것이 밝혀짐에 따라 이 분야에 대한 연구가 최근 10여년간 활발하게 이루어져 있다. 지식표현은 단순한 지식 자체의 표현뿐 아니라 표현된 지식을 처리하는 방법도 동시에 연구되어야 한다. 즉 지식표현은 자료구조와 그것을 해석하여 컴퓨터가 지적인 행위를 할 수 있게 하는 해석과정의 결합이라 할 수 있다. 인공지능에서 다루는 지식은 주로 다음 네가지로 객체에 대한 지식, 사건에 대한 지식, 성능에 대한 지식, 그리고 지식을 위한 메타지식 등이다.

지식표현을 위한 원리로는 크게 두가지, 절차적(procedural) 표현방식과 선언적(declaration) 표현방식이 있다. 절차적 표현방법은 전통적인 알고리즘이나 프로시저를 지식으로 표현한 것과 마찬가지로서 각 선언적 지식간의 관계를 정의해 주는 지식을 나타내는 방식이다. 선언적 표현방법으로는 술어논리(predicate calculus) 방식, 생성규칙(production rule), 의미네트워크(semantic network), 프레임(frame)이나 스크립트(script) 등이 있다.

#### (1) 논리적 표현방식

인간의 지식을 표현하는 방식으로 논리(logic)는 이미 철학자나 수학자들이 오래전부터 사용하던 것으로 지식베이스는 논리적공식(logical formula)의 집합이다.

논리적 표현방법에는 현재 알고 있는 지식에서 다른 새로운 지식을 추출할 수 있는 추론 규칙이 잘 연구되어 있으며 표현방식이 단순하고 그 의미가 분명하다. 그러나 지식베이스를 구성하는데 조직화 할 수 없다는 단점도 있다. 논리적 표현방식은 주로 1차술어논리(first-order logic)를 많이 이용해 왔으며 현재 지식습득(knowledge acquisition), 믿음(belief), default 등의 문제로 확장되면서 더욱 중요한 분야로 연구되고 있다. 논리를 이용한 방법의 예로 STRIPS와 Omega 등이 있다. STRIPS(stanford research institute problem solver)는 로봇트가 물체를 재배치

하고 어지럽게 늘려져 있는 환경에서 돌아다니는 문제를 계획하고 해결하는 것을 목적으로 하는 시스템이다.

### (2) 의미 네트워크(semantic network) 표현방식

이 방식은 의미가 있는 노우드(node)와 아크(arch)로 구성한다. 노우드는 해당분야의 개체, 개념, 상황 등을 나타내며 아크는 그들간의 관계를 표현한다. 이런 의미 네트워크는 60년대 말부터 심리학자들에 의하여 인간 기억의 심리학적 모형으로 개발되어 왔다. 의미 네트워크가 갖는 가장 큰 특징은 상속기법이 쉽게 제공되는 것으로 논법이 주로 복잡한 분류(taxonomy)에 바탕을 둔 문제는 이 방식이 자연스러운 표현방식이다.

그러나 이 방식의 문제점은 표현된 구조가 무엇을 의미하는가 하는 정형적인 의미가 없다는 점이다. 네트워크 구조에 주어지는 의미는 그 네트워크를 다루는 과정의 성격에 의해서 결정되고 추론은 주로 매칭(matching)에 의해서 이루어진다. 이 방식을 이용한 시스템의 예로는 1968년 Raphael의 SIR 프로그램이 최초의 것이고 1976년 Walker가 SRI에서 주제영역을 표현하는데 이 방식을 사용했으며 BBN의 Woods는 자연어 처리에서 이 방식을 사용했다. Mylopoulos는 동료들과 TORUS 시스템을 위한 PSN이란 네트워크를 개발하였으며 MIT의 Winston은 시각적 지식을 표현하는데 일종의 의미 네트워크를 사용하였다. Bradman은 구조적 상속 네트워크라는 보다 복잡한 구조적 의미 네트워크를 사용하여 KL-1 언어를 개발하였다.

### (3) 프레임(frame)과 스크립(script)

MIT의 M. Minsky가 1975년 프레임 이론을 발표한 이후 이 방식은 지식표현의 수단으로 가장 널리 쓰이게 되었다.

프레임은 일단의 slot로 구성되어 있으며 각 slot은 프레임이 나타내고자 하는 개념 기술의 한 측면을 나타낸다. 이 slot의 값은 전형적인 값을 갖거나 특정상황에 맞는 특정한 값을 나타낸다. 이러한 프레임을 바탕으로 과거 경험에서 얻어진 개념을 통해서 새로운 데이터를 이해하는 구조가 제공된다. 한편 프레임은 전형적인 개념을 나타내고 비교에 의해 특정한 개념을 표현할 수 있기 때문에 심리학에서 발전된 원형(prototype) 이론을 쉽게 반영할 수 있다. 즉, 일반적인 프레임은 바로 객체의 원형을 나타내며 실증하는 이 원형과의 비교에서 이루어지는 것이다. 이것이 프레임 이론이 각광을 받는 가장 큰 장점중 하나이다.

스크립(script)은 프레임과 유사한 지식표현 수단으로 특정한 문맥에서 전형적인 사건의 연속을 기술하기

위한 구조이다. 스크립은 프레임보다 복잡한 구조로서 실제 세계에서 사건의 발생에 대한 유형을 갖고 있기 때문에 사건의 연속관계에 대한 기술에 더욱 유용하다. 즉 스크립은 사건이 어떻게 서로 관계를 가지면 어떤 순서로 이루어지는가를 나타낼 수 있는 장점이 있다.

스크립은 Yale 대학의 R. Schank팀이 주로 자연어 처리와 이해를 위한 프로그램에 지시기 표현 수단으로 사용하였다. 그 대표적인 것으로 SAM이 있다.

프레임은 발표된 이래 Bobrow팀이 GUS라는 자연어 이해 시스템을 개발하는데 이용되었다. 이와 동시에 Bobrow와 Winograd는 프레임에 바탕을 둔 프로그래밍 언어로서 KRL(knowledge representation language)를 개발하고 여러분야에 실험적으로 적용시켜 보았다. 또다른 이용으로 Goldstein과 Roberts의 NUDGE 시스템을 들 수 있다. 이 시스템은 불완전하고 모순이 있는 경영 예정 요구를 이해하여 기존의 예정 알고리즘에 대한 완전한 명세를 제공하는데 사용되었다. 이때 사용된 지식표현 언어는 FRL(frame representation language)로서 1977년 MIT의 B. Bruce Roberts와 Ira P. Goldstein에 의해 개발된 것이다.

지금까지 지식표현에 관한 것으로 몇가지 방법에 대하여 살펴보았다. 최근 지식표현의 경향은 지식베이스를 일의 성격에 따라 분류하고 각기 일의 성격을 가장 잘 대변할 수 있는 표현방법으로 나타내고 그위에 여러가지 표현기법을 조정하는 소프트웨어 도구가 많이 개발되고 있다.

### 3. 계획수립(planning)

계획수립이라는 것은 어떤 일을 실행하기 이전에 문제해결 행위의 전과정을 결정하는 것을 의미한다. 결국 계획은 행위과정에 대한 표현이다. 계획수립의 장점은 탐색량을 줄이고 목적의 상충을 해결하며 오류 회복의 기반을 마련하는데 있다.

인공지능에서 계획수립에 대한 접근방식은 크게 네 가지로, 비계층적(nonhierarchical), 계층적(hierarchical), 스크립(script), 기회적(opportunistic) 계획수립으로 분류할 수 있다.

비계층적 계획수립이라는 것은 문제의 각 목적을 성취하기 위하여 문제해결 행위의 연속을 개발하는 것이다. 이를 위하여 원래의 목적을 보다 단순한 부목적으로 축소시키거나 현 상황과 목적상황과의 차이를 줄이기 위한 Meanends 분석방식을 사용하기도 한다. 대표적인 비계층 계획자료는 STRIPS, HACKER, INTER-

PLAN 등이 있다.

계층적 계획수립 방식은 먼저 계획은 모호하지만 완전하게 윤곽을 잡은 뒤 이 모호한 부분을 좀더 상세한 부분계획으로 기술하며 이를 상세한 문제해결 연산자의 연속으로 구성될 때까지 반복한다. 이 방식의 장점은 처음부터 계획을 지나치게 상세할 필요가 없다는 것이다. 예로서 ABSTRIPS, NOAH, MOLGEN 등이 있다.

세번째 스크립(script) 계획방식은 개략적인 계획은 사용하지만, 계층적 계획수립방식에서 발견한 기회적 방식이다. 이들은 인간계획 수립을 모형화하기 위한 Blackboard라는 제어구조를 채택하였다. Blackboard는 계획단계에 대한 제어하는 것으로 이 제어는 계획수립을 잘하는 전문가가 만든다. 각 전문가는 특정한 종류의 계획수립을 결정하도록 하지만 특별한 순서로 운영하지는 않는다. 즉 어떤 이유가 발생했을 때만 계획수립의 결정을 내리게 된다.

네번째 접근방식은 Barbara Hayes-Rothe와 Friederick. Hayes-Roth가 인간의 계획수립 방식에서 발견한 기회적 방식이다. 이들은 인간계획수립을 모형화하기 위한 Blackboard라는 제어구조를 채택하였다. Blackboard는 계획단계에 대한 제어하는 것으로 이 제어는 계획수립을 잘하는 잘하는 전문가가 만든다. 각 전문가는 특정한 종류의 계획수립을 결정하도록 하지만 특별한 순서로 운영하지는 않는다. 즉 어떤 이유가 발생했을 때만 계획수립의 결정을 내리게 된다.

#### 4. 컴퓨터 학습

컴퓨터가 단순히 지시한 일만 따르지 않고 새로운 것을 배우고 새로운 상황에 적응하게 될때 비로소 컴퓨터가 지능을 가졌다고 할 수 있다. 일반적인 학습에는 크게 두가지 형태로 영어나 수학을 배우는 것처럼 "지식의 습득" 형태와 피아노를 배우는 것처럼 반복을 요하는 "숙련"이 추가 되는 형태가 있다. 컴퓨터 학습의 분야는 "지식의 습득"에 중점을 두는 것으로 다음과 같은 방법이 있다.

##### (1) 단순한 암기에 의한 학습

이 방법은 컴퓨터가 경험과는 관계없이 단순히 외부에서 넘겨주는 지식을 기억하고 있다가 그와 꼭 같은 상황이나 조건을 만났을 때 꺼내 쓰는 가장 간단한 방법이다. 이 방법은 Samuel의 체커 프로그램에 사용하여 어느정도 효과를 거두었지만 알고있는 방법을 응용할 수 없다는 단점이 있다.

##### (2) 가르침에 의한 학습

이 방법은 교사나 트레이너가 컴퓨터를 가르치는 것

으로 컴퓨터는 배우는 입장에서 입력된 지식을 스스로 이용 가능한 형태로 변환시켜 간다. 그리고 새로운 지식을 효과적으로 사용하기 위해서 기존의 지식과 서로 상충되지 않도록 잘 결합하여야 한다. 그러나 학습자인 컴퓨터가 스스로 추출해서 배울수 있는 양이 그리 많지 않으며 교사의 부담이 크다는 점이 있다.

##### (3) 유사성을 이용한 학습

이 학습방법은 새로운 개념이나 기술과 매우 유사한 기존의 지식을 새로운 상황에 유용한 형태로 변환시켜 새로운 지식을 습득케 하는 방법이다. 유사성에 의해 배우는 컴퓨터는 자기가 가지고 있는 기존 프로그램을 약간만 변형함으로써 다른 문제를 해결할 수 있기도 하다.

##### (4) 예를 통한 학습

이는 어떤 개념에 대한 긍정적인 예와 부정적인 예만으로 구성되어 있을때 이들로부터 긍정적인 예는 다 포함하지만 부정적인 예는 하나도 포함될 수 없는 그런 일반화된 하나의 개념을 스스로 도출함으로써 컴퓨터가 배우는 방법이다. 이 방법은 교사에 의해 일반개념 자체가 바로 주어지지 않고 유사성을 이용할 수 없기 때문에 더 많은 주론을 해야한다.

##### (5) 발견에 의한 학습

이는 가장 높은 차원의 학습으로 수학처럼 정형적이고 체계적인 영역에서만 응용할 수 있다. 이 방법은 이미 알려진 사실과 시도해 보지 않았던 방법을 결합하여 새로운 사실을 밝혀낸다. 이 경우 과거의 사실로부터 새로운 결론을 끌어낼 수 있는 통찰력이 필요하나 이는 인간의 인식분야에서도 분명하게 규정되어지지 않은 부분이다. 현재의 인공지능 시스템에서는 광범위한 지식의 습득이나 일반적인 문제풀이 기술이 충분하지 못하기 때문에 가까운 장래에 실현하기는 힘들다.

### Ⅲ. 인공지능 언어 및 도구

인공지능 프로그램이란 경험적인 지식을 문제 해결에 이용하는 것으로 주로 부호조작(symbol manipulation)이나 규칙(rule)을 이용하는 프로그램이다. 이와 같은 프로그램을 개발하기 위한 환경으로 고급 프로그래밍 언어, 소프트웨어 도구와 같은 프로그램이나 LISP 기계와 같은 하드웨어가 필요하다.

#### 1. 인공지능용 언어

인공지능 언어는 그림 2에서 보는 바와 같이 C, LISP, PROLOG와 같은 고급 프로그래밍 언어에서부터 KEE나 EMYCIN과 같은 전문가 시스템 구축 도구에 이

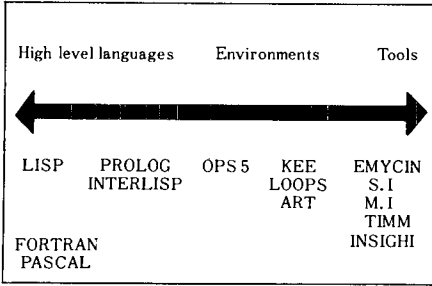


그림 2. The Language-Tool Continuum

르기까지 다양하다.

그러나 여기에서 주의해야 할 점은 LISP나 PROLOG, 혹은 소프트웨어 도구를 이용하여 프로그램했다고 해서 전부가 인공지능 프로그램이 되는 것은 아니다. 인공지능 프로그램은 FORTRAN이나 COBOL, C, 심지어는, 어셈블리어로도 개발할 수 있다. 중요한 점은 개발하는 도구에 있는 것이 아니고 프로그램 자체가 문제해결을 위하여 경험적인 지식을 이용했느냐 하는 점이다. 인공지능 언어는 단지 하나의 도구로서 프로그램의 생산성을 향상시키거나 효과적으로 수행하기 위하여 선택된 것일 뿐이다.

한편 인공지능 응용분야를 보면 초기에는 부호조작을 이용하는 것이 주류를 이루었기 때문에 부호처리에 용이한 LISP가 많이 사용되었다가 지식을 이용하는 방법이 달라짐에 따라 규칙을 처리하기에 편리한 PROLOG도 주의를 끌고 있다. 또 한편 기존의 개발된 소프트웨어와의 호환성을 고려하여 C나 FORTRAN도 많이 사용되고 있다.

(1) LISP

부호처리에 적합한 최초의 언어는 1957년도에 Carnegie-Mellon 대학의 Newell과 Shaw, 그리고 Simon에 의해서 개발된 IPL이다. 이듬해 IPL의 list 처리 방법을 이어받은 LISP가 McCarthy에 의해 개발되었다.

순수한 함수언어로서 LISP를 살펴보면 LISP가 처리하는 데이터 구조는 list이고 이것을 다루는 CAR, CDR, CONS와 같은 몇개의 매우 간단한 명령어로 이루어져 있다. 이외에 LISP의 구성요소로, 조건적으로 수행하는 COND와 새로운 명령어를 정의할 수 있는 기능이 있고 프로그램하기 쉽도록 여러가지 새로운 기능이 추가되었다.

LISP가 인공지능 언어로서 인정을 받게 된 배경은 첫째 대화식으로 프로그램할 수 있어 터미널에서 자유로

이 LISP 프로그램이나 데이터를 변경시킬 수 있다는 점이다. 둘째 개발환경에 관한 것으로 프로그램하기 편리한 소프트웨어 도구가 LISP로 많이 개발되어 있다는 점이고, list processing이라는 원어가 말해주듯이 부호처리에 적합하다는 점이다. 마지막으로 LISP는 프로그램과 데이터가 같은 형식으로 나타내고 데이터도 프로그램처럼 수행될 수 있다는 점이다. 그림 3에서 보는 바와같이 LISP는 크게 MAC-LISP계와 INTER-LISP계로 각각 발전되어 왔으며 현재는 표준화의 일환으로 COMMON LISP가 개발되었다.

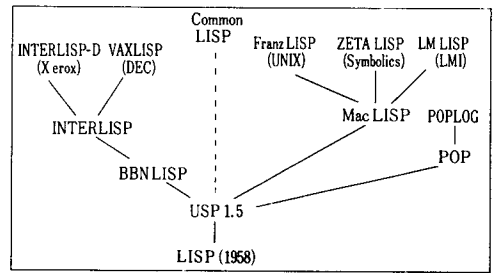


그림 3. The LISP Family Tree

(2) PROLOG

PROLOG는 서술논리 (predicate logic)에 기반을 둔 언어로서 Kowalski가 기본개념을 제안하였고, 1972년 Marseille 대학에서 Colmerauer와 Reussell에 의해 처음으로 제작되었다. 그후 Pereira와 Warren은 DEC-10에 컴파일러를 개발하였으며 유럽 각지에서 확장 발전되어 오다가 최근 일본에서 제 5세대를 컴퓨터의 핵심언어로 채택되면서 각광을 받기 시작했다.

프로그래밍 언어로서 PROLOG는 종래의 컴퓨터 언어와는 달리 논리에 기반을 둔 언어이다. 즉 논리를 문제표현 도구로 이용하고 질문에 답하기 위해 논리의 추론방법을 이용하였다. 일반적으로 PROLOG 프로그램은 지식베이스(프로그램), 추론방법, 그리고 제어전략이 통합된 일종의 생성시스템 (production system)이다.

사용자는 제어전략이나 추론방법에 대하여 알필요 없이 단지 지식베이스인 PROLOG 프로그램을 선언하기만 하면 된다.

PROLOG가 인공지능 프로그램에 적합한 점은 첫째 규칙을 처리하기가 용이하다는 점이다. 둘째 PROLOG는 일반 컴퓨터 언어와는 달리 문제해결 방법에 대하

여 프로그램하는 것이 아니고 문제를 PROLOG 언어로 묘사하면 된다는 점이다. 마지막으로 추론기능을 포함하고 있어 주어진 사실과 규칙들간의 관계를 이용하여 새로운 규칙만 정의하면 된다는 점으로 지식베이스의 지식만 수정하면 프로그램 전체를 수정하지 않고서도 확장할 수 있다는 점이다.

그러나 PROLOG는 제어기능의 비효율성과 지식을 표현하는 능력에 한계점을 지니고 있기 때문에 여러방향으로 수정보완되고 있으며 현재는 PROLOG 전용기계를 제작하기 위한 연구도 진행중이다.

2. 소프트웨어 도구

앞절에서 살펴본 고급 프로그래밍 언어만으로는 많은 양의 지식을 처리하는 응용 시스템을 개발하는데 불편할 뿐만 아니라 실험모델(prototype)을 빨리 개발해야 하는데 적합하지 않으므로 EMYCIN이나 KEE와 같은 소프트웨어 도구가 등장하게 되었다. 소프트웨어 도구는 주로 전문가 시스템 개발용 패키지들이 대부분으로 표1에 정리되어 있다.

이외에도 최근에 Teknowledge사의 M.1과 S.1이 있다. M.1은 지식공학의 실제 적용을 위한 소프트웨어 도구로 IBM PC에서 수행되도록 개발되었다.

S.1은 보다 광범위하고 전문적인 지식시스템을 개발하기 위한 복합적 소프트웨어로 Xerox 1100과 1108 Workstation에서 수행된다. Intellicorp사는 인공지능 응용을 위한 대표적인 도구의 하나로 KEE(knowledge

engineering environment) 시스템을 개발하여 제공하고 있으며 Inference사는 ART(automated reasoning tool)를 제공하고 Xerox사는 loops시스템을 제공하고 있다.

3. 하드웨어 도구

인공지능 언어는 전통적인 컴퓨터의 Von-Neumann 구조에서는 효과적으로 수행될 수 없으므로 LISP나 PROLOG를 위한 전용기계가 개발되어 보급되고 있다.

(1) Symbolics 3600

Symbolics, Inc.의 이 시스템은 1983년 소개된 것으로 MIT의 AI Lab.의 연구를 바탕으로 개발된 시스템이다. 중앙처리장치는 고성능의 lisp전용 처리기이고 가상기억장치가 있으며 30메가 바이트의 주기억 장치와 64K MOS ROM을 가지고 있다. 입출력 보조시스템으로 MC 68000을 이용한 전위처리기가 있어 Ethernet, SMD디스크, 고상도 흑백화면과 같은 고속 입출력 장치를 처리한다. 주요한 소프트웨어로는 Zeta-Lisp, Flavor에 입각한 윈도우 시스템, INTER LISP와 호환적인 패키지, 강력한 대화식 소프트웨어가 있다.

(2) LAMBDA

LAMBDA는 다중처리기 구조를 가지고 있어 LISP는 LISP 전용처리기에서 수행되고 다른 68000처리기는 4.8BSD UNIX OS를 수행한다. 또 다른 한개의 68000 처리기는 시스템 진단장치 (SDU)로서 사용된다. 이러한 부품들은 NuBus로 연결되고 multibus와의 연

표 1. 전문가시스템 구축도구

Tool	Organization	Nature
OPS 5	CMU	A programming language built on top of LISP designed to facilitate the use of production rules.
EMYCIN	Stanford U.	A domain independent version of MYCIN, which accompanies the back-ward chaining and explanation approach with user aids.
KAS	SRI	Supervises interaction with an expert in building or augmenting an expert system knowledge base in a network form implemented for PROSPECTOR.
ROSIE	RAND	A general rule-based programming language that can be used to develop large knowledge bases. Translates near-English into LISP.
AGE	Stanford U.	A sophisticated expert system to aid users in building expert systems.
HEARSAY III	USC/ISI	A generalized domain-independent extension of HEARSAY II. Includes a "context" mechanism, and an elaborate "blackboard" and scheduler.
UNITS	Stanford U.	A knowledge representation language and interactive knowledge acquisition system. The language provides both for "frame" structures and production rules.
TEIRESIAS	Stanford U.	A expert system that facilitates the interactive transfer of knowledge from a human expert to the system via a(restricted) natural language dialog.

결도 가능하다. 현재 LAMBDA 시스템은 개인용 뿐만 아니라 다수의 사용자가 이용할 수 있는 LAMBDA 2×2/PLUS, 그리고 LAMBPA 4×4 시스템을 계속 발표하고 있다. 또한 병렬처리를 위한 LISP나 ZETA-LISP, 그리고 고속으로 prolog를 수행할 수 있는 LM-PR ROLOG 등을 갖추고 있다.

### (3) EXPLORER

이 기계는 Texax Instruments사에서 개발된 32비트 개인용 LISP 기계로서 LISP전용 마이크로 프로세서를 이용하여 책상밑에도 설치할 수 있을 정도로 규모를 줄이는데 성공한 기계이다. 소프트웨어는 고해상의 그래픽스 윈도우, 마우스등을 기반으로 하여 객체지향 프로그래밍 기법을 COMMON LISP에 결합하였고 MIT의 LISP 기계의 많은 라이브러리 프로그램과 결합되어 있다.

### (4) SUN Workstation

SUN Micro사에서 제공하는 SUN Workstation에는 인공지능 응용을 위하여 LISP나 PROLOG을 제공하고 있다. 이 시스템은 4.2BSD UNIX OS를 기반으로 하는 것으로 전통적인 소프트웨어와 호환성이 필요한 인공지능 프로그램을 개발하기 위하여 적합한 Workstation이다. 최근에 LISP나 PROLOG 처리속도를 높이기 위하여 플로팅포인트 가속기와 같은 가속기를 제공하기도 한다.

## IV. 인공지능 응용기술 및 분야

일반적으로 인공지능 응용분야는 전통적인 프로그래밍 방법으로 해결하기 힘든 분야를 목표로 한다. 즉 문제해결 방법을 알고 있어도 기존의 컴퓨터에 수행되기 불가능하거나 해결방법은 모르지만 전문적인 지식을 가진 인간만이 해결할 수 있는 분야이다.

### 1. 전문가시스템 (expert system)

전문가 시스템은 현재 인공지능 분야에서 가장 강조되고 있는 분야중의 하나로서 대부분의 인공지능 응용 프로그램이 일종의 전문가 시스템이거나 혹은 지식기반 시스템이라고 생각할 수 있다.

전문가시스템이란 지능적인 컴퓨터 프로그램으로서 전문적인 지식을 가진 인간만이 해결할 수 있는 정도의 어려운 문제를 풀기 위하여 인간의 경험적인 지식이나 추론과정을 이용하는 것이다.

전문가시스템은 그림 4에서와 같이 지식베이스와 추론방법, 그리고 워킹 메모리로 구성되어 있다. 그러나 전문가 시스템의 승패는 얼마나 정교한 추론방법과 지식표현 방법등을 사용하느냐에 있는것이 아니고 얼마

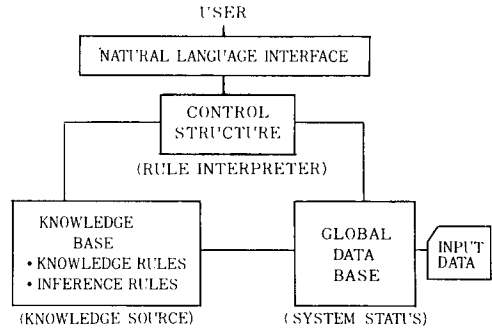


그림 4. Basic Structure of an Expert System

나 좋은 지식을 모을 수 있느냐에 달려있다.

현재 개발되고 있거나 사용중인 전문가시스템을 성능과 응용분야별로 대별하여 살펴보면 표 2와 같다.

### 2. 자연어 처리 및 음성인식

자연어란 사람들이 일상생활을 하는 중에 사용하는 언어로서 컴퓨터를 사용하는 인간에게 가장 이상적이고 자연적인 컴퓨터와의 대화도구다. 이 분야에 대한 연구는 인공지능의 역사에서 볼 수 있듯이 자동번역기 개발을 목표로 초창기부터 계속해온 분야이다. 그러나 자연어란 지능적인 존재사이의 대화수단으로서 서로간의 "정신적인 구조(mental structure)"를 전달하는 것이기 때문에 그 의미를 이해하기 위해서는 아주 복잡한 처리를 해야만 한다. 전통적인 방법으로는 이런 처리를 하지 못하기 때문에 경험적인 지식을 이용하는 인공지능 기술로 해결하려고 한다. 한편 음성인식은 컴퓨터와의 대화를 음성이나 말로서 할려는 것으로 자연어 처리기술을 바탕으로 음성처리 및 이해시스템을 개발하려는 시도이다. 현재 개발되었거나 개발중인 자연어 처리분야의 시스템을 살펴보면 다음과 같다.

#### (1) ARTHUR

ARTHUR(a reader that understand reflectively)는 미국의 Irvine에 있는 캘리포니아대학에서 Richard H. Granger 팀이 개발한 컴퓨터 프로그램으로 짧고 간단한 대화의 내용을 이해하여 내용상 모순이 있는 줄거리는 수정도 할 수 있는 프로그램이다.

#### (2) ROBOT-INTELLECT

이 프로그램은 Dartmouth 대학의 Larry R. Harris에 의해서 개발된 것으로 데이터베이스에 저장된 정보에 관한 질문을 자연어로 받고 이해하고 자연어로 대답해 주는 시스템이다. 현재 시스템은 실제 응용분야에서 데이터베이스의 user interface 기능으로 활용중에 있다.

## 丑 2. Existing Expert Systems by Function

	Domain	System	Institution
Diagnosis	Medicine	PIP	M. I. T.
	Medicine	CANSET	Rutgers U.
	Medicine	Interist/Caduceus	U. of Pittsburg
	Medicine	MYCIN	Stanford U.
	Medicine	PUFF	Stanford U.
	Computer Faults	DART	Stanford U./IBM
	Medicine	MDX	Ohio State U.
	Computer Faults	IDT	DEC
	Nuclear Reactor Accidents	REACTOR	EG & G Idaho Inc.
Data Analysis and Interpretation	Geology	Dipmeter Advisor	M. I. T. /Schlumberger
	Chemistry	DENDRAL	Stanford U.
	Chemistry	GAL	Stanford U.
	Geology	PROSPECTOR	SRI
	Protein Crystallography	CRYBALIS	Stanford U.
	Determination of Casual Relationships in Medicine	RX	Stanford U.
	Determination of Casual Relationships in Medicine	ABEL	M. I. T.
Analysis	Oil Well Logs	ELAS	AMOCO
	Electrical Circuits	EL	M. I. T.
	Symbolic Mathematics	MACSYMA	M. I. T.
	Mechanics Problems	MECHO	Edinburgh
	Naval Task Force Threat Analysis	TECH	Rand/NOSC
	Earthquake Damage Assessment for Structures	SPERIL	Purdue U.
Design	Digital Circuits	CRITTER	Rutgers U.
	Computer System Configurations	RI	C. M. U.
	Automatic Programming	PECOS	Yale
	Circuit Synthesis	SYN	M. I. T.
Planning	Chemical Synthesis	SYNCHEM	SUNY Stoneybrook
	Chemical Synthesis	SECHS	U. of Ca. Santa Cruz
	Robotics	NOAH	SRI
	Robotics	ABSTRIPS	SRI
	Planetary Flybye	DEVISER	JPL
	Errand Planning	OP-PLANNER	Rand
	Molecular Genetics	MOLGEN	Stanford U.
	Mission Planning	KNOBS	MITRE
	Job Shop Scheduling	ISIS-II	CMU
Design of Molecular Genetics Experiments	SPEX	Stanford U.	
Learning from Experience	Chemistry	METADENDRAL	Stanford U.
	Heuristics	EURISKO	Stanford U.
Concept Formation	Mathematics	AM	CMU
Signal Interpretation	Speech Understanding	HEARAY II	CMU
	Speech Understanding	HAPPY	CMU
	Machine Acoustics	SU/X	Stanford U.
	Ocean Surveillance	HASP	System Controls Inc.
	Sensors On Board Naval Vessels	STAMMER-2	NOSC, San Diego/SDC
	Medicine-Left Ventrical Performance	ALVEN	U. of Toronto
Monitoring	Patient Respiration	VM	Stanford U.
Use Advisor	Structural Analysis	SACON	Stanford U.
	Computer Program		
Computer Aided Instruction	Electronic Troubleshooting	SOPHIE	B. B. N.
	Medical Diagnosis	GUIDON	Stanford U.
	Mathematics	EXCHECK	Stanford U.
	Steam Propulsion Plant Operation	STRAMER	BBN
	Diganostic Skills	BUGGY	BBN
	Causes of Rainfall	WHY	BBN
	Coaching of a Game	WEST	BBN
	Coaching of a Game	WUMPUS	M. I. T.



Domain	System	Institution
Knowledge Acquisition	Medical Diagnosis	TEIRESIAS
	Medical Consultation	EXPERT
	Geology	KAS
Expert System Construction		Stanford U.
		Rutgers
		SRI
		Rand
		Stanford U.
		USC/ISI
		Stanford U.
		CMU
		IBM
		U. of Mo
Consultation/Intelligent Assistant	Battlefield Weapons Assignments	BATTLE
	Medicine	Digital Therapy Advisor
		NRL AI Lab
		M. I. T.
	Radiology	RAYDEX
	Computer Sales	XCEL
	Medical Treatment	ONCONIN
Nuclear Power Plants	CSA Model-Based Nuclear Power Plant Consultant	
	Rutgers U.	
	CMU	
	Stanford U.	
	GA Tech	
Management	Automated Factory	IMS
		CMU
Automatic Programming	Modelling of Oil Well Logs	φNIX
		Schlumberger-Doll Res
Image Understanding		VISIONS
		ACRONYM
		U. of Mass.
		Stanford U.

(3) Hearsay- II

Carnegie-Mellon 대학에서 개발된 이 프로그램은 1000개의 단어로 구성된 말을 90% 정도로 정확하게 인식할 수 있는 프로그램이다. 기본적인 방법은 signal processing을 위하여 인공지능의 symbolic reasoning 방법을 이용하였다. 이것과 비슷한 방법을 위하여 BBN에서는 HWIM(Hear What I Mean)이라는 시스템을 개발하였다.

(4) The SRI System

이 시스템은 제어전략(control strategy)를 강조하여 구분분석기나 지식베이스, 의미분석기 등 각 부분들 사이의 통신이 중앙 집중적인 제어전략을 통하여 간접적으로 이루어지도록 하는 것이 특징이다.

(5) 기타

그 외에도 자연어 처리 시스템이 많이 개발되어 있지만 대표적인 것으로 Yale대학에서 개발한 VEGE와 BORIS, Brown 대학의 PARAGRAM과 BRUIN, State 대학의 SNePs 등이 있다. VEGE(vegetable gardening expert)는 자연어를 사용하여 채소경작에 관한 질문에 응답하는 시스템이고 BORIS(in-depth understanding of narratives)는 어떤 이야기를 자세히 이

해하고 그 이야기에 관한 질문에 대하여 자연어로 대답해 주는 프로그램이다.

3. Artificial Vision

비전시스템은 눈의 역할을 하는 카메라와 비디오 영상을 처리해 주는 분석기로 구성되어 있다. 이 시스템의 전형적인 구성은 그림 5에서와 같이 Octek 2200 비디오 영상분석기를 보면 잘 알 수 있다. 이 시스템은

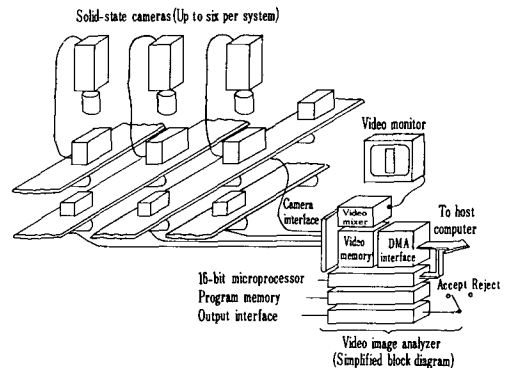


그림 5. Configuration of Octek Vision System

고속의 마이크로 프로세스를 사용하여 카메라 신호를 모아 그 영상을 분석하고 계산하여 보내줄 신호를 제공한다. 그런데 이 영상분석 및 계산은 굉장히 복잡하고 많은 시간이 소요되기 때문에 인공지능 기술을 이용하여 보다 효과적인 처리를 하기 위하여 이용된다. 한편 사용화된 비전시스템으로 ORS(object recognition system) 비전시스템이나, Stanford Acronym system 등이 있다.

4. Robotics

지능형 로봇이라면 생각하고 느낀 것에서 동작할 수 있어야 한다. 감각적인 기능과 동작하는 기능은 인간의 육체적인 기능에 해당되겠지만 생각하는 기능은 인간의 뇌기능에 해당되는 것으로 바로 인공지능 기술이 그대로 이용되는 분야이다. 인공지능과 robotics 와의 통합된 모델은 그림 6 에서와 같이 생각할 수 있는데 인공지능은 뇌나 육체적인 기능중에도 계획수립 및 업무수행을 하는데 이용된다.

한편 mobile robot에 대한 예로서 SRI Institute에서 개발한 Shakey와 JPL(Jet Propulsion Laboratory)에서 NASA 우주프로그램을 위한 moving robot와 Carnegie-Mellon 대학과 Ohio 주립대학에서 개발한 legged robot 등이 있다.

상용화 시스템으로는 Unimation Inc.의 PUMA(programmable universal machine for automation) 개발을 시작으로 Machine Intelligence Corp.의 Mode VS-100(1978), General Motors의 CONSIGHT 시스템(1978), CRC Welding Systems, Inc.의 THRUARC(1981), Object Recognition Systems, Inc.의 I-bot-1(1982), IBM Corp., Control Automation, Intelledex 등의 PCB 부품조립과 같은 전자회사에서 사용되는 산업용 로봇(1982) Automatrix Inc.의 Statistical process control S/W Package(1983) 등이 있다.

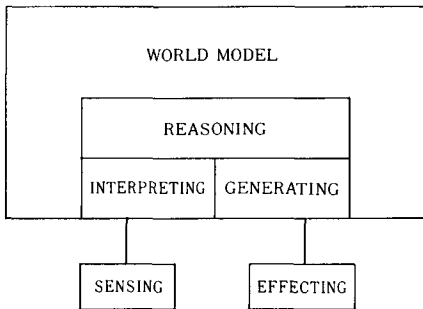


그림 6. A Unified Model of Artificial Intelligence and Robotics

5. 기타 응용분야

(1) VLSI Design

VLSI Chip을 설계하는 것이 복잡해짐에 따라 인간이 자세한 모든 것을 고려할 수 없게 되어 이를 위한 인공지능 프로그램을 개발하려는 사업이 시작되었다. Stanford대학에서는 Heuristic Programming Project의 일환으로 EURISKO project와 KBVLSI project 등을 추진하고 있다. 그리고 MIT AI Lab.에도 VLSI 설계를 위한 여러개의 project가 진행되고 있고 Maryland대학에서도 SADD project를 추진하고 있다. 그리고 GOULD와 같은 회사에서도 VLSI design 및 verification을 위하여 인공지능 기술을 이용한 project가 오랜전부터 시작되었다.

(2) Automatic programming

이 분야는 자동적으로 프로그램을 개발하거나 개발된 프로그램을 검증하기 위한 것으로 인공지능 기술을 이용한 방법개발이 연구되고 있다. 예로서 Stanford 대학의 PSI(program synthesis), PECOS, DEDALUS(Deductive Algorithm U-synthesizer) 등이 주어진 규정을 이용하여 컴퓨터 프로그램을 만들어내는 프로그램이다.

(3) 군사적 응용

많은 사람 및 물적자원을 지리학적인 요건을 빠른 시간내 일관성있게 연속적으로 다루기 위하여 많은 돈을 투자하고 있다. 중요한 군사용 인공지능 project는 DARPA(defense advanced projects agency)에서 주관하고 있는데 1985년에는 \$ 100million을 투자했다. 주요한 인공지능 project를 보면 Air/Land Battle 2000, Battle field Uses, Military Maintenance, Intercepting Enemy Aircraft, Robotic Reconnaissance Vehicle 등이 있다.

參 考 文 獻

[1] Nils J. Nilsson, Principles of Artificial Intelligence, Springer-Verlag, 1982.  
 [2] Richard K. Miller, Artificial Intelligence: vol 1- Technology and Applications, Technical Insights, Inc., 1984.  
 [3] UNIX/WORLD: Special Artificial Intelligence Issue, August 1986.  
 [4] 경영과 컴퓨터, 1985. 1~8 호.  
 [5] 정보산업, 1986. 4 월호. \*