

지식기반 시스템

林 榮 煥

(正 會 員)

韓國電子通信研究所 知能網研究室 室長

I. 서 론

지식기반 시스템은 종래의 시스템 개발 방법과는 달리 데이터 뿐만 아니라 인간의 지식을 문제해결을 위한 시스템 개발에 이용 할려는 시도이다. 주로 이러한 요구는 기존의 시스템이 보다 기능이 우수하거나 사용하기 편리한 지능을 부여할 필요에서 출발하였다. 그리고 현재는 기존의 컴퓨터로는 해결이 불가능하지만 인간은 해결할 수 있는 분야에 까지 도전하게 되었다. 이 분야의 대표적인 것으로 전문가 시스템을 들 수 있다. 즉 문제를 해결할 수 있는 방법이 개발되었다고 하더라도 기존의 컴퓨터로는 기억장치의 부족이나 수행속도가 느려 해결할 수 없는 문제라도 인간이 해결했던 지식을 이용하여 기존의 컴퓨터가 문제해결을 위한 대안(최선의 방안은 아닐지라도)을 찾을 수 있다는 점이다.

이러한 특징 때문에 인공지능 분야에서 해결해야 할 문제들도 많지만 지금 당장이라도 기존의 컴퓨터 기술과 결합하여 실용가능한 지식기반 시스템을 개발 할 수 있다는 매력때문에 세계적으로 연구개발이 활발하다.

본 고에서는 지식기반 시스템을 소개하기 위하여 우선 지식기반 시스템의 구성에 대하여 살펴 보았다. 지식기반 시스템은 지식베이스와 처리기 그리고 사용자 접속부와 지식수집기 등으로 구성되어 있고 시스템의 성능과 가장 밀접한 관계가 있는 것은 지식베이스의 양과 질이다. 다음으로 지식기반 시스템이 잘 적용 될 수 있는 분야에 대한 고찰로 기존의 시스템에 지능을 부여하는 분야와 인간만이 해결했던 분야에 적용한 사례에 대하여 살펴 보았다. 그리고 이를 개발하기 위하여 가장 핵심적으로 해결해야 할 문제점에 관하여 논의 하였다. 그리고 지식기반 시

스템을 개발하기 위하여 일반적으로 고려해야 할 개발 단계 및 팀 구성과 개발환경에 대하여 살펴 보았다. 마지막으로 컴퓨터 센터와 같은 개념을 좀더 개선한 지식공학센터의 필요성을 강조하고 끝을 맺었다.

II. 지식기반 시스템의 구성

지식기반 시스템(knowledge based system)은 기본적으로 지식을 이용하여 만들어진 시스템이기 때문에 구성 자체가 지식을 저장하거나, 처리하기에 편리하도록 되어있다. 지식기반 시스템의 기본구성은 아래 그림과 같이 크게 네부분, 즉 지식 또는 데이터 베이스와 지식처리기, 그리고 사용자 접속기(man-machine interface)와 지식수집부 등으로 나눌 수 있다.

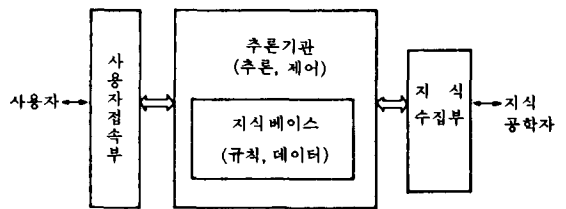


그림 1. 지식기반 시스템 구성

이 개념은 시스템 사용자를 위한 사용자 접속부와 끊임없이 새로운 지식을 모아서 수정 보완하는 지식수집기 그리고 모아진 지식을 활용하여 처리해 주는 지식처리부 등이다. 이 장에서는 각 모듈의 역할 및 기술 동향을 소개하려고 한다.

1. 지식베이스

이것은 지식이나 데이터를 저장하는 곳으로 기본적으로는 데이터베이스 관리시스템에서의 데이터 베이스와 유사한 개념으로 생각하면 된다. 여기서 중요한 문제는 어떻게 하면 시스템의 효율을 높일 수 있게끔 데이터나 지식을 저장하느냐 하는 문제이다. 데이터를 저장하는 방법은 잘 알려진 바와 같이 network model과 hierarchical model, relational model 등이 있으나 지식베이스로 확장이 용이한 relational model을 많이 이용하고 있다. 지식을 저장하는 방법은 지식을 컴퓨터에 표현하는 방법 및 추론방법과 실과 바늘의 사이만큼 밀접한 관계가 있다. 지식표현은 단순한 지식자체의 표현 뿐만 아니라 표현된 지식을 처리하는 방법도 동시에 고려하여야 한다. 즉 지식표현은 자료구조와 그것을 해석하여 컴퓨터가 지적인 행위를 할 수 있게하는 해석과정의 결합이라 할 수 있다. 지식기반 시스템에서 고려할 수 있는 지식은 주로 네가지 유형으로, 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 마찬가지로써 각 선언적 지식과의 관계를 정의해 주는 지식을 나타내는 방식이다. 선언적 표현방법으로는 서술논리(predicate calculus) 방식, 생성규칙(production rule), 의미네트워킹(semantic network), 프레임(frame)이나 스크립(script) 등이 있다. 최근 지식표현의 경향은 지식베이스를 일의 성격에 따라 분류하고 각기 일의 성격을 가장 잘 대변하고 처리해 줄 수 있는 표현방법으로 나타내고 그 위에 여러방식을 통합처리해 주는 방식을 많이 이용하고 있다. 즉 하나의 소프트웨어 패키지내에 생성규칙이나 프레임을 동시에 사용가능도록 처리해 줄 수 있는 것이 많이 이용되고 있다.

2. 추론방법

추론방법은 사용자의 질의에 답하기 위하여 지식을 재결합 새로운 지식을 유도하고 또다시 어떤 결론에 도달할때까지 지식을 결합하여 생성해가는 과정을 반복하는 것이다. 추론방법은 지식표현 방법에 따라 각기 다르나 일반적으로 추론방향에 의하여 순방향 추론(forward chaining)과 역방향 추론(backward chaining) 등으로 나눌 수 있다. 순방향 추론은

주어진 사실과 알고 있는 지식에서 어떤 결론에 도달하게 하기 위하여 추론해 가는 방법이고, 역방향 추론은 주어진 현상에서 미리 어떤 결론일 것 이라고 가정하고 그 가정을 확인하며 추론해 가는 방법 이라고 볼 수 있다. 순방향 추론방법은 좀 더 구체적으로 얘기하면 상태공간방법(state-space search)에 해당하고 역방향 추론방법은 문제축소 (problem reduction approach)와 관계가 있다. 이 방법은 다 같이 시작점과 끝점을 어떤 상태로 표시하고 두점사이 에 도달하기 위한 길을 찾아가는 방식으로 볼 수 있는데 여기서 탐색공간을 최소화 하기 위한 제어전략이 필요하다. 즉 시스템의 효율이 좋다는 것은 탐색공간을 최소화 하면서 옳은 답을 내는 것으로 볼 때 이 제어 전략이 중요하고 궁극적으로는 이것을 위하여 지식이 이용된다고 볼 수 있다.

3. 사용자 접속부(Man-Machine Interface)

사용자는 사용하기 편리한 시스템을 원한다. 아무리 시스템의 기능이 우수하다고 해도 사용하기 불편하면 그것은 현장에서 살아 남을 수 없게 되기 쉽기 때문이다. 사용하기 가장 편리한 시스템은 사용자와 기계가 서로 얘기 하면서 일을 추진해 간다고 볼 수 있다. 그러나 현재의 기술로는 인간의 언어로 기계와 자연스럽게 대화하여 서로를 이해하기 위한 연구는 많이 하고 있지만 실용화 하기는 아직 요원한 상태이다. 그러므로 현재 컴퓨터 기술로 많이 이용하고 있는 것은 고해상도CRT를 이용한 다중 윈도우(multi-window)에 마우스를 이용한 메뉴 선택방식이 많이 이용되고 있다. 그리고 시스템의 상태를 표시하기 위하여 컬러모니터에 컴퓨터 그래픽스를 이용하여 현장감이 넘치도록 현상황을 표시하는데 이용하기도 한다. 앞으로 인간의 자연어로 기계와 대화할 수 있는 방법이 개발 되겠지만 기계와의 인터페이스를 원활히 하기 위하여 음성인식과 자연어 처리가 동시에 해결되어야 하고 그 각각의 기술은 또다시 지식기반 시스템 접근방법을 도입해야 하게 되어 있다.

4. 지식 수집기

컴퓨터가 단순히 지시한 일만 계속 반복하는 것이 아니고 새로운 것을 배우고 새로운 상황에 순응하게 될때 비로소 컴퓨터가 지능을 가졌다고 볼 수 있다. 지식기반 시스템은 지식베이스를 끊임없이 수정보완하여 양질의 지식을 가지고 있어야 좋은 시스템이 될 수 있다. 이를 위하여 지식수집기는 지식공학자

가 수집한 지식이나 데이터를 지식베이스에 첨삭하기 용이하게 하는 기능을 가지고 있다. 이 모듈은 새로운 지식을 첨가할때 마다 기존의 지식과 중복되는지 또는 모순이 발생하지 않았는지 등을 검사하고 이 새로운 지식이 추론과정에서 어떻게 이용되는지 등을 보여 준다. 현재 기술은 단지 지식베이스를 편집하는 기능이나 중복성 및 일치성 검사정도의 기능을 가지고 있으나 가까운 장래에는 지식기반 시스템 자체가 학습하는 기능을 보유하게 되어 실제 현장에서 사용되면서 점점 숙련된 결과를 도출할 수 있게 성장해 가게 될 수 있게 된다.

Ⅲ. 적용분야

지식기반 시스템 접근방법으로 해결하기에 적당한 적용분야는 크게 두분야로 나누어 볼 수 있다. 첫째는 지식기반 시스템의 특성 그대로 기존의 방법으로는 해결이 불가능했던 분야의 문제를 해결하려는 시도이다. 즉 문제해결을 위한 방법론(algorithm)은 존재하지만 기존의 컴퓨터 처리 속도의 제한으로 만족할 만한 결과를 계산하는데 너무 오래 소요되어 해결이 불가능하거나 또는 해결 방법 자체는 모르지만 장기간 축적된 경험을 바탕으로 일종의 감에 의해 해결할 수 있는 분야에 적용할 수 있다. 둘째로 지식기반 시스템의 적당한 적용분야로는 기존의 활용중인 시스템에 지능을 제공함으로써 시스템의 기능을 향상시키거나 사용하기 편리하게 하는 분야 등이다. 각각의 구체적인 예를 찾아 보면 다음과 같다.

1. 전문가 시스템

지식베이스 시스템의 대표적인 적용분야로서 전문가 시스템은 인간인 전문가를 모방하려는 시스템이다. 즉 지식기반시스템의 지식베이스를 인간인 전문가의 지식을 수집, 저장하여 인간인 전문가 처럼 처리할 수 있도록 할려는 것이다. 그러므로 전문가가 존재하는 분야는 실제 적용가능한 분야라고 생각할 수 있을 정도로 적용범위가 광대하다. 현재는 좁은 분야의 깊은 전문성을 요구하는 문제를 해결하기 위한 전문가 시스템은 많이 개발되어 실용중에 있는 것도 있고 지금도 개발이 증가일로에 있다. 이 분야의 중요한 기술은 지식수집으로 전문가 시스템의 능력은 저장된 지식의 양과 질에 달려있다. 그러나 앞으로 해결해야 하는 문제는 시스템이 스스로 배우는 능력을 보유하게 하는 것으로 인공지능 분야의 연구

자들은 이 문제를 활발히 연구하고 있다. 이 분야에 대한 상세한 소개는 본 특집호에서 자세히 다룰 예정이기 때문에 여기서는 생략하기로 한다.

2. 자연어 처리 및 음성인식

자연어란 사람들이 일상생활을 하는 중에 사용하는 언어로서 컴퓨터를 사용하는 인간에게는 가장 편리하고 자연적인 컴퓨터와의 대화 도구이다. 이 분야에 대한 연구는 인공지능의 역사에서 볼 수 있듯이 자동번역기 개발을 목표로 초창기부터 계속 연구되어 온 분야이다. 그러나 자연어란 지능적인 존재사이의 대화 수단으로 정신적인 구조를 전달하는 것이기 때문에 그 의미를 이해하기 위해서는 아주 복잡한 처리를 해야 한다. 기존의 방법으로는 계산능력의 부족으로 한때 중단되기도 했지만 지식기반 시스템의 연구가 활발해 짐으로 경험적인 지식을 이용하여 대화의 의미를 전달 하려고 하는 노력이 제한된 범위나마 성공을 거두고 있다.

3. 로봇 및 비전

로보틱스 분야는 처해 있는 상황에 적절하게 대응할 수 있는 장치를 개발하려는 시도로서 제어분야에 지식기반 시스템을 적용하는 것이라 볼 수 있다. 이것은 주변환경에 직접 행동하기 위한 기계적인 장치와 그 행동을 수행하기 위한 계획을 수립하는 시스템 개발이 필요하다. 현재 공장자동화 시스템과 같이 엄격하게 제어할 수 있는 상황에는 다양하게 사용되고 있다. 앞으로는 비전 기술의 발전으로 좀더 융통성이 필요한 분야에서도 이용할 수 있도록 상황 지식을 이용한 지식기반 시스템이 개발되고 있다.

4. 기타

지식기반 시스템은 지식을 이용할 수 있는 곳이면 어디서나 필요한 것으로 음악을 작곡하거나 연주하는 것에서부터 법적인 문제를 처리하는 변호업무, 공장의 스케줄링, 경영자의 의사결정, 의사의 진단 및 처방, 군사용의 조기경보 시스템이나 시설관리 시스템, 무인 자동차, 자동프로그래밍 등에 이르기 까지 매우 다양하다.

Ⅳ. 핵심적인 문제

지식기반 시스템의 기본적인 구성과 그 적용분야에서 나타난 바와 같이 이 시스템의 가장 핵심이 되는 문제는 어떻게 좋은 지식베이스를 구축할 것인가 하는 것이다.

1. 지식수집이 가장 중요

사람 개인 또는 집단이 머리속에 막연히 간직하고 있는 지식이나 필요한 데이터를 종이위에 적는것이 가장 큰 문제이다. 실제 막연히 경험적인 판단을 정형적인 언어로 표현 하는것 자체가 많은 정보를 잃어 버리게 되고 또한 많은 시간을 가지고 지식이나 데이터를 수집하기 때문에 모아진 지식이 중복되거나 일관성을 잃게 되는 경우도 허다하다. 일반적으로 좋은 개발도구는 이 모듈의 성능이 우수해야 한다. 현재는 모아진 지식을 검사하거나 새로운 지식을 추가 할때마다 기존의 지식과 관계를 검사하고 지식을 제공하는 사람이 이해 하기 쉽도록 그래픽스를 사용하여 지식들간의 관계를 도식으로 나타내 주는 정도이다.

2. 지식표현은 그 다음 문제

다음으로 중요한 것은 모은 지식을 분석하여 일의 순서를 정하고 문제를 해결하기 위한 방법을 정립하는것 및 기존의 컴퓨터가 효율적으로 수행할 수 있도록 조직하여 저장하는 것이다. 현 추세는 데이터를 저장하기 위하여는 관계형 데이터관리 시스템을 이용 저장하고 지식을 저장하기 위하여는 전문가 시스템 구축 도구를 이용한다. 일반적으로 지식기반 시스템의 적용분야의 문제들이 전문지식만을 가지고 해결되는 것이 아니고 기존의 데이터 처리나 관리 시스템과 같이 결합될때 좋은 결과를 낼 수 있기 때문에 기존의 소프트웨어와 호환성을 유지하기 위하여 관계형 데이터 베이스 관리시스템과 전문가 시스템간의 인터페이스가 자연스럽게 이루어져야 한다.

3. 현명한 도구선택

지식을 모아서 문제해결을 위한 방안이 정립되면 실제 구현을 위한 도구를 선택하게 된다. 이때는 사용자가 사용하기 편하게 하기 위한, 그리고 현재 가능한 컴퓨터 및 소프트웨어 도구를 현재의 일을 해결하는데 가장 적합한 도구를 선택하여 구현 하여야 한다.

V. 개발 방법

이 장에서는 지식기반 시스템을 개발하는데 필요한 팀 구성 및 개발환경 그리고 개발 단계에 대하여 살펴본다.

1. 개발단계

지식기반 시스템을 개발하는 방법은 여러가지가

있을 수 있으나 전문가 시스템 개발 단계를 소개하면 다음과 같이 5 단계로 나눌 수 있다.

- 문제 설정단계 :

개발 가치가 인정되는 어느 한 분야 선정 및 연구

- 개념화 단계 :

선정된 분야의 문제 및 특성을 구체화 하는 단계

- 표현화 단계 :

모아진 지식을 지식 공학적인 측면에서 분석 및 설계하는 단계

- 구현화 단계 :

설계된 시스템을 전산화 하는 단계

- 시험단계 :

현장 적용 및 결과검증 단계

이 단계를 거쳐서 완성된 시스템의 수준은 완전한 시스템이라기 보다는 이제야 겨우 지식을 이용할 수 있는 수준이 되었다고 생각하면 된다. 그러므로 실용가능한 시스템이 되기 위해서는 실제 적용분야에서 사용되면서 경험적인 지식을 꾸준히 쌓아 가야만 한다. 위의 전체 과정에서 간과하기 쉬우면서도 중요한 단계는 개념화 단계라고 할 수 있다. 지식기반 시스템의 개발 주역은 인공지능을 전공한 지식공학자 같지만 실제로는 응용분야의 흐름과 그 분야의 지식이라는 것을 잊지 말아야 한다.

2. 개발 팀

개발 팀은 크게 적용분야의 전문가 팀과 지식공학자 팀으로 대별 할 수 있다. 적용 분야의 전문가 팀은 적용분야의 문제점을 파악하고 데이터 및 지식을 수집 분석하여 일의 흐름을 파악하고 시스템을 모델화 하면 지식공학자 팀은 수집된 지식을 검사하고 컴퓨터에 저장하고 일의 흐름을 전산화 하는 작업을 한다. 그리고 적용분야의 담당자들은 개발된 시스템이 제대로 성능을 내고 있는지 검증하고 수정보완해 가는 것이다. 앞절에서도 설명했듯이 지식기반 시스템의 성능은 개발된 후 사용하면서 얼마나 좋은 지식을 수정보완해 갈 수 있는나에 달려 있다고 해도 과언은 아니다. 그러므로 적용분야의 운영자가 처음 개발단계부터 이 프로젝트에 참여하여 시스템 개발이 완료된 후에 지식의 수정보완이 만족할 만한 수준에 이르기 까지는 끊임없이 계속 되어야 한다.

3. 개발도구

개발 환경은 아래와 같이 3 가지로 구분하여 생각할 수 있다.

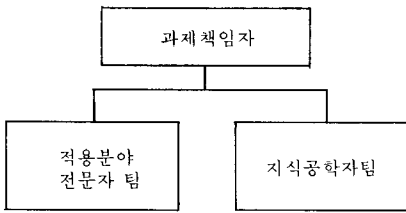


그림 2. 개발팀 구성

- 하드웨어
- 지식프로그램용 언어
- 소프트웨어 도구

1) 하드웨어 도구

컴퓨터는 기존의 범용 컴퓨터로는 지식기반 시스템을 구축하기에 적합하지 않기 때문에 아래와 같은 특별한 기계를 기존의 컴퓨터와 연결하여 사용하고 있다.

(1) Symbolics 3600

Symbolics, Inc. 의 이 시스템은 1983년 소개된 것으로 MIT의 AI Lab.의 연구를 바탕으로 개발된 시스템이다. 중앙처리장치는 고성능의 LISP 전용 처리기이고 가상 기억장치가 있으며 30메가 바이트의 주기억 장치와 64K MOS ROM 을 가지고 있다. 입출력 보조시스템으로 MCG 3000을 이용한 전위처리기가 있어 Ethernet, SMD 디스크, 고상도 흑백화면과 같은 고속 입출력 장치를 처리한다.

주요한 소프트웨어로는 ZETALISP, Flavor에 입각한 윈도우 시스템, INTER LISP와 호환적인 패키지, 강력한 대화식 소프트웨어가 있다.

(2) LAMBDA

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

이러한 부품들은 NuBus로 연결되고 multibus와의 연결도 가능하다. 현재 LAMBDA 시스템은 개인용 뿐만 아니라 다수의 사용자가 이용할 수 있는 LAMBDA 2×2/PLUS, 그리고 LAMBDA 4×4 시스템을 계속 발표하고 있다. 또한 병렬처리를 위한 LISP나 ZETALISP, 그리고 고속으로 PROLOG를 수행할 수 있는 LM-PROLOG등을 갖추고 있다.

(3) EXPLORER

이 기계는 Texas Instruments 사에서 개발된 32비트 개인용 LISP 기계로서 LISP 전용 마이크로 프로세서를 이용하여 책상 밑에도 설치할 수 있을 정도로 규모를 줄이는데 성공한 기계이다.

소프트웨어는 고해상의 그래픽스 윈도우, 마우스 등을 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 PROLOG를 제공하고 있다.

이 시스템은 4.2 BSD 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.

2) 지식프로그램용 언어

지식프로그램용 언어는 C, LISP, PROLOG와 같은 고급프로그래밍 언어에서부터 KEE나 EMYCIN 과 같은 전문가 시스템 구축도구에 이르기까지 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 개발 할 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 지식프로그램의 응용분야를 보면 초기에는 부호조작을 이용하는 것이 주류를 이루었기 때문에 부호처리에 용이한 LISP가 많이 사용되었다가 지식을 이용하는 방법이 달라짐에 따라 규칙을 처리하기에 편리한 PROLOG도 주위를 끌고 있다.

또 한편 기존의 개발된 소프트웨어와의 호환성을 고려하여 C나 FORTRAN도 많이 사용되고 있다.

3) 소프트웨어 도구

앞절에서 살펴본 고급 프로그래밍 언어만으로는

많은 양의 지식을 처리하는 응용시스템을 개발하는데 불편할 뿐만 아니라 실험시제품을 빨리 개발해야 하는데 적합하지 않으므로 EMYCIN을 효시로 소프트웨어 도구가 많이 개발되었다. 현재 대표적인 개발 도구로는 KEE(knowledge engineering environment)나 ART(automated reasoning tool), Knowledge Craft 등 대형 소프트웨어 도구에서 OPS5와 같이 PC에 이용되는 것에 이르기까지 매우 다양하게 개발되어 있다. 이 도구의 선택방법 및 기능에 관한 상세한 것은 본 특집의 다른 원고에서 상세히 기술될 예정임으로 생략한다.

VI. 결 론

지식기반 시스템에 대한 연구가 국내에서도 인공지능이나 전문가 시스템 개발을 하는 분야에서 연구가 활발히 진행되고 있다. 즉 학교나 연구소, 그리고 기업체에서 다양한 분야의 전문가 시스템 시제품이 속속 개발되어 발표되고 있다. 그러나 현재 국내에

서는 기대에 만족할 만한 실용제품이 잘 나오지 않는 것도 사실이다. 이는 우선 개발자들의 기술 및 경험 부족도 문제가 되겠지만 이보다 적용분야의 전문가들이 자기의 지식을 선뜻 내어 놓으려고 하지 않는 배타성과 자금을 지원하는 측에서 성급한 결론을 유도하는데도 기인한다고 본다. 그리고 각 개발과제별로 각기 서로 다른 개발환경을 구축하려고 하기 때문에 중복투자로 인한 낭비도 문제이지만 서로의 경험을 교환할 수 없는 것이 더 큰 문제이다. '70년대 이후부터 컴퓨터의 계산능력을 제공하는 전자계산실의 개념이 현재까지 주종을 이루어 왔지만 지금부터는 지식처리에 필요한 컴퓨터 능력 및 표준화된 소프트웨어를 제공하고 지식기반 시스템 개발방법을 제공할 수 있는 지식공학센터와 같은 개념의 도입 및 실현이 이루어져야 한다. 이 센터에서 한국내에 표준화된 소프트웨어 도구를 개발 보급할 뿐만 아니라 거의 무한할 정도로 다양한 적용분야의 잠재력을 깨우기 위해서 교육도 필요하고 표준화된 개발 방법론 정립 및 보급도 절실히 요구된다. ☺

♣ 用語 解説 ♣

Schema(스키머)

데이터 모델에 의거하여 데이터 베이스의 성질을 형식적으로 기술한 것을 스키머라고 한다. 각 이용자에게 제공되는 스키머를 외부 스키머라고 한다. 모든 외부 스키머를 통합한 논리적 스키머를 개념 스키머라고 하며, 파일 편성에 대응한 스키머를 내부 스키머라고 한다. 이들 스키머로 기술되는 데이터를 각각 외부 모델, 개념모델, 내부 모델이라고 한다. 각 외부 스키머는 개념 스키머의 일부분에 대응하는 일이 많으므로 부분 스키머 또는 뷰(view)라고도 한다.

Semantic network(의미네트워크)

지식 표현법의 일종으로 지식과 개념등을 마디점으로 나타내고 이들 사이의 관계를 방향이 있는 가지(有向枝)로 표시한 유향(有向)그래프로 나타낸 것

Functional language(함수형 언어)

비절차형 언어중 프로그램을 만들어가는 조립방법 또는 결합의 형식을 대수에 대한 연산이라고 간주할 때 이 치역(值域)에 해당하는 프로그래밍 언어를 말한다.