지식기반 시스템 개발도구

金泳煥, 金鎭衡
(正 會 員)
韓國科學技術院 電算學科

Ⅰ. 서 론

근래 20여년 동안 연구소나 학교 실험실에서 인공지능에 대한 활발한 연구와 기술 개발을 이룬 결과 지금까지 컴퓨터가 풀 수 없다고 생각했던 문제들을 인공지능을 이용하여 풀 수 있다는 사실을 입증하게 되었으며 최근에는 인공지능의 기법들이 현실 세계의 많은 문제에 실제로 적용되기 시작하였고 상업적인 제품으로 등장하기에 이르렀다. 이 결과 인공지능기술의 파급효과가 산업계 전반에 걸쳐 널리 확산되었으며 인공지능 기술이 컴퓨터 상품시장의 큰 경쟁세력으로 등장하게 되었다. 본 고에서는 인공지능 개발시스템의 등장 배경과 기능에 대해서 살펴본 뒤에 현재 상용화되어 사용되고 있는 대표적인 시스템들을 비교 분석하고 이를 바탕으로 인공지능 개발시스템의 개발추세와 필요한 세부 기술 사항들을 정리하였다. 그리고, 국내 동향을 알아본 후, 현재 한국과학기술원 전산학과 인공지능 연구실에서 개발중에 있는 복합형 인공지능 개발시스템, HyKET을 소개하였다.

Ⅱ. 인공지능 개발시스템이란?

인공지능을 이용한 응용분야 중 가장 성공한 분야가 지식기반 시스템이라 불리우는 전문가 시스템이다. 전문가 시스템은 특정 분야의 전문가들이 가지고 있는 전문지식을 컴퓨터내에 구현한 후에 이 지식을 바탕으로 추론하여, 일반 사용자들에게 전문적인 도움을 제공하는 대표적인 인공지능 프로그램으로서, 인간 전문가가 지니는 인간적인 약점을 방지할 수 있을 뿐만 아니라 인간 전문가와 같은 수준의 전문지식과 추론능력을 발휘할 수 있어서 의학진단, 재정계획, 광물탐사, 법률문제, 고장진단, 조립계획 등 다양한 분야에 걸쳐서 상당한 성공을 거두고 있다.[1,2] 전문가 시스템은 지식을 컴퓨터에 입력하고 이를 이용하여 추론 과정을 거친 후 주어진 문제를 해결한다는 점에 있어서 지금까지 개발된 기존의 프로그램과는 큰 차이가 있으며 개발방법도 다르다. 전문가 시스템의 특성 중 두드러진 몇가지를 나열하면 다음과 같다. 첫째 전문가의 경험적인 지식을 기반으로하여 문제를 해결한다. 전문가 시스템은 일반적인 전문지식뿐만 아니라 문제를 해결하는데 중요한 역할을 담당하는 전문가의 경험적인 지식(heuristics)을 이용하여 불충분하고 상호 상충되며, 불확실한 입력자료를 근거로하여 믿을 만하고 타당성 있는 결론을 이끌어낼 수 있다. 이는 기존의 시스템이 알고리즘만을 이용했던 것과 비교될 수 있으며, 따라서 지식표현 시스템이 큰 비중을 차지하게 된다. 둘째 프로그램 제어가 불확실하다. 기존의 프로그래밍 방식에서는 대개 머리속으로 해당문제의 해를 완전히 구한 뒤에 단순히 연산만을 컴퓨터에 맡기는데 반해 전문가 시스템에서는 문제해결방식의 결정까지도 시스템 자체에 맡긴다. 세째 기존의 많은 프로그램들이 수치적 연산을 이용한 방법으로 문제를 푸는데 반하여 전문가 시스템에서는 기호를 이용하여 문제를 표현하고 해결한다. 즉 주어진 문제를 기호를 이용하여 표현하고 컴퓨터에 미리 인식시켜둔 패턴과 비교 분석함으로써 결론을 도출한다. 실제로 인간이 실세계의 문제를 해결할 때 이러한 문제 해결방식을 사용한다는 것이 심리적인 연구를 통하여 발견되었다.

이와같이 전문가 시스템은 분명히 기존의 프로그램과 그 성격이 다르며 이로 인하여 기존의 프로그램 개발방법과는 다른 개발방법을 택하고 있다.[3] 기존의 프로그램은 프로그램을 하기 전에 벌써 그 해

결방법을 알고 있는 문제들을 다루고 있다. 즉 미리 알고 있는 문제해결 절차나 알고리즘을 프로그램으로 바꾸는 것이다. 따라서 기존의 소프트웨어 개발은 주로 top-down 개발방법을 쓰기에 적합하다. Top-down 개발방법은 프로그램 개발과정 중에서 발생하는 설계의 변경을 최소화 하는데 그 의미가 있다. 하지만 프로그램 개발도중이나 개발된 후 설계상의 오류로 인하여 변경이 필요한 경우가 발생하면 굉장히 많은 시간과 경비를 낭비하게 된다. 반면에 전문가 시스템을 개발하는 경우에는 top-down 개발방법이 부적합하다. 전문가 시스템을 개발할 때에는 기호, 상호관계, 전략 등을 포함하는 경험적 지식을 시스템내에 표현해야 하며 이러한 개념들을 기존의 PASCAL 이나 C와 같은 프로그래밍 언어를 이용하여 표현하는 것은 힘든 일이다. 더우기 전문가 자신도 인간 전문가가 어떻게 결론을 이끌어내는지에 대해 정확하게 알고 있지 못하다. 따라서 전문가 시스템의 개발은 여러번의 반복작업을 통하여 점점 향상된 시스템으로 변화시켜가는 개발방법을 사용해야 한다.

전문가 시스템은 전문가의 지식을 간직하고 있는 지식베이스와 이러한 지식을 효과적으로 적용할 수 있는 추론기관을 이용하여 불확실한 사실로부터 결과를 추론할 수 있는 능력을 가지고 있다. 특정한 응용분야와 관련되는 전문가의 경험으로부터 얻을 수 있는 사실, 규칙과 이들 간의 상호관계를 포함하고 있는 지식베이스는 기존의 프로그래밍 기법으로는 구축하기가 매우 힘이 들고 개발과정의 어려움만 증가시키게 된다. 따라서 전문가 시스템을 개발할때 이러한 지식을 어떻게 용이하게 프로그램할 수 있느냐가 중요한 문제가 되며 이에 대한 많은 도구들이 개발되어 사용되고 있다. 전문가 시스템 개발은 반복적인 과정을 거쳐 시스템을 점진적으로 향상시키는 개발 방법을 사용해야 한다. 그리고 이와 같은 개발기법들을 적용하기 위해 기존의 개발도구와는 다른 전문가 시스템 개발도구가 필요하게 되었다. 초창기의 전문가 시스템 개발자들은 LISP과 같은 인공지능 프로그래밍 언어를 사용하여 직접 전문가 시스템을 개발하였다. 이러한 노력의 결과 LISP 환경위에 전문가 시스템을 용이하게 개발할 수 있는 특별한 도구들이 나타나게 되었으며 이러한 개발도구들은 전문가의 지식을 편리하게 표현할 수 있고 표현된 지식을 이용하여 결론을 추론할 수 있는 기본적인 골격을 제공하게 되었다.

초기의 인공지능 개발시스템들은 일차술어논리,[4] 규칙(rule),[5] 프레임,[6] 의미 네트워크, 스크립트(script) 등과 같은 한가지의 지식 표현방법을 사용한 단순한 시스템이었다. 그러나 인공지능 시스템이 다루는 문제가 어느 한가지의 지식표현 방법으로는 자연스럽게 여러 분야의 전문지식을 표현할 수 없다는 사실을 발견하게 되어 최근에는 다양한 지식표현 방법을 제공하는 복합형 인공지능 개발 시스템이 많이 사용되고 있다. 복합형 인공지능 개발시스템은 여러가지의 지식표현 방법을 사용자에게 제공함으로써 사용자가 전문지식을 자연스럽게 표현할 수 있도록 하였으며 이로 인하여 전문가 시스템의 응용 범위를 더욱 다양하고 복잡한 영역까지 확장할 수 있게 하였다. 뿐만 아니라 복합형 인공지능 개발시스템은 윈도우 시스템, 그래픽 시스템, 객체 지향 언어, LISP 개발 도구 등으로 구성된 강력한 개발 환경을 제공함으로써 인공지능 시스템을 개발하는데 드는 시간과 노력을 감소시켜 준다.[12] 이와 같이 개발시스템들은 점점 발전되어 현재는 다양한 지식 표현 방법과 강력한 프로그램 개발 환경을 제공하는 복합형 인공지능 개발시스템이 주류를 이루고 있다.

Ⅲ. 상용 인공지능 개발시스템의 비교분석

최근들어 전문가시스템과 이의 개발을 위하여 사용되는 소프트웨어 도구, 즉 인공지능 개발시스템에 대한 관심이 고조되고 있다. 그러나 현재 상품화되어 판매되고 있는 상용 인공지능 개발시스템에 대한 정보가 국내에는 충분히 알려져 있지 못한 실정이다. 이러한 상용 시스템들은 주로 응용시스템을 개발하거나 이의 시제품을 만드는데 사용되며 앞으로 수요가 계속 증가될 것이다. 따라서 사용자가 여러 시스템 중에서 사용목적에 알맞는 시스템을 선택하는데 도움이 되도록 각 시스템의 특징을 비교분석하는 것이 필요하다.

이 비교분석을 위하여 평가기준을 정하였으며 이를 바탕으로 현재 가장 많이 사용되고 있는 대표적인 상용 시스템 4 가지(ART, KEE, Knowledge Craft, S1) 를 분석하였다. 분석을 위한 자료로는 주로 기술보고서, 잡지 기사, 사용자 지침서, 교육자료, 선전책자들을 이용하였다. 여기에서 이용한 자료는 주로 1986년도 경에 발표된 자료들임을 밝혀둔다.

1. 평가기준

개발 시스템은 사용목적을 우선적으로 반영하여 선정되어야 한다. 한 분야의 응용에 아주 우수한 시스템이 다른 응용분야에 대해서는 좋지 않은 경우가 대부분이다. 평가 기준은 각 개발시스템의 특성을 잘 반영할 수 있어야 하며 가격 대 성능의 비교도 꼭 필요한 평가 기준이 된다. 특히 개발시스템은 사용자에게 얼마만큼 편리한 개발환경을 제공해 주느냐가 중요한 기준이 된다.[13]

1) 기본 특성

기본 특성은 지식표현 방법과 이에 사용되는 지식표현언어 그리고 추론방법과 제어 구조의 특성을 나타낸다. 모든 인공지능 개발시스템들은 지식을 표현할 수 있는 방법을 제공하는데 보통 술어논리, 논리 프로그래밍, 규칙, 의미망, 프레임 또는 스키마 등이 있으며 각 방법은 그 방법이 표현하기에 효과적인 지식의 형태가 정해져 있다. 최근에는 여러 형태의 지식을 표현할 수 있는 복합형 인공지능 시스템이 개발되는 추세이다. 또한 인공지능 개발시스템들은 연역 기능, 순방향, 역방향 규칙 해석기, 계층 상속기능 등의 내장된 추론기능을 제공한다.

2) 개발환경

개발환경이란 인공지능 개발시스템을 사용하여 응용시스템을 개발할 때 개발과정을 용이하게 하여 주는 프로그래머 인터페이스로서 지식베이스 편집기, 규칙 컴파일러, 지식베이스 Browser, 성능평가 도구등으로 구성된다. 지식베이스 편집기는 프로그래머 또는 지식공학자가 지식베이스를 구축하는 과정을 도와준다. 한편 지식베이스내의 상충되는 지식들을 찾아 주거나 누락된 정보들을 지적하여 주며, 지식베이스의 내용을 그래픽으로 나타내어주어 지식공학자로 하여금 전체 내용을 쉽게 파악할 수 있게 하는 지식습득 도구도 개발환경을 이루는데 중요한 역할을 한다.

근래에 들어 그래픽 도구들은 인공지능 개발시스템에 필수적인 도구가 되어가고 있으며 지식베이스뿐만 아니라 에러 수정도구, 모니터 도구등과 결합되어 이들 도구의 성능을 향상시켜 주는데 이용되고 있다.

LISP 개발환경으로는 대개 원하는 곳에서 현재의 binding 환경을 알아볼 수 있는 Break 기능, 명령을 순서대로 하나씩 수행하여 그 과정을 보여주는 Stepper, 한개의 LISP 객체를 일목요연하게 보여주는 Inspector등이 있는데 이러한 기능과 도구들이 단순한 프로그램언어가 아닌 인공지능 개발시스템으로써 제공되고 있다.

3) 기타 사항

각 개발시스템이 사용될 수 있는 하드웨어의 종류, 사용의 난이도, 사용방법 습득의 어려움, 응용범위, 응용시스템의 크기 등도 인공지능 개발시스템의 중요 평가기준이 된다. 아울러 교육, 사용 방법 문서화, 정비보수 체계 등의 지원사항과 시스템의 가격 및 기타 부가적인 비용도 중요한 평가기준이 된다.

2. ART(Automatic Reasoning Tool)[3,7,8,9,10]

ART는 Inference Corporation사에서 개발한 인공지능 개발시스템으로서 강력한 규칙기반 시스템을 바탕으로 하여 가상적 추론기능, 규칙, 프레임, 논리, LISP 프로그래밍과 같은 다양한 특성들을 한 시스템에 통합시킨 시스템이다. Common loops를 기초로한 객체 지향적 프로그래밍 특성이 ART의 프레임 시스템에 추가되었다.

1) 기본 특성

ART에서는 선언적 지식(declarative knowledge)을 표현하기 위하여 명제(proposition)를 사용하며 어떤 사실(fact)은 명제와 이 명제가 어떤 상황 또는 관점(viewpoint)에서 유효한 지를 나타내는 extent로써 표현된다. 변수를 포함한 명제를 패턴이라고 부르는데 이 패턴에는 심볼의 값, 논리 연산자, 임의의 술어 등을 포함할 수 있다. 스키마타(schemata)를 사용하여 객체(object)와 객체들의 클래스(class)들을 표현할 수 있으며 슬롯에 대하여 추론할 수 있다. 하지만 슬롯 값에는 제약을 줄 수 없고 슬롯에 LISP의 함수와 같은 순차적 명령(procedure)을 결합할 수 없다. 한편 사용자는 relation을 정의할 수 있으며 정의된 relation에 의해 새로운 상속관계가 형성된다. ART의 규칙시스템은 절차적 지식을 표현하는데 주로 사용되며 순방향, 역방향 추론이 가능하고 conflict resolution 전략을 사용자가 임의로 정의할 수 있다. 또한 규칙의 조건부분과 실행부분을 임의로 정의할 수 있으며 규칙의 조건부분과 실행부분에서 임의의 LISP 함수들이 불려질 수 있다. ART의 가장 두드러진 특징중의 하나는 가설적 추론(hypothetical reasoning)을 위한 viewpoint라는 기능을 제공하는 것이다. ART는 많은 수의 viewpoint를 동시에 처리할 수 있다. Viewpoint는 기능상으로 De Kleer의 Assumption-Based Truth Maintenence 시스템[11]과 비슷하다. 한편 ART는 불확실성 처리를

위하여 사용자로 하여금 각 viewpoint의 유효성을 수치로 표현하여 믿음의 정도를 표현할 수 있게 하였으며 지식 베이스내의 각각의 사실에 대해서도 Certainty Factor를 이용하여 믿음의 정도를 표현할 수 있게 하였다.

2) 개발 환경

ART의 개발환경은 크게 사용자 인터페이스 및 그래픽 패키지와 지식베이스 편집 및 디버거 등으로 나눌 수 있다. 사용자 인터페이스 및 그래픽 패키지는 ART Studio라 불리는데 지식베이스 Browser, 수행 모니터(execution monitor), 그래픽 패키지 등을 포함한다. ART는 내장된 도움말 기능이 있어서 사용자가 언제든지 시스템의 명령이나 언어 특성 등에 대해서 설명을 받을 수 있다. 한편 명령어 입력은 철자법 교정이나 약어 입력 등의 기능으로 인하여 매우 쉽게 할 수 있다. 그래픽 패키지는 ARTIST라 불리는데 사용자가 필요한 아이콘(icon)을 만드는 것을 도와주며 각 아이콘들은 스키마타로 표현되고 그래픽 에디터를 이용하여 사용자가 아이콘을 만들면 이에 해당되는 코드가 자동적으로 ARTIST에 의하여 만들어진다.

계층적인 지식구조를 그래프 형태로 화면에 나타낼 수 있는 기능이 있으며 구조적 지식베이스 편집기도 제공된다. 또한 시스템 개발자가 수행환경에서 시스템의 상태를 관찰하고 디버깅할 수 있는 모니터가 제공되며, 수행을 추적하고 break point를 정하여 수행상태를 관찰할 수 있는 디버깅환경을 제공한다.

3) 기타 사항

ART는 Common LISP으로 쓰여진 개발도구로서 Symbolics, LMI lambda, DEC Vax, TI Explorer 와 같은 LISP 머신 상에서 사용되며 1986년말에 IBM-PC/RT용이 소개되었다. 그리고 SUN-3 워크스테이션에서도 사용 가능한 버젼을 개발하였다.

ART는 강력한 프로그래밍 환경을 제공하므로 규모가 큰 문제를 풀기에 적당하다. ART를 이용하여 개발된 프로그램은 실시간 처리 응용에 사용되지만 일반적인 범용시스템에 비해 훨씬 많은 컴퓨터 자원을 요구한다. 따라서 Inference사에서는 적은 용량을 필요로 하면서 LISP 버젼과 같은 수준의 성능을 발휘할 수 있는 C-버젼을 UNIX 시스템하에서 개발하였다.

3. KEE(Knowledge Engineering Environment)[8,10,12]

KEE는 Intellicorp사에서 개발하여 판매하며 프레임기반 지식표현방식을 바탕으로하여 규칙기반 추론, 논리기반 추론, 논리 표현, Data-driven 추론, 객체지향적 프로그래밍, 대화형 그래픽스, LISP프로그래밍 등과 같은 다양한 인공지능 기법들을 하나의 시스템 개발환경으로 통합한 대표적인 복합형 인공지능 개발시스템이다. 가설적 추론기능(hypothetical reasoning)과 시제 추론기능(temporal reasoning)이 포함되어 있다.

1) 기본 특성

KEE에서는 프레임을 유닛(unit)이라고 부른다. 유닛은 유닛명과 여러 슬롯(slot)에 의해 정의되는 속성(attribute)들의 집합으로 이루어진다. 각 슬롯에는 그 슬롯이 사용되거나 슬롯의 값에 대한 제약조건(constraints), 그리고 그 슬롯이 사용되거나 슬롯의 값이 변화할 때 불리어지는 프로시듀어로 이루어진다. 슬롯은 크게 own slot과 member slot으로 구분되는데 own slot에 의해서 실제의 인스턴스가 표현되고 member slot에 의해서 객체 집단(class)의 부집단(subclass)이 표현된다. 따라서 유닛은 속성의 상속관계를 유지하는 계층구조(class taxonomy)를 형성하게 된다.

규칙(rule)에는 논리 연산자를 포함한 논리식이나 임의의 LISP기호식을 포함할 수 있다. 규칙해석을 위해 역방향 추론과 순방향 추론기능 모두를 기본적으로 제공하며 사용자가 정의한 함수로써 규칙해석을 사용자 용도에 맞게 수정할 수 있다. 논리식은 KEE에서 슬롯값에 대한 제약조건을 표현하는데 사용되며, 규칙내부에 있고 술어논리에 기반을 둔 질의어인 TellAndAsk에서 사용된다. TellAndAsk가 기본적인 relation은 제공하고 있으나 필요에 의해서 사용자가 새로운 relation을 정의하여 사용할 수 있다.

KEE에서 아이콘(icon)은 하나의 객체 또는 객체의 특성값을 나타낼 수 있다. 아이콘으로 이루어진 그래픽 표현은 지식베이스내의 객체의 프레임 표현 방법과 연결되어 있어서 사용자가 대화식으로 그래픽 이미지를 변경하면 실제 지식베이스내의 값이 변화하게 되고 이와 반대로 지식베이스내의 슬롯값이 어떤 방법에 의해서 변화하게 되면 그 변화가 아이콘을 통해 화면에 나타나게 된다. 이와 같은 기능을 제공하는 것을 통틀어 active image라 부르며 이를 이용하여 시뮬레이션 작업을 효과적으로 수행할 수 있다.

2) 개발 환경

KEE는 시스템 개발자를 위해 강력하고 융통성있

는 사용자 인터페이스를 제공하며 시스템 개발자가 응용시스템의 사용자를 위한 인터페이스를 쉽게 구성할 수 있도록 도와 준다. 하지만 아주 숙달된 사용자라야만 사용자 인터페이스를 충분히 사용할 수 있으며 경험없는 사용자에게는 사용하기가 어렵다는 단점이 있다.

또한 그래픽 지식베이스 편집기를 제공하는데 이는 프레임내의 객체와 집단간의 상호 계층구조를 목구조 형태의 그래프로 화면에 보여줄 수 있으며 사용자는 마우스를 이용하여 화면에 출력된 그래프의 노드를 선택함으로써 지식베이스를 편집할 수 있다. 그리고 추론이 진행되는 동안 어떤 속성의 값이 결정되는 과정과 시스템이 사용자에게 정보를 요구할 때 그 이유등을 설명할 수 있는 기능이 'Backward-Chainer Explanation Window'를 통해서 제공된다.

Rule Class Graph를 이용하여 한 집단내 규칙들 간의 상호관계를 표시할 수 있다. 또한 규칙해석 과정을 추적할 수 있는 추적기도 제공되며 디버깅을 위한 다양한 기능을 제공한다.

3) 기타 사항

KEE는 LMI Lambda, Xerox 1100s, Symbolics, TI Explorer, SUN-3, DEC VAX와 같은 컴퓨터에서 사용될 수 있으며, HP 9000 시리즈와 IBM-PC/RT 상에서 사용할 수 있는 버젼이 개발되고 있다. KEE는 다양한 응용시스템 개발에 사용될 수 있도록 설계 되었으며, 현재 재정계획, 작업계획, 감시, 시뮬레이션, 고장진단 등 다양한 응용분야에 사용되고 있다.

Simkit란 시뮬레이션 패키지를 Symbolics와 TI Explorer에서 이용할 수 있으며 PC/HOST 분산 체제 환경을 갖추면 KEE의 인터페이스만을 IBM-PC에서 수행되도록 하고 나머지는 DEC VAX에서 수행되도록 할 수 있다.

4. Knowledge Craft[8,10,13]

Knowledge Craft는 카네기그룹에서 개발된 KEE와 유사한 인공지능 개발시스템으로서 프레임, 규칙, 논리, 객체지향언어등과 LISP 프로그래밍이 하나로 결합된 복합형 시스템이다. 그외 비망록 메카니즘(agenda mechanism), 스키마기반 그래픽 인터페이스, 대형 데이타 베이스처리를 위한 다 사용자 (multi-user) 데이타베이스 처리기능을 가지고 있다. Knowledge Craft는 SRL(schema representation language)을 기본으로 개발되었으며 SRL+가 Knowledge Craft의 이전 이름이다.

카네기그룹에서는 Knowledge Craft를 이용하여 지식기반 시뮬레이션, 소프트웨어 공학 및 제품관리, 고장진단 등 더 실제적인 문제를 더 쉽게 해결할 수 있는 도구(shell)를 개발하고 있는 중이다.

1) 기본 특성

Knowledge Craft에서 프레임에 해당하는 것은 스키마이다. 스키마는 한 객체 또는 같은 클라스의 특성을 서술하는 슬롯을 가진다. KEE에서와 마찬가지로 슬롯의 다중상속을 허락하며 슬롯의 값에 제한을 가할 수 있다. 그러나 Knowledge Craft는 슬롯의 값을 제한하는 정보가 분리된 스키마에 저장된다는 점이 KEE와 다르다. KEE와 마찬가지로 슬롯에 method, 규칙, KEE의 active value에 해당하는 demon이 저장될 수 있다. 규칙시스템으로는 OPS5에 기초하고 OPS5와 호환성 있는 CRL-OPS가 순방향 추론기능을 제공하며, LISP 문법형태만 제외하면 DEC-20 PROLOG와 호환성 있으며 PROLOG를 기초로 하는 CRL-PROLOG가 역방향 추론기능을 제공한다. 그러나 이러한 CRL-OPS 또는 CRL-PROLOG 규칙들은 스키마로 표현될 수 없으며 이것이 KEE나 ART 등과 다르다. 그리고 시뮬레이션 시각에 따라 필요한 동작을 순서대로 진행시키키 위한 제어구조로 사건 queue가 있다. 각 사건은 스키마로 표현되며 agenda라 불리는 queue에 놓아지고 지정된 시간에 queue의 사건들이 수행되게 된다. 이러한 기능은 복잡한 프로세서의 사건기반 시뮬레이션(event-based simulation)을 가능하게 한다.

2) 개발환경

Knowledge Craft의 개발환경은 사용자 인터페이스를 위한 그래픽 패키지, 지식베이스 편집, 대화식 명령체제와 그외 버젼경영 및 데이타베이스 체제등으로 분류될 수 있다.

사용자에게 2D 그래픽 화면을 제공하기 위해 표준화 되고 있는 CORE와 스키마 기반 그래픽 패키지가 제공된다. 이것은 다중윈도우 화면을 다루기 위한 함수를 제공하는 표준화 패키지이다. Knowledge Craft에서는 선과 문자를 포함하는 모든 그래픽스 객체는 스키마로 표현된다. 한 문자열이 여러 윈도우에 보여지고 있을 때 한 곳의 수정은 다른 곳에서도 자동적으로 바뀌어 보여지게 되어 일관성을 유지하게 되는데 이는 KEE, ART, S1에서는 아직 볼 수 없는 편집기능이다.

Knowledge Craft의 지식베이스 편집을 위해서 그래픽 인터페이스가 역시 사용된다. 그래픽 인터페이스는 스키마를 browse하고 수정가능하게 하며 스키

마를 그래프로서 수정 할 수 있게 해주기 때문에 사용자가 지식베이스를 편집하기가 매우 편리하다. CRL-PROLOG와 CRL-OPS의 사용을 편리하게 하여주는 Workbench도 있다.

대화식 명령어시스템으로서 중요어, 제어문자, 마우스를 사용할 수 있으며 온라인 도움말기능을 제공한다. 특히 사용자는 마우스만을 사용하여 키보드 명령과 동등한 명령을 수행할 수 있다. 또한 스키마 기반 에러운용시스템을 제공한다.

그외 비젼 경영시스템과 대규모 데이타베이스 경영시스템을 제공한다. 데이타베이스 경영시스템은 현재 DEC VAX에서만 가능하다.

3) 기타 사항

Knowledge Craft는 Common LISP으로 쓰여졌으며 DEC VAX와 Symbolics등 LISP 머신에서 사용할 수 있다. 카네기그룹은 그외 Common LISP을 제공하는 다른 기계에도 설치할 계획을 가지고 있다.

5. S1[8,10,14]

S1은 Tecknowledge사에서 개발된 인공지능 개발 시스템으로서 규칙 기반 시스템이다. S1은 주로 많은 해들이 존재하는 문제에서 구조화된 선택(structured selection) 문제를 풀기 위해서 설계되었다. S1의 설계는 EMYCIN에 기초하였으며 제어블록(control block) 등은 ONCOCIN에서 유래된 것이다. S1은 ART, KEE, Knowledge Craft등에 비해서 단순한 시스템이기 때문에 적용 영역이 보다 제한된다. 그러나 Tecknowlege사는 S1이 고장진단, 목록 선택, 공학용 설계 문제 등에 사용하기에 적당하다고 말하고 있다. 현재 Release 2.0이 제공되며 Release 2.1이 개발 중이다.

1) 기본 골격

규칙 기반 시스템의 규칙에 Certainty Factor를 사용할 수 있으며 규칙에서 LISP함수가 사용될 수 있으나 변수는 허락하지 않는다. 규칙 해석시에는 역방향 추론이 기본적인 제어체제이다. 규칙안의 하나의 절은 객체-특성이름-특성값 세가지로 구성되고 역시 여기에 Certainty Factor로 믿음의 정도를 표시할 수 있다. S1의 큰 특징은 절차적인 지식을 잘 표현한다는 것이다. 높은 수준의 절차적인 지식을 표현하기 위하여 제어블록(control block)이란 것이 존재한다. 이곳에서 하나의 객체를 만들거나 한 속성의 값을 결정할 수 있으며, 또다른 제어블록을 호출할 수 있다. 객체간의 관계를 표시하기 위해 프레임 시스템을 사용하는데 사용자는 클라스와 클라스의 특성을 정의할 수 있다. 그러나 상속기능과 수행시 슬롯의 제한을 가하는 기능, method, demon 등은 존재하지 않는다. 이것이 S1의 큰 약점이다. 그러나 객체간의 관계는 이 프레임 시스템을 통하여 전파될 수 있다. 또한 프로시듀어호출은 기반 운영체제에 의존하며 LISP, C등을 제공한다.

2) 개발 환경

S1의 인터페이스는 대부분 메뉴를 통해 이루어진다. 다중 윈도우를 제공하며 도움말 윈도우, 상태표시 윈도우가 있어서 현재의 질문, 명령, 옵션 등을 서술한다. 지식베이스에 있는 객체들은 문자열과 관계를 가지고 있다. 이것은 if-then 형식의 규칙이 쉽게 자연어 비슷한 언어로 바뀔 수 있는 것을 의미한다. 이러한 기능은 설명 기능에 매우 유용하다.

S1의 지식베이스 수정기는 문법과 일치성을 검사할 수 있는 기능을 가지고 있다. 지식공학자가 지식베이스와 수행의 상태를 수행 중에 browse할 수 있다. 사용자는 규칙 시스템을 추적하면서 일어나는 사건들을 추적 윈도우에서 볼 수 있으며 break 기능등도 가지고 있다.

3) 기타 사항

S1은 LISP으로 쓰여진 Xerox, Symbolics 등과 DEC-VAX, Apollo, Sun, IBM PC/RT, HP9000, NCR Tower 등의 UNIX기반 시스템 등에서 사용할 수 있으며 KEE, ART 등에 비해 훨씬 다양한 하드웨어에서 수행한다는 것이 큰 장점이다.

Tecknowledge사는 S1 사용자에게 선택에 따라 교육을 하고 있으며 소프트웨어와 documentation도 계속적으로 제공하고 있다.

Ⅳ. 개발추세 및 세부 기술 사항

1. 개발추세

아직까지 인공지능 개발 시스템은 대부분이 고가이며 또한 고가의 하드웨어에서 사용되고 있다. 하지만 VLSI 기술의 발달로 인하여 하드웨어 값이 싸지게 되면 현재의 LISP 머신과 같은 고가의 인공지능 워크스테이션도 일반화되리라 예상된다. 실제로 TI에서는 Explorer라는 LISP머신의 프로세서를 한개의 칩(chip)으로 만든 "Mega Chip"이라는 것을 개발한 상태이다. 그리고 전문가 시스템의 가치가 점점 인정되어 이의 시장이 앞으로 계속 급격히 늘어날 전망이므로 자연히 이의 연구와 개발에 투자

하는 비용이 늘어나게 될 것이고 이로 인하여 다양한 제품이 개발 생산되어, 경쟁으로 인한 가격하락을 초래하게 될 것이다. 또한 인공지능 시스템의 값이 고가인 가장 큰 원인은 이를 개발하는 전문 지식공학자가 드물고 이들의 인건비가 굉장히 비싸기 때문인데 하드웨어와 소프트웨어 가격하락으로 인하여 인공지능 시스템 개발환경이 보편화 되면 전문 지식공학자도 많이 배출할 수 있게 되어 개발에 따른 인건비도 많이 줄어들게 될 것이다.

이러한 여러 추세로 볼 때 인공지능 개발시스템의 세계시장은 현재 도입기라고 볼 수 있으며, 국내에서는 이러한 시스템의 개발을 시작하려는 상태이며 이의 개발이 끝나서 수출을 할 수 있을 때가 되면 세계시장은 성장기에 접어 드리라고 생각된다.

현재 인공지능 개발시스템의 시장은 다수 지식표현 방법 및 추론방법, 가설적 추론기능(hypothetical reasoning), 진리유지 기능(truth maintenance), 객체지향적 프로그래밍 환경, 강력한 디버깅도구 및 그래픽 인터페이스를 제공하는 몇개 업체의 시스템에 의해 점유되고 있다.[10] 현재 판매되고 있는 상용 인공지능 개발시스템들은 제 3 세대에 속한다.[15] 1 세대 시스템은 주로 연구환경에서 개발된 것으로서 지식과 제어구조를 직접 LISP 프로그램으로 작성한 것이다. 2 세대 시스템도 1 세대와 마찬가지로 주로 연구 개발용으로 사용되었으며 지식과 추론기능을 따로 분리하여 개발에 용이하게 설계되었으며 도구의 형태를 갖추게 되었고 특정 응용분야별로 적합한 개발도구가 나타나게 되었다. 제 3 세대 시스템은 1 세대, 2 세대에서 연구용으로 개발된 시스템을 확장, 개선하여 상업용으로 개발한 시스템으로서 다양한 지식표현 방식과 강력한 프로그래밍 환경을 제공하고 있다.

2. 세부 기술 사항[16]

현재의 대표적인 상용시스템을 살펴볼 때 인공지능 개발시스템을 특징짓는 세부기술 사항은 크게 지식표현, 추론 및 제어, 개발환경의 세가지로 구분할 수 있다.

1) 지식표현

지식표현 방식으로서는 규칙, 프레임, 객체 등이 있다. 규칙은 지식을 표현하기가 쉽고, 정형적인 표현 방법을 제공하므로 가장 많이 사용된다. 하지만 복잡한 문제에 적용하기에는 비효율적이고 유지 및 보수가 어렵다는 단점이 있다. 프레임은 클래스(class)의 골격을 제공하는 것으로서 상속기능이 있고 프로시듀어와 결합할 수 있다. 객체는 프레임에 의해 표현되며 객체들간에는 메시지를 전송함으로써 서로 통신할 수 있다. 객체는 복잡한 관계를 표현하기에 효과적이다. 앞으로 새로이 요구되는 지식표현 기능은 다음과 같다.

- 지식 습득 도구
- 인과모델의 표현
- 공간적 시간적 관계의 표현
- 불확실성의 표현
- 지적 지식베이스 유지 보수 기능

2) 추론 및 제어

규칙기반 추론기능으로 역방향 추론기능과 순방향 추론기능이 있다. 추론시 여러개의 대상규칙 중에서 하나의 규칙을 선택하는 상충해결방식(conflict resolution)이 필요하며 규칙 컴파일러가 필요하다. 그외에 프로시듀어를 접속시켜 사용할 수 있는 기능이 필요하고 객체 지향언어에서 상속기능과 메소드(method)를 이용한 추론기능이 필요하다. 그리고 접근지향 프로그래밍(access-oriented programming)이 가능해야 하고 진리유지 기능(truth maintenance) 기능이 제공되어야 한다. 앞으로 새로이 요구되는 추론제어 기능은 다음과 같다.

- 지적인 제어 방법
- 여러 단계의 추상화
- 추론기능의 성능 평가를 위한 방법론

3) 개발환경

지식베이스를 관리하기 위해 새로운 지식의 추가, 삭제를 용이하게 하는 유틸리티들이 필요하고 지식베이스의 순수성을 검사할 수 있는 기능이 필요하다. 한편 강력한 디버깅 환경이 필요하며 윈도우 시스템과의 결합이 요구된다. 앞으로 새로이 요구되는 개발환경 세부기술 사항은 다음과 같다.

- 지식베이스의 의미를 이해하는 지식베이스 관리 프로그램
- 지식베이스 버전-콘트롤(version-control) 유틸리티
- 성능 평가도구
- 새로운 지식공학 방법론

V. 국내동향

국내에서도 인공지능 시스템을 개발하고자 시도하고 있으며 이를 위한 개발시스템은 대부분 외국에서 상업적으로 개발된 시스템을 구입하였거나 이의 구입을 계획하고 있는 실정이다. 국내에 소개되어 사

용되는 시스템은 규칙기반 시스템인 OPS5가 가장 보편적인 시스템이고 PC에서 사용되는 GURU, INSIGHT2 등과, MRS, LMA 등의 시스템들이 일부 기업체나 연구소, 학교등에서 연구용으로 사용되고 있으며, KEE와 같은 고가의 복합형 시스템은 현재 한 두 기관에서 사용되고 있다. 하지만 이러한 상용 인공지능 개발시스템의 사용이 점점 늘어나고 있는 추세이다.

현재 인공지능에 대한 인식이 날로 새로와지고 있고, 이에 대한 투자도 급격히 늘어날 전망이어서 앞으로 인공지능 개발시스템의 수요가 증대되리라고 생각된다. 하지만 이러한 시스템들은 국산품이 아직 없고 대부분 값이 매우 비싸서 구입하기에 많은 어려움이 있다. 비록 이러한 시스템을 구입하였다 하더라도 실무에 적용하다 보면 문제의 특이성에 따라 수정보완이 많이 필요하게 되는데 그 내부의 구성을 이해하기가 용이하지 못하므로 수정작업이 힘들게 된다. 따라서 국내에서도 강력한 인공지능 시스템 개발환경을 갖춘 복합형 인공지능 개발시스템의 개발이 필요하다. 국산 시스템이 개발되면 외화의 낭비를 줄일 수 있고 아울러 국산 워크스테이션에서 사용할 수 있도록 하여 수출상품화도 할 수 있을 것이다. 현재 몇몇 기관에서 인공지능 개발시스템의 개발을 위한 연구과제를 수행하고 있다. 하지만 이들 기관들의 연구목표 및 방법이 서로 유사하여 잘못되면 얼마되지 않는 국내 인공지능 연구인력을 낭비하는 결과를 초래할 수 있다. 이들간의 정보교환과 의견 조정을 통해 중복된 연구개발을 막을 수 있도록 노력해야 한다.

Ⅵ. HyKET (Hybrid Knowledge Engineering Tool) 의 소개[17]

국내에서도 인공지능에 대한 열의와 관심이 고조되고 있고 인공지능 시스템의 필요성이 학교, 연구소, 기업체 등에서 증대되고 있으며 이에 대한 연구와 투자가 늘고 있는 실정이다. 하지만 현재 국내의 인공지능 개발환경은 아주 미비한 상태이며 이러한 환경 구축을 위해 고가의 하드웨어 및 소프트웨어들을 수입해야 하고 또 이의 사용이 어려워서 쉽고 저렴한 가격의 국산 인공지능 개발시스템이 절실히 요구되고 있다.

이에 따라 한국과학기술원 전산학과 인공지능연구실에서는 국가 특정 연구과제로서 복합형 인공지능 개발시스템 HyKET를 개발 중에 있다. HyKET는 프레임을 기반으로 한 객체지향적 프로그래밍 환경, 규칙기반 추론시스템, 사용자 인터페이스를 위한 그래픽 패키지, 강력한 윈도우시스템, Common LISP 프로그래밍 환경과 같은 인공지능 개발기술들을 하나의 시스템 형태로 묶어놓은 복합형 인공지능 개발시스템이다. HyKET의 구성은 그림 1과 같다.

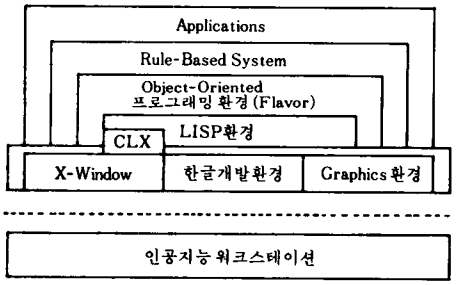

그림 1. 복합형 인공지능 개발시스템의 구성도

1. LISP 개발환경

HyKET는 기호처리기능이 가능하고 개발환경이 뛰어난 LISP 프로그래밍 언어를 기본적인 프로그래밍 언어로 채택하며 모든 환경을 LISP 프로그래밍 환경 위에 구축한다.

LISP은 인공지능 연구 및 인공지능 시스템 개발에 가장 많이 사용되고 있는 프로그래밍 언어로서 객체(object)의 개념을 실현하기가 용이하고 절차적 지식을 표현하기에도 적합하다. HyKET에서는 현재 LISP의 표준안으로 채택되고 있는 Common LISP을 사용하고 편집기, 디버거, 인스펙터 등으로 구성된 LISP 개발환경을 구축한다. 그리고 한글 기호처리를 위한 한글 LISP 환경을 구축하고[18] Common LISP으로는 KCL(kyoto common LISP)을 기본적으로 사용할 계획이다. 이에 따라 KCL의 원시코드를 분석하여 내부구조를 이해하였으며 이를 바탕으로 KCL의 단점을 보완, 개선하여 HyKET의 Common LISP 개발환경을 구축중에 있다. 그리고 한글처리 환경을 구성하였으며 입력 편집기, 한글 GMACS등을 개발하였고[19] KCL과 GNU-EMACS를 결합할 계획이다.

2. 객체지향적 프로그래밍환경

프레임을 기반으로한 객체지향적 프로그래밍 환경을 HyKET에 구축한다. 이 환경에서는 여러가지 객

체들을 표현할 수 있고 이러한 객체들이 계층적인 구조로 연결되어 상속관계를 갖는다. 또한 규칙내에서도 객체의 슬롯값이나 속성값들을 사용할 수 있으며 그래픽 인터페이스도 프레임을 기반으로 구축하고 윈도우 시스템과도 연결을 시킨다. 결국 LISP머신의 Flavor 시스템과 유사한 객체지향적 환경을 구축한다. 이의 일환으로 Flavor 시스템을 PC위의 GCLISP상에 이식하였으며 이를 바탕으로 규칙기반 시스템을 설계, 개발하였다.[20]

3. 규칙 기반 시스템

LISP 프로그래밍환경과 객체지향적 프로그래밍환경을 바탕으로 한 규칙기반 시스템은 일반적인 규칙기반 시스템의 특성을 지니면서 프레임과 결합되어 다양한 지식표현 방법을 제공한다. 역방향 추론기능과 순방향 추론기능을 제공하며 규칙의 패턴으로 Flavor의 instance를 가질 수 있다. 또한 규칙의 각 문장에서 논리연산자를 표현할 수 있고 임의의 LISP함수를 호출할 수 있다. 이의 일환으로 KAPS (KAIST Production System)를 개발하였다.[20] 한편 일반적인 규칙 시스템인 OPS5의 기능을 개선하였으며 한글처리가 가능하게 하였고,[21] PC위의 GCLISP상에 이식하여 사용하게 하였다.

4. 윈도우시스템

기본적인 윈도우시스템의 기능을 가지며 규칙 시스템, Flavor 시스템, LISP 환경과 결합되어 원활한 사용자 인터페이스 역할을 수행하고 효율적인 프로그래밍 개발환경을 제공하는 윈도우시스템을 구성한다. 현재 윈도우시스템은 세계적으로 X 윈도우로 표준화되고 있으며 특히 Common LISP 환경위의 윈도우시스템으로 CLX(Common LISP X Window)가 표준화 되고 있다.[22,23] HyKET의 윈도우시스템은 CLX를 기본으로 KCL 위에 구현할 것이다. 한편 PC상에서 프로그래머가 편리하게 윈도우를 생성, 관리할 수 있는 윈도우 시스템을 개발하였다.[17]

5. 그래픽 인터페이스

Flavor 시스템과 규칙 시스템의 내용을 그래픽으로 나타낼 수 있도록 하여 편집 등을 용이하게 하고 이들 시스템과 결합되어 HyKET의 한 객체가 아이콘(icon)으로 표시될 수 있도록 한다. 즉 active image와 유사한 그래픽 인터페이스를 구축하며 주로 마우스를 사용할 수 있게 한다. 이의 일환으로 프로그래머가 메뉴식의 시스템을 개발할 때 이를 도와주는 메뉴시스템을 개발하였다.[17]

Ⅶ. 결 론

인공지능 시스템들이 점점 복잡한 문제들을 풀 수 있게 됨에 따라 이들의 개발에 사용되는 개발도구들도 보다 더 다양한 기능과 개발환경을 요구하게 되었다. 복수의 지식 표현 방법과 다양한 추론기능, 디버깅 도구와 지식베이스 관리 기능을 포함한 강력한 개발환경 및 사용자와의 입출력을 용이하게 하여 주는 사용자 인터페이스 등으로 특정지워지는 제3세대 인공지능 개발도구들이 현재의 주류를 이루고 있다. 이러한 추세에 맞추어 국내에서도 현재 기술수준에 뒤지지 않는 개발도구에 대한 연구개발이 활발히 진행되고 있어서 멀지 않은 장래에 국내 인공지능 시장을 주도할 수 있는 인공지능 개발시스템이 개발될 것으로 기대된다.

參 考 文 獻

[1] Daniel G. Bobrow, Sanjay Mittal, and Mark J. Stefik, "Expert systems: perils and promise," *Communications of the ACM,* vol. 29, no. 9, pp. 880-894, September 1986.

[2] Donald A. Waterman, *A Guide to Expert Systems,* Addison-Wesley, Reading, MA, 1986.

[3] Chuck Williams, "ART the advanced reasoning tools - conceptual overview," Inference Corporation, 1984.

[4] M. Genesereth and M. Ginsberg, "Logic programming," *Communications of the ACM,* vol. 28, no. 9, pp. 933-941, September 1985.

[5] Frederick Hayes-Roth, "Rule-Based systems," *Communications of the ACM,* vol. 28, no. 9, pp. 921-932, September 1985.

[6] R. Fikes and T. Kehler, "The role of frame-based representation in reasoning," *Communications of the ACM,* vol. 28, no. 9, pp. 904-920, September 1985.

[7] John C. Kunz, Thomas P. Kehler and Michael D. Willams, "Applications development using a hybrid AI development system," The AI Magazine, pp. 41-54, Fall 1984.

[8] John F. Gilmore and Kirt Pulaski, "A survey of expert system tools," The Second Conference on AI Applications, pp. 498-502, February 1985.

[9] Chuck Williams, "Expert systems, knowledge engineering, and AI tools-an overview," IEEE EXPERT, Winter 1986.
[10] Mark H. Richer, "An evaluation of expert system development tools," *Expert Systems,* vol. 3, no. 3, pp. 166-183, July 1986.
[11] J. de Kleer, "An assumption-based TMS," *AI Journal,* vol. 28, no. 2, pp. 127-162, 1986.
[12] SPERRY, Intellicorp KEE Software Development System User's Manual (KEE Version 2.1), July 1985.
[13] Carnegie group, Knowledge Craft Manual (Version 3.1), August 1986.
[14] Tecknowledge, S1 User's Guide, 1985.
[15] Frederick Hayes-Roth, "The knowledge based expert system: a tutorial," *IEEE Computer,* vol. 17, no.9, pp.11-28, September 1984.
[16] William Mettrey, "An assessment of tools for building large knowledge-based systems," *AI Magazine,* vol. 8, no. 4, pp. 81-89, Winter 1987.
[17] 복합형 인공지능 개발시스템에 관한 연구: 최종 연구 보고서, 한국과학기술원, 4. 1988.
[18] 김두현, 김진형, "한글 LISP의 개발," 한국정보과학회 춘계학술발표회 논문집, 4. 1986.
[19] 김석원, 홍진수, 김영환, 김진형, "Common LISP상의 한글개발 환경구축," 한국정보과학회 추계학술발표회 논문집, 10. 1987.
[20] 송은강, 객체지향언어의 구현 및 규칙기반 시스템의 설계와 구현에 관한 연구, 한국과학기술원 석사학위논문, 1988.
[21] 김영환, 이현규, 송종수, 김진형, " 인공지능 연구를 위한 한글개발환경 구축 및 시험용 전문가 시스템 구현," 한국정보과학회 추계 학술발표회 논문집, 10. 1986.
[22] R.W. Schifler and J. Gettys, "The X window systems," *ACM Transaction on Graphics,* vol. 5, no. 2, pp. 79-109. April 1986.
[23] Kerry Kimbrough, "Windows to the future," *LISP Pointers,* vol. 1, no. 4, pp. 13-22, 1987.

♣ 用 語 解 說 ♣

Procedural language (절차형 언어)

프로그램을 작성할 때 그 실행순서를 지정하게 되는 프로그래밍 언어의 총칭. 이에 대하여 실행 순서를 지정할 필요가 없는 언어를 비절차형 언어라 한다. 이것에는 LISP 등의 함수형 언어나 프로로그(prolog) 등의 논리형 프로그래밍 언어가 포함되어 있다.

Production system (생성시스템)

前提 및 適用條件에 結論을 대응시켜 生成規則의 형태로 나타내고 그것을 바탕으로 하여 어떤 조건으로부터 결론을 유도하는 일종의 지식표현, 추론(推論)시스템. 인공지능의 분야에서는 결론이 行爲를 나타내며 短期記憶의 내용을 바꿔 쓴다. 이 내용은 생성 규칙의 전제 및 적용 조건에 영향을 미친다.

Data base language (데이타 베이스 언어)

데이타 베이스 언어에는 데이타 모델에 바탕을 두고 데이타 베이스의 성질을 형식적으로 기술하기 위한 데이터 기술언어와 데이터를 조작(操作)하기 위한 데이타 조작 언어가 있다. 데이터 조작 언어에는 단말 이용자 언어와 프로그래밍 언어가 있고, 데이타 베이스 언어에는 그 자체가 독립되어 있는 것과 COBOL, PL/I 등에 포함되어 사용되는 것이 있다.