韓國經營科學會誌
第19卷 第2號
1994年 8月

KADS 지식모델의 의미론

전윤재*

A Semantics for KADS Model of Expertise

Yoon-Chae Chon*

Abstract

KADS is currently a best known methodology for expert system development in Europe. KADS world sees the expert system development as a modelling activity and uses models to control complexity of the development process. Four layered KADS expertise model is used to describe expert konwledge. But this expertise model in high abstraction level is conceptual and not formalized. This paper has formalized KADS expetise model using set theory and mathematical semantics combined in order to reduce the ambiguity of informal models of expertise, provide a precise means of communication about the model of expertise and point out incompleteness and inconsistency of the model of expertise. Instead of first order predicate calculus, set theory and mathematical semantics are used because they are a more general and have compositive quality.

1. 서 론

보다 능률적인 프로그램 개발을 위해 1970년대에는 주로 일반적인 문제 해결 방법(general problem solver)을 개발하는 데 역점을 두었는데, 그 이후로는 세부적인 전문 지식의 중요성이 점점 부각되기 시작하였다[19]. 여기서 가장 핵심적인 문제는, 어떻게 하면 전문가가 소유하고 있는 지식을 손실없이 프로그램에 옮기느냐 하는데 있는데, 우리는 이과정을 지식 습득(knowledge acquisition)이라는 개념으로 이해하고 있다. Hayes-Roth는 이 과정을 전문가 시스템 개발에 있어서 가장 중요하고 어려운 과정이라고 말하고, 특히 이 과정을 "병의 목"에 비유하고 있다[13]. 지식 습득의 어려움은 아래와 같은 이유에 근거를 두고 있다[16].

- 전문지식(expertise)이란 무엇인가?
  일반적으로 전문지식이 무엇이며 또한 이

* 서울시립대 대학원 및 고대강사

전문 지식이 무엇으로 이루어 졌는지에 대한 정확한 정의가 없으며, 그에 상응하여 무엇이 지식 습득 과정에서 입력되어야 하는지 모호하다. 다시말해, 지식공학자(knowledge engineer)와 영역 전문가(domain expert)는 "어떤것들이 전문가 시스템에 입력되어야 하고" 또한 "언제 이 입력프로세스가 끝나는지" 정확히 모른다.

- 전문지식은 어떻게 조직되어 있는가?
  우리들은 전문가가 자신들의 지식을 어떤 형태와 개체로 표기하였는지 많이 알지 못하고 있다. 이와 더불어 지식 유도(knowledge elicitation)기술을 통한 지식 접근이 대단히 어렵다.
- 함축되어 있는 지식을 어떻게 해석하는가?
  전문 지식은 많은 경우에 명확하게 존재하지 않고, 단지 구체적인 상황과의 대치관계에서 하나의 행위를 나타나는 경우가 많다. 현재로서는 이 함축되고 복합된 지식을 명확한 구조로 유도할 수 있는 정형화된 방법이 존재하지 않는다.
- 전문지식을 어떻게 표기하는가?
  오늘날 우리가 사용하고 있는 표기형식들은 지식의 표기를 위해서 적합하지 않는 경우가 많다. 예를 들어 "비단조 추론", "시간과 공간에 관한 추론" 및 "상식"(common sense)을 위한 적합한 표기형식은 현재 부재 상태이다.

이러한 전문가 시스템 개발에 있어서의 어려움들을 부분적으로 나마 극복하기 위한 여러가지 지원 도구들이 지속적으로 연구 되었는데 초기의 연구활동 영역은, 어떻게 하면 전문지식을 컴퓨터에 적절한 형태로 표기하는가 하는 표기법 연구에 주로 중점을 두었다([5], [18]). 그러나 시간이 흐름에 따라 이러한 지식의 단순한 표기와 더불어 "표기 형식으로의 이동 과정" 또한 중요하다는 것을 인식하게 되었다[12]. 이러한 경향은 많은 지식습득 지원도구들에서 엿볼수 는데, 본 논문의 기초가 되는 KADS를 포함하여 현재 알려진 전문가시스템 개발지원도구들은 유형별로 분류하면 다음과 같다([3], [20], [28]):

- 전문가 시스템구현 도구:
  EMYCIN, KEE, OPS5, InterLisp-D, PROLOG 등
- 지식습득지원 도구:
  지식 습득 지원도구들은 영역 전문가들과 직접 대화를 나누면서, 지식베이스를 자동으로 습득하고 생성 하는 것을 특징으로 한다. 이 지식 습득 지원 도구들은 사용되어진 기술에 의해서 다시 "귀납법을 근거로 한 도구"와 인터뷰를 근거로한 도구"들로 분류될 수 있는데, AQUINAS와 같이 두 기술을 동시에 사용하는 경우도 있다.
  - 귀납법을 근거로 한 도구:
    AQINAS, OPAL, MOLE, STUDENT 등
  - 인터뷰를 근거로 한 도구:
    AQINAS, MUM, ROGET, KRITON 등
  - 지식베이스 편집도구:
    KREME, NEXPERT 등
  - 과제 및 방법의 표기언어:
    KADS, Generic Task, Expertise Specification, MUM 등

2. KADS

2.1 KADS란 무엇인가?

일반적으로 전문가 시스템 개발 접근 방법에는 크게 두가지가 있는데, 한 방법은 "Rapid Prototyping"이고 다른 한 방법은 "모델 지향적 방법"이다. "Rapid Prototyping"에서는 우선 몇 개의 예를 바탕으로 하나의 단순한 시스템 원형을 구축한 뒤, 진척될 지식유도(knowledge elicitaion)를 근거로 이 시스템 원형이 원하여진 요구사항들을 만족시킬 때마다 점진적으로 이 원형을 확장 및 개량 시켜나가는 방법인데 대표적인 예로 Hayes-Roth의 제안[14]을 들수 있다. "Rapid Prototyping" 방법은 여러가지 단점들을 안고 있는데

- 지식공학자가 데이터의 해석과 구현을 동시에 수행하여 지식의 손실이 많고
- 지식공학자의 사고 방향이 너무 빨리 구현의 세부사항에 치중되며
- 영역전문가 추론프로세스의 명확한 추상 모델이 존재하지 않는다.

"모델 지향적 방법"([2], [8], [11])에서는 반면에 데이터 분석 및 해석을 시스템 구현과 엄격히 구별하고 있는데, "knowledge level"을 바탕으로한 "지식 모델"과 시스템을 바탕으로 한 "디자인 모델"이다.

본 장에서는 본 논문의 기초가 되었던 "KADS 방법론"(Knowledge Acquistion and Documentation System)을 소개하겠는데, KADS는 현재 유럽에서 가장 널리 알려진 전문가 시스템 개발 방법론이다. KADS는 앞에서 언급한 "모델 지향적 방법"을 기초로 하고 있는데, KADS 세계에서는 전문가 시스템 개발을 주로 모델 작성행위로 보고, 모델을 전문가 시스템 개발 과정의 복잡성을 극복하는데 이용하고 있다. 여기서는 전문가시스템 구축 과정을 효율적으로 지원할 수 있는 포괄적인 모델, 기술 및 도구들이 연구개발되었는데, 구체적으로 다음과 같은 세종류의 지원을 제공하고 있다[15]("KADS 프로젝트"는 유럽 공동체에서 지원한 "ESPRIT 프로그램"의 하나인데 유럽의 여섯 파트너 회사와 대학이 참여하였으며, 진행기간은 1985년부터 1990년까지였고, 현재 후속 프로젝트인 "KADS-II"가 1994년까지 진행되고 있음[26]):

- 모델작성언어
  KADS 방법론의 기본 아이디어는 전문가시스템 개발을 실 세계 모델들이 하나의 인공세계 모델로 사상(mapping)하는 프로세스로 보는 것이다. 그래서 KADS는 두 종류의 모델작성 언어를 제공하고 있는데, 하나는 본 논문에서 다루어질 전문지식의 표현을 위한 4층으로 이루어진 지식 모델[28]이고 다른 하나는 시스템 설계를 위한 디자인 모델[22]이다.
- 생활주기(life cycle)
  전문가 시스템 개발 프로세스는 여러 단계들로 이루어졌는데, KADS 방법론에서는 우선 개념적 지식 모델 및 같은 추상 수준의 디자인 모델을 다루고 있다[24].
- 작업대(workbench)
  KADS에서는 KADS방법론 사용자에게 하나의 집적된 작업대를 제공하고 있는데. 이 작업대는 KADS방법론 지원을 위하여 KADS-

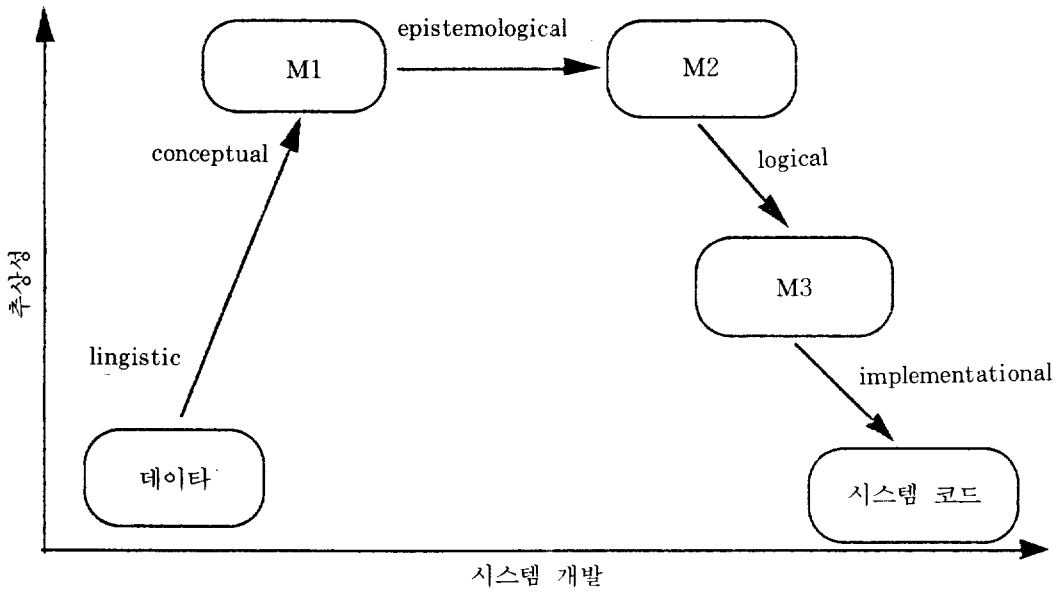
프로젝트에서 연구 모델로 개발되었다[7].

[그림 1]은 KADS의 전문가 시스템 개발 과정을 데이타에서부터 시스템코드까지 나타내고 있는데, Y-축은 모델들의 추상 수준(level)을 그리고 X-축은 모델들의 변환과정을 나타내고 있다. 이 그림에서는 의미 망 모델을 다섯 단계로 나타낸 Brachman 표기법을[4] 사용하고 있는데, 여기서 이미 지식공학프로세스의 확실한 단계 구별이 표출된다.

M1:M1은 개념과 인식론(conceptual and epistemological)의 수준에서 문제해결을 위한 모델이다. 이 모델은 영역의 전문지식 모델인데, KADS에서는 이 모델을 "개념적 지식 모델"이라 부른다.

M2:M2는 디자인 모델인데, M1과 같은 추상 수준에 있으며, 상위 수준 시스템 설계모델이다. M1과 M2의 중요한 차이점은, M2에서는 외적 요구 사항과 사용자 인터페이스가 고려된다는 점이다.

M3:M3은 앞으로 사용 되어질 구현 도구를 염두에 두고 구축되어진 "세부 디자인 모델"이다.

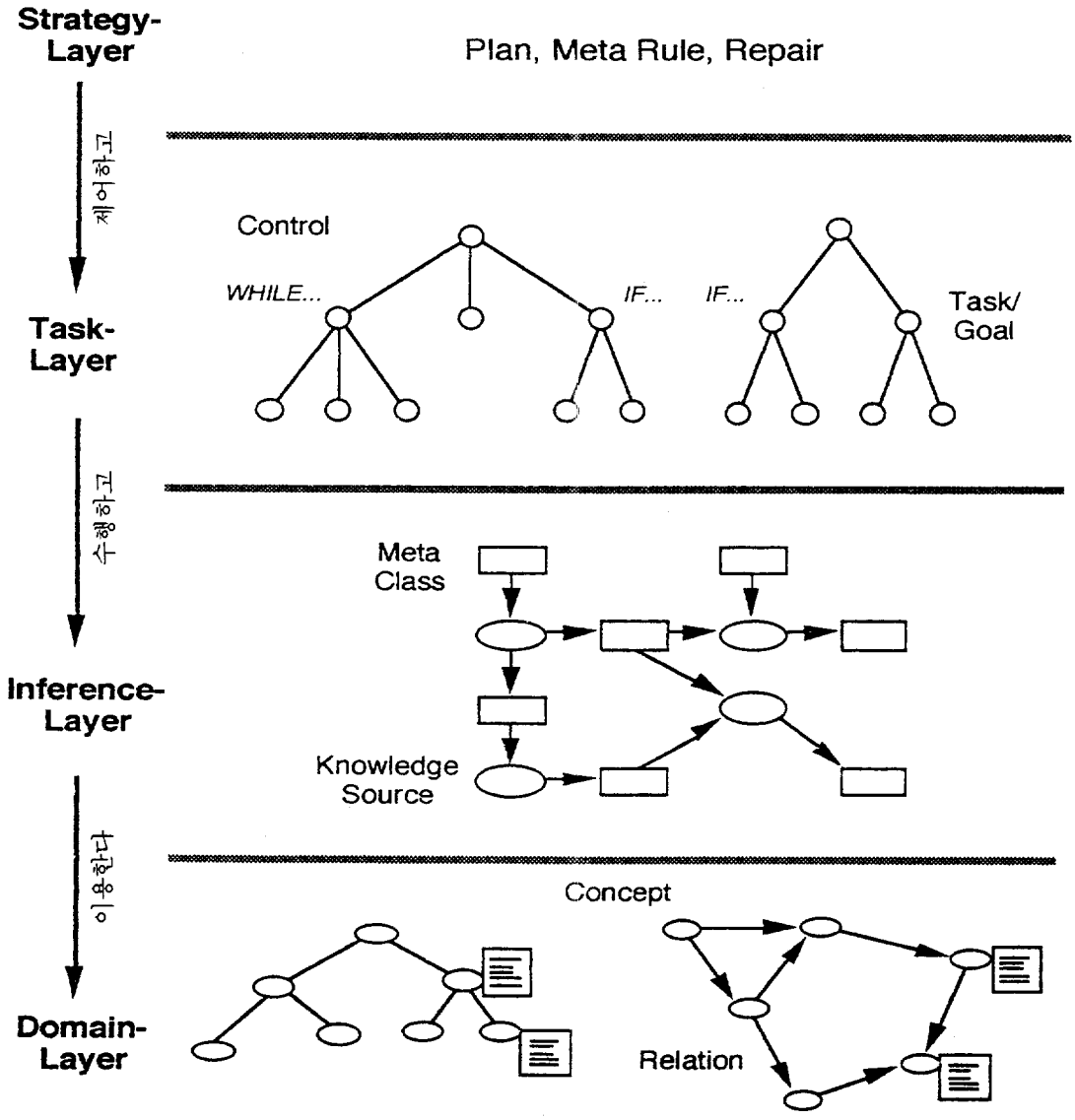


[그림 1] KADS 세계의 모델들[27]

2.2 KADS 지식모델

전문가 시스템 개발이란 전문 지식의 모델화를 의미한다. 다시말해서 전문 지식의 구조화, 효율적인 추론 전략의 수립 그리고 전문지식의 유연한 이용을 의미한다. 이 모델을 KADS용어로는 개념적 지식모델이라 부른다. 이 KADS지식 모델에서는 전문지식을 아래 [그림 2]에서와 같이 4개의 층으로 나타낸다[21].

영역 층:본 층에서는 정적인 영역지식을 개념, 관계 그리고 복합된 구조 (complex structure)로서 표기한다. 이 영역 층은 Brachman의 KL-ONE[6]에 비교될 수 있다.



[그림 2] KADS 지식 모델[21]

추론 층:추론 층에는 잠재적인 추론을 규정하고 영역층의 추상적 개념을 수행하고 있다. 이 층의 지식 원(knowledge source)은 원시함수(primitive function)로 해석할 수 있으며, 메타 클라스(meta class)는 구체적인 문제 해결 과정에서 영역 층의 개념들이 수행하는 역할을 나타내고 있다. 예를 들어, 동일한 개념이 동일한 진단 과정에서 어떤때는 증세 어떤때는 가정의 역할을 한다는 것이다. 다시말해서, 기침은 감기의 증세일수도 있고 또한 감기라는 진단에 대한 가정일 수도 있다. KADS에서는 13유형의 지식 원을 제공하고 있는데, 그들은 다음과 같다 : instantiate, classify, generalize, abstract, specify, assign_value, compute, compare, match, assemble, sort, decompose, transform.

타스크층:세번째 층은 타스크 층(task layer)인데, 이 층에서는 하나의 목적(goal)을 달성하기 위해, 어떻게 지식 원이 결합되는냐가 표기 된다. 하나의 타스크 층 구조는 지식 원의 이용 그리고 사용자 및 주위 환경의 상호작용을 제어하는 고정된 전략이다. 이 구조의 기본 요소는 "목적 문"(goal statement)과 "제어문"(control statement)이다.

전략 층:마지막 층은 전략 층(strategic layer)이다. 여기서는 전형적인 계획과 수선의 과제가 달성되어야 하는데, 현재로서는 아직 성숙되지 않은 분야이기에 때문에 본 논문에서는 제외 되었다.

4. KADS 지식모델의 취약점

앞 장에서 KADS에서 제공한 개념적 지식 모델을 간단히 소개하였는데, 본 장에서는 이 지식 모델이 안고 있는 몇가지 취약점들을 설명하겠다. KADS 프로젝트가 진행되는 동안에 KADS에서 제공한 모델, 기술 및 도구들을 평가하고 또한 평가 결과를 토대로 KADS 방법론을 개선하기 위하여 각각 서로 다른 영역에 KADS 방법론이 사용되어졌는데, 여기서 여러가지 개선안들이 지적 되었다[15]. 또한 제 1차 "KADS 사용자 만남"에서도 여러가지 취약점들이 지적 되었다[25]. 이 여러가지 취약점 중 본 논문에서는 본인이 KADS 방법론을 사용하여 하나의 프로세스 진단 시스템인 "요판 인쇄기 오류 조기발견 프로토타입 시스템"을 설계 구현하면서 발결한 취약점 보완에 역점을 두었다([9], [10]).

KADS지식 모델은 주로 전문가 시스템 개발의 초기단계, 즉 지식의 유도 및 지식의 개념화와 해석 단계에서 지식 공학자와 영역 전문가의 대화의 수단으로 이용되고 있다. 그러나 이 지식 모델은 개념적이여서 정형화되지 않했는데, 이로 인하여 이 두개체사이에서 이루어진 지식 모델과 이 모델을 바탕으로 나중에 구현되어질 시스템이 처음 의도한 요구 사항에 정확히 일치하는지 검사할 근거가 빈약하다. 다시말해서 이 비정형화된 지식 모델에서는 모델의 정확성(correctness)과 완전성(completeness)을 기대할 수 없다. 또 다른 취약점으로는, 지식 모델 층 구조간의 비집적성(not integrated)을 들수 있는데, 이 말은 각 층의 구조들이 서로 톱니바퀴처럼 맞 물리지않고 따로

따로 놀 수 있다는 것이다.

본 논문에서는 이러한 KADS의 비정형화된 지식모델을 한 단계 더 발전 시키기 위하여, 언어의 정형화를 위해 주로사용되던 방법인 1차 술어 논리(first order predicate calculus)를 피하고, 보다 보편적이고, 특히 조합성(compositive quality)을 ([17], [23]) 갖는 집합 논리와 수학적 의미론(mathematical semantics)을 복합하여 KADS 지식모델의 영역층, 추론 층 그리고 타스크 층을 각각 정형화 하였다. 또한 위에서 언급한 지식모델 층 구조간의 비 집적성을 제거 하기위해 본 논문은 정형화된 세 층 구조간에 4종류의 새로운 관계 유형을 도입하였다.

4. KADS 지식 모델의 의미론

4.1 영역 층

앞절 2.2에서 설명하였듯이, 영역 층에서는 정적인 영역 지식을 개념, 관계 그리고 복합된 구조로서 표기한다. 그래서 이 정적 영역 지식을 표현하는 "영역 층 구조"는 영역구성 성분 사이의 구조적인 관계를 표현하는 계층(hierarchy) 즉 "is_a" 및 "consist_of" 계층의 집합이다. 이 영역 층 구조의 구문(syntax)을 하나의 변형된 BNF으로 상술하며는 다음과 같다(비 단말 기호는 꺽쇄 괄호로 표기하였고, 사전에 정의된 단말 기호는 이태릭체로 표기하였다. A$^+$는, 최소한 하나의 구문적 범주 A에 속하는 성분으로 이루어졌다는 것을 의미함).

⟨Domain−Structure⟩ ::=
⟨Hierarchy⟩$^+$
⟨Hierachy⟩ ::=
⟨"is_a"−Hierarchy⟩
|⟨"consist_on"−Hierarchy⟩
⟨"is_a"−Hierarchy⟩ ::=
"is_a"−Hierarchy
⟨HierarchyName⟩
《{⟨End−Concept⟩|⟨"is_a"−Tree⟩}$^+$
subtype_of ⟨Root-Concept⟩》
⟨"is_a" −Tree⟩ ::=
({⟨End−Concept⟩|⟨"is_a"−Tree⟩}$^-$
subtype_of ⟨Between−Concept⟩)
⟨"consist_of"−Hierarchy⟩ ::=
"consist_of"
−Hierarchy ⟨HierarchyName⟩
《{⟨End−Concept⟩|
⟨"consist_of"−Tree⟩}$^+$
part_of ⟨Root−Concept⟩》
⟨"consist_of"−Tree⟩ ::=
《{⟨End−Concept⟩|
⟨"consist_of"−Tree⟩}$^+$
part_of ⟨Between−Concept⟩》

다음으로 이 영역층 구조(또한 나중의 추론 층 구조 및 타스크 층 구조)의 의미론을 상술하기 위하여 먼저 하나의 확장 사상(extention mapping) ε를 도입하겠다.

$$\varepsilon : S_n \rightarrow D^n$$

표기법:

1) $\varepsilon(s_n)$는 하나의 $s_i \in S_n$에 대한 D의 부분 집합을 의미 한다.

2) $\varepsilon[\![S_p]\!]$는 하나의 언어성분 집합 S_p에 대한 D의 부분 집합을 의미한다.

보기:

1) ε ("*is_a*"−*Hierarchy* Human
(Man Woman *subtype_of* Human))
={"*is_a*"−*Hierarchy* Human
(Man Woman *subtype_of* Human)}

2) $\varepsilon[\![$ "is_−Hierarchy $]\!]$
={"*is_−Hierarchy*" Human
(Man Woman *subtype_of* Human)},
"*is_a*"−*Hierachy* Living Things
(Plant Animal *subtype_of*
Living Things),,}

위에서 BNF으로 상술된 두 계층의 의미론은 이 확장 사상을 기초로 하여 재귀적(recursive)으로 다음과 같이 상술되어 질 수 있다.

1a) ε ("is_a−Tree)
$=\varepsilon(x_1, \cdots x_n$ *subtype−of*
Root−Concept)
$=\varepsilon(x_1, \cdots, \varepsilon x_n$ *subtype−of*
Root−Concept)
if x_i End−Concept,
then $\varepsilon(x_i)=\varepsilon$ (Concept)→
$\{\varepsilon(x_i)\}\subseteq\{\varepsilon$ (Root−Concept)$\}$
if x_i "is−a"−Tree,
then $\varepsilon(x_i)=\varepsilon(x_{i1}, \cdots x_{in}$
subtype_of Between-Concept$_i$)
$=\varepsilon(x_{i1}), \cdots, \varepsilon(x_{in})$
subtype_of ε (Concpt$_i$)
→{ε (Concept$_i$}$\subseteq\{\varepsilon$ (Root−Concept)$\}$
∧Concpet$_i$→Root−Concept$_i$

1b) ε ("is_a"−Hierarchy)
ε ("*is_a*"−*Hierarchy h* "is_a"−Tree)
$=\{(h, b) \mid \in H \wedge b=\varepsilon$ ("is_a"−Tree)$\}$
(H는 계층의 실제 이름들의 집합)

또한 관계 *subtype_of*는 이항적(transitive)이다. 다시말해

if $S_p \models\{\varepsilon(\text{Concept}_1)\}\in\{\varepsilon(\text{Concept}_2)\}$ and
$S_p \models\{\varepsilon(\text{Concept}_2)\}\subseteq\{\varepsilon(\text{Concept}_3)\}$
then $S_p \models\{\varepsilon(\text{Concept}_1)\}\subseteq\{\varepsilon(\text{Concept}_3)\}$

보기 : 앞의 3장에서 언급한 "요판 인쇄기 오류 조기 발견 프로토타입 시스템"에서는 인쇄 고장의 증세를 세가지 유형으로 분류하고 있는데, 그것은 "인간", "인쇄라인", "제어시스템"이다. 이들 개념들은 다음의 집합관계로 나타낼 수 있으며:
{Human, Printing Line, Control System}$\subseteq${Symptom}
또한, "is_a"−Hierarchy "증세"는 다음과 같다.
"*is_a*"−*Hierarchy* Symptom
(Human, Printing Line, Control System *subtype_of* Sympton)

2a) ε ("consist_of"−Tree)
$=\varepsilon(x_1, \cdots x_n$ *part−of* Root−Concept)
$=\varepsilon(x_1, \cdots \varepsilon x_n$ *part−of* ε (Concept)
if x_i End−Concept,
then $\varepsilon(x_i)=\varepsilon$ (Concept)
→{$\varepsilon(x_i)$}$\in\{\varepsilon$ (Root−Concept)$\}$
if x_i "Consist−of"−Tree
then $\varepsilon(x_i)=\varepsilon(x_{i1}\cdots, \varepsilon(x_{in})$
part_of ε (*Between−Concept$_i$*)
$=\varepsilon(x_{i1}), \cdots\varepsilon(x_n)$
part_of ε (Concep$_i$)

$\rightarrow\{\varepsilon(\text{Concept}_i)\}\in\{\varepsilon(\text{Root}-\text{Concept})\}$

$\wedge\text{Concept}_i\rightarrow\text{Root}-\text{Concept}_i$

2b) ε ("consist_of"−Hierarchy)

$=\varepsilon$ (*"consist_of"−Hierarch* h "consis_of"−Tree)

$=\{(h, b)|h\in H\wedge b=\varepsilon(\text{"consist\_of"}-\text{Tree})\}$

(H는 계층의 실재 이름들의 집합)

보기:앞의 3장에서 언급한 "요판 인쇄기 오류 조기발견 프로토타입 시스템"에서의 인쇄 프로세스는 2개의 부프로세스 즉 "종이 운반"과 "종이 프로세스"로 이루어졌으며, 또한 부 프로세스 "종이 프로세스"는 다시 "인쇄"와 "건조"로 이루어졌다. 이 개념들은 집합원소관계로 다음과 같이 나타낼수 있으며:

{Paper Transport, Paper Process}
∈{Printing Process}
{Printing, Dry}∈{Paper Process}
또한, "consist_of"−Hierarchy "인쇄프로세스"는 다음과 같다.
"consist_of"−Hierarchy
Printing Process
((Printing, Dry *part_of* Paper Process)
(Paper Transport, Paper Process
part_of Printing Process))

3) $\varepsilon[\![\text{Hierarchy}]\!]=\varepsilon[\![\text{"is\_a"}-\text{Hierarchy}]\!]$
$\cup\varepsilon[\![\text{"consis\_of"}-\text{Hierarchy}]\!]$

4.2 추론 층

4.2.1 지식 원(knowledge source)

본 절에서는 지식 모델 층 구조가운데 가장 중요한 부분인 추론 층 구조의 지식 원 정형화를 설명하겠다. 앞절 2.2에서 언급하였듯이 추론 층 구조의 지식 원은 원시 함수로 간주할 수 있는데, 이것은 추론 층과 영역 층 구조를 연결하는 연결 고리(link), 언어를 정의하는 기호(signature) 그리고 함수의 특성을 규정짓는 공리(axiom)들로 이루어진 이론(theory)으로 해석 할 수 있다.

Theory=(Link, Signature, Axiom)

그래서 이 이론을 바탕으로 정형화된 지식 원은 아래와 같이 보인다 A*는, 표현이 공집합이거나 A^+임을 의미함).

⟨knowledge−Source⟩ ::=
Theory ⟨Knowledge−SourceName⟩
Link ⟨input−role⟩⁺⟨output−role⟩
Signature
Sort ⟨Meta−Class⟩⁺
Operation ⟨Knowledge−SourceName⟩:
⟨Input−MetaClassName⟩
{,⟨Input−MetaClassName⟩}*
→⟨Output−MetaClassName⟩
Variable ⟨Var−Symbol⟩:
⟨Meta−Class⟩{⟨,Var−Symbol⟩:
⟨Meta−Class⟩}⁻
Axiom
⟨Causal−Axiom⟩
|⟨Functional−Axiom⟩
|⟨Associative−Axiom⟩

추론층 구조와 영역 층 구조사이의 연결 고리는 두 종류가 있는데, 하나는 "input−role"이고 다른 하나는 "output−role"이다. 연결고리 "input−role"과 "output−role"은 영역객체(domain object), 지식 원 그리고 메타 클라스 사이의 세자리 관계이다.

정의 4.1 (연결고리 "input-role") 연결고리 "input-role"은 영역 객체 집합, 하나의 지식 원 그리고 하나의 메타 클라스 사이의 세자리 관계인데, 이 영역 객체집합은 지식원의 추론 과정에서 입력 역할을 한다:

$\varepsilon[\![\text{Input-Role}]\!] \subseteq \varepsilon[\![\text{Domain-Object}]\!] \times \varepsilon[\![\text{Knowledge-Source}]\!] \times \varepsilon[\![\text{Meta-Class}]\!]$

정의 4.2(연결고리 "output-role") 연결고리 "output-role"은 영역 객체 집합, 하나의 지식 원 그리고 하나의 메타 클라스 사이의 세자리 관계인데, 이 영역 객체 집합은 지식 원의 추론 과정에서 출력 역할을 한다:

$\varepsilon[\![\text{Input-Role}]\!] \subseteq \varepsilon[\![\text{Domain-Object}]\!] \times \varepsilon[\![\text{Knowledge-Source}]\!] \times \varepsilon[\![\text{Meta-Class}]\!]$

여기서 지식 원, 메타 클라스 및 영역 객체의 의미는 다음과 같다:

1) $\varepsilon[\![\text{Knowledge-Source}]\!] \subseteq \{x \mid x$는 가능한 지식 원의 실례(instance)$\}$

2) $\varepsilon[\![\text{Meta-Class}]\!] = \varepsilon[\![\text{input-MetaClassName}]\!] \cup \varepsilon[\![\text{Output-MetaClassName}]\!]$

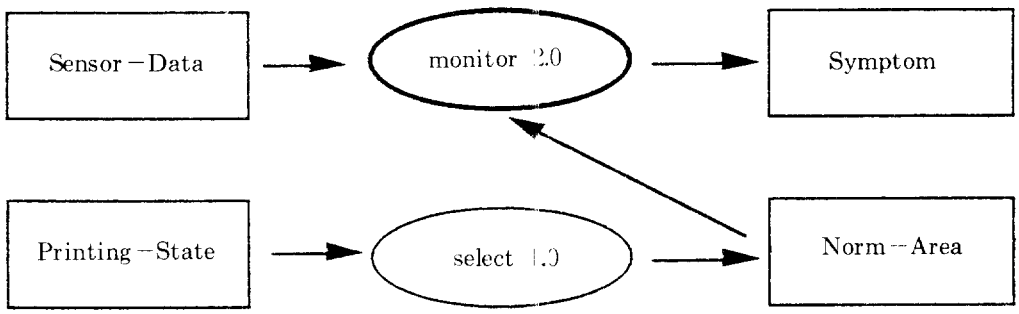
2a) 지식 원 y의 ε(Input-MetaClassName) $= \{z \mid (x, y, z) \in \varepsilon[\![\text{Input-Role}]\!] \rightarrow \varepsilon[\![\text{Knowledge-Source}]\!)\}$

2b) 지식 원 y의 $\varepsilon[\![\text{Output-MetaClassName}]\!) = \{z \mid (x, y, z) \in \varepsilon[\![\text{Output-Role}]\!] \rightarrow y = \varepsilon[\![\text{Knowledge-Source}]\!]\}$

3) $\varepsilon[\![\text{Domain-Objcet}]\!] = \varepsilon[\![\text{Concept}]\!] \cup \varepsilon[\![\text{Attribute}]\!]$

기호(signature)는 이론의 언어를 구성하는데, 쏘트(sort)와 오퍼레이션으로 이루어졌으며, 쏘트는 케리어 집합(carrier set)을 위한 표기이다. 예를들어서 Peano는 자연수 계산구조를 0자리와 1자리 오퍼레이션 즉 successor로 나타내고 있는데 여기서는 자연수가 이 계산구조의 케리어 집합이다. 본 논문에서는 메타 클라스가 이 케리어 집합이 되는데, 그 이유는 이 메타 클라스가 추론 층에서 영역 층의 영역 객체 역할을 하기 때문이다. 변수(variable)는 공리들의 상술을 위해 도입하였다. 이 변수의 도입은 영역객체의 자리 역할을 위해 필수적이었는데, 그 이유는 공리가 실제에 있어서 영역 객체들의 동적 관계를 표시하기 때문이다. 지식 원의 구체적인 의미는 공리들을 통하여 주어진다. 공리는 인과 공리(causal axiom) 함수 공리(function axiom) 그리고 연관 공리(associative axiom)로 이루어졌는데, 인과 공리는 개념속성 사이의 인과 관계를 나타내며, 함수 공리는 개념 속성사이의 함수관계를 나타내고, 연관 공리를 통하여서는 개념 속성 사이의 고정된 관계를 나타낸다.

[그림 3]은 위에서 언급한 "요판 인쇄기 오류 조기발견 프로토타입 시스템"의 최상층을 간략하여 KADS 추론층에 모형화 한 것이다. 먼저 파라메터들의 특정한 기준영역값이 인쇄공정 주변상황에 따라 선택되어진다(select 1.0). 다음에는 실제 파라메터 센서값과 그들 기준 영역이 비교되는데, 만약 센서 값이 기준 영역을 벗어 나며는 이 차이값을 근거로 인쇄공정의 이상 증세가 추론 되어진다 (monitor 2.0). 여기서 monitor 2.0은 다시 추론 층 구조가 되는데, 지식 원과 구별하기 위하여 태두리를 이탤릭체로 하였다.



[그림 3] 간략화된 인쇄공정 진단의 최상층

[그림 3]의 지식 원 "select 1.0"을 앞에 설명한 정형화식에 의해 옮겨 쓰면 다음과 같다.

Theory "select 1.0"
Link
input－role(Prerequisite : Printing－State, "select 1.0", Printing－State)
output－role (Printing－Process : *End－Concept* : Norm－Area, "select 1.0," NormArea)
Signature
Sort Printing－State, Norm－Area
Operation "select 1.0" : Printing－State →Norm－Area
Variable p : Printing－State, n : Norm－Area
Axiom
p[1]=paper-in, p[2]=paper, p[3]=b/w, p[4]=ds, [p5]=0km/h →n :=("norm－area 1"(min max));
p[1]=normal－run, p[2]=paper, p[3]=color, p[4]=ds, p[5]=35km/h →n :=("norm－area 2"(min max));
p[1]=roller－change, p[2]=paper, p[3]=color, p[4]=ds, p[5]=20km/h →n :=("norm－area 3"(min max));

4.2.2 추론 층 구조

앞절 2.2에서 언급하였듯이 추론 층에서는 잠재적인 추론을 규정하고 영역층의 추상적 개념을 수행하고 있다. 추론 층 구조는 지식 원 그리고 입력 및 출력 메타 클라스들의 집합인데, 이 구조는 지식 원 즉 원시 함수들의 연결로서 상술 되어진다. 그래서 추론 층의 구조적 성분 구성 표기를 위한 추론 층 구조의 정형화는 다음과 같다(밑줄 그어진 비 단말 표시는 마크로(macro)표시 임)

〈Inference－Structure〉 ::=
Theory 〈Inference－StructureName〉
Signature
Sort 〈Meta－Class〉
Operation (〈Operation〉 {,〈Operation〉}*:
〈Input－MetaClassName〉
{,〈Input－MetaClassName〉}*
→〈Output－MetaClassName〉
Axiom
{〈Operation〉(〈Meta－Caiss〉{,〈Meta－Class〉}*)
→〈Meta－Class〉:}*
〈Operation〉 ::=〈knowledge－SourcesName〉
|〈Inference－StructureName〉

ε [[Inference-Structure]] ⊆{x|x는 가능한 추론 층 구조의 실례(instane)}

모든 추론 층 구조는 고유 이름을 갖고 있으며, 오퍼레이션은 지식 원 또는 추론 층 구조 자신이다. 지식 원과 추론 층 구조의 구별을 위하여 추론 층 구조 이름에는 밑줄이 그어졌는데, 이 추론 층 구조는 결국에 지식 원으로만 구성되어진다. 다음에서는 [그림 3]의 "요판 인쇄기 오류 조기발견 프로토타입시스템"의 간략된 최상층을 정형화하였다.

Theory "Simplified Top-Level of Printing Process Diagnosis"

Signature

Sort Printing-State, Norm-Area, Sensor-Data, Symptom

Operation (select 1.0, "monitor 2.0") : Printing-State, Sensor-Data → Symptom

Axiom

"select 1.0"(Printing-State)→Norm-Area ; "monitor 2.0"(Norm-Area, Sensor-Data)→ Symptom

Theory "monitor 2.0"

Signature

Sort Norm-Area, Sensor-Data, History, Tendency, Difference, Data-State, Symptom

Operation ("store 2.1", "compare 2.2", "compute 2.3", "match 2.4", "abstract 2.5"): Norm-Area, Sensor-Data → Symptom

Axiom

"store 2.1"(Sensor-Data) → History ;

"compare 2.2"(Norm-Area, Sensor-Data) → Difference ;

"compute 2.3"(History) → Tendency ;

"match 2.4"(Difference, Tendency) → Data-State ;

"abstract 2.5"(Data-State)→Symptom ;

4.3 타스크 층

전문지식 모형화의 최종목적은 궁극적으로, 가능한 지식의 손실없이 전문가 시스템을 구현하는데 있다. 그러나 인간의 인식론에 근거를 둔 지식 모델에서 인공적인 시스템에 근거를 둔 디자인 모델로의 변이(transition)는 두 모델의 서로 다른 성향때문에 간단하지 않다. 특히 KADS 지식 모델의 아래 두 층, 즉 추론 층과 영역 층은 구현되어질 시스템의 고려없이 구축 되어지기 때문에 디자인 모델로의 다리역할로는 적합하지 않다. 그래서 한편으로는 문제 해결 프로쎄스를 고려하고 또 한편으로는 시스템 구현을 감안하여 설계되어진 층의 도입이 필요 불가결하다. 이 층이 바로 타스크 층이다. 이 타스크 층에서는 구현될 시스템의 시각에서 과제가 분할되고 또한 추론 프로쎄스의 제어가 수행되어 진다. 이 층은 일반 소프트웨어 프로그램에 비교하면 주 프로그램(main program) 역할을 한다. 그래서 타스크 층은 다음 두가지 요구사항을 충족시켜야 한다.

1) 타스크 층은 구현될 시스템으로의 변이 층으로서 시스템의 구조적 특성을 고려하여야 한다. 예를 들어서 제어 구조와 블록 구조.
2) 타스크 층은 또한 추론 층과의 연결을 보증하고 그 위에 절차형 지식(procedual knowledge)을 표현하기 위하여 추론 층 구조와 동일한 추상수준의 구조를 제공하여야 한다.

KADS 타스크 층은 무엇보다도 위에서 언급한 두가지 요구사항을 충족시키기 위하여 [29]

도입 되었음에도 불구하고 이에 부응하지 못하였다. 그래서 본 논문은 이 두가지 요구사항을 충족시켜주는 새로운 타스크 층 구조를 제안한다.

타스크 층 구조는 4. 2절의 추론 층 지식 원에서 처럼 타스크 층과 추론 층 구조를 연결하는 연결 고리, 언어를 정의하는 기호 그리고 오퍼레이션의 특성을 규정짓는 공리들로 이루어진 이론으로 해석할 수 있으며

Theory = (Link, Signature, Axiom),

이 타스크 층 구조를 이 이론으로 정형화하면 다음과 같다.

〈Task-Structure〉 ::=
Theory 〈Task-StructureName〉
Link 〈execute−st〉* 〈execute−pt〉+
Signature
Operation
〈Operation〉 {,〈Operation〉 }*
Axiom 〈Task-Steering〉

〈Operation〉 ::=
〈Primitive-TaskName〉
|〈Task-StructureName〉
〈Task-Steering〉 ::=
(〈Task〉*[〈Control-Relation〉]〈Task〉*)
〈Task〉 ::=〈Primitive-TaskName〉
|〈Task-StructureName〉
|(〈Task〉 {|| 〈Task〉}+)
〈Control-Relation〉 ::=〈IF-Condition〉
|〈WHILE-Loop〉
|〈Control-Relation〉
{||〈Control-Relation〉}+

과제(task)의 세분화는 최하급 과제가 추론 층 지식 원의 단계에 이를때까지 계속되는데, 타스크 층 구조의 최하급 과제를 원시 과제(primitive task)라 부른다. 이 원시 과제는 추론 층 구조의 지식 원에 해당한다.

ε (Primitive-Task) = ε (Knowledge-Source)

이러한 과제 세분화는 추론 층 구조와의 정확한 연결을 위한 필수 조건인데, 타스크 층 구조와 추론 층 구조와의 사이에는 두 종류의 연결고리가 있다. 하나는 "execute−st"이고 다른 하나는 "execute_pt"이다.

정의 4. 3(연결고리 "excute_st")연결고리 "execute−st"는 하나의 타스크 층 구조와 하나의 추론 층 구조사이의 두자리 관계이다 :

ε [[Execute-St]] ⊆ε [[Task-Structure]]
×ε [[Inference-Structure]]

정의 4. 4(연결고리 "execute_pt") 연결고리 "execute−pt"는 하나의 원시 과제와 추론 층 지식 원사이의 두자리 관계이다 :

ε [[Execute-Pt]] ⊆ε [[Primitive-Task]] ×
ε [[Knowledge-Source]]

타스크 층 구조를 4.1절에서 도입한 확장 사상으로 상술하면 다음과 같다.

1) ε [[Task]] = ε [[Task-Structure]] ∪
ε [[Primitive-Task]]
2) ε [[Task-Structure]]
⊆{x|x는 가능한 타스크 층 구조의 실례(instance)}
3) ε [[Primitive-Task]]
=ε [[Knowledge-Source]]
4) ε [[Task-Steering]]
=ε [[$(t_1 \cdots t_n)$]] ∪ ε [[$t_1 || \cdots || t_n$)]]

5) $\varepsilon[\![(t_1 \cdots t_n)]\!]$

$= \{(t_1, \cdots, t_n) | t_1, \cdots, t_n \in \varepsilon[\![\text{Task}]\!] \cup \varepsilon[\![\text{Control-Relation}]\!]$들이 순차적으로 실행되어진다}

6) $\varepsilon[\![t_1 || \cdots || t_n)]\!]$

$=\{(t_1, \cdots, t_n) | t_1, \cdots, t_n \in \varepsilon[\![\text{Task}]\!] \cup \varepsilon[\![\text{Control-Relation}]\!]$들이 병렬로 실행되어진다}

7) $\varepsilon[\![\text{Control-Relation}]\!] = \varepsilon[\![\text{IF-Condition}]\!] \cup \varepsilon[\![\text{WHILE-Loop}]\!]$

앞절 4.2.2의 간략된 인쇄 공정 진단 최상층과 그의 부 추론 층 구조 "monitor 2.0"의 타스크 층 구조는 다음과 같다.

Theory "Simplified Top-Level of Printing Process Diagnosis"

Link

execute−pt("select 1.0"^{TL}, "select 1.0"^{IL})

execute−st("monitor 2.0"^{TL}, "monitor 2.0"^{IL})

Signature

Operation

"select 1.0", "monitor 2.0"

Axiom

("select 1.0 "monitor 2.0")

Theory "monitor 2.0"

Link

execute_pt("store 2.1"^{TL}, "store 2.1"^{IL})

execute_pt("compare 2.2"^{TL}, "compare 2.2"^{IL})

execute_pt("compute 2.3"^{TL}, "compute 2.3"^{IL})

execute_pt("match 2.4"^{TL}, "match 2.4"^{IL})

execute_pt("abstract 2.5"^{TL}, "abstract 2.5"^{IL})

Signature

Operation

"store 2.1", "compare 2.2", "compute 2.3", "match 2.4", "abstract 2.5"

Axiom

while(system="on")||

if(Data-State="suspicious")

do(("store 2.1"||"compare 2.2")

then("abstract 2.5")"compute 2.3"

else(Skip)*fi*

"*match 2.4*")*od*

5. 본 논문과 (ML)^{2}의 비교 및 결론

마지막으로 본 논문과 KADS−II의 (ML)^2, "A Formal Language for KADS Models of Expertise"[1]을 비교하여 보겠다. (ML)^{2}에서는 KADS 지식 모델을 확장된 1차 술어논리의 표현으로 번역하였는데, [표 1]에서 볼수있듯이, 영역층은 "order-sorted logic"으로, 추론층은 "meta-logic"으로 그리고 타스크 층은 "dynamic logic"으로 각각 번역하였다. 그러나 본인은, (ML)^{2}에서 크게 두가지 단점을 지적할 수 있는데:

첫째, 1차 술어 논리의 "조합성 결여"(not compositive)이다. 조합성 결여라는 말은, 만약 어떤 모델이 조합적 의미론의 정의를 허락 한다면, 어떤 언어를 정의하기위해 그 언어의 기본 성분에 기본 의미를 부여하는 것으로서 충분하다. 1차 술어 논리에는 언어를 정의하는데 가장 중요한 이 조합성이 결여되어 있으나, 집합논리와 수학적 의미론은 이 조합성을 갖고있다([17], [23])

둘째, KADS지식 모델은 영역 전문가와 지식 공학자가 수집한 데이터를 [그림 1]

에서와 같이 개념적(conceptual)추상수준에서 분석 및 해석할 수 있는 도구를 제공하는데 목적이 있으므로, 정형화된 지식 모델도 본래의 추상 수준에서 모델의 완전성, 정확성 및 집적성을 보완 하는데 기여하여야 하는데, (ML)²은 [그림 1]의 세부적 디자인 모델인 M3에 가까울 정도로 너무 구체적이어서 본래의 지식 모델 추상 수준에 맞지 않는다. 예를 들어서 (ML)²은 영역층 표기를 위하여 "order-sorted logic"에 "module concept"를 사용하였는데, 본 논문에서는 단지 개념들의 분류를 위한 두 종류의 계층("is_a", "consist_of")만을 사용하였으며, (ML)²의 계층간의 엘레베이터는 따로 정의되어 엘레베이터가 어느 개체에서(from) 어느 개체로(to)로 가는지 변수를 사용하여 별도로 명시한 반면, 본 논문에서는 추론 층 및 타스크 층에 집적시켜 표기하였다.

[표 5] KADS-Ⅱ의 (ML)²과 본 논문의 비교

	지 식 모 델			
	영 역 층	추 론 층	타스크 층	층간의 연결
(ML)²	"order-sorted logic"	"meta logic"	"dynamic logic"	엘리베이터 개념
본 논문	집합이론 + 수학적 의미론	집합이론 + 수학적 의미론	집합이론 + 수학적 의미론	연결관계 도입

끝으로 본 논문의 전망과 과제를 언급하자면, 오늘날에는 아무리 좋은 이론이라도 그 이론을 사용하는데 컴퓨터의 지원이 없으면 거의 쓸모없는 이론이 되어버린다. 그래서 가능하면 본 논문의 내용을 KADS 생활 주기에 포함시켜 집적된 작업대(integrated workbench)에 구현시키고 싶은 욕심이다. 더 나아가서는 본 논문에서 제안한 정형화된 지식 모델에 연결시켜 객체지향적 방법을 근거로한 디자인 모델 및 세부 디자인 모델을 개발해서, 만약에 KADS 생활 주기를 지원하는 작업대에 최소한 하나의 객체 지향적 프로그래밍 시스템을 집적시켜 세부 디자인 모델을 오퍼레이션화 할 수 있으면 전문가 시스템을 개념화에서 부터 시스템 코딩까지 모델 지향적 방법을 토대로 하여 좀더 효과적으로 구현할 수 있을 것이다.

참 고 문 헌

[1] Akkermans, J. M., Aben, M., Balder, J. and van Harmelen, F., (ML)² : *A Formal language for KADS models of expertise, Deliverable D 1. 2. 1, Esprit-Ⅱ Project P5248*, 1991.

[2] Bennett, J. S., "ROGET : a knowledge-based system for acquiring the conceptual structure of a diagnostic system", *Journal of Automated reasoning* 1 (1985), pp. 49-74.

[3] Boose, J. H., "A research framework for knowledge acquisition techniques and tools", *Proceedings of the European Knowledge Acquisition Workshop*(EKAW 1988), GMD-Studien Nr. 143(1988), Germany.

[4] Brachman, R. J., "On the epistemological status of semantic networks", In N. V. Findler [Editor] : *Associative Network-The Representation and Use of Knowledge by Computers*, Academic Press, New York, 1979.

[5] Brachman, R. J., Levesque, H. : *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.

[6] Brachman, R. J., and Schmolze, J. G., "An Overview of the KL-ONE Knowledge Representation System", *Cognitive Science 9*(1985), pp. 171-216.

[7] Brunet, E., Wood, M. and Anjewier den, A., *An Overview of the Shelley Workbench. Internal Review Paper. Esprit Project P1098*, 1989.

[8] Bylander, T. and Chandrasekaran, B., "Generic Tasks in Knowledge-Gased Reasoning : the right level of abstraction for knowledge acquisition", *Knowledge Acquisition for Knowledge-Based Systems Vol. 1*, Academic Press, pp. 65-77, 1988.

[9] Chon, Y. C., Streng, K. and Nobis, R. : *Experiment F10(Process Monitoring and Preventive Maintenance). Deliverable E10.1. Esprit Project P1098*, 1990.

[10] Chon, Y. C., Streng, K. and Nobis, R., *Experiment F10(Process Monitoring and Preventive Maintenance). Deliverable E10.2. Esprit Project P1099*, 1990.

[11] Clancy, W. J., "Heuristic Classification", *Artificial Intellgence 27* (1985), pp. 289-350.

[12] Gaines, B. R., "An Overview of Knowledge-Acquistion and Transfer", *Knowledge Acquisition for Knowledge-Based Systems Vol. 1*, Acacemic Press, pp. 3-22, 1988.

[13] Hayes-Roth, F., Waterman, D. A. and Lenat, D. B.(eds.) : *Building Expert Systems*, Addison-Wesley, pp. 128, 1983.

[14] Hayes-Roth, F., Waterman, D. A. and Lenat, D. B.(eds.) : *Building Expert Systems* Addison-Wesley, pp. 140-149, 1983.

[15] Hesketh, P. and Barrett, T., *An introduction to the KADS methodology. Deliverable M1. Esprit Project P1098*. 1989.

[16] Karbach, W., *Methoden und Techniken des Knowdlege Engineering Arbeitspapiere der GMD Nr. 338*, Germany, pp. 54-55, 1988.

[17] von Luck, K. and Owsnicki-Klewe, B., "KL-ONE : Eine Einfuerung", *Wssensrepreesentation*, Oldenbourg, pp. 102-121, 1991.

[18] McCalla, G. and Cercone N., "Guest Editors *Introduction : Approaches to Knowledge Representation*". *Computer Vol. 16 No. 10*(1983), pp. 12-18.

[19] Newell, A., "The Knowledge Level" *Artificial Intellgence 18*(1982), pp. 87-127.

[20] Patel, J., "On the road to automatic knowledge engineering", *Proceedings of the 11th international Joint Conference on Artificial Intellgence*(1989), Detroit, USA.

[21] Schachter-Radig, M. J., Chon, Y. C., Krickhahn, R., Nobis, R. and Streng, K. H. : *Eine Methodologie zur Entwicklung der WBSe. NTE internal Paper*. Muenchen, Germany, 1990.

[22] Schreiber, G., Bredeweg, B., de Greef, P., Terpstra, P., Wielinga, B., Brumet, E., Simonin, N. and Wallyn, A. : *A KADS approach to KBS Design. Deliverable B6. Esprit Project P1098*, 1989.

[23] Stoy, J. E., *Denatational Semantics : The Scott-Strachey Approach to Programming Language Theory*, MIT Press, Cambridge, Mass., 1977.

[24] Taylor, R. M., Porter, D., Hickman, F., Streng, K. H., Tansely, S. and Dorbes, G., *System Evalution-Principles and Methods (The Life-Cycle Model). Task G9 Deliverable. Esprit Project P1098*. 1989.

[25] Ueberreiter, B. and Voss, A. : *Materialien KADS-Benutzer-Treffen*. Siemens AG, Muenchen-Perlach, Germany, 1991.

[26] Wielinga, B. J., Schreiber, A. Th. and Breuker, J. A., KADS : *A Modelling Approach to Knowledge Engineering, Synopsis of KADS-II. Esprit Project P5248*. Uni. of Amsterdam, Netherlands, 1991.

[27] Wielinga, B. J., Schreiber, G. and de Greef, P., *Synthesis report. Deliverable V3, Esprit Project P1098*, 1989.

[28] Wielinga, B. J., van Someren, M., de Hoog, R., Schreiber, G., de Greef P., Bredweg, B., Wielemaker, J. and Billeaut, J.-P., *Model-driven knowledge acquisition interpretation models Deliverable A1. Esprit Project P1098*. 1988.

[29] Wielinga, B. J. : van Someren, M. ; de Hoog, R. : Schreiber, G. ; de Greef P. : Bredweg, B. : Wielemaker, J. ; Billeaut, J.-P. : *Model-driven knowledge acquisition interpretation models, Deliverable A1. Esprit Project P1098*. pp. 42-44, 1988.