

KEE : 도메인 전문가를 위한 자동지식공학 틀 개발에 관한 연구

강병호, 김길곤
호서대학교, 에이치엔케이

KEE : Knowledge Engineering Tool for Domain Experts

Byeong Ho Kang, Kim Kil Gon

Hoseo University, HNK

요약

전문가시스템은 지식베이스를 이용하여 지식을 추론하는 추론엔진과 해당분야의 전문지식이 쌓여 있는 지식베이스로 구성되어진다. 전문가시스템연구에 있어서 지식베이스에 저장되는 지식획득과정의 규명이 가장 핵심적인 연구분야라고 할 수 있다. 전문지식은 일반적으로 지식공학자들이 전문가로부터 전문지식을 획득하여 구현하는 것이 가장 일반적인 방법으로서 많은 전문가시스템 개발방법 이론들은 지식획득과정에서 지식공학자들의 역할을 필수적인 것으로 이해하고 있다. 그러나 지식획득에 관한 광범위한 해결책은 여전히 제시되고 있지 못하며, 이러한 문제점은 전문가시스템의 상용화에 가장 큰 어려움으로 지적되고 있다. 최근에는 이러한 지식획득병목현상을 해결하기 위하여 상황인식론과 같은 지식에 대한 새로운 해석을 기반으로한 지식획득 방법 이론들이 소개되고 있다.

Multiple Classification Ripple Down Rules 이론은 1995년 소개된 이론으로서 지식의 획득과정을 지식의 유지보수라는 소프트웨어 공학적 개념에서 접근하고 있다. 지식의 획득과정에서 완전무결한 지식의 모델 찾기를 지양하고 지식이란 단계적 확장개념에서 진화한다고 이해한다. 즉 지식베이스의 구축 단계를 개발과 완료가 불가능하다는 관점에서 지식베이스는 끊임없이 유지보수가 필요한 대상으로 이해하고 이러한 유지보수를 가능하게 하는 방법론을 제시한다. MCRDR에서 가장 핵심적인 부분은 지식공학자의 역할을 최소화하고 시스템 내부에서 지식의 관리와 획득을 수행하여 전문가로 하여금 직접적인 지식 입력이 가능하도록 하는 부분이다. MCRDR 이론의 경우 여러 가지 연구실험을 통하여 실용성이 입증되었고 의료분야에서 상용화 시스템 개발 틀로서 사용되어져 왔다. 그러나 MCRDR 이론이 적용된 전문가시스템 들의 경우 MCRDR 이론을 기본으로한 개발 틀로서 개발된 시스템들이 아니고 해당분야에서 MCRDR 이론을 적용한 엔진을 직접 설계 구현하여 온 것이 사실이다. . KEE(Knowledge Engineer for Experts) 시스템은 최근 개발된 MCRDR 기반 전문가시스템 개발 틀로서 본 논문에서는 이러한 분야별 전문가시스템 개발을 지양하고 MCRDR 이론을 기반으로 한 범용성 있는 전문가시스템 개발 틀의 개발에 관한 연구를 소개한다.

Keyword : expert system, knowledge acquisition, knowledge engineer, 전문가시스템, 지식획득, 지식공학

1. 서론

전문가시스템은 컴퓨터의 발달과 함께 많은 분야에서 연구되거나 상업적 개발이 활성화된 인공지능 분야로서 정보화 사회에서 이러한 전문가시스템을 이용한 고급 정보에 대한 수요는 기하급수적으로 증대될 것으로 예측되어진다. 그러나 이러한 수요에 대응하기 위한 전문가시스템 개발의 기술적 발전은 미흡한 수준에 머물러 있는 것이 현실이다. 추론엔진과 지식베이스로 구성되어지는 전문가시스템에서 지식은 지식베이스에 저장되어 있으면서 추론엔진에 의하여 사용되어진다. 전문가시스템에 사용되어지는 지식을 획득하여 시스템에 구현하는 과정을 지식공학이라는 표현으로 사용되어지며 이러한 역할을 수행하는 사람을 지식공학자(Knowledge Engineer)라고 한다. 전문가시스템 개발과정에서 지식공학자와 전문가 및 컴퓨터시스템등의 이질성으로 인하여 효과적인 지식획득이 어려운 현상을 지식획득병목현상 (Knowledge Acquisition Bottleneck)이라고 한다. 이러한 지식획득병목현상을 해결하기 위하여 여러 가지 전문가시스템 개발 이론들이 제시되었으며, 대부분 전문가시스템 개발이론들은 지식공학자의 역할을 필수적인 요소로 가정하고 있다.

Ripple Down Rules 이론은 의료전문가시스템인 Garvan-ES1의 개발과정에서 기존 전문가시스템 개발 이론들이 내포하고 있는 문제점들을 해결하기 위하여 개발되었다. RDR 이론은 전문가의 지식을 종합적으로 분석하여 획득하는 것이 가능하다는 전제하에 개발된 기존 이론들을 비판하고 지식에 대한 새로운 이해에서 출발한다. RDR 이론에서는 지식이란 가변적인 것으로 전문가가 제공하는 지식은 상황에 따라 항상 변화한다는 상황인식론[2,10]에서 출발한다. 이러한 관점에서 본다면 지식공학자가 전문가로부터 획득하는 지식도 획득시점의 여러 가지 상황등에 따라 가변적일 수 밖에 없으며 특히 전문지식이 부족한 지식공학자의 입장에서 이해된 이러한 불완전한 지식을 시스템에 구현하는 것은 불가능할 수도 있다. RDR 이론은 지식공학자를 지식획득과정에서 배제하고 전문가가 직접 시스템에 지식을 학습시키도록 하고 있다. 시스템에 의한 지식획득과정에서 전문가는 사례별로 시스템에 자신의 판단 이유들을 시스템에 설명하게 되고 이러한 설명을 바탕으로 시스템은 학습을 수행하게 된다. 이 과정에서 전문가는 지식공학

혹은 컴퓨터에 관한 지식이 없이도 성공적으로 시스템에 지식을 입력시킬 수 있으며 지식을 관리 유지 할 수 있다. RDR은 지식획득과정을 시스템 개발 초기로 국한시키지 않고 운영과정에서 불완전한 지식을 발견할 때마다 지식학습과정을 수행하도록 하고 있다. 여기서 선결과제는 지식학습을 위한 시간이 지식베이스의 복잡성이나 규모에 관계없이 일정하여야 하며 지식학습 과정이 단순화되어야 한다는 점이다. RDR 이론은 호주 시드니의 세인트 빈센트병원의 임상병리검사실에 병리검사소견서 자동 생성 시스템(Pathology Expert Interpreting System) [11]에 최초로 적용되어 5년 동안 상용시스템으로서 사용되어져 그 가능성을 증명하였다. 그러나 RDR 이론은 사례당 1개의 결론만을 생성하는 분야에 적용이 가능했었으며 이것은 RDR 이론의 확산에 걸림돌로 작용되었다. 1995년에 RDR의 이와 같은 문제점을 해결한 범용적인 Multiple Classification Ripple Down Rules 라는 이론이 소개되었다. (MC)RDR 이론은 여러 가지 연구 결과 및 상용시스템의 운영을 통하여 가능성을 보여주었으나 MCRDR 관련 연구들이 본격적인 상용시스템으로의 확산에 크게 기여하지는 않았다.

본 논문은 (MC)RDR 이론을 적용한 상용전문가시스템개발틀인 (Knowledge Engineer for Experts(KEE)의 개발에 관한 내용이다. (MC)RDR이 이론으로서 정립되어 있으나 이것을 범용적인 상용 전문가시스템 개발틀로의 개발에는 고려되어야 하는 중요 요소들에 대한 논의와 실제 적용 결과등을 본 논문에서 기술되어 있다. 이론의 기본 개념 이해를 위하여 RDR 이론이 간단히 소개되어 있으며 KEE 시스템의 여러 가지 구성 요소들이 설명될 것이다.

2. Ripple Down Rules

RDR 이론은 전문가시스템에 있어서 핵심적인 knowledge base의 구축과정에서 기존의 이론들과는 달리 전문가의 지식을 추출하거나 분석하는 것은 불가능하다는 가정하에서 개발되었다. 이와 같은 가정의 근거는 초기의 의료전문가시스템인 Garvan-ES1의 수년간에 걸친 구축과 운영과정에서 관찰된 결과에 따른 것이다[3]. 시스템의 운영과정에서 관찰한 바에 따르면, 주어진 문제의 결론에 대한 전문가(의사)들의 설명은 어떻게 그것들이 유도되었는가를 설명하기보다는 왜 그들의 결론이 적합한지를 정당화하는데 중점을

맞추고 있었다. 또한 동일한 문제에 대해서도 주어진 context (상황이나 문맥)에 따라 유동적으로 전문가의 설명이 변하는 것이 관찰되었다. 이것은 전문가가 제공하는 지식의 설명은 정형화된 상태로 고정되어 있는 것이 아니라 상황에 따라 변하는 것을 보여주며 이것은 인지학에서 설명하는 Situated Cognition(or Situated Action)[1,2]과 일맥상통하는 면이 있다. 이러한 가정에 근거한다면, 전문가시스템의 구축방법은 이러한 유동적인 전문가의 변화에 대응할 수 있어야 함을 의미한다. RDR 시스템에서는 일련의 규칙들의 배열을 이용하여, 같은 문제에 대해서는 시스템이 전문가가 사용하고 입력한 같은 배열의 규칙들을 적용할 수 있도록 하여 시스템이 항상 전문가와 동일한 context 상에서 문제를 해결하게 된다.

RDR 이론에서의 지식베이스는 'If conditions Then conclusion'의 형식을 가진 일련의 규칙(rule) 들로서 구성되어져 있고, 이것은 전문가시스템이론에서 가장 일반화된 지식의 표현방식이다. 특히 RDR 이론에서는 incremental learning에 의하여 지식을 축적하기 때문에, 지식습득의 기본적인 기능인 지식의 첨가와 교정 그리고 삭제는 새로운 규칙의 첨가로 달성하게 된다. 일반적으로 새로운 규칙을 knowledge base에 첨가하는 경우에는 rule의 위치와 조건(condition)의 선택이 가장 중요한 요소들이다.

그림 1은 간단한 RDR 구조와 시스템의 inference 과정을 보여주고 있다. 시스템은 이진 tree 구조의 처음부터 'Test Case'를 rule의 조건들과 비교해 간다. 그래서 rule이 만족되어지는 경우에는 true 방향의 node들을 계속 검증하게 된다. 그림 1에서 회색으로 채워진 Rule들이 Case들을 만족한(true) Rule들이며, 넓은 화살표로 표시된 경로가 'Test Case'를 입력으로 하여 실행하였을 때, 지식베이스에서의 Rule들의 진행 경로이다. 이 경우에 Rule 7이 진행경로에서 마지막으로 true가 되었으므로 Concl 5이 'Test Case'에 대한 결론으로서 사용자에게 제시되어지게 된다.

지식습득 (Knowledge Acquisition)에 대하여 대하여 설명하자. 지식습득은 시스템이 제공한 문제 (Test Case)의 결론에 대해서 전문가가 동의할 수 없는 경우에 발생하게 되고, RDR 시스템에서는 이것을 새로운 rule을 knowledge base에 첨가하

는 것으로 해결한다. RDR에서는 새로운 rule의 위치와 규칙조건의 선택을 위하여 위에서 언급되어진 rule들의 배열과 knowledge acquisition 과정에 사용되어졌었던 case들을 이용한다. 'Test Case'에 대한 정답이 Concl 8이라고 가정하면 현재의 지식베이스에 근거한 결론 Concl 5는 잘못된 결론이다. 이 경우에 RDR 시스템은 전문가에게 현재의 결론이 도출된 Rule 7이 시스템에 추가되어질 때 사용되어진 Case 7과 Test Case간의 차이점(표 1)들중에서 Test Case의 결론이 Concl 8인 이유(rule conditions)를 선택하게 한다. 이 때 전문가가 g를 선택한다면, RDR 시스템은 '10 : g -> Concl 8'을 rule로서 Rule 7의 아래쪽 위치 A에 첨가하게 된다. 또한 Test Case는 Rule 10번의 첨가시 사용되어진 Case로서 그림 1의 아래쪽 Case Base에 첨가되어진다.

Case 7	Test Case	Case 7 과 Test Case 차이점
c,d	c,f	
k	g	g NOT (k)

표 1. 케이스 차이표

3. Knowledge Engineer for Experts (KEE)

KEE는 전문가시스템의 기능 개념에서 추론엔진과 지식베이스를 관리하는 지식관리부분으로 나누어진다. 그리고 적용분야의 모델링을 위한 데이터(Case)관리와 결론관리 부분으로 나누어진다. 데이터관리부분은 항목관리와 그룹관리가 있으며 각각의 케이스(객체)별 관리가 있다. 복잡한 데이터 모델링기능을 위한 여러 가지 기능들이 제공되어져 있다.

기초 Data 관리

기초 Data 관리는 전문가시스템지식을 관리하는데 필요한 기본 항목들을 정의한다.

그림 2는 기초 데이터를 관리하는 화면이다. 항목들은 실제 값들을 저장할 수 있는 변수형태의 항목들과 항목들의 그룹으로 정의되어진다. 항목은 숫자형과 문자형 그리고 유한개의 문자열로 구성되어진 리스트형으로 선언되어진다. 그룹이 선언되어진후 항목들이 추가되거나 혹은

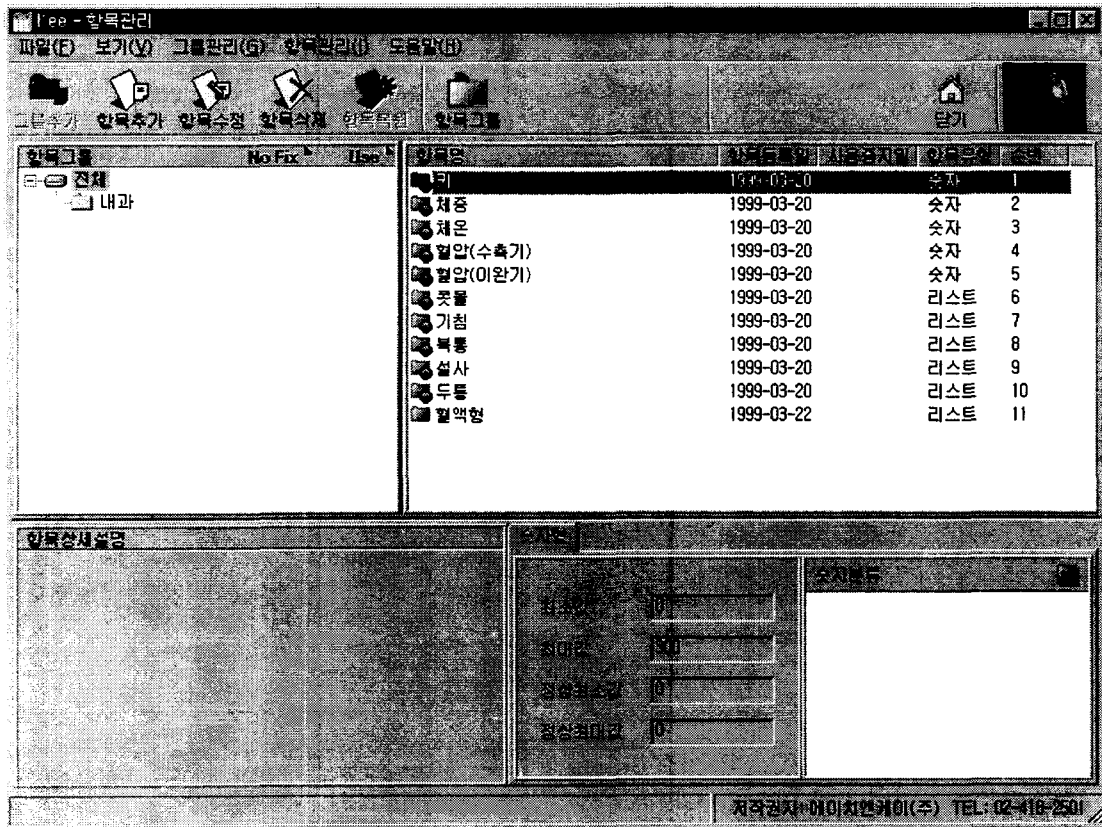


그림 2. 기초 데이터 관리 인터페이스

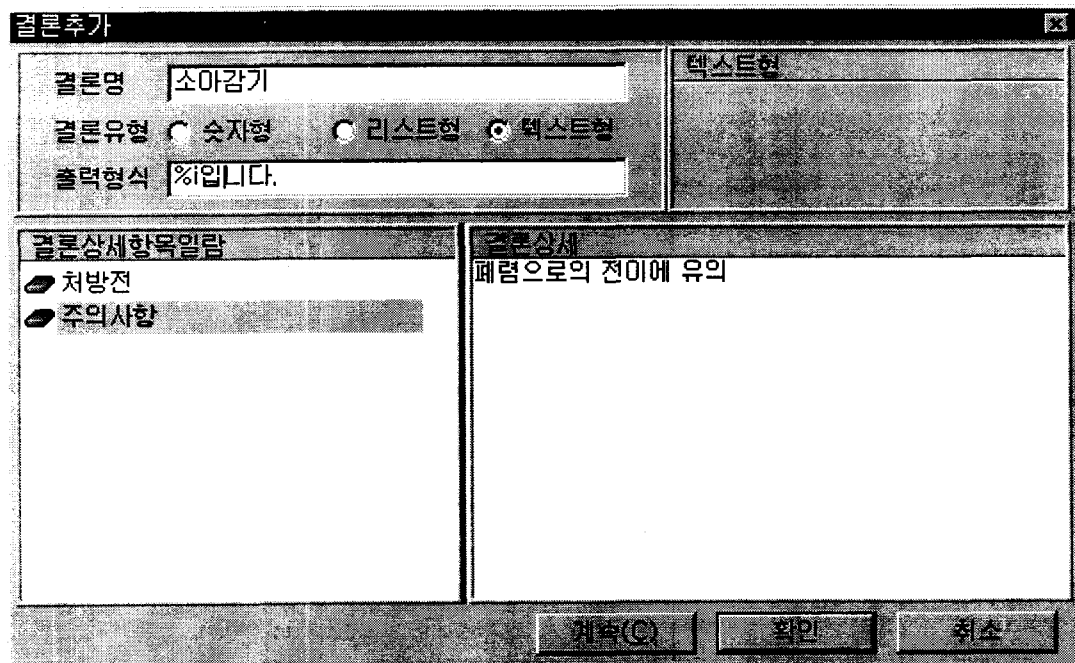


그림 3. 결론 추가 관리 인터페이스

추가된 항목들로 그룹을 구성할 수 있도록 설계되어있다. 여기서 항목의 삭제와 추가 혹은 수정이 가능하도록 되어있다. 삭제된 항목은 복원이 가능하지만 시스템의 지식베이스에 사용되어진 항목명은 수정이 불가능하고 삭제 역시 불가능하다. 숫자형의 경우 최소값과 최대값의 지정이 가능하고, 정상수치의 범위가 지정될 수 있다. 항목과 관련된 것으로 함수관리모듈이 따로 존재한다. 함수관리모듈의 경우 단순 입력된 값들로부터 전문가가 직접 함수들을 선언하고 선언된 함수들은 지식베이스의 룰에 사용되어진다. 또한 가지 기능으로는 차수관리를 들 수 있다. 대개의 경우 동일 항목이 하나의 값을 가진 경우외에 복수의 값을 가지는 경우가 있다. 예를 들면 임상병리 검사의 경우 여러 차례 반복되는 검사값들을 관리해야만 하고 이러한 여러 차례의 검사들은 그 나름대로 다른 형태의 함수를 가지고 있다. 예를 들면 특정항목값의 증가 혹은 감소, 최소 최대등이 여러 개의 차수로부터 계산되어질 수 있다.

결론 관리

전문가시스템이 사용자에게 제공할 진단 혹은 정보를 관리하는 항목이다. 구성방법이나 구조는 기초 Data와 비슷한 형태를 유지한다. 즉 결론들은 그룹으로 분류하여 관리가 가능하다. 각각의 결론은 진단외에도 상세처방등을 유지할 수 있도록 설계되어있다. 결론 및 그룹 역시 추가 삭제 수정 기능이 모두 제공되어지면 삭제의 경우 지식베이스에 사용되어진 이후 삭제가 불가능하도록 디자인 되어있다.

추론화면

추론화면은 실제 KEE 시스템의 가장 중요 인터페이스로서 데이터의 입력 및 추론등이 여기서 이루어진다. 추론화면에서 입력된 데이터에 대하여 추론엔진을 실행시켜 진단결과를 나타낼 수 있으며 추론화면으로부터 위에 설명된 항목 관리 및 결론 관리 인터페이스로 연결할 수 있다. 또한 진단 결과가 정확하지 않거나 수정이 필요한 경우 지식베이스 관리 인터페이스로 연결하여 지식베이스를 관리할 수 있다.

지식베이스관리

주어진 데이터에 대한 결론이 부정확한 경우 시스템은 세가지 다른 지식관리 기능을 제공한다.

첫번째는 주어진 데이터에 대한 결론이 없는 경우 결론을 추가할 수 있다. 두번째는 시스템에 의하여 제공된 결론이 부정확하여 삭제를 원하는 경우이다. 세번째는 결론이 부정확하여 수정을 원하는 경우이다. 이 세가지 지식베이스 수정요구는 추론엔진에서 요청되어지지만 지식베이스 관리는 동일모듈에서 이루어진다. 추론엔진에서 선택되어진 기능에 따라 시스템은 내부적으로 지식이 추가될 위치와 새로 입력될 지식의 검증에 위한 준비를 마치고 전문가로부터 지식을 수정하기 위하여 필요한 정보를 입력된 데이터를 기준으로 받아들이게 된다. 전문가시스템은 현재 입력데이터 항목들 중에서 지식베이스를 수정하기 위하여 필요한 것들을 선택하거나 제공되어진 함수 기능등을 이용하여 결론에 대한 조건들을 입력하도록 전문가에게 요청한다. 전문가가 제공한 조건들로 규칙을 생성했을 경우 기존의 지식베이스의 규칙들과의 모순이 여부를 점검하여 모순이 발생한 경우 전문가로 하여금 확인을 하도록 다시 요청한다. 이러한 작업을 반복하여 모순이 모두 해소된 경우 시스템은 전문가의 승인을 받아 규칙을 지식베이스에 입력하고 연관된 정보들을 수정한다. 이것은 세가지 지식수정 유형에 동일하게 적용되며 지식수정을 마친 지식베이스 관리 모듈은 다시 추론엔진모듈로 전환된다.

4. 결론

KEE 시스템은 1년여에 걸친 개발과정을 거쳐 개발되었다. 본 논문에서 소개된 시스템의 경우 가능한 범위내에서 많은 적용분야에 이용될 수 있도록 여러가지 기능들을 일반화하여 설계되었다. 최근 장애진단 Help Desk System 개발에 최초로 KEE 시스템이 적용되었으며, 품질관리 전문가시스템 및 건강검진 시스템등의 개발이 진행중이다. 이러한 여러가지 분야에의 적용과정에서 여러가지 새로운 문제점들이 발견되고 있고 이를 수용하기 위한 이론의 확장 및 개발틀의 지속적인 업그레이드가 진행되고 있다. 그러나 이와 같은 개발과정에서도 기본 (MC)RDR 이론의 추론엔진과 지식획득에 관한 알고리즘은 기본 엔진으로서 인터페이스와 독립적으로 디자인 되었다. 그러므로 KEE 시스템은 논문에서 소개된 인터페이스 뿐만이 아니라 필요한 경우 기존 시스템에 부분 모듈로 적용이 가능하다. 또한 같은 속성을 가진 여러가지 도메인에 적용이 가능하다. KEE 시스템은 지식공학자의 도움

없이 전문가가 직접 지식을 관리할 수 있는 전문가시스템 개발툴로서 전문가시스템의 대중화에 긍정적인 영향을 미칠 것으로 예상되어진다.

참고문헌

1. Clancey, W.J., *A boy scout, toto, and a bird: How Situated Cognition is Different from Situated Robotics*. 1991.
2. Clancey, W.J., *Situated Cognition: Stepping out of representational flatland*. AI communications, 1991.
3. Compton, P., et al. *Ripple down rules: possibilities and limitations*. in 6th Knowledge Acquisition for Knowledge Based Systems Workshop. 1991. Banff, Canada: SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Canada.
4. Compton, P. and R. Jansen, *A philosophical basis for knowledge acquisition*. *knowledge acquisition*, 1990. 2: p. 241-257.
5. Compton, P., et al., *Local patching produces compact knowledge bases*, in A Future for Knowledge Acquisition, L. Steels, G. Schreiber, and W.V.d. Velde, Editors. 1994, Springer-Verlag: Berlin, German. p. 104-117.
6. Guha, R.V. and D.B. Lenat, *CYC: a midterm report*. *AI magazine*, 1990. 11(3): p. 32 - 59.
7. Lenat, D., M. Prakash, and M. Shepherd, *CYC: using common sense knowledge to overcome brittleness and knowledge acquisition bottleneck*. *AI magazine*, 1986. 6(4): p. 65 - 85.
8. Morik, K., *Underlying assumptions of knowledge acquisition and machine learning*. *Knowledge Acquisition*, 1991. 3(2): p. 137-156.
9. Piaget, J., *The child's conception of the world*. 1929, London: Routledge and Kegan Paul.
10. Popper, K., *Conjectures and refutations*. 1963, London: Routledge and Kegan Paul.
11. Preston, P., G. Edwards, and P. Compton. *A 1600 rule expert systems without knowledge engineer*. in World Congress on Expert Systems. 1994. Lisbon, Portugal: Macmillan New Media License.
12. Ross, S.P., *Case-Based Reasoning*, 1989, Department of Computer Science, University College London.
13. Slade, S., *Case-Based Reasoning: a research paradigm*. *AI magazine*, 1991. 12: p. 42-55.
14. Stottler, R.H., A.L. Henke, and J.A. King. *Rapid retrieval algorithms for case-based reasoning*. in The Eleventh International Joint Conference on Artificial Intelligence. 1989. Detroit, Michigan USA: Morgan Kaufmann.
15. Sycara, K.P. and K.D. Ashley, *Case-Based Reasoning* : Tutorial Note of the Twelfth International Joint Conference on Artificial Intelligence. 1991: IJCAI-91.