

# 지식의 일관성이 보장된 지식 기반 시스템

서 의 현  
목원대학교 컴퓨터공학과  
ehsuh@mokwon.ac.kr

## Knowledge Based System with Consistency of Knowledge

Euy-Hyun Suh  
Dept. of Computer Engineering, Mokwon University

### 요 약

생성 규칙은 지식 기반 시스템의 지식표현 기법 중 가장 많이 사용되고 많은 장점을 갖고 있음에도 불구하고 지식의 일관성 유지가 어렵다는 단점을 가지고 있다. 그런데 지식의 일관성이 추론 결과의 신뢰도에 직결된다는 점에서 지식의 일관성을 유지할 수 있는 시스템이 필수적이다. 따라서 본 논문에서는 새로 첨가될 지식이 기존 지식과 일관성이 유지될 경우에만 지식을 첨가하는 일관성 유지시스템과 지식베이스는 물론 데이터 베이스 내의 선언적 지식과 절차적 지식 등 다양한 지식을 호출할 수 있는 기능을 갖춘 지식 기반 시스템을 구축하였다.

### 1. 서론

지식기반시스템은 지식베이스, 추론엔진과 사용자 인터페이스로 구성된다. 이 시스템은 사실과 지식베이스의 규칙을 이용하여 추론을 수행하고 그 결과 새로운 지식들을 생성하여 사용자에게 제공한다. OPS5, KES, EXPERT, CLIPS[3] 등은 이러한 지식기반시스템을 쉽게 구축할 수 있는 전문가 시스템 셸(shell)이다. 이러한 셸들은 추론 메카니즘을 제공하며 지식베이스를 쉽게 구축하고 유지할 수 있는 개발 환경을 제공하고 있어 지식 공학자의 노력을 많이 절감시켜준다. 그러나 이러한 셸에서 전문가 시스템의 신뢰도의 성능을 좌우하는 지식베이스 내의 지식의 검증은 극히 기본적인 것에 불과하며 지식의 일관성 및 정확성 여부는 전적으로 지식의 제공자에 의하여 결정된다.

그런데 지식의 양은 방대한 경우가 많기 때문에 지식 제공자(전문가)나 지식 공학자가 내용 전체를 파악하기가 어렵고, 전문가는 때때로 직관적으로 생각하므로 추론에서의 많은 과정을 생략한 채 지식을 지식베이스에 삽입할 수 있다. 또한 지식베이스는 점진적인 방식으로 오랜 기간에 걸쳐 구축되고 여러 명의 전문가가 지식을 함께 저장하기 때문에 지식의 중복, 순환과 모순의 문제가 야기될 수 있다. 이러한 문제들을 해결하기 위해서는 전문가 시스템이 지식베이스를 사용하기 전에 정

확성을 검증하여 발견된 오류를 자체 수정하거나 자체 수정이 불가능할 경우 전문가에게 문의하여 수정하는 시스템이 요구된다.

따라서 본 논문은 새로운 지식이 기존의 지식베이스에 첨가되기 전에 새로운 지식의 정확성 및 일관성을 검증한 후 일관성이 유지될 경우에만 지식을 지식베이스에 첨가하는 일관성 유지 시스템을 구축하였다. 이와 함께 지식베이스 및 데이터 베이스의 다양한 지식을 호출할 수 있는 지식기반시스템을 구축하여 일관성 유지 시스템과 통합하였다. 모든 시스템은 C++언어와 Oracle SQL을 사용하여 Sun Sparc 5.5에 구축되었다.

본 논문에서는 신뢰도를 증진시키기 위한 방법들을 2장에서 설명한 후 3장에서는 시스템 구성요소에 대해 소개한다. 4장에서는 일관성 유지 시스템에 대해서 자세히 설명하고 결론에서는 일관성 유지시스템을 갖춘 지식 기반시스템을 구축함으로써 확인된 결과를 요약하고 향후 과제를 다룬다.

### 2. 관련연구

신뢰성 있는 시스템 구축을 위하여 지식 검증을 위한 많은 시도가 있었다.

전통적인 방법으로 의존 관계표를 이용하여 일관성과 완결성을 점검하는 방법[8]이 있는데 이는 열거 방식으

로 처리하기 때문에 시간이 많이 걸리는 단점이 있다.

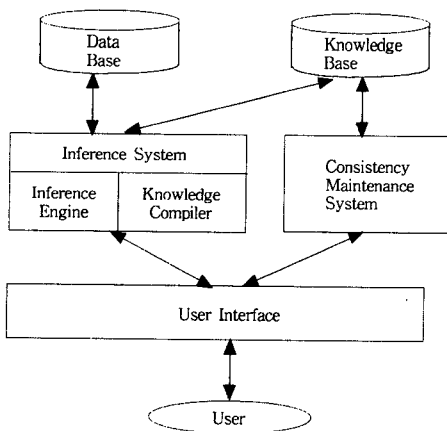
결정표(Decision Table)를 사용하는 방식에서는 간혹 오류의 존재만을 발견하고 오류의 위치를 정확하게 찾아 내지 못하는 단점이 있다. 또한 연결된 규칙이 많아지면 지수 함수적인 성능을 나타낸다[6].

쌍 검증(Pairwise Checking) 방식에서는[5] 경우에 따라서는 가치 있는 규칙이 생략되고 의미 없는 규칙이 취급되는 일이 발생할 수 있으며 규칙의 수가 많아질 경우 시간이 많이 걸리는 단점이 있다[4].

그래프 표현 방식은 프레미스의 개념적인 의존 관계를 나타내기 쉬운 표현법으로서 방향 그래프(Directed Graph), 추론 그래프(Inference Graph), 하이퍼 그래프(Hypergraph)등의 방법이 있다. 그러나 이 방법들은 복잡절과 단순절 사이의 의존 관계를 나타내기 어렵다[8]. 따라서 그래프 표현 방식은 노드를 첨가하거나 전이를 묘사하고 또는 하이퍼 아크를 사용하여 복잡절을 묘사함으로써 중복과 포함을 정확히 발견하여 수정한다. 그렇지만 모순과 불완전성은 여전히 완벽하게 처리되지 못하는 단점이 있다. 방향 하이퍼 그래프(Directed Hypergraph)[8]는 이러한 그래프의 단점을 보완하였지만 고려해야 할 하이퍼 노드의 조합의 수가 많아지면 실제로 이들을 다 묘사하지 못함으로써 오류가 생성될 수도 있다. 또한 이 방식은 구조적 오류만을 검색할 수 있을 뿐이며 규칙이 많아질 경우에는 시간이 많이 소요된다. 개념 그래프(Conceptual Graph)는 계층적 구조로서 지식을 체계적으로 분류하여 계층적으로 저장하고 편집 방법을 통해 중복을 제거한다[1,2]. 그러나 이 방식은 표현의 다양성 때문에 극히 제한적인 일관성만을 검증할 뿐 여러 곳에 있는 비 일관성 오류를 점검하기 어렵다.

### 3. 시스템 개요

시스템은 [그림 1]과 같이 구성된다.



[그림 1] 시스템 구조

#### 3.1 데이터 베이스

관계형 데이터 베이스로서 Oracle 7.3.3을 사용하여 일반적인 정보들을 표현하였다. 전문가 시스템은 필요한 경

우 데이터 베이스의 정보들을 호출할 수 있도록 하였다.

#### 3.2 지식베이스

지식베이스의 기본 구조는 생성 규칙이다. 생성규칙은 규칙들을 모듈화하여 나타낼 수 있고, 비절차적 방식으로 규칙들로서 저장된 지식을 사용할 수 있을 뿐 아니라 여러 종류의 지식을 쉽게 표현할 수 있기 때문이다. 지식은 조건 부분과 실행 부분으로 표현되며 일반 지식과 절차적 지식도 같은 방식으로 나타낼 수 있도록 하였다. 이러한 생성 규칙들의 실행 효율을 증가시키기 위하여 내부 표현 구조는 네트워크 구조를 사용하였다.

생성 규칙 표현법의 장점들 중 하나는 다양한 지식을 표현할 수 있다는 것인데 이 시스템에서는 필요한 경우 함수를 호출하여 비결정적인 값을 계산할 수 있는 절차 부가의 개념을 사용하였다. 예를 들면 다음과 같다.

```

rule1 (0.9) <modeler>
if type of model, s=triangle
    surface, x=calcul_surface_tri(base, height)
    surface, r >= 100.0
then triangle_model_no = 1
    
```

이 경우 함수는 0개 이상의 매개변수를 가진다.

추론 도중 이러한 지식을 만나면 먼저 함수 지식 인터프리터가 이 문장을 분석하여 함수명과 매개 변수의 개수 및 매개 변수를 분리한 뒤 해당 함수를 진행시킨다.

지식기반시스템에서 경우에 따라서는 사실의 양이 많아지고 데이터 베이스에 저장되어 있기 때문에 데이터 베이스 내에 저장된 사실들을 호출해야 할 경우가 있다.

따라서 본 시스템은 데이터 베이스 내의 정보의 호출, 추가, 제거 및 수정할 수 있는 기능을 갖추었다. 관계(relation), 엔티티(entity), 키(key), 레코드(record)와 호출된 값을 보관할 수 있는 변수의 이름들을 가지고 조작할 수 있다. 예를 들면 다음과 같다.

```

rule2 (0.8) <modeler>
if
    eq1, i y a(relation_name, equation, key_col:edge1_id, return_value)
    eq2, i y a(relation_name, equation, key_col:edge2_id, return_value)
    calcul_state, x = calcul_parall(eq1, eq2)
    calcul_state, s = intersection
then delete_edge, i n s(relation_name, key_col, edge2_id)
    
```

이것은 절차 부가의 일종이므로 수행과정은 절차 부가와 동일하다.

#### 3.3 추론 시스템

추론 시스템은 추론 엔진과 지식 컴파일러로 구성된다. 생성 규칙 기반 시스템에서 많이 사용하는 추론 엔진의 기법은 전방 추론(forward chaining)과 후방 추론(backward chaining)이다. 본 시스템에서는 확신도에 의해 추론의 순서가 결정되는 전방 및 후방 추론을 구축하였다. 실행시킬 추론 방식은 사용자가 선택할 수 있다.

지식 컴파일러는 지식기반시스템이 실행되기 전에 지식을 네트워크의 내부 표현 방식으로 컴파일한다. 네트워크 표현 방식은 하나의 규칙 내부에 이 규칙과 연결되어지는 규칙의 정보를 포함하고 있다. 이 방식은 추론이 실행될 때 이러한 정보를 이용함으로써 모든 지식을 검색할 필요

없이 꼭 필요한 지식만 검색해 봄으로써 추론의 속도를 증가시킬 수 있고 도달될 수 없는 규칙, 더 이상 연결되지 않는 규칙들을 쉽게 찾아낼 수 있는 장점을 가진다. 지식은 오류를 점검하는 편집 과정을 거쳐 오류가 없으면 네트워크 구조에 추가되며 최종적으로 지식베이스에 저장된다.

3.4 지식의 일관성 유지시스템

일관성 유지 시스템은 새로운 지식을 지식베이스에 추가하고자 할 경우, 추론 결과의 신뢰도를 높이기 위해 지식이 추가되기 전에 정확성을 검증한다. 이는 만약 오류가 존재한다면 오류를 자체 수정하거나 자체 수정이 불가능할 경우 전문가에게 문의하여 오류를 수정하도록 한 후 지식이 첨가되도록 하는 시스템이다.

3.5 사용자 인터페이스

사용자가 전문가 시스템을 쉽게 이용할 수 있도록 대화형의 인터페이스가 구축되었다. 전문가 시스템이 사용자에게 질문을 하면 사용자는 주어진 답안 중에서 하나를 선택하거나 간단한 문자열이나 숫자를 입력하든지 혹은 예(true), 아니오(false) 형태로 대답하면 된다.

4. 지식의 일관성 유지 시스템

일관성 유지시스템에서는 확실한 특성의 리스트와 우발적 특성의 리스트를 생성하고 검증 방법 및 단계를 개선하는 형태로 쌍 검증 방식을 보완한다. 이 리스트들과 개선된 검증 방법은 오류를 검색할 때 전방 추론이나 후방 추론을 해야 하는 과정의 대부분을 생략할 수 있게 함으로써 시간이 많이 걸리는 단점을 보완함과 동시에 오류 점검 단계에서 모든 가능성을 고려함으로써 생략되는 규칙이 없어 정확한 오류 점검을 수행한다.

특히 확실한 특성의 리스트에는 일반적인 지식이나 상식 등도 포함될 수 있어서 의미적 오류도 찾아 낼 수 있다. 본 논문은 부정의 프레미스를 포함한 네트워크로 지식을 표현하고 보완된 쌍 검증 방식을 이용하여 구조적 측면에서만뿐만 아니라 의미적 측면에서도 중복, 모순, 순환의 일관성 오류는 물론 도달될 수 없는 규칙과 더 이상 연결되지 않는 규칙 등의 완전성 오류를 찾아내어 자체 수정하거나 자체 수정이 불가능할 경우 전문가에게 수정하도록 메시지를 전달하는 일관성 유지시스템을 제시, 구

축하였다.

5. 결론

본 논문은 지식의 일관성 유지 시스템이 통합된 지식 기반 시스템을 제안하고 구축하였다. 이 시스템은 지식베이스 내의 지식과 데이터 베이스의 데이터 및 절차적 지식을 호출하여 새로운 지식을 추론하는 과정에서 추론 결과의 신뢰도를 높이기 위해서 지식의 일관성 검증을 위한 시스템이 통합되었다. 일관성 유지 시스템에서는 새로운 지식이 첨가되기 전에 기존의 지식과의 중복, 모순, 순환, 도달되지 않는 규칙과 더 이상 연결되지 않는 규칙 등의 오류를 판별한 뒤 오류가 없는 경우에만 새로운 지식이 지식베이스에 첨가된다. 따라서 신뢰도가 전적으로 지식 제공자에 의해 결정되는 방식을 피하고 오류 판별이 자동적으로 행해진다. 그 결과 지식 제공자는 새로운 지식을 일관성을 유지시키며 쉽고 간단히 첨가할 수 있어서 인간과 컴퓨터사이의 인터페이스의 기능도 향상되었다고 할 수 있다.

또한 지식을 네트워크 구조로 컴파일하여 꼭 필요한 지식만을 탐색함으로써 추론 결과의 효율성을 증가 시켰다. 향후 특정 분야에 대한 지식베이스를 구축하여 구축한 시스템을 실제로 적용해 보는 연구가 계속 진행될 것이다.

6. 참고 문헌

[1] Walling R. Cyre, "Capture, integration, and analysis of digital system requirements with conceptual graphs," IEEE Trans. on Knowledge and data engineering, Vol.9, No.1, pp.8-23, jan-feb, 1997.  
 [2] Gerard Ellis, "Compiling Conceptual Graphs," IEEE trans. on Knowledge and data engineering, Vol.7, No.1, pp.68-81, feb, 1995.  
 [3] Giarratano, Riley, "Expert Systems: Principles and Programming," PWS Publishing Company ed., 1997.  
 [4] N.K.Liu, "Formal Verification of Some Potential Contradictions in Knowledge Base Union a High Level Net Approach," Applied Intelligence, Vol.6, No.4, Oct, 1996  
 [5] P.Morizet-Mahoudeaux, "Maintaining Consistency of Database During Monitoring of an Evolving Process by a Knowledge-Based System," IEEE Trans. on systems, man and cybernetics, Vol.21, No.1, pp.47-60, 1991.  
 [6] D.L.Nazareth, "Issues in the Verification of Knowledge in Rule-Based Systems," I.J.Man-Machine Studies, Vol.30, pp.255-271, 1989.  
 [7] T.A. Nguyen, N.A.Perkins, T.J.Laffey and D.Pecora, "Checking an Expert system knowledge base for consistency and completeness," Proc. 9th IJCAI, LA, California, pp.375-378, 1985.  
 [8] M. Ramaswamy, S.Sarkar, Ye\_Sho Chen, "Using directed hypergraphs to verify rule-based expert systems," IEEE Trans. on Knowledge and data engineering, Vol.9, No.2, march-april, pp.221-237, 1997.

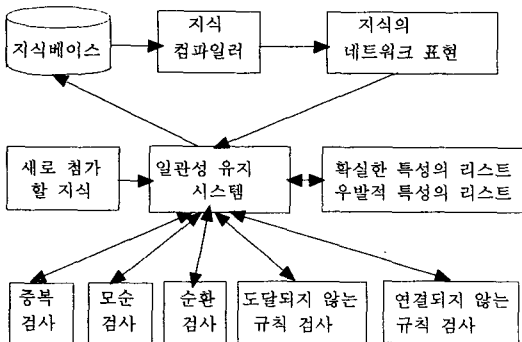


그림 2. 일관성 유지 시스템