

지식획득, 추론, 지식정제의 통합적 설계를 위한 규칙모델의 구축

이 계 성[†]

요 약

전문가시스템의 개발은 지식획득, 적합한 추론기구의 설계 및 구현, 지식의 정제 등 다단계 과정으로 이뤄져 있다. 각각을 하나의 연구이슈로 다양한 연구가 진행되어 왔으나 전체를 하나로 연계해 통합적 개발에 관해서는 상대적으로 연구가 활발히 진행되지 못한 실정이다. 지식획득은 전문가에 의해 수행되는 추론과정에서 특정 응용분야의 필요한 지식이 전달되어야 하므로 지식획득과 추론은 서로 밀접한 연관성을 갖는다. 지식의 정제는 추론과정에서 일어나는 문제의 제기과 이의 해결을 통해 지식베이스의 불완전하거나 논리적 모순을 찾아 해결함으로써 지식베이스를 보다 완벽하고 정확한 것으로 만드는 것이다. 본 연구에서는 서로 연관된 다단계 과정이 통합적으로 개발될 수 있는 환경의 설계에 대한 하나의 방안을 제시하려한다. 특히 도메인 모델이 잘 정립되기 어려운 분야에 학습기법을 활용해서 초기 지식베이스를 구성할 수 있는 점진적 지식획득방법과 이를 통해 만들어진 지식베이스의 규칙들을 학습기법의 일종인 개념적 클러스터링 기법을 이용하여 규칙모델을 구축하고 이를 이용해 효율적인 추론이 가능하게 하며, 지식획득 과정에서는 규칙의 오류를 제시할 수 있고 이에 대한 규칙의 수정이나 새로운 규칙이 기존의 지식구조에 합당한지를 결정하는 통합적 설계방안에 대해 연구한다. 지식의 정제는 설명기구와 규칙모델을 활용하여 문제의 원인을 찾고 해결점을 제시해 그에 대한 유효성을 검증함으로써 이뤄지게 한다.

Rule Models for the Integrated Design of Knowledge Acquisition, Reasoning, and Knowledge Refinement

Gyesung Lee[†]

ABSTRACT

A number of research issues such as knowledge acquisition, inferencing techniques, and knowledge refinement methodologies have been involved in the development of expert systems. Since each issue is considered very complicated, there has been little effort to take all the issues into account collectively at once. However, knowledge acquisition and inferencing are closely related because the knowledge is extracted by human experts from the inferencing process for solving a specific task or problem. Knowledge refinement is also accomplished by handling problems caused during the inferencing process of the system due to incompleteness and inconsistency of the knowledge base. From this perspective, we present a method by which software platform is established in which those issues are integrated in the development of expert systems, especially in the domain where the domain models and concepts are hard to be constructed because of inherent fuzziness of the domain. We apply a

※ 이 논문은 한국과학기술원 인공지능 연구센터의 1995년 기초연구비 지원에 의해 연구되었음.

† 정 회 원: 단국대학교 전자계산학과

논문접수: 1996년 9월 3일, 심사완료: 1996년 10월 29일

machine learning technique, conceptual clustering, to build a knowledge base and rule models by which an efficient inferencing, incremental knowledge acquisition and knowledge refinement are possible.

1. 서 론

전문가 시스템의 일반적인 구조는 크게 세 부분으로 나뉜다: 지식베이스, 추론기구, 사용자 인터페이스. 지식베이스는 문제해결지식을 나타내는 휴리스틱과 규칙들을 특정 코드형태로 표현하게 되고 추론기구는 문제해결 연산을 지식베이스로부터 필요한 지식을 선택해서 어떻게 적용하느냐를 제어하는 과정을 수행한다. 전문가 시스템의 설계 및 제작은 지식공학자라 불리는 개발자와 한 명 또는 그 이상의 전문가가 참여하여 이뤄진다. 보편적으로 지식공학자는 다음과 같은 단계를 거쳐 전문가시스템을 구축한다.

- 1) 문제영역의 이해
- 2) 적절한 지식표현방식의 선택과 추론기구의 설정
- 3) 전문가와의 인터뷰 또는 케이스 분석을 통해 지식의 획득
- 4) 선택된 지식표현형태로 획득된 지식의 코드화된 지식베이스구성
- 5) 성능개선, 기능확장을 위해 지식베이스의 테스트, 수정 및 확장.

첫 번째 과정은 문제영역의 주특성을 파악하고 이를 바탕으로 두 번째 과정의 지식표현방식을 결정할 수 있고 주어진 문제영역에 적합한 추론기구를 결정할 수 있는 기반을 마련할 수 있다. 일반적인 지식표현방식에는 규칙들(rules), 의미적 네트워크구조(semantic nets), 프레임(frame), 객체 지향적 지식구조 등이 있다. 성공적인 전문가 시스템이 완성되기 위해서는 효율적인 추론기구의 설계 및 제작이 중요하다. 초창기의 전문가 시스템이 수용했던 범용 추론방식(순방향, 역방향)은 문제의 유형과 속성에 관계없이 문제를 설정된 범용 추론방식에 맞추어 사용하였기 때문에 문제해결 방식이 비효율적이 되었다. 그러나 최근에는 소프트웨어 공학의 기법과 아울러 문제의 특성에 적합한 추론기구를 개발함으로써 문제의 특성과 문제해결 방법의 밀접함하여 주어진 문제를 가장 효과적으로 해결할 수 있는 방법이 보편적으로 사용되고 있다.

세 번째 단계, 지식획득은 전문가로부터 지식공학자에게 구체적 문제해결 지식의 전달이 일어나는 과정이고 지식공학자는 이 과정에서 생성되는 지식을 지식베이스에 적당한 형태로 저장하여 문제해결을 위해 효과적으로 사용되도록 한다. 그러나 지식공학자는 전문영역에 대한 지식을 이해하기는 매우 어렵고 반대로 전문가는 컴퓨터나 전문가시스템 개발에 대해서는 지식이나 경험이 없게 마련이다. 따라서 전문지식을 컴퓨터에 이전시키는 과정에서 이 두 영역 간의 연결을 원활히 해주는 것이 주요 관건이 된다. 지식의 이전에 따른 이 문제는 다음과 같은 구체적인 원인에 의해 발생된다.

- 전문가의 자기 지식에 대한 이해와 그것이 컴퓨터 프로그램에 구현되어야 하는 방식과의 차이 [10].
- 경험을 통한 케이스 분석의 한계, 즉 케이스 분석을 통해 모든 경우를 다 포함하기는 어렵다. 이것은 지식베이스의 불완전성으로 나타나게 된다.
- 전문가가 자기 지식에 대해 명료하고 체계적으로 표현하지 못함으로 인한 문제해결 지식의 모호성, 또한 개별 케이스로부터 얻어지는 지식들 간의 상호 연계성을 파악하는데 따른 어려움. 이것은 지식베이스의 모순(inconsistency)의 형태로 나타나게 된다.

이 문제들을 해결하기 위한 방안의 하나는 전문가로 하여금 중간 매개자 없이 직접 지식을 시스템에 전이시킬 수 있는 자동 지식획득 도구의 개발이 있다 [13]. 이 도구들은 미리 응용분야에 적합한 지식구조나 모델을 정립하여 전문가로 하여금 지식구조를 직접 구축할 수 있게 하고 그것을 바탕으로 새로운 지식의 첨가 또는 기존의 지식에 대한 수정 및 유지보수 등에 대해 보다 편리하고 체계적인 환경을 제공함으로써 중간 매개자가 있으므로 발생하는 문제를 최소화하도록 시도하였다. 지식베이스 개발의 어려움은 다른 소프트웨어 개발과 마찬가지로 그 개발이 점진적 개발이라는 점에서 잘 나타나 있다. 특히 '지식은 경험에 따라 발전하고 진화한다' [14]라는 사실에 비추보면 전문가 시스템 개발에서 지식베이스의 유효성과 논리적 일관성(consistency)을 유지하기 위해

지속적인 수정과 보완이 이뤄져야함을 알 수 있다.

위에서 설명한 전문가시스템 개발과정을 기능적인 관점에서 구분해보면 세 가지 부분으로 나뉜다: 지식획득 (Knowledge Acquisition), 추론 (Reasoning), 지식정제 (Knowledge Refinement). 각각의 과정이 매우 복잡하고 다양하여 이들을 연계해서 개발하기보다는 개별로 개발하여 통합하는 방식을 취하거나 각각을 위해 틀을 활용하여 전문가 시스템 개발에 사용해왔다. 추론기구는 지식획득을 통해 얻어진 지식을 활용하여 주어진 문제를 해결해야 하고 지식획득과정은 전문가에 의해 수행되는 추론과정에 기반을 두고 그 과정에서 적용되는 지식의 부분을 추출해 내야 하는 문제이므로 지식획득과 추론기구는 아주 밀접한 관계를 갖는다. 지식의 정제는 추론과정에서 일어나는 문제의 제거와 이의 해결을 통해 지식베이스의 불완전성이나 논리적 모순을 찾아 해결하는 과정으로 지식베이스의 문제는 추론에 의해 밝혀지고 이의 수정은 추론과정을 분석함으로써 가능해진다. 이 연구에서는 전문가 시스템 개발시 서로 밀접하게 연관된 세부부분을 통합적으로 연결하여 설계하는 하나의 방안을 제시한다.

2. 범용타스크 구조와 문제해결 방법

추론기구 설계가 전문가 시스템 전체의 성능을 결정하는데 있어 가장 중요한 요소로 알려져 왔다. 시스템 성능은 추론과정의 계산적 속도, 즉, 주어진 문제에 대해 얼마나 빠르게, 동시에 얼마나 효과적으로 결론에 도달할 수 있는지 여부에 의해 결정된다. 초기의 범용적 추론방법에서 더 발전하여 NEOMYCIN [4] 시스템에서 사용된 추상적 제어지식 (abstract control knowledge), TEIRESIAS [5]에서 사용된 메타 지식 (meta knowledge) 등이 개발되어 시스템의 성능 향상에 크게 기여하였다. McDermott와 여러 연구가 [12]들에 의해 문제 해결 기법은 특정 타스크의 유형에 의해 밀접히 연관되어 있다고 이해하였고 추론과 관련된 지식의 획득을 위하여 문제해결방법 (Problem Solving Method, PSM)이라는 방안을 소개하였다. 문제해결을 위한 제어지식이 추론기구에 코드화되어 사용되는 방법을 탈피하여 문제해결에 적합한 명시적인 PSM을 정의하고 이를 이용하여 문제해결에 맞

는 특정형식으로 지식을 획득하는 방법으로 지식획득과 추론을 결합하였다. 예로, 고장, 또는 질병 진단에서 사용되는 'cover and differentiate' PSM은 가설 (병명 또는 고장원인)과 증상 (symptoms)을 연결시키기 위해 두 가지 종류의 문제해결 지식을 요구한다. 하나는 '주어진 여러 증상에 대해 어떤 가설들이 이 같은 증상을 설명할 수 있는가'이고 다른 하나는 '주어진 가설집합에 대해 어떤 추가적인 정보가 사용되어 이들을 차별할 수 있는가'이다. 첫째 지식은 'cover'라는 PSM을 두 번째 지식은 'differentiate'라는 PSM을 나타낸다. 여기서 사용된 지식은 PSM 적용을 통한 추론에 직접 사용될 수 있고 역으로 PSM의 활용으로 추론에 필요한 지식을 획득할 수 있다. McDermott는 지식역할 (knowledge roles)이라는 개념을 사용하여 주어진 문제해결 상황에서 특정역할 (cover 또는 differentiate)을 하는 지식이 필요하고 이는 지식베이스를 구축하는데 사용된다.

문제의 특성과 해결기법의 관계를 정립하는 PSM 개념과 유사한 것으로 범용타스크 (generic task)라는 개념이 있다 [2]. 대상문제에 대한 지식구조나 추론구조는 문제해결 유형에 따라 일반적 유형으로 분류할 수 있다는 개념이다. 범용타스크는 세 가지 부분으로 구성되어진다: 문제, 해결방법, 부분제 (subproblem). 문제는 하나 또는 그이상의 방법을 동원해 해결해야 하고 개개의 방법은 지식의 형태, 해당 추론 구조와 문제해결을 위한 부분제의 정의에 의해 결정된다. 전형적인 범용타스크의 예로 진단, 분류, 설계 등이 있다. 타스크의 특성은 다음 정보에 의해 결정된다:

- 타스크 함수라 불리는 입출력 양태,
- 해당 응용영역의 도메인 지식,
- 타스크에 적합한 추론,
- 지식의 적용과 추론을 제어하는 방법.

예로 계층적 분류 (hierarchical classification)는 진단에서의 주요 타스크가 된다. 일반적인 결론 (가설)으로부터 구체적인 결론으로 유도해나가는 것이 계층적 분류에서의 주된 타스크이다. 지식구조도 이에 따라 계층구조를 갖는 가설들의 집합으로 표시할 수 있다. PSM과 마찬가지로 범용타스크도 특정 지식이 주어진 상황과의 관계를 결정할 수 있으며 이를 통해 시스템은 어떻게 적용할 것인가를 결정할 수 있다. 범용타스크는 주어진 타스크와 연관된 방법에 필요

한 지식을 식별할 수 있고 이에 따라 적당한 지식의 획득이 일어날 수 있다.

일반적으로 지식획득과 지식정제 (Knowledge Refinement)는 그 구분이 뚜렷하지 않다. 일반적으로 지식획득이 개발과정에서 추론과 함께 사용하면서 지식베이스의 문제를 찾고 이를 해결해 가는 과정에서 일어나거나 또는 완성된 시스템으로 사용되다가 필요한 경우 수정 또는 새로운 지식의 보완이 이뤄지기도 한다. 초기의 지식베이스를 구축하기 위해서 필요한 것이 지식획득이라면 그 이후에 점진적으로 일어나는 지식획득 및 지식베이스 유지보수는 지식정제에 포함하는 것이 보다 정확한 정의일 것이다. 대표적 지식정제의 예로 TEIRESIAS [5]가 있다. 지식정제는 미리 결과를 알거나 예측 가능한 case를 사용하여 만일 결과가 예측한 것과 차이가 있을 때 지식정제의 과정이 시작된다. 지식정제는 세단계로 이뤄진다: i) 오류 감지 ii) 수정 제안 iii) 검증. 첫 번째 단계인 오류 감지 단계에서는 원인 추적이 매우 중요하다. 원인 추적에는 전문가에 의한 방법 (ASK [10], MOLE [6])과 시스템의 감지에 의한 방법 (ODYSSEUS [16])이 있다. 전문가에 의한 원인추적은 주로 추론기구가 수행한 추론과정을 추적함으로써 이뤄지는데 이를 위해 설명도구의 활용이 필수적이다. 추론과정에서 적용된 규칙들을 추적하는 간단한 방법으로부터 추론과 관련된 제어지식과 도메인 지식 등을 이용해 보다 상세한 설명을 통해 원인추적을 효과적으로 수행할 수 있는 방법이 있다. 시스템에 의한 추적은 이것을 가능하게 하는 메타규칙이라는 제어지식이 있어 이의 적용이 실패할 때 일어난다. 실패한 규칙의 적용에 필요한 수정을 시스템이 제안하는 방법이다.

지식정제도 PSM 또는 범용타스크 구조를 이용한 추론과정을 통해 지식베이스의 문제점을 발견하고 이의 해결은 추론과정의 분석을 통해 원인 추적과 수정보완에 대한 제시가 이뤄져야한다. 결국 전문가 시스템 설계관점에서 보면 PSM 또는 범용타스크구조에 기반을 둔 전문가 시스템은 지식획득, 추론, 지식정제를 통합적으로 구성한 형태를 취할 수 있다. 그러나 많은 연구조사를 통해 파악한 바로는 기존의 연구가 적용되는 응용이 정의나 원인적 모델 (causal model) 등 해당 분야의 도메인 지식이 잘 정립되어 있

는 분야에서 성공적으로 활용된 것을 살펴볼 수 있다. 반면 도메인 모델이 잘 정립되어있지 않고 주로 경험에 의한 판단에 의해 해결되는 분야 (의학, 지질학, 법학, 경영 분야 등)들이 있다. 이런 분야에서는 PSM으로 해결되기 어렵거나 간단히 분류될 수 없는 복합적 고장 또는 질병이 있을 수 있다. 본 연구에서는 위의 PSM 구조 내에서 모델의 부재를 규칙모델을 추가하여 전문영역의 특성이나 모델이 잘 정의되기 어려운 응용에 적절히 사용될 수 있는 방안을 제시한다. 이를 위해 개념적 학습기법을 이용하여 규칙모델을 구축하고 이를 활용하여 전문가 시스템 개발의 지식획득, 추론, 지식정제에 적용할 수 있는 통합적 설계를 설명하기로 한다.

3. 규칙모델의 구축

3.1 규칙모델

규칙모델은 MYCIN 시스템의 개발과정에서 필요로 하는 지식획득과정을 처리하는 시스템인 TEIRESIAS에서 처음 소개되었다. TEIRESIAS는 전문가로 하여금 직접 문제해결과정을 추적하면서 시스템의 실행과정상 문제를 파악할 수 있게 도와주고 동시에 필요한 지식의 획득이 일어나게 된다. 초기의 지식획득 과정과 지식 정제과정을 통해 모아진 규칙들은 그들이 추론하는 결론에 따라 나뉘어져 하나의 그룹을 형성하고 그 그룹은 그들의 구성요소들에 의해 집단적 특성을 갖게 된다. TEIRESIAS의 규칙모델은 규칙들의 부분집합에 대해 그들의 공통적 요소, 규칙성 등을 간단한 통계적 방법을 통해 요약함으로써 구축된다. 예로 규칙베이스의 규칙들에 대해 어떤 특성의 임상 변수 (clinical variable)가 규칙의 조건부에 30% 이상 사용되었을 때 이들은 규칙모델의 특성을 결정짓고 만일 일련의 변수들의 집합이 규칙의 80% 이상을 차지하게 되면 이들도 규칙집합을 특성 짓는 구성요소로 사용된다.

TEIRESIAS에서의 규칙모델은 주로 지식정제의 목적으로 사용되어졌다. 각 규칙의 조건부와 결론부분의 요소들은 특정 문제해결상황과 기존의 규칙집합의 배경을 나타내므로 이들의 요약은 새로운 규칙에 대한 유효성을 결정할 수 있다. 새로운 규칙이 입력될 때 이 규칙은 기존의 규칙집합을 모델화한 규칙

모델에 합당해야하는 제약을 받는다. TERESIAS의 규칙모델의 한계는 이것이 지식획득시 기존의 규칙과 유사성, 또는 모순 점 등 구문적인 조사에 치중하는 검사에 한정적으로 쓰여졌다는 점에 있다. 또한 규칙모델의 구성이 매우 간단한 통계적 조사로 이루어졌기 때문에 그 유용도나 신뢰도가 낮음을 알 수 있다.

규칙은 그 자체가 문제해결 지식을 내포하고 있으므로 이들을 요약한 규칙모델을 이용하여 추론을 보다 효율적으로 할 수 있으며 새로운 지식의 기존의 지식과의 적합도를 결정할 수 있고 지식의 수정에도 사용될 수 있으리라 예상된다. 우리는 여기서 보다 유용한 정보를 내포하고 활용도를 높이는 규칙모델을 기계학습 기법의 일종인 개념적 클러스터링을 활용하여 구성하고 이를 추론, 지식획득, 및 지식정제에 적용함으로써 규칙모델의 유용도를 검증하고자한다.

3.2 개념적 학습기법

최근 기계학습 연구에서 계층적 클러스터링을 사용한 개념형성 시스템이 많이 논의되고 있다. 개념 학습 시스템은 레이블(각 자료 object에 대해 미리 정의된 분류, unsupervised learning)이 없는 자료들의 집합으로부터 개념적 표현인 클래스구조를 유추해내는 개념적 클러스터링 방법에서 발전한 시스템으로 볼 수 있다. 이 개념 형성과정은 학습과정으로 인식될 수 있고 따라서 그 과정이 점진적 절차에 의해 이뤄진다. 각 object가 시스템에 의해서 순차적으로 처리되고 이렇게 추가적으로 관측되고 처리됨에 따라 클래스 표현구조도 점진적으로 변화한다. 점진적 학습기법이 갖는 문제점으로 편향(bias)이라는 것이 있다. 이것은 구성되는 학습의 결과가 입력되는 자료순서에 의해 영향을 받는 문제로 자료의 입력순서에 따라 다른 결과를 생성해 내기 때문에 결과에 대한 최적여부를 결정짓기 어려운 경우가 생긴다. 이 연구에서는 전형적인 개념적 클러스터링 시스템인 COBWEB [7]을 모형으로 설정하고 편향문제를 해결하는 WOB 알고리즘을 고안해 규칙모델 구축에 활용한다.

3.3 WOB 알고리즘

WOB알고리즘은 점진적 클러스터링 방법의 바이어스 문제를 해결하기 위한 방안으로 점진적인 구축

대신 자료집단 전체를 분석하여 입력되는 자료가 가장 상이하도록 인위적으로 미리 배열함으로 편향 문제를 해결하려는 것이다. WOB의 알고리즘은 다음과 같이 요약될 수 있다:

- 비유사성 순서화 [15] 방식으로 자료집단을 순서화하고,
- [8]에서 정의된 분리점수 (partition score)를 이용해 계층적 분류 트리를 생성하고,
- 이 트리로부터 개념을 대표할 수 있는 노드를 선택하여 초기 클래스 집단을 구성한다.
- 카테고리 일치도 (match measure) [7]를 이용해 자료 각각을 가장 근사한 클래스에 재배치한다 (재배치 알고리즘).
- 위의 과정을 통해 재배치를 반복하는데 반복은 자료 이동이 없을 때까지 계속된다.

알고리즘의 첫 번째 단계의 순서화가 편향(bias) 문제를 완전히 해결할 수는 없다고 보나 많은 개선이 있음을 [15]에서 잘 보여준다. WOB는 카테고리 유용도 (utility measure)를 기준함수로 사용하여 2번 단계의 계층적 분류 트리를 생성한다. 클래스 C_k 에 대한 카테고리 유용도 CU_k [8]는 아래와 같이 정의된다.

$$\sum_i \sum_j P(A_i=V_{ij}|C_k)^2 - P(A_i=V_{ij})^2$$

여기서 A_i 는 속성을 나타내고 V_{ij} 는 속성값을 나타낸다. $P(\cdot)$ 는 확률을 나타낸다. CU_k 는 두개의 항의 곱으로 표시된다. 첫 번째 항, $P(C_k)$ 는 클래스 k에 있는 object 수를 전체 object 수로 나눈 값으로 이것은 큰 사이즈의 클래스에 새로운 object가 첨가되도록 유도하는 것이다. 두 번째 항

$$CU_k = P(C_k) \sum_i \sum_j (P(A_i=V_{ij}|C_k)^2 - P(A_i=V_{ij})^2)$$

은 주어진 클래스 C_k 에 대한 특성이 예측될 수 있는 기대값의 증가분으로 이것이 클수록 그 클래스의 결집도가 높아짐을 알 수 있다. 결집도를 높이려다보면 되도록 작은 사이즈의 클래스를 선호하게 되어 파편 현상 (fragmentation)이 일어나게 된다. 결국 이 두개의 항을 조화 있게 조절하여 각 클래스내의 object간에는 유사성을 최대한 하고 클래스간은 비유사성을 최대한 하는 클래스들의 그룹이 만들어지게 된다. 네

제 재배치 알고리즘은 자료 d와 클래스 k간 유사도에 의해 자료 d가 위치할 클래스 k가 결정되고 유사도는 카테고리 일치도에 의해 정의된다 [7].

$$CM_{dk} = P(C_k) \sum_{i, j \in (A_i)_k} (P(A_i = V_{ij} | C_k)^2 - P(A_i - V_{ij})^2).$$

3.4 규칙모델 구축

지식베이스에 있는 규칙들을 대상으로 규칙모델을 만든다. 대상은 지식구조나 모델 등이 매우 모호하게 정의되거나 개념 정립이 어려운 응용을 택하였다. 그 이유는 도메인이 잘 정의된 분야에 대해서는 개념이나 모델이 완벽하게 정의되어 있어 규칙모델없이 직접 잘 정립된 도메인 모델을 활용하여 문제해결을 시도할 수 있기 때문이다. 도메인의 설정은 유전지역의 지질학적 특성분석에서 사용되었던 규칙들[3]을 대상으로 하였고 위의 WOB 알고리즘을 이용하여 트리구조의 규칙모델을 완성한다. 각 규칙의 조건부와 결론을 속성과 속성값 세트로 구성하여 같은 속성을 결론짓는 규칙들을 모아 자료집단으로 설정한다. 이 분야에서 사용된 규칙들은 하나의 속성이 다양한 특성을 암시하고 또한 여러 속성들이 모여 하나의 특성을 의미하는 양태를 갖는다. 예로 Facies라는 지질특성은 14개의 Facies 구조(그림 1)를 갖고 이를 구분하는 속성은 16개로 이뤄져 있다 <표 1>. 이들로부터 만들어진 규칙이 144개가되므로 Facies를 분석하여 분류하는 작업이 얼마나 어려운 일인지를 짐작할 수 있다. 규칙의 예를 살펴보면 다음과 같다.

rule 219:

```

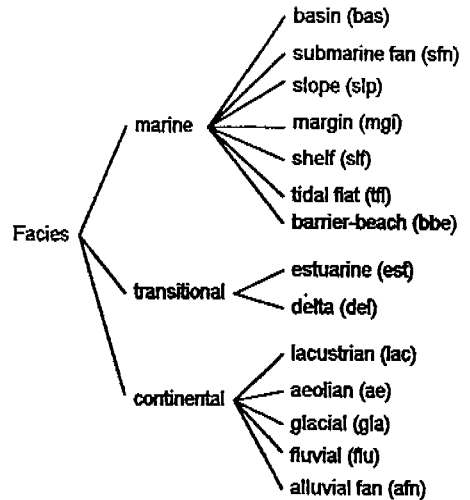
if ((depositional_setting basin)
    (updip_sediment_association sandstone)
    (primary_bedding_type horizontal)
    (sediment_type shale)
    (paleo_environment_indicator marine))
then
    ((facies basin) 0.4)
    
```

이 규칙은 <표 1>의 16개 속성에 대해 속성-값 쌍으로 아래와 같이 표현된다.

(DSE BAS) (PBS NA) (DSA NA) (ISA NA) (LTY NA) (BTH NA) (AGM NA) (PEI MAR) (PBT

HOR) (VSV NA) (USA SDS) (STY SHL) (PMA NA) (FNA NA) (STX NA) (SST NA)

여기서 NA는 'not applicable'을 나타내는 약어로 해당 속성에 대해 값이 없음을 나타낸다. 이 규칙 집단으로부터 만들어진 규칙모델은 트리의 구조를 갖고 각 노드는 유사한 특성을 갖는 집단으로 구성되어지고 각각은 그것을 구성하는 규칙들의 특성에 의해 규칙모델로 정의된다. 트리의 상위에 있는 노드는 일반적인 개념을 나타내는 규칙모델이고 이들에 대해 같은 알고리즘을 재귀적으로 적용하여 좀더 구체적인 노드의 규칙모델을 만들어 나가 트리를 구성하게 된다. 각 규칙모델은 그것을 구성하는 규칙들의 모임에 대해 확률적 요약으로 표현된다.



(그림 1) Facies 구조
(Fig. 1) Facies Structure

실세계 데이터에서 불완전한 정보는 여러 이유에 의해 발생하나 두 가지로 요약될 수 있는데 하나는 무작위적 예러나 무작위 관측에 의해 발생하는 것이고 다른 하나는 관측체계와 필요에 따른 체계적 (systematic) 불완전 정보가 있다. 전자는 오류에 의한 정보 미비인 반면 후자는 불필요한 정보의 수집을 방지함으로써 정보의 질을 높이는 결과와 자료수집에 따른 비용절감을 가져올 수 있다. 예로 의사는 진단을 하기 위해

〈표 1〉 Facies구조를 위한 속성들
 (Table 1) Attributes for Facies Structure

depositional setting	DSE
primary bedding shape	PBS
downdip sediment association	DSA
interbedded sediment association	ISA
lithology	LTY
bedding thickness	BTH
regional geometry	AGM
paleo environment indicator	PEI
primary bedding type	PBT
vertical sediment variation	VSV
updip sediment association	USA
sediment type	STY
paleomarker	PMA
fauna	FNA
sediment texture	STX
sediment structure	SST

환자의 증상에 대한 자료수집과 의료검사를 시행한다. 이때 의사는 환자의 질병종류와 상태에 따라 수집되는 자료의 종류와 양이 다르고 이는 순전히 의사의 의견에 달려있다. 주어진 Facies 규칙들을 WOB 알

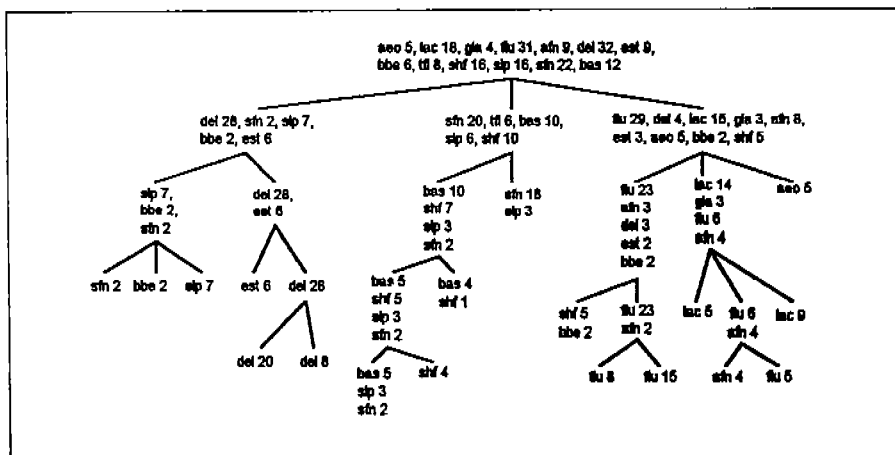
고리즘에 맞는 자료형태로 전환했을 때 약 60-70% 정도가 값을 갖지 않은 자료형태를 취한다. 이같은 체계적 불완전 정보의 처리를 위해 카테고리 유용도를 수정한다. 우선 카테고리 유용도를 측정하기 위한 수식에서 $P(A_i=V_{ij})$ 와 $P(A_i=V_{ij}|C_k)$ 를 다음과 같이 재정의한다. $P(A_i=V_{ij})$ 은 $\frac{A_i=V_{ij}$ 를 갖는 object 수}{관측된 A_i 를 갖는 object 수}로 재정의된다. 관측된 자료에 대해 처리하기 위해 현저성(salience) 요소를 정의하는데 이것은 속성에 대한 값이 특정 클래스에 대해 관측될 확률을 나타내게 한다. 이를 고려한 카테고리 유용도는 아래와 같이 정의된다:

$$P(C_k) \left(\sum_i P(\text{관측된 } A_i | C_k) \sum_j P(A_i=V_{ij} | C_k)^2 - \sum_i P(\text{관측된 } A_i) \sum_j P(A_i=V_{ij})^2 \right)$$

여기서

$$P(\text{관측된 } A_i | C_k) = \frac{A_i \text{에 대해 관측값을 갖는 } C_k \text{의 object 수}}{C_k \text{의 object 수}}$$

WOB 알고리즘을 이용하여 Facies 규칙 집합에 적용한 결과 (그림 2)와 같은 규칙모델이 만들어졌다. 규칙모델트리의 각 노드는 해당 Facies를 결정짓는 규칙들로 구성되어 있고 각각의 Facies 옆에는 그들



(그림2) Facies 규칙모델
 (Fig. 2) Facies Rule Model

의 빈도를 숫자로 나타내었다. (그림 1)의 구조와 달리 이 규칙모델트리는 문제해결 목적으로 구성되어 있다. (그림 1)과 같은 도메인 모델은 전체적인 분류를 퇴적환경이라는 하나의 관점에서 분류한 트리여서 Facies의 구조를 결정할 때 필요한 다양한 특성에 따른 분류는 불가능하다. 반면 전문가에 의해 주어진 규칙들을 재구조하여 Facies 구조 결정을 효율적으로 가능하게 하는 규칙모델을 이용하여 주어진 정보가 입력될 때마다 이에 적합한 규칙모델 노드를 설정하고 이 노드에 초점을 맞춰 필요한 정보의 요구하는 방식의 추론이 가능하게 된다. 즉, Facies 구조를 분류하는 과정은 일반적인 자료수집에서 시작하여 그 자료들이 암시하는 현재의 상태를 가장 잘 나타내는 상황을 규칙모델트리에서 찾아 가장 유사한 규칙모델 노드로 선정한다. 이를 기반으로 현재의 가설에 대해 추가의 필요한 정보의 요구를 결정하고 좀더 구체적인 결론에 접근하도록 유도한다.

3.5 규칙모델을 이용한 추론

규칙모델트리는 일반적인 개념에서부터 점차 구체적인 개념으로 세분화 되가는 구조를 가지므로 규칙모델과 PSM을 이용한 추론은 다음과 같은 세단계 과정으로 요약된다. 1) 규칙모델의 선택, 2) PSM 선정, 3) 증거(evidence)수집을 위한 query 선정. 첫 번째 단계는 규칙모델의 선택으로 규칙모델 트리에서 문제 해결상황을 가장 잘 나타내는 규칙모델을 찾는다. 규칙모델을 찾는 방법은 A* 알고리즘과 유사한 함수 f를 정의한다.

$$f = g + h$$

$$g = \sum bel(\text{fired rules})$$

$$h = \frac{\sum_{mat-av} bel(\text{unfired rules}) - \sum_{unmat-av} bel(\text{unfired rules})}{\sum_{tot\ poss\ mat} bel(\text{unfired rules})}$$

g 함수는 현재 상황과 가장 적합한지를 결정하는 함수로 각 규칙모델에서 사용된 규칙들의 신뢰도의 합으로 표시된다. 두 번째 함수 h는 일종의 예측함수로 해당 규칙모델이 최종결론에 이를 가능성을 나타내는 함수이다. mat-av는 현재 제공된 증거와 일치한 조건을 포함한 규칙들을 의미하고 그렇지 않은 규칙들

은 unmat-av로 표시한다. 또한 tot poss mat는 해당 규칙모델의 총 가능한 규칙들을 의미한다. 함수 f는 현재의 상황에 가장 적합한 규칙모델을 결정짓고 이 규칙모델에 포함된 규칙들이 사용되어 보다 최소의 증거에 의해 최종결론에 도달하도록 query를 선택한다. 규칙모델의 선택은 규칙모델 트리의 첫 번째 레벨의 노드로부터 시작한다. f 값은 트리에서 하향으로 내려가면서 값이 증가하다가 감소하는 형태를 취하고 실행이 진행될 수록 하위레벨에 있는 노드의 f 값이 증가함을 알 수 있다. 이는 증거가 수집 되는데로 적용되는 규칙모델이 보다 구체적이기 때문이다. f값을 이용한 규칙모델의 선정은 경험에 의해 설정한 임계값에 의해 결정된다. 부모노드에 달려있는 자식노드들중 f값이 부모노드의 f값 보다 크고 자식노드의 최대 f값 (f-max)의 50% 이상인 노드가 선택된다. 구체적인 규칙모델 (하위 노드)로의 확장은 자식노드의 f값이 부모노드의 평균 f값, $f_{avg} = \frac{\sum_{parent}}{\#\ child\ node} + h_{parent}$ 보다 클 때 이뤄진다. 규칙모델이 선택되면 PSM이 선정되는데 이 PSM은 3개의 PSM, 즉, maxhyp, filter, differ중 하나가 선택되어 사용된다. 각각이 선택되는 조건이 다른데 maxhyp의 경우는 하나의 가설이 다른 가설에 비해 신뢰도가 높아 그 가설로 결론지어질 가능성이 매우 높고 하나의 규칙모델이 선정될 때 선택된다. 이는 현재 가장 가능성이 높은 가설을 집중적으로 공략하는 문제해결방법이고 filter는 하나 이상의 다중가설이 서로 경쟁하고 있는 상황에서 하나의 규칙모델이 선택되거나 혹은 대상 가설이 두 개(최대 선택될 규칙모델은 두 개로 제한하였음)의 규칙모델에 고루 분포될 경우 filter PSM을, 그렇지 않은 경우에는 McDermott의 'differentiate' PSM [12]와 같은 differ PSM을 적용한다. 만일 두 개의 규칙모델이 선정되고 해당 가설의 신뢰도가 낮은 경우 (실행의 초기부분의 경우) differ PSM을 선택한다. 선택된 규칙모델과 현재의 가설을 조사하여 셋중 적합한 하나가 선정되고 선택된 PSM을 처리하기 위해 증거수집을 위한 Query 선정이 뒤따르게된다 [1].

3.6 지식획득과 지식실제

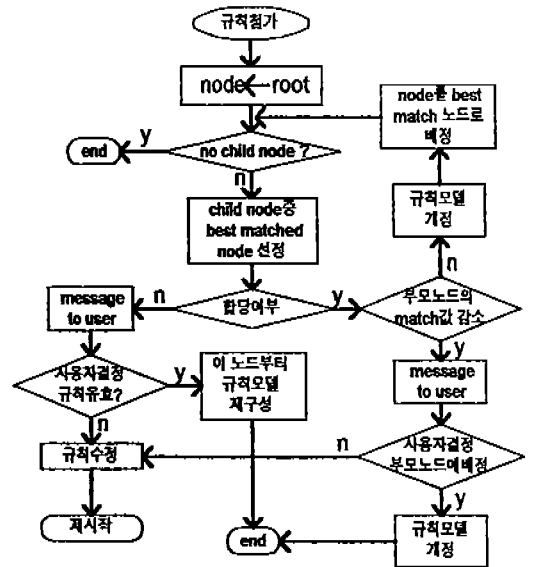
일정수준의 규칙이 전문가에 의해 편리한 사용자 인터페이스를 통한 규칙입력 도구에 의해 모아지면

최초의 규칙모델 트리가 구성될 수 있다. 일단 언어진 규칙들은 자료집단으로 표현되고 이들은 WOB 알고리즘에 의해 규칙모델트리로 구성된다. 완성된 트리는 전문가에 의해 면밀히 조사되고 노드로 정의되는 규칙모델의 구성과 확률적 요약을 분석하여 오류의 유무를 확인할 수 있게 된다. 예로 하나의 노드에 포함되어 있는 규칙들이 동일한 결론을 갖지 않고 극소수의 다른 결론을 갖는 규칙들이 포함되어지는 경우가 있게 되면 전문가는 이 규칙들에 대해 특별한 케이스를 처리하는 규칙인지 아니면 잘못된 규칙인지를 결정하여 필요한 조치를 취하게 된다.

위의 정적인 지식베이스의 유지관리와 함께 케이스 라이브러리와 설명기구를 통해 지식베이스의 오류를 해결하는 동적 유지관리 방법인 지식정제가 있다. 케이스 라이브러리에 저장된 케이스들은 결과를 이미 알고 있는 케이스들로 실행을 통해 올바른 결론이 도출되지 않으면 전문가에 의해 오류에 대한 원인 추적이 시작된다. 전문가는 설명기구[11]를 통해 추론 과정에 적용된 PSM과 규칙모델, 그리고 시스템에 의해 선택된 query를 조사하여 오류의 원인을 추적하게 되고 발생한 오류에 대한 수정정보만이 전문가에 의해 제안되고 이에 대한 검증과정으로 지식정제의 한 사이클이 완료된다. 추론기구는 문제해결지식을 규칙모델로부터 추출하여 활용하기 때문에 실행중 발생하는 오류에 대한 원인 추적은 결국 규칙모델에서의 문제로부터 발생한다고 보고 (추론기구의 문제해결 알고리즘에는 오류가 없다고 가정한다) 해당 규칙모델을 구성하는 규칙을 대상으로 오류를 조사한다. 지식정제에는 규칙의 수정, 새로운 규칙의 첨가, 또는 삭제가 있다. 각각에 대해 이들은 모두 확인과정을 거치는데 확인과정은 규칙모델을 기준으로 이뤄진다. 새로운 규칙이 첨가될 때 새 규칙은 기존의 규칙모델트리에서 가장 유사한 규칙모델에 포함되는데 이는 일치도 (match value), $\sum_{i, j \in object} P(A_i = V_{ij} | C_k)$ 에 의해 그 규칙이 소속할 모델을 결정한다. 이때 이 규칙이 수용되는 모델은 새로운 구성으로 갱신되어야 한다. 새 규칙의 입력에 대한 처리는 (그림 3)과 같이 기존의 규칙들에 의해 구축된 규칙모델에 합당여부를 통해 규칙의 유효성을 검증하고 필요에 따라 규칙의 수정 또는 규칙모델의 재구성 등 해당 상황에 따라 적절한 재조정이 필요하게 된다. 합당여부는 규칙

모델에 포함되어있는 다른 규칙들과 유사한 결론을 가지는지에 의해 결정된다. 규칙이 기존의 모델에 합당하지 않다는 판결이 나면 그것은 사용자(전문가)에 의해 규칙에 대한 오류여부를 최종 확인하게 되고 이 단계에서 오류라 판정되면 규칙을 수정하여 전 과정을 다시 시작해야한다. 만일 유효하다는 판결이 나면 이 노드로부터 규칙모델의 재구성이 필요하게된다.

새로 입력되는 규칙이 위치할 노드가 규칙모델 트리의 상위노드로부터 하위노드로 진행해 가면서 결정되는데 일치도 (match value)가 하위노드로 갈수록 증가한다. 그 이유는 하위노드일수록 구체적인 모델이기 때문이다. 이럴 경우 그 규칙은 세부적인 분류를 하는 것이기보다는 상위개념을 구분하는데 적합한 규칙일 수도 있기 때문에 전문가의 의견에 따라 부모노드에 남아 있을 수도 있고 아니면 재수정의 대상이 될 수 있다.



(그림 3) 규칙 첨가에 따른 유효성 검사 (Fig. 3) Validity Test for A Rule Addition

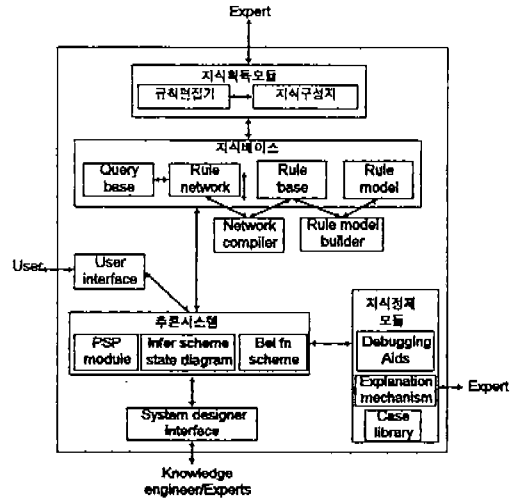
삭제의 경우 해당 규칙이 포함된 노드들로부터 규칙을 제거하고 그에 따른 규칙모델도 갱신된다. 이 경우 삭제가 일어난 노드에 현격한 변화가 있는지를 조사해야한다. 예로 삭제가 여러 번에 걸쳐 일어났다면 규칙모델이 제대로 구성되어 있는지가 의문이 될

수 있다. 현격한 변화를 조사하기 위해 삭제가 일어난 노드의 규칙들에 대해 WOB의 재배치과정을 적용하여 만일 특정값(예로 20%)이상의 변화가 있을 때는 그 노드 이하의 모든 노드에 대해 WOB를 적용하여 새로이 규칙모델을 구축하도록 한다. 규칙의 수정에 대해서는 그 규칙을 제거한 후 새로 첨가하는 두 과정의 연속으로 처리할 수 있다.

4. 논의 및 결론

본 논문에서는 개념적 학습기법을 이용하여 규칙 모델을 구성하고 이의 활용을 통해 추론, 지식획득 및 지식정제를 통합적으로 해결하는 하나의 방안을 제시하였다. 모델의 정립이 어렵거나 개념이 모호하여 전문지식의 획득이 어렵고 결국 올바른 지식베이스 구축이 힘든 분야에 기계학습 기법, 그 중 unsupervised 학습기법인 개념적 클러스터링 방법을 활용하여 전문가가 제공한 지식을 새로이 재 구축하여 효율적인 문제해결지식을 유출해내고 이를 이용해 문제 해결을 위한 추론과정에 적용할 수 있었고 또한 규칙 베이스를 구축하는 과정에서 새로운 규칙의 유효성을 시스템에 의해 검증할 수 있는 방안을 제시하였다. 지식정제를 위해 규칙모델은 문제발생의 원인 규정은 물론 수정한 지식베이스의 검증하는데 효과적으로 사용될 수 있음을 보였다. 규칙모델을 사용하여 추론의 효율을 14%의 추론 시간을 단축하였음을 실험을 통해 보였다 [1]. 기존의 Facies 규칙들을 이용해 규칙모델을 구축하면서 발견한 오류 규칙과 지식정제 과정을 통해 발견한 규칙들은 총 4개 정도였다. 이는 전체 Facies 규칙의 3% 정도에 해당하는데 이와 같이 매우 적은 양의 규칙이 수정된 때에는 이 규칙들이 장기간에 걸쳐 자세히 검토되고 정련되어 왔기 때문에 오류의 가능성이 매우 적었다. 현재 여기서 제안된 사항들을 포함하여 개발도구의 구축이 진행 중에 있다. 이미 구현된 부분과 계속 구현해나가는 부분이 있다. 이 개발 시스템(IESDT, Integrated Expert System Development Tool)의 전체 구성도는 (그림 4)에 나타나있다. 현재 이 시스템을 새로운 응용에 적용시켜보려는 시도를 하고 있고 지질학 전문가와 함께 새로운 지식베이스 구축 및 전문가 시스템 구축을 계획하고 있다.

이 연구의 또 다른 결과는 지식의 재발전이다. 규칙모델 (그림2)과 Facies 구조(그림1)를 비교해보면 (그림 1)의 Facies 구조는 지질학에서 교육용으로 가르치는 구조로 퇴적층의 형성 환경이라는 관점에서 분류한 단순 구분이다. 반면 규칙모델은 Facies 구조의 지질학적 특성분석을 통해 Facies 구조를 파악하는 문제해결에 중점을 둔 구조라 할 수 있다. 결국 규칙모델은 문제해결을 위한 지식을 함축하고 있어 이는 지식의 재구조를 통한 문제해결 지식의 획득이라 할 수 있다.



(그림 4) IESDT 시스템 구조
(Fig. 4) System Structure of IESDT

참고 문헌

[1] Biswas, G. and G. Lee, "Knowledge reorganization :A rule model scheme for efficient reasoning," IEEE Tenth Conference on Artificial Intelligence for application, 1994.
 [2] Chandrasekaran, B., "Task-structures, knowledge acquisition and learning," Machine Learning 4, pp. 339-345, 1989.
 [3] Cheong, D. K., et al., "PLAYMAKER, a knowledge based expert system," Geobyte, vol. 7, no. 6, pp. 28-41, 1992.

[4] Clancey, W. J., "Representing control knowledge as abstract tasks and metarules," Expert system applications, Bolc, L. and Coombs, M. J. (eds.), Springer Verlag, 1988.

[5] Davis, R., "Use of meta-level knowledge in construction, maintenance, and use of large knowledge bases," Ph. D. Dissertation, Stanford University, 1976.

[6] Eshelman, L., "MOLE:a knowledge acquisition tool for cover and differentiate systems," Automating knowledge acquisition for expert systems, S. Marcus (eds.), pp. 37-80, 1988.

[7] Fisher, D. and P. Langley, The structure and formation of natural categories, Psychology of learning and motivation, Bower, G. H. (Ed.), Academic Press, 1990.

[8] Fisher, D. H., "Knowledge acquisition via incremental conceptual clustering," Machine Learning 2, pp. 139-172, 1987.

[9] Gennari, J. H., "Focused concept formation," Proceedings of the 9th International Workshop on Machine Learning, Aberdeen, Scotland, pp. 464-469, 1992.

[10] Gruber, T. R., "Automated knowledge acquisition for strategic knowledge," Machine Learning 4, pp. 293-336, 1989.

[11] Lee, G. and G. Biswas, "A new version of MIDST for building PLAYMAKER, a knowledge-based system characterizing hydrocarbon plays," Conference on AI in Petroleum Exploration and Production, 1992.

[12] Marcus, S. (ed.), "Automating knowledge acquisition for expert systems," Kluwer Academic Publishers, Boston, Mass., 1988.

[13] Musen, M. A., "Automated Support for building and extending expert models," Machine Learning 4, pp. 347-376, 1989.

[14] Newell, A., "The knowledge level," Artificial Intelligence 18, pp. 87-127, 1982.

[15] Weinberg, J. and Biswas, B., "The role of the order bias in concept formation," Tech. Report

CS-93-01, Dept. of Comp. Sci., Vanderbilt University, Nashville, TN, 1993.

[16] Wilkins, D. C., Apprentice learning techniques for knowledge based systems, Ph. D. Dissertation, University of Michigan, 1987.



이 계 성

1980년 서강대학교 전자공학과 졸업(학사)
 1982년 한국과학기술원 전산학과(석사)
 1994년 Vanderbilt대학 전산학과(공학박사)
 1982년~1985년 경제기획원 조사통계국 전산처리관
 1994년~1996년 대구대학교 전산정보학과 전임강사
 1996년~현재 단국대학교 전자계산학과 전임강사
 관심분야:전문가 시스템, 기계학습, ITS, Data mining