규칙 기반 전문가 시스템에서의 설명 기법에 관한 연구

전도홍†

요 약

규칙 기반의 전문가 시스템에 있어서 사용자에게 추론 결과를 설명하는 기능은 중요한 기능들 중의 하나이다. 설명은 크게 문제 분야 지식에 관한 설명과 문제 풀이 방법에 관한 설명의 두 가지로 나눌 수 있으며 여기에서는 문제 풀이 방법에 관한 설명 기법을 제안하였다. 이 논문에서는 규칙 전문가 시스템으로 구현된 응용 분야 업무를 태스크(task)라는 계층적이고 독립적인 구조를 갖는 형태로 구성한 후 이를 이용하여 사용자에게 좀더 인지적인 설명을 제공해 줄 수 있는 새로운 설명 생성 기법을 제안하였다. 제안한 기법에서는 문제 풀이 방법에서 전통적으로 중요한 '왜 이 작업을 수행하는가(why)?'하는 것과 '이 작업을 어떤 식으로 수행할 것인가(how)?'하는 두 가지 질문에 대한 설명이 가능함을 보였다.

A Study on an Explanation Method in Rule-based Expert SYSTEM

Do Hong Jeon†

ABSTRACT

In Rule-Based expert system, the function which explains the reasoning result to user is one of the important functions. These explanations are separated into two kinds. One explanation is about domain knowledge source, the other is about problem-solving strategy. In this paper, the method of the problem-solving strategy is proposed. After constructing the form of applying domain service which embodyed to Rule-Based expert system into the form of the hierarachical and independent which is called task, the generation method of a new one which can offer recognizable explanation to user. The proposed method show possibilities about two questions which are "Why achieve this work?" and "How method have to achieve this work?".

1. 서 론

전문적 수준의 문제를 해결하기 위하여 지식과 추론을 이용하는 지능적 컴퓨터 프로그램을 전문가 시스템이라 한다[1]. 전문가 시스템에는 문제 분야에 관한 특정한 지식을 다량으로 주입시켜야 하며 이러한 전문가 시스템은 현실적으로 그 유용성이 인정되어 많은 분야에 걸쳐 만들어졌다[2]. 전문가 시스템에서 설명 기능은 시스템이 특정한 결론에 도달하게 된 과정(process), 즉 시스템이 어떤 방식으로 추론을 했는가를 보여주는 것이다[3].

전문가 시스템이 개발되기 시작한 이래로 설명 시스템은 전문가 시스템에서 가장 중요한 기능중의 하나로 인식되어져 왔다. 설명 기능은 최종 사용자에게는 시스템의 지식과 추론과정을 보여

† 정회원: 관동대학교 전자계산공학과 부교수
논문접수:1998년 3월 12일 심사완료: 1998년 5월 27일

줌으로써 확신을 줄 수 있을 뿐 아니라 지식 공학자 또는 시스템 개발자로 하여금 시스템을 개발하는 동안에 시스템을 개선하고 지식을 테스트하는데 도움을 준다[4].

설명 생성 기능의 세 가지 연구 분야는 첫째, 설명 내용을 발생시키는 것이다. 둘째, 사용자의 지식 수준에 맞추어 발생된 설명 내용을 적절히 변형하는 것이다. 셋째, 사용자와의 인터페이스 구현이다. 이중 가장 중요한 분야는 설명 내용의 생성이다[5]. 초기의 설명 생성에 대한 연구는 단순한 규칙의 추적을 통하여 설명을 제공하는 것이었다[6].

그 이후 문제 풀이 방법(problem solving strategy) 레벨의 설명의 필요성을 인식한 몇몇 사람들에 의해서 문제 풀이 방법에 대한 연구가 있었다[7][8].

이 논문에서는 문제 풀이 방법(problem-solving strategy) 레벨의 설명에 대하여 태스크(task)라는 새로운 개념을 이용한 설명 기법을 제시하였다. 즉, 1) 설명 시스템에 대한 개요를 설명하고 2) 기존의 설명 시스템을 분석하였으며, 3) 제안한 태스크 기반의 전문가 시스템 구조를 제안하고 4) 태스크 기반의 설명 생성 기법에 대하여 제안하였다. 끝으로 6장에서 결론을 맺었다.

2. 설명 시스템의 개요

많은 사람들이 설명의 종류를 나누는 작업을 수행하였다. Shortliffe는 how와 why로 질문의 종류를 나누고 이에 따른 설명을 구현하였다[6]. 사용자는 다음과 같은 두 종류의 질문을 할 수 있다.

첫째는 how이다. 이는 '어떻게 이 결론에 도달했는가?' 하는 질문이다. 둘째는 why이다. 이는 '왜 이와 같은 질문을 하는가?'하는 질문이다. 첫 번째 질문에 대해 시스템은 결론에 이른 추론 과정을 역추적 함으로써 설명을 해준다. 두 번째와 같은 질문을 받았을 때 시스템은 내고자 하는 결론을 보여줌으로써 사용자의 질문에 대답한다. 이와 같은 질문과 설명의 형태는 주로 상담(consultation) 시스템에서 나올 수 있는 것이다. 특히 위와 같은 질문과 설명은 Shortliffe의 MYCIN이라는 의료 상담 시스템에 적용된 것이다. 위에서 why 설명은 "왜 이러한 추론 과정을 좇아가고자 하는가? (why the system is pursuing a line of reasoning)"에 대한 설명이다. 이 설명은 오늘날의 문제 풀이 방법 차원의 설명으로 발전되었다고 볼 수 있다.

Waterman은 다음과 같은 세 가지 부류로 설명을 나누었다[1]. 첫째는 회상 방식(retrospective) 설명으로 이는 시스템이 결론에 이르게 된 과정을 보여주는 설명이다. 둘째는 가설적 추론에 대한 설명으로서 이는 어떤 사실이나 규칙이 다르다면 어떤 다른 결과가 나타나는지에 대한 설명이다. 셋째는 모순적 추론에 대한 설명이며 이는 예상되는 다른 결론이 왜 나오지 않았는지에 대한 설명이다. 이중 가설적 추론에 대한 설명은 시스템이 결론에 이르게 된 추론 과정을 보여주는 것이 아니라 상황을 다르게 하여 추론한 결과를 보여 주는 것이므로 전통적인 설명의 범주에는 해당되지 않는다고 볼 수 있다.

Chandrasekaran은 다음과 같이 설명을 나누었다[5]. 첫째는 판단에 대한 설명(explaining decision)이다. 이는 시스템이 왜 이러한 결정을 했는지에 대한 설명으로서 시스템은 결정에 쓰인 지식을 보여줌으로써 설명해 준다. 둘째는 지식에 대한 정당성 제공으로서 시스템의 문제 풀이 지식이 왜 옳은지에 대한 설명이다. 이와 같은 질문을 받았을 때 시스템은 심층 지식(deep knowledge)을 통해 설명할 수 있다고 제안하였다. 셋째는 전략 설명(explaining strategy)이며 이는 문제 풀이 방법을 설명하는 것을 말한다.

Chandrasekaran은 이와 같은 설명을 위하여 태스크 생성이라고 하는 개념을 도입하였다.

이 논문에서 제안한 문제 풀이 방법 레벨의 설명이란 시스템이 왜 이와 같이 문제를 풀었는지 또 전체적인 일 처리의 방법과 흐름이 무엇인지를 보여주는 설명이라고 할 수 있다. 좋은 설명 시스템은 시스템이 무엇을 했는지 하는 것에 대한 설명뿐만 아니라 그와 같은 일을 왜 했는지에 대한 설명을 제공해 줄 수 있어야 한다[9]. 일반적인 사용자는 일 처리의 흐름에 관심을 많이 가

지고 있다. 시스템이 전체적으로 어떤 식으로 문제를 푸는지를 알면 더욱 좋은 설명 시스템이 될 것이다. 그러나 문제 풀이 방법의 설명은 일반적으로 제공해 주기가 어렵다. 왜냐하면 전문가 시스템의 지식 구성 자체에 관련된 문제이기 때문이다.

3. 기존 설명 시스템의 고찰

3.1 MYCIN

Davis와 Shortliffe등이 개발한 MYCIN은 혈액을 통한 전염병 등을 진단하고 치료하는데 조언을 해 주는 시스템이다[6]. MYCIN은 규칙 기반 시스템으로 자신의 문제 풀이 과정을 규칙 레벨에서 역추적 함으로써 자신이 한 일에 대한 설명을 제공해 주고 있다. MYCIN의 단점은 수행한 일에 대해서는 설명해 줄 수 있지만 '왜 그와 같이 일을 수행했는가?'하는 질문에 대해서는 설명을 제공해 줄 수 없다는 것이다. 이는 MYCIN이 규칙 단위의 지식 표현만을 가지고 있기 때문이다. 문제 풀이 전략은 규칙 기반으로 이루어진 지식 속에 내재적으로 들어 있기 때문이다

3.2 NEOMYCIN

Clancey는 NEOMYCIN을 통해서 문제 풀이 방법 레벨의 설명을 시도하였다[8]. NEOMYCIN은 MYCIN의 후속 모델로서 MYCIN의 휴리스틱한 규칙 구조에 내재적으로 들어 있던 문제 풀이 전략을 외적으로 명시하여 주었다. NEOMYCIN은 의학지식과 이 의학 지식을 관리하는 전략적 지식을 메타 지식으로 구별하였고 문제풀이 전략 레벨에서 how와 why에 대한 설명을 구현하였다. Clancey가 제안한 태스크 구조는 근본적으로 상담 시스템에 맞게 구성되어 있어 일반적으로 적용되기 어렵고 그의 문제 풀이 방법 설명은 단순히 태스크에 매달린 메타 규칙을 보여주는 것이었다. 그 메타 규칙은 현재 태스크를 고른 이유에 대해서만 나와 있다.

3.3 Generic Task System

Chandrasekaran은 작업의 유형을 6개로 나누고 이를 generic task라 하였다[12]. 각각의 generic task를 위하여 개별적인 지식 구축 언어를 만들어서 지식을 구축하였다. 이를 통하여 효과적인 일 처리 및 설명을 구현하고자 하였다. 그러나 이는 다른 유형의 작업에는 적용될 수 없는 유연하지 못한 태스크 구조이다.

Chandrasekaran은 일 처리를 하는 각각의 객체들이 서로 다른 구조를 가지도록 하였고 서로 계층적인 구조를 가지도록 하였다. 또 이들을 각각 기술하는 DSPL이라는 언어를 만들었다. 이 언어는 주로 설계 및 계획 분야에 이용되도록 고안되었기 때문에 어떤 특정한 문제 분야에서는 효율적일 수 있지만 응용 분야가 바뀌면 일반적으로 적용하기 어려운 구조임을 알 수 있다.

이밖에도 문제 분야의 지식 이외에 다른 차원의 처리 규칙을 통한 설명을 제공하는 시스템은 Swartout의 XPLAIN이 있다[10]. Swartout와 Moore는 고질의 설명을 위하여 심층지식을 제안하였고 자동 프로그래밍 기법을 사용하여 사용자와 상담하는 동안에 사용자의 관심에 맞는 문제 풀이의 정당성을 제공하였다[10][11]. 그러나 이러한 방법은 문제 풀이 지식 외에도 정당성 및 전문 용어 정의 지식을 사용하므로 다른 실용화된 시스템에는 적용시키기가 어렵다[4].

이 논문에서는 이상에서 설명한 기존의 시스템들과는 달리 태스크라는 것이 문제 풀이 지식 체계 내에서 독립적으로 처리되도록 하고 독자적인 목표와 이 목표를 위한 지식 표현과 처리 기능을 가지는 새로운 문제 풀이 방법의 설명 기법을 제안하였다.

4. 태스크 기반 전문가 시스템의 구조

대부분의 기존 전문가 시스템 설명 기능은 문제 풀이 방법(problem-solving strategy)의 설명을 제공해 주지 못하는 단점이 있다. 풀이 방법 레벨의 설명 기능은 지식 구축 때부터 정교하게 구축해야 가능하기 때문에 일반적으로 제공하기가 매우 어렵다.

문제 풀이 방법의 설명과 지식의 구성과는 떼

어놓을 수 없는 관계이다. 즉 어떤 식으로 지식을 구축했느냐에 따라서 문제 풀이 방법에 관한 설명은 많은 차이를 나타낼 수밖에 없다.

문제 풀이 방법의 설명을 위해서는 지식 기반의 구성이 태스크를 이용한 규칙 기반이 되어야 한다. 단순한 규칙 기반 시스템은 "왜 이 시점에서 이런 추론을 하지 않고 이런 추론을 행했는가" 등에 대한 설명을 제공하지 못한다. 태스크는 문제 풀이의 단위이기 때문에, 지식 기반의 구성이 규칙 기반이 아니라 문제 풀이의 단위인 태스크 기반일 때 문제 풀이 방법에 대한 설명을 제공해 줄 수 있다. 태스크 기반의 시스템에서 문제풀이 방법에 대한 설명은 각 태스크의 목적과 역할로부터 도출되어진다.

4.1 태스크의 정의

문제 풀이의 단위, 작업 단위를 태스크라고 한다. 각 태스크는 풀어야 할 문제를 정의하고 있다. 지식 기반을 작업 단위로 작성함으로써 일 처리 단위의 설명을 제공해 줄 수 있다(예를 들면, 일 처리의 흐름, 태스크의 선택 등).

이러한 태스크 단위는 문제풀이 지식 체계 내에서 독립적으로 처리될 수 있어야 하고 독자적인 목표와 이 목표를 위한 지식 표현과 처리 기능을 가져야 한다. 또한 태스크 단위는 전체 시스템의 목표를 달성하기 위한 하나의 구성 요소이기 때문에, 전체 지식 체계 내에 수행 시기, 수행에 필요한 자료의 구비 요건, 수행 결과에 따른 다른 표현 단위에의 영향들의 묘사가 포함되어 있어야 한다. 태스크가 이러한 기능들을 가짐으로써 문제 풀이 방법 차원의 명료한 설명의 제공이 가능하다. 태스크 기반 시스템의 수행을 위하여서 태스크에 대한 구조적, 동적, 기능적 측면에 대한 성격이 정의 내려져야 한다.

첫째, 태스크의 구조적 측면은 하나의 작업이 어떤 부분 작업들로 구성되었는가에 관련된 특징들로 정의된다. 이것은 주 태스크를 이루는 부 태스크들을 단순히 열거하는 것이 아니라 주 태스크를 이루는 부 태스크들이 어떻게 구성되는지를 기술하는 것이다. 부 태스크들의 구성 방법은 주 태스크가 성취해야 할 목표에 따라 달라진다.

둘째, 태스크의 동적 측면은 태스크의 결과가 다른 태스크에 어떤 영향을 미치는가 하는 것을 말한다. 태스크는 자신의 성공 또는 실패 및 태스크 수행 후 발생되는 결과 정보들에 따라 다른 태스크에 영향을 미친다. 다른 태스크의 활성, 비활성, 초기화, 대기화, 수행화 등 태스크 선정과 수행에 영향을 미친다.

셋째, 기능적 측면은 태스크가 활성화되었을 때 태스크 수행에 필요한 자료와 태스크에 의해 실제로 행해지는 처리를 기술한다. 전문가 시스템의 지식 표현에서 자료 구조는 전역 변수로 표현되고 이들은 태스크의 처리 규칙에 의해서 행해진다. 그러므로 태스크의 수행에 필요한 자료는 따로 기술되지 않고 제어를 위한 자료인 경우에만 기술한다. 태스크가 수행될 때 실제적인 처리는 추론 엔진에 의해서 처리된다.

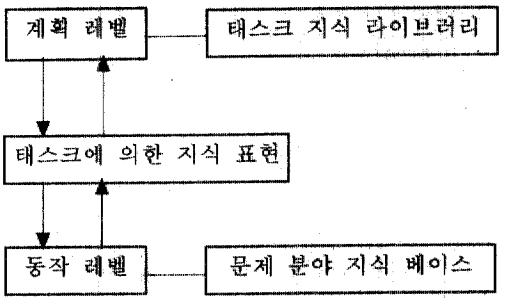
세 가지 측면에 의해 정의된 태스크는 다음과 같은 순환을 반복한다. 태스크가 기능적 측면에 따라 전체 시스템의 수행 중에 현재 성취되어야 할 것으로 선정되면, 그 태스크는 '활성' 상태가 되고 이 태스크의 부 태스크들이 있다면 구조적 측면에 따라 부 태스크들을 활성화하고 이 부 태스크의 수행을 기다리는 동안에 '대기'상태가 된다. 부 태스크들은 동적 측면에 따라 그들간의 태스크 활성화 선정에 의해 수행된다. 그리고 부 태스크의 수행이 끝난 후, 주 태스크는 부 태스크의 구성방법에 따라 수행 결과가 '성공'이나 '실패'로 결정된다. 이러한 '성공'과 '실패'의 수행 결과에 따라 다른 태스크가 활성화되면 이 태스크는 '수면'상태가 된다. 이와 같이 문제풀이 과정에서 태스크는 세 가지 측면에서 기술된 정보가 복합적으로 연결되어 활성화되고 수행된다. 이와 같은 작업은 태스크 제어 지식에 의해서 처리된다.

4.2 태스크 기반 시스템의 처리 구조

태스크를 기반으로 지식 시스템을 구축할 경우 크게 두 가지의 처리 레벨을 생각할 수 있다. 하나는 문제 분야를 처리하는 레벨이고 다른 하나는 문제 분야 지식 이외의 태스크를 관리 감독하

는 처리 레벨이다.

태스크 기반으로 이루어진 시스템은 문제분야에 대한 추론 뿐 아니라 태스크 레벨에서의 추론을 수행하여야 한다. 즉 태스크를 관리하고 감독하는 처리가 필요하다. 따라서 다음 그림과 같은 구조를 가지게 된다.



동작 레벨은 당면한 문제를 해결하기 위한 문제분야 지식 처리를 담당한다. 계획 레벨은 메타 지식을 이용하여 동작 레벨의 처리를 관리/감독한다. 계획 레벨의 처리를 담당하는 지식을 태스크 제어 지식이라 하며 이것은 태스크 지식 라이브러리를 참조하여 일을 처리한다.

4.3 태스크 기반 지식 시스템의 구성

이 논문에서 사용한 태스크들은 매우 독립적이며 다른 태스크들을 부를 수도 있고 다른 태스크들에 의해 호출될 수도 있다. 태스크의 외형적인 형태는 동일하며 태스크내의 정보만 다른 정보를 가지고 있다. 각 태스크에는 태스크의 특성을 결정지어 주는 태스크 구조 정보가 있다. MKS는 이러한 태스크 구조 정보를 보고 수행할 태스크를 선정한다.

4.3.1 시스템의 구성 요소

태스크 기반 설명 시스템은 다음과 같은 요소로 구성된다.

1) MKS(Meta knowledge source)

태스크들을 관리하는 지식이다. 언제 어떤 태스크를 수행할 것인지에 관한 지식이다. MKS는 태스크들의 구조 정보를 보고 다음에 수행할 태스크를 알아내며 태스크의 활성화, 비활성화 등을 수행한다.

2) DKS(Domain Knowledge Source)

문제 영역 지식(domain knowledge)을 위에서 제안한 태스크 방법에 의해 구축하였다. 각 태스크는 문제 영역 지식을 일 처리 단위로 나눈 것이다. 일 처리 단위(task 단위)로 문제를 나누는 것이 문제 풀이 및 설명에 있어서 가장 중요한 요소 중의 하나이다. 각각의 태스크는 풀어야 할 문제를 정의하고 있다. 각 태스크에서 정의된 문제는 작게 쪼개어진 쉽게 풀 수 있는 단위이다.

각 태스크들의 성격은 다음과 같은 속성들에 의해서 특징지어 진다. 각 태스크들은 외형적 구조는 모두 똑같지만 태스크들마다 가지고 있는 속성 정보들에 의해서 서로 다른 동작 특성을 나타낸다.

3) 태스크 정보

: name(이름)
object(목표)
주태스크(super task)
부태스크들(sub tasks)
decomposition type(태스크 분류)
precede정보(선행정보)
precondition(선행조건)
output(출력)
explanation(설명정보)

위에서 decomposition type(부 태스크들로 나누는 방법)은 AND, OR, DONE 등이 있다. AND는 부 태스크들을 모두 수행하여 모두 성공일 경우에 주 태스크가 성공이 되는 경우이고 OR는 부 태스크들 중 하나만 성공이면 주 태스크가 성공인 것이고 DONE은 부 태스크들을 수행하기만 하면 주 태스크가 성공인 것을 말한다.

4) 추론 엔진

메타 지식(MKS)과 문제분야지식(DKS)을 이용하여 추론을 수행하는 부분이다. 추론엔진은 전향추론과 후향 추론을 모두 사용하여 외부 상황 정보와 시스템 내에 있는 지식을 사용하여 새로운 정보를 발생시킨다. 추론엔진은 처음에 메타 지식을 사용하여 어떤 태스크를 선정할 것인지를 결정하는데 이것은 위에서 언급한 계획레벨의 작업이 된다. 추론엔진은 그 다음에 태스크내의 문제영역 지식을 이용하여 추론을 수행한다. 이것

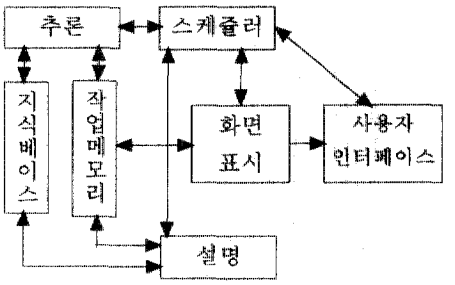
은 동작 레벨의 작업이 된다.

5) 설명생성기

설명생성기(explainer)는 위에서 언급한 MKS와 DKS를 이용하여 문제 풀이 방법을 설명하는 것으로 설명 시스템 안에 있다. 이러한 정보를 기초로 설명을 생성해 줄 설명 생성기가 따로 있어야 한다. 설명생성기는 설명내용을 문장구조로도 만들어야 한다. 현재는 자연어 처리가 안되기 때문에 간단히 몇 가지 템플릿 구조를 이용한다.

5. 제안한 설명 생성 기법

5.1. 설명 생성 구조



(그림 2) 전체 시스템 구조

태스크의 단위로 문제 분야 지식이 구성되었으며 이를 제어하는 메타 제어 지식과 규칙을 해석하고 수행하는 추론 엔진이 있고 설명 기능이 있다.

5.2 문제풀이 방법의 설명

문제 풀이 방법에 대한 질문은 크게 두 가지로 나눌 수 있다. 하나는 '왜 이러한 방법으로 문제를 푸는가?' 하는 질문이고 다른 하나는 '이 문제는 어떤 방식으로 풀 것인가?'하는 질문이다.

본 연구에서는 문제 풀이 방법 레벨에서 위의 두 가지 종류에 대한 설명을 제공하여 주는 방안에 대하여 연구하였다.

5.2.1 태스크 선정 이유에 대한 설명

태스크 기반 시스템에서 '왜 이러한 방법으로 문제를 푸는가?' 즉 '왜 이러한 태스크를 선정하였는가?'에 대한 설명은 다음과 같은 절차로 제공한다.

① 태스크 제어 지식을 참조하여 현재 태스크를 선정한 이유들을 파악한다.

② 태스크 선정을 위해 필요한 태스크의 구조적, 동적, 기능적 정보들 중에 설명을 위해 필요한 정보들만을 추출해 낸다.

③ 얻어낸 정보들을 바탕으로 이에 적당한 설명 template를 선정한다.

④ template를 해석해 가면서 설명을 생성해 낸다.

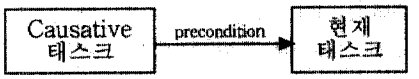
1) 태스크 제어 지식 참조

여러 태스크가 복잡하게 얽혀 있는 상황에서 앞으로 처리해야 할 태스크를 고르는 작업은 정교하게 구축된 태스크 제어 지식에 의해서 수행된다. 태스크 제어 지식은 태스크의 구조적 정보, 동적 정보, 기능적 정보들을 이용하여 수행해야 할 태스크를 선정하는 작업을 한다. 그러므로 사용자가 태스크 선정의 이유에 대한 질문을 할 경우 시스템은 태스크 제어 지식을 참조하여 설명을 발생시켜야 한다.

2) 필요 정보 추출

태스크 제어 지식은 태스크의 구조적 정보 이외에 상당히 많은 다른 정보들을 이용하기 때문에 필요한 정보만을 골라서 참조할 필요가 있다. 특히 태스크 제어를 위한 전문가의 지식들이 들어 있기 때문에 이 많은 정보가 모두 다 태스크 선정의 이유를 설명하는데 실질적으로 사용자에게 필요하지는 않다. 따라서 사용자에게 필요한 정도의 정보들만을 가지고 설명을 생성해 내게 된다. 대체적으로 사용자에게 필요한 정보로는 태스크 구조 정보들 중에 'precondition','precede', 'super task'등의 정보가 있다. 이 세 가지 정보는 태스크의 선정에 주로 사용되는 정보들이다. 'precondition'정보는 만족되는 선행 조건이 있어야 태스크가 수행되는 경우를 말하고 'precede'정보는 태스크의 순서를 정할 때 사용하는 정보이다. 'super task'정보는 주 태스크 정보로서 주 태스크를 이루기 위해서 수행되는 경우를 말한다. 세 가지 경우에 대한 태스크 선정 다이어그램을 표현하면 다음과 같다. 수평적 화살표와 수

직적 화살표와 아크(arc)형 화살표로 구별된다.



(그림 3) 선행 조건이 있는 태스크

(그림 4) 부 태스크

(그림 5) 선행 태스크

3) 설명 형태 선정

태스크 선정의 이유에 대한 필요한 정보를 얻었으면 이를 바탕으로 설명을 제공해 주기에 적당한 설명 template를 선정한다. 다음은 세 가지 경우에 대한 설명 template들이다.

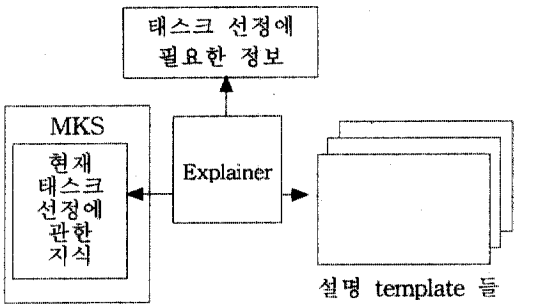
① precondition인 경우의 설명 template
<causative-task>를 수행한 결과
<precondition>이기 때문에
<current-task>를 수행한다.

② precede인 경우의 설명 template
<preceding-task>를 먼저 수행해야
<following-task>를 수행할 수 있다.

③ super task인 경우의 설명 template
<super-task>를 수행하기 위해선
<sub task1>
...
<sub taski>
를 수행하여야 한다.
이러한 것들을 수행하기 위해서
현재 <current-task>를 수행하고 있다.

4) template 해석 및 결과 출력

설명 template를 얻었으면 그 template를 해석해 가면서 설명을 생성해 낸다. 단순한 문장인 경우에는 그대로 출력하여 주면 되고 새로운 내용이 삽입되어야 하는 변수들인 경우에는 원하는 정보를 얻을 수 있는 함수를 구현하여야 한다. 위의 예에서는 '< >'로 표시된 변수들이 새로운 정보를 필요로 하는 부분들이다. 많은 경우에 있어서 태스크의 이름과 태스크들의 목적이 설명에 필요하다. 따라서 DKS에 태스크의 이름과 목적을 문장으로 명시하는 것은 태스크기반 설명 시스템에 있어서 필수적인 것이다.



(그림 6) 태스크 선정 이유 설명을 위한 처리 구조

5.2.2 태스크 수행 방법에 대한 설명

'이 문제는 어떤 방식으로 풀 것인가?' 즉 ' 이 작업을 어떤 식으로 수행할 것인가?'에 대해서는 다음과 같은 방법에 따라서 태스크 수행 방법에 대한 설명을 제공한다.

① 문제 영역 지식을 참조하여 현재의 태스크 수행에 필요한 정보들을 얻어낸다.
② 태스크 수행 방법과 관련된 태스크의 구조적, 동적, 기능적 정보들 중에 설명을 위해 필요한 정보들만을 추출해 낸다.
③ 얻어낸 정보들을 바탕으로 이에 적당한 설명 형태를 선정한다.
④ 형태를 해석해 가면서 설명을 생성해 낸다.

1) Domain Knowledge Source 참조

태스크를 수행 방법(method)에 대한 설명시에는 메타 지식인 태스크 제어 지식을 참조하기보다는 영역지식을 참조하여야 한다. 왜냐하면 태스크 제어 지식은 문제 영역 지식과는 무관하게 구성되어 있으며 단지 문제 영역 지식을 이용하여 태스크들을 선정, 활성, 비활성하는 등의 일을 하기 때문이다. 태스크의 문제 풀이 방법에 관한 지식은 DKS에 들어 있다. DKS는 태스크 형식을 이용하여 지식이 구축되어 있기 때문에 태스크 구조 정보를 통하여 어떤 식으로 작업이 수행되는지를 알 수 있다.

2) 필요 정보 추출

'어떤 식으로 이 태스크를 처리하는가?'하는 질문은 앞으로 '어떤 일들을 할 것인가?' 하는 질문과도 같은 맥락이라고 볼 수 있다. 따라서 태스크가 앞으로 수행하여야 할 일을 보여줌으로써 설명을 제공한다. 이러한 정보는 sub-task에 들어있다. sub-task들을 나열해 주는 것만으로는 좋은 설명이 되지 않을 수도 있다. 만일 sub-task들 사이에 어떤 동적인 측면에서의 관계가 있다면 이 정보들도 역시 이용하여서 설명을 생성해 주어야 할 것이다. sub-task들 사이의 precede 및 precondition 정보들을 이용하여야 하는 것이다.

만일 태스크가 말단 태스크(leaf task)이면 그 밑에 부 태스크들이 없기 때문에 이러한 태스크는 단순히 기능적 측면에서 자신이 수행해야 할 일들만을 수행하고 끝마친다. 이러한 경우 이러한 경우 태스크 수행 방법에 관한 설명은 설명 슬롯을 이용한다.

3) 설명 형태 선정

태스크 수행 방법(method)에 대해 필요한 정보를 얻었으면 이 정보에 합당한 설명 template를 선정한다. 다음은 설명 template의 예들이다.

① sub task들 정보와 sub task사이에 precede정보가 있을 때 설명 template

<current task>를 위해서

<sub task 1>를 하고

...

<sub task n>를 한다.

② sub task들 중 precondition이 만족해야 동작하는 태스크가 있는 경우 다음과 같은 template가 사용된다.

<precondition>일 경우 <sub task i>를 하고

4) 설명형태 해석 및 결과 출력

설명 template를 얻었으면 그 template를 해석해 가면서 설명을 생성해 낸다.

6. 결 론

이 논문에서는 각각의 구조가 독립적인 태스크를 이용한 전문가 시스템에서 문제 풀이 방법에 대한 설명을 제공하는 기법을 제안하였다. 문제 풀이 방법에 대한 설명은 크게 두 가지로 나누어 이에 대한 설명을 구현하였다. 하나는 "왜 이러한 작업을 수행하는가?"하는 것이고 다른 하나는 "이 작업을 어떤 방식으로 수행할 것인가?"하는 것이다. 이 두 가지 분류는 과거 Clancey의 분류와 비슷하지만 그의 설명 생성 구조는 단순히 태스크에 속한 메타 규칙을 보여주는 것이었다. 이 논문에서는 독립적인 태스크들이 선정되고 활성화되는 구조적, 동적, 기능적 측면들을 통하여서 다양한 문제 풀이 방법 레벨의 설명을 제공하여 줄 수 있었다. 설명 생성 시스템은 태스크의 선정, 활성, 비활성, 대기 등을 관리하는 태스크 제어 지식과 태스크 구조로 이루어진 영역지식을 보고 설명을 생성해 내었다. 생성된 설명은 작업의 분해에 대한 설명, 작업의 선후 관계에 대한 설명, 작업의 수행 조건 등에 대한 설명 등 작업의 선정 이유와 작업의 수행 방법에 관한 것이다. 이를 위해서는 문제 풀이 지식 이외에 설명을 위한 지식이 추가적으로 많이 필요하다. 예를 들어 상황에 맞는 설명 template들이 필요하고 규칙이나 다른 지식들을 정당화시켜 주는 심층 지식이 필요함을 알 수 있었다.

참고문헌

[1] D. A. Waterman (1986). *A Guide to Expert Systems*. Addison-Wesley.

[2] J. Durkin (1994). *Expert Systems Design And Developmen*. Macmillan.

[3] M. C. Tanner A. M. Keuneke(1991). The roles of the Task Structure and Domain Functional Models. *IEEE Expert System*. JUNE pp. 50-57.

[4] M. R. Wick (1989). J. R. Slagle, An Explanation Facility for Today's Expert Systems. *IEEE Expert Systems*. SPRING. pp. 26-36.

[5] B. Chandrasekaran, M. C. Tanner, J. R. Josephon(1989). Explaining Control Strategies in Problem Solving. *IEEE Expert Systems*. SPRING pp. 9-24.

[6] E. H. Shortliffe, B. G. Buchanan (1984). *Rule-Based Expert Systems*. Addison-Wesley.

[7] D. W. Hasling (1983). *Abstract Explanation of Strategy in a Diagnostic Consultation System*. AAAI, pp. 157-161.

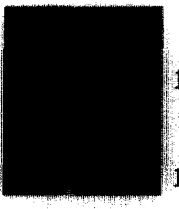
[8] W. J. Clancey (1983). The Epistemology of a Rule-Based Expert System - a Framework for Explanation. *Artificial Intelligence 20*. pp. 215-251.

9] W. R. Swartout, C. Paris, J. Moore (1991). Design for Explainable Expert Systems, *IEEE Expert Systems*. JUNE pp. 58-64.

[10] W. R. Swartout (1983). XPLAIN : a System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence 21*. pp. 285-325.

[11] J. D. Moore, W. R. Swartout (1989). A Reactive Approach to Explanation. *IJCAI*. pp. 1504-1510.

[12] B. Chandrasekaran, W. Swartout (1991). Explanations in Knowledge Systems. *IEEE Expert Systems*. JUNE pp. 47-49.



전 도 홍

1983 ~ 1985 미국 OCU (Oklahoma City University) 컴퓨터과학 학사

1985 ~ 1987 미국 Florida Institute of Technology 컴퓨터과학 석사

1987 ~ 1990 미국 Florida Institute of technology 컴퓨터교육공학 박사

1990 ~ 1994 관동대학교 조교수

1994 ~ 현재 관동대학교 부교수

1996 ~ 현재 관동대학교 중앙도서관장

E-Mail: dhjeon@kdccs.kwandong.ac.kr