

# 컴퓨터 엔지니어의 입장에서 본 전문가 시스템의 구현방법론

0  
조승호\* 이광형\*\*  
\*순천향 대학교 \*\*한국과학기술원

Methodology of Expert System  
from computer Engineer point of view

CHO Choogn-Ho\*, LEE Kwang-Hyung\*\*  
\* Univ. Soonchunhyang, \*\* KAIST

## 요 약

본 연구는 처음 전문가시스템을 만들때, 컴퓨터 엔지니어의 입장에서 현장전문가들과 대화 속에서 발생할 수 있는 어려움을 제거하기 위한 방법론을 전문가 시스템 AID [1]를 만들면서 느낀 실제 경험을 토대로 서술한다. 이를 위해 개발기간을 여러 단계로 나누어 각 단계별로 생길 수 있는 문제점과 이를 해결하기 위해 필수적으로 거쳐야할 과정과 해결 방법을 제시한다. 따라서 제시된 방법론은 전문가시스템을 만들고 있거나 앞으로 만들려고하려는 사람들에게 참고가 될 수 있을 것이다.

## I. 서론

본논문은 처음 전문가시스템(Expert System)을 만들어고 시작할때, 현장전문가(domain expert:전문가로 칭함)와 컴퓨터 엔지니어(computer enginner)의 역할을 비교하면서 이들의 대화 속에서 발생할 수 있는 어려움을 제거하기 위한 방법론을 전문가 시스템 AID [1]를 만들면서 느낀 경험을 토대로 서술해 보고자한다 우리의 경험과 느낌(reflection)이 전문가시스템을 만들고 있거나 앞으로 만들려고하려는 사람들에게 참고가 될 수 있으면하는 것이 목적이다.

AID는 원자력발전소의 전기배전판 운영 시스템의 사고발생시 전체시스템의 안전을 보장할 수 있도록 사고처리 및 예방을 위한 전문가 시스템이다. 이시스템은 원자력 발전소가 발전중 어느 부분의 갑작스런 고장이 생기거나 정기적인 수리 기간중 어떤 부분을 손질할때, 그곳에 전기를 제공해주는 배전판을 결정해주고, 그배전판 자만에 의해 발생할 수 있는 모든 위험을 신속하게 제거해 그에 따르는 영향을 최소화시키기 위해 현장 운영자(operator)에게 필요한 모든 조치들을 짧은 시간내에 주게된다.

이 시스템은 1987년 10월에 시작해서 1989년 9월에 완성되었으며, INSA de Lyon(France)의 컴퓨터 엔지니어와 NERSA(고속증식로 방식의 핵 발전소:France, Germany, Italy 합작)의 전기운영 전문가들과 이발전소의 시공을 맡은 NOVATOM(France)의 전기배선 및 안전시설 전문가등 3개팀에의해 수행되었다.

이들 3팀의 기본적인 차이는 다음과 같다:

- 컴퓨터 엔지니어 : 발전소의 전기시설 및 운영 방법을 전혀 모름.
- 현장 운영 및 시설 전문가 : 전문가 시스템의 이름과 생성규칙(production rule)에 의한 지식(knowledge) 표현의 가능성 정도 알고 있음.

이시스템을 완성시켜 가동시키기 까지 우리는 여러 단계의 분석과 대화를 했다. 이는 일의 시작과 기본 골격(form)을 만들기 위해 필요하며 지식(knowledge)을 어떻게 만드는가 하는것보다 더 중요한 일이라 생각된다. 따라서 여기에서는 이주제를 가지고 시스템 개발자의 입장에서 느낀바를 요약하겠다.

## II. 문제수집(acquisition)단계

왜 현장 전문가는 주어진 문제를 해결하기 위해 전문가시스템을 생각했느냐?

이 첫번째 질문에서 이미 전문가와 컴퓨터 엔지니어는 상당한 시각차를 보일것이다. 전문가는 전문가시스템이 무엇인지조차 잘모르고, 단지 신문이나 정보전달 매체를 통해 단편적으로 알고있다. 그는 이를 자신의 문제를 해결하기 위한 수단으로 사용하고자 한다.

또한 전문가는 자신의 문제를 AI방법을 도입하여 해결하면 막연히 더 좋은 결과가 나올 것이며, 전문가 시스템을 모든 문제를 해결해주는 요술상자처럼 생각한다. 그래서 처음에 전문가는 컴퓨터 엔지니어에게 자신의 문제에 대해 경리가 되지 않은(컴퓨터 엔지니어 입장에서 볼때 문제 의식이 분명하지 않은) 상태로 이야기를 한다

이때까지도, 컴퓨터 엔지니어는 전문영역에 대한 지식이 미미한 상태이므로, 전문가가 전문가시스템에서 처리해줄기를 바라는 내용을 알고리즘한 방법으로 설명할 수 있도록 유도해야한다. 이러한 방식으로 문제가 설명되면 전체적인 문제해결 방법에 대한 논리적인 모순이나 불일치성(incoherence)이 컴퓨터 엔지니어 입장에서 쉽게 파악이되며, 제각각 문제를 논리적으로 해결(resolve)될 수 있는가를 알 수 있게 된다.

전문가는 잠시동안 자신의 문제를 상상력을 가미시켜 광범위하게 말할 수 있으며, 이때 컴퓨터 엔지니어는 필요한 요소를 적시에 끌어내야하고, 대화의 방향을 원하는 방향으로 잘 유도해야한다.

이 단계에서 필요하다고 생각되는 대화는 다음과 같다:

\* 완전히 알고리즘한 방법이 가능하지 않고, 바람직하지도 않다는 것을 전문가에게 확인시켜야한다. 이를 위한 기술적인요소로서.

- Procedural한 방법으로 실패한 경우의 예
- 다양하고 여러가지 의미를 가진 지식에 의해 구성 되어야할 많은 양의 데이터.
- 많은 Heuristic Rule의 사용

\* 전문가가 그문제를 해결할 수 있는 다른 방법이 있다는 것을 인식시켜 주어야 한다. 컴퓨터 엔지니어는 이점에 있어서 전문가에게 분명히 AI 기술의 한계를 이해시켜야 한다.

\* 전문가의 지식(know-how)을 모델화시키는 것이 가능한 것을 보여주어야 한다. 이때 시스템을 위해 필요한 지식의 분량, 복잡성, 표현될 지식의 정확성등이 고려되어야 한다.

이 단계의 결론은 다음 단계인 가능성 분석(feasibility analyse) 단계를 계속 진행할 것인가를 결정하고, 좋은 인상을 가지고 해어지는 것이다. 보통 처음 이단계는 비교적 짧고 어떤 의미에서는 약간 촉박하게 짜여진다 따라서 어떤 결론도 아직 이른 단계이다.

III. 가능성 분석(feasibility analyse)단계

이단계에서는 재정과 인원이 구성되어 팀이 형성되었다고 가정하자. 그러면, 주어진 기간동안 이문제에 대한 세부적인 분석결과를 얻기위해 일을 추진하게 될 것이며, 이결과를 가지고 연구를 계속 할것인가에 대한 최종 결정을 내려야 한다.

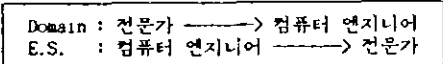
가능성 분석을 위해 구체화 시켜 나가야 할 사항은 다음과 같다:

- 구성할 전문가 시스템의 전반적인 구조
- 지식을 표현할 언어
- 지식의 형태 및 그것의 표현방법
- 추론(reasonning)의 형태

이분석을 위해 무슨 방법을 쓸것이며, 어려움은 어떤것들이 있을까?

1. 보완적인 정보(mutual information)

양자 합의에 의해 결정된 주제를 가지고 ES를 만들어 나가자면, 먼저 필요한 정보를 상호 교환해야 한다. 이를 위해 컴퓨터 엔지니어를 위해서는 전문분야(domain)를, 전문가를 위해서는 전문가 시스템을 각각 이해시켜 주어야 한다. 정보 교환의 흐름은 다음과 같이 요약 된다:



이는 물론 컴퓨터 엔지니어가 전문가로, 역으로 전문가가 컴퓨터 엔지니어로 전환되는 것은 아니며, 이 상호 정보교환은 가장 중요한 기본사항 이다.

이단계에서 전문가와 컴퓨터 엔지니어 지식베이스를 구성하고, 이를 서로 교환할 수 있도록 하는 것은

꼭 최대한의 지식을 전달해 주어야하며, 전문가는 또한 전문가 시스템의 기본기능, 지식베이스의 구성방식과 각기 다른 모듈(module)의 기능 및 역할을 이해해야한다. 이점을 소홀히 함으로서 완성된 전문가 시스템이 제구실을 못하고 사장되어 버리는 경우가 많다.

전문가에게 설명되는 전문지식은 컴퓨터 엔지니어에게는 좋은 지식베이스를 구성할 수 있을 만큼 완전하게 이해되지 못하는 경우가 종종 발생한다. 컴퓨터 엔지니어는 전문가가 어렵푸시나마 지식베이스의 구성과 시스템의 운영 메카니즘(exploitation mechanism)을 점진적으로 이해할 수 있도록 도와주어야 한다. 그렇게 함으로서 전문가도 보다 빨리 정확하게 주어진 틀(context)안에서 지식을 전달시킬 수 있다.

또한 우리가 간과할 수 없는 중요한 문제는, 전문가는 자신이 수년동안 배우고 터득한 지식(know-how)을 한낱 자신을 대신할 기계를 만들기 위해 제공해야 하는 대에서오는 문제로, 현실적으로 이지식을 전달하는 것을 무의식중에 꺼려하게 된다. 따라서 컴퓨터 엔지니어는 이문제를 잘 이해하여야 하며, 전문가 시스템의 기본기능을 전문가에게 이해시켜야 한다. 즉 전문가 시스템은 전문가를 대신하기 위해 만드는 것이아니라 전문가를 돕기위한 시스템으로서, 궁극적용 전문가는 이 시스템의 지식 베이스를 운영(control)하는 일을하게 되며, 이 지식베이스를 수정하고 보완하는 일로서 시스템 입장에서 가장 중요한 역할을 하는 사람으로 남아있게 된다.

따라서 이러한 정보는 미래시스템의 전반적인 운영 계획(global plan)을 정리할때 고려되어야 할 점이다.

2. 실제상황의 연구

앞에서 우리는 지식추출에 있어서의 어려움을 강조하였다. 맨처음 단계로 실제문제의 추론방법(reasonning)을 만들어 수 있어야 한다. 전문가는 종종 너무나 남연하다고 생각되는 단계(step)를 건너 뛰게 될 때가있다. 반면 이단계는 컴퓨터 엔지니어가 추론 메카니즘을 구성하는데 있어서는 꼭 이해되어야 할경우가 있다 또한 자주 전문가는 자신이 상식적으로 알아서 처리하는 추론단계를 인식하지 못하는 경우가 있다. 이 경우는 간단한 예를 만들어 문제를 해결 해보고, 되도록 피상적으로 문제를 분석함으로써 야기될 수 있는 잘못을 피해야 한다 어떤경우에는 현재 접근방식으로 추론을 해낼 수 없는 경우가 나타난다 따라서 이 경우는 현재의 방법으로는 실현가능하지 않다는 결론에 이르게 된다

여러가지 예가 만들어져 약간 상세(detail)하면서도, 어느정도의 복잡성(complexity)을 갖는 시스템의 장래의 기능(functionality)을 손으로 모의실험(simulation)할 수 있어야 한다. 이실험은 처음 손으로(manually) 만들었던 지식 베이스에 의해 이루어 져야 한다. 이는 완벽하지 않을 뿐 아니라, 어떠한 컴퓨터의 기술도 이용되지 않는다. 다만 표현해야 할 여러 지식 형태를 나타낼 수 있는가 하는 것이 목적이다. 따라서 이점은 오로지 이 지식 베이스안으로 전문가의 추론과정을 만들어 내는 것이 가능하다.

확인된 결합이나 실수가 발견되면 이를 처리할 수 있어야 한다. 이를 위해 복잡한 예제가 다루어질 수 있다. 이경우 생성규칙(production rule)에 의한 지식 표현과 추론과정을 이용한다.

생성 규칙의 구조는 전문가에게 대단히 쉽게 이해되어진다. 이는 일반적으로 그 지식(know-how)을 표현하는 자신의 방법과 어떤경우에는 일치한다. 이는 다음의 형태로 요약 된다.

```

IF <조건> THEN <결론>
혹은
IF <조건> THEN <결론>
IFESLE <결론>
    
```

혹은

<사실> IF NOT <조건>

이 경우 기본적으로 종이 위에 지식을 만드는 것이기 때문에 어떤 제약이 가해진 구조가 강요되지는 안된다. 만약 전문가가 자신의 추론을 달리 표현할 방법을 찾지 못할 때는 여러다른 방법을 시도해 보아라. 만약 그 지식의 표현이 컴퓨터의 처리 방식에 부합되지 않을지라도 일단 받아 드려야 한다.

이를 위해 2 단계로 나누어 처리하는 것이 중요하다:

첫째, 전문가에 의한 지식의 표현. 완전하고 유용한 지식베이스를 얻기 위한 가능성을 최대화시키려면 대단히 자유롭게 표현되어야 한다.

둘째, 컴퓨터에 적용할 수 있는 구조 안에서 지식 표현.

따라서 처음 이 생성규칙의 형태 안에서 지식 표현은 이형태로 최종 지식베이스 안에서 사용됨을 의미하지는 않는다.

비로소 수작업한 데이터와 추론기능이 만족한 상태에 이르렀을 때 구현하고자하는 시스템의 전체적인 구조와 지식표현을 위한 언어를 제안할 수 있다.

예를 들어 시스템 단계에서, 생성규칙에 의한 표현을 유지하도록 하는 결정을 할 수 있다면, 이 경우는 지식을 표현하는 언어와 시스템의 지식베이스가 거의 일치할 것이다. 그러나 그반대의 경우는 그데이터를 위해 다른 표현 방법을 사용 해야하기 때문에 보다 복잡한 번역 시스템이 개발 되어야 할 것이다.

이장에서의 일은 전적으로 컴퓨터 엔지니어가 책임을 가지고 있다. 전문가에 의해 취급되고 조작되어질 지식을 표현할 언어의 선택은 특히 중요한 일이다. 또한 최종 사용자의 Interface문제도 이단계에서 고려되어야 할 점이다.

여기에 이르러 가능성 분석이 내려질 수 있다. 만약 가능하다면, Prototype구현을 시작할 수 있다.

IV. 프로토타입(prototype) 구현

이 단계를 위해 여러다른 Actions들을 크게 3 단계로 묶어서 생각할 수 있다:

- 합의된 언어로 지식베이스의 표현
- 컴퓨터 엔지니어에 의한 Prototype 구현
- 실행 결과 분석

1. 지식베이스의 표현

이 단계에서는 합의된 언어의 표현능력의 범위 안에서 전문가에 의해 자유스런 형태로 표현된 지식을 그대로 옮겨쓰는 단계로 전문가에 의해 많이 주도되어진다 아직 지식의 입력작업을 도와주는 모듈(entry module)이 없기 때문에 수작업으로 입력을 한다. 이 모듈 구현은 컴퓨터 엔지니어에 의해 Prototype의 구현과 동시에 진행된다.

이때 지식을 베이스로 옮기는데 있어서 그 지식이 완전해질 때까지 기다릴 필요가 없다. 이는 가능성 분석 단계중에 표현된 지식의 목적어, 이들사이의 관계나 다양하게 다루어져야할 Objects들을 설명하기 위한 것이기 때문이다. 이때 베이스로 옮기는 작업은 합의된 언어의 틀 안에서 추기될 지식을 설명할 직접적인 좋은 기회가 된다. 이 과정은 이론적으로는 간단한 논리적인 질문이지만 전문가는 실제적으로 이과정을 수행하는데 상당한 어려움에 봉착하게 되므로 전문가는 이를 정확히 수행할 수 있도록 준비해야 한다.

따라서 여러가지 규칙(rules)들이 합의된 언어의 틀 안에서 다시 쓰여져야 한다. 예를 들어 표현된 규칙이 전문가가 생각했던 반대 의미로 쓰여지기도하며, 논리적으로 차후 전문가가 이 규칙을 수정할 필요를 느끼지 못하게도 한다. 더욱 전문가를 혼돈시키는 것은

그 자신이 설명하고자 했던 것과 전혀 같지않은 표현이다. 이러한 현상은 규칙의 초기형태가 불완전했음을 의미한다.

이과정은 전문가에게 다음 3가지의 일을 부여한다:

- 지식의 정확한 표현
- 지식사이의 불일치성(Incoherence) 제거
- 합의된 언어에 친밀해침

이는 다음 3절에서 결과 분석 단계를 쉽게 해줄 수 있는 가장 좋은 준비가 된다

2 Prototype 구현

이과정에서는 일의 진행 권한이 컴퓨터 엔지니어에게 주어지며, 컴퓨터 엔지니어는 선택된 프로그래밍언어를 가지고 시스템을 구성하는 여러 모듈을 작성한다. 각각의 모듈은 운영 매카니즘(exploitation mechanism)과 지식 베이스 뿐만 아니라 전문가(혹은 컴퓨터엔지니어)의 지식입력을 돕기 위한 Interface 기능과 전문가나 최종 사용자(final users)의 시스템 사용을 돕기 위한 Interface들을 포함 해야한다. 따라서 전문가가 시스템과의 대화를 가능하게 해주는 것을 이 단계에서 중요하게 다루어져야 할 것이다. 즉 전문가는 지식 표현에 있어서 Object의 구조, 사용될 수 있는 어휘 및 단어, 취급할 수 있는 값들(values)을 알고 있어야 한다. 특히 지식을 표현할 때 전문가가 이 구조를 머릿속에 가지고 있다면 상당히 효과적으로 일을 할 수 있다.

이과정에서의 많은 어려운 부분은 전문가가 지식을 잘 입력할 수 있도록 Interface Module을 얼마나 잘 만들어 주느냐에 따라 해결될 수 있다.

3 실행 결과 분석

이과정은 먼저 전문가가 지식베이스를 얼마나 빠르게 입력할 수 있는가하는 것이 문제인데, 이는 입력 모듈을 통해서 처리되고, 이 입력된 지식은 논리성(logic)과 유효성(variable)이 검증되어야 한다.

논리성의 검증과정은 지식베이스가 가지고있는 사실명제(facts)들중 서로 어떤 상호관계가 있는 사실들을 모아 각 부분별로 그들의 관계가 만족하고 있는가를 검증하는 과정이다. 예를들어 어떤 사실들이 Tree 구조로 구성될때 Root가 아닌 Node(사실)들은 그의 조상(parent node)을 추적해 보면 언젠가는 그의 Root에 도달하게 되며 만약 그렇지않다면 이 사실은 그들의 관계를 만족하지 않다고 말할 수 있다.

이 유효성 검증과정은 전문가가 컴퓨터 엔지니어가 실제 상황에서 여러경우로 나타날 수 있는 데이터를 가지고 질문과 대답을 반복해서 모의 실험을 해봄으로써 가능하다.

AID 개발에서는 이 시스템이 잘 진행되었다. 즉 비교적 여러개의 데이터 입력및 검증 Interface 모듈들과 이 모듈들의 처리순서를 정해줌으로서 여러가지 일을 동시에 처리해줄 수 있었고, 전문가와 그의 보조들이 비교적 단시간 내에 그들이 경험하지 못한 일에 익숙해졌다. 한편으로는 약간의 프로그래밍상의 오류나 실수로 바트된 데이터도 있었지만 이과정을 수행하는데 있어서 결정적으로 영향을 미치는 것은 못되었다.

이 과정에서 전문가는 컴퓨터에 대한 약간의 지식을 습득하게 된다.

여기에서의 goal은 AID개발에 있어서의 특별한 경우임을 밝힌다.

V. 시험 및 실제상황하에서의 수정

앞의 Prototype과정에서, 전문가에 의해 제시된 실제 상황에 대한 검증을 해보아야하는데 이는 다음을 위한

것이다

- 지식베이스를 지속적으로 실행해봄으로써 전문가 시스템의 성능 향상
- 시스템의 Interface나 출력(output: 특히 그래픽 출력이 필요한 경우에도 포함)의 향상.

이과정은 2단계로 구분할 수 있다.

- 전문가가 이미 대답을 알고 있는 실제상황에 대한 시험으로서, 현재 구성된 지식베이스를 가지고 전문가 시스템이 정확히 작동하여 예상되는 답이 정확히 추론되어 지는가를 검증하는 단계이다. 이는 시스템이 최소한으로 주어야 할 최소한의 결과이다.
- 전문가가 아직 연구해보지 않은 상황에 대한 시험으로서, 다른 전문가들에 의해 시스템의 성능이 평가되어 진다. 또한 그들은 이미 얻어진 답에 대하여 질문을 할 수도 있다.

각 단계에서, 검증은 무엇보다도 먼저 전문가의 처리능력 안에서 수행되어야 하며, 바람직한 결과를 유도해내기 위해서 베이스가 수정되어야 할 수 있다.

그러나 지금까지는 한꺼번에 모든 데이터를 입력할 수 없었기 때문에 부분적으로 지식베이스를 추가하고 이를 검증하여 실제 지식베이스로 확장하였다. 그러나 이들 지식들은 부분적으로는 검증이 되었지만 시스템 전체 입장에서도 다시 검증이 이루어져야 한다.

이단계는 대단한 주의를 가지고 수행해야 하는 가장 힘든 단계중의 하나이다. 이를 검증하기 위해, 여러가지 실제상황을 복합시켜 문제를 제기해 보아야 한다. 이때 자주 기대한 답을 얻을 수 없거나, 기대하지 않은 답을 얻게될 수도 있다. 실제로 이 지식베이스를 수정하기 전에 우리는 어떤 어떤 경우에 시스템이 제기들을 발휘하지 못하는가를 찾아야 한다. 그러나 방대한 분량의 전체 지식베이스를 대상으로 어디가 잘못되었는가를 찾는 일 또한 쉽지 않다.

이는 지식이 생성규칙에 의해 표현될 때 구문상 강점(strong syntactic)이 있으나 의미상 약점(weak semantic)이 생겨 많은 규칙이 한꺼번에 같이 사용될 때 이들을 논리적으로 정당화할 증명하는 것이 쉬운 일이 아니기 때문이다.

특히 전문가에게 쉬운 문제가 컴퓨터 위에서 처리될 때는, 그 추론과정의 프렌이 대단히 복잡해질 수 있다. 이런 문제에대한 질문은 대단히 광범위하게 해석 되어질 수 있고 가능한 여러가지 다양한 답을 가지게되므로 신중하게 처리되어야 한다. 왜냐하면 사용자는 이것을 증명하기 위해 반드시 여러가지 다양한 질문을 하게 될것이다.

또한 이단계에서, 최종 사용자를 편리하게하기 위한 Interface의 형태나 (종이나 화면위에)출력된 답들의 구성형태가 보다 이해가 쉽도록 약간의 수정 보완이 있어야 한다.

전문가에 의해 제안된 정확한 실제 상황(real cases)을 시험하여 결과를 분석해서, 우리는 이를 토대로 시스템의 성능평가를 내릴 수 있다.

## VI. 결론

컴퓨터 엔지니어로써 가장 어려웠던 과정은 처음에 전문가들로부터 완전한 지식을 얻어내는 과정이었다.

특히 높은 안전성을 유지해야하는 원자력 발전소의 경우, 그들은 그들의 안전을 기계에 맡기고 싶어하지 않았다. 이러한 이유는 컴퓨터 자체에 대한 불신보다는, 발전소 자체에 대한 불신이 더 있었다. 왜냐하면, 그 발전소가 가지고 있는 수많은 (약 20000개) 전기적 작동기들(Motor, pompe, pan, 냉각기, 등)들의상

호 관계를 모델화시키고, 이들의 운영 메카니즘을 지식화시키는 것에 대한 확신이 부족했었다.

이 방법론은 컴퓨터 엔지니어와 전문가사이에 지식의 공통점이 거의 없이 시작하여 한 시스템을 만들기 위해서는 그들사이에 존재하는 문제 해결을 위한 많은 대화의 방법을 제시하고 있다.

또한 미래에 전문가 자신이 이 시스템을 운영할 때를 위해 지식 베이스를 수정하거나 추가할 때 간편성과 다른 사용자가 이 시스템을 이용할 때 사용의 편의성을 위해 사용하기 쉬운 Interface 모듈이 필요함을 보았다.

따라서 이 연구는 컴퓨터 엔지니어가 전문가시스템을 만들때 생길 수 있는 여러가지 다양한 문제를 전문가와 함께 단계적으로 해결 방법을 제시하고 있다.

## > 참고 문헌 <

- [1] CHO C. H., BAPTISTE P., FAVREL J., DOREY R., D UPONT D "AID :Système expert d'aide à la coupure de tableau électriques pour la Centrale Nucléaire de Creys-Malville (Decision Making Expert System to shut down superphenix nuclear power plant electronic supplies for maintenance during outage)", 1990, janvier, Convention Intelligence Artificielle 90, Ed HERMIS PARIS, p729-744
- [2] CHO C.H Technical note for Expert System AID "Rapport de Synthèse pour le Système Expert d'Aide à la Mise Hors Tension de Tableaux Electriques pour la Centrale Nucléaire de CREYS-MALVILLE, Creys-Malville (Fr.): Société NERSA, 1989,feb 140p.