

# 저장조 설계 지원 전문가시스템

조향덕\*·조근식\*\*

## Expert System for Supporting the Design of Storage Tanks

Huang Duk Jo\*·Geun Sik Jo\*\*

### 요 약

본 논문에서는 저장조 설계에 필요한 설계용 입력데이터의 자동생성을 위한 지식베이스 구축과 그래픽 사용자 인터페이스를 개발하였다. 지식베이스는 전문가의 오랜 경험을 통해 축적된 지식을 전문가의 도움을 받아 객체지향 모델링 기법들을 이용하여 구축하였으며, 추론기법은 전문가시스템 내에 전향추론 기법과 후향추론 기법을 모두 갖고서 사용자가 사용 목적에 따라서 추론기법을 선택할 수 있도록 하는 혼합형추론 기법을 사용하였다. 또한 사용자와 지식베이스간의 인터페이스와 사용자가 쉽게 시스템을 사용할 수 있게 하기 위하여 윈도우 환경에서 그래픽 사용자 인터페이스를 개발하였다. 이렇게 개발된 전문가시스템을 과거 수행했던 저장조 설계 프로젝트에 적용해 봄으로써, 비전문가도 전문가의 도움 없이 저장조를 설계할 수 있었으며, 표준화된 절차에 의하여 설계함으로써 설계상의 오류를 줄일 수 있었고, 약 88%의 비용을 감소시킬 수 있었다.

## 1. 서 론

일반적으로 저장조(Storage Tank) 초기 설계 단계에서는 설계자의 과거 경험에 의존하여 설계자의 수작업으로 설계용 입력데이터를 작성하

고 설계도면을 설계자의 육안으로 분석하였다.

그러나, 설계자가 직접 수작업으로 설계용 입력데이터를 작성하고 육안으로 분석하는 과정의 한계성 때문에 시간적인 효율성과 정확성이 결여되고 있는 문제점을 낳고 있으며, 각 설계단계에서는 설계작업의 복잡성과 다양성 그리고

\*Netting 연구원

\*\*인하대학교 전자계산공학과 부교수

기존 프로그램의 한계성으로 인하여 타분야와 긴밀한 협조가 이루어지지 못하고 이중작업을 하는 불편함이 있다. 또한 저장조 설계를 위한 전문가의 부족과 설계과정에서의 전문적인 지식이 체계화 되어 있지 못함으로 인하여 합리적인 설계가 이루어지지 못하고 있다.

그러나 근래에 와서는 이러한 문제점을 해결하기 위해 여러 전문가들과 연구진에 의하여 새로운 방법을 도모하고 있으며, 구조물의 설계를 진행해 나가는 과정에서 가장 효율적인 대안을 찾기 위한 방법을 얻고자 많은 노력을 하고 있다. 외국의 경우 여러 연구자에 의해 수많은 연구발표를 하고 있으며, 엔지니어링 디자인 데이터 유출을 자동적으로 관찰하고 평가할 수 있는 메커니즘과 공학에서 데이터의 교환, 분배는 통합에서 중요한 역할을 한다고 기술함으로써 일괄구조시스템의 설계에 있어서 중요성을 말하고 있다.

국내에서도 한국전문가시스템학회의 창립과 본격적인 전문가시스템의 사용화와 더불어 구조물에 대한 설계분야에서도 전문가시스템에 관한 연구가 활발히 진행되고 있다[12]

본 연구에서는 전문가의 경험적인 지식을 바탕으로 설계시방서 및 각종 연구자료들을 통해 수집한 저장조 설계를 위한 문제해결 방법을 객체지향 모델링 기법을 이용하여 지식으로 모델링한 후 전문가시스템(Expert System)으로 개발하였다.

이 전문가시스템을 이용함으로써 ①저장조 설계를 위한 도면설계용 입력데이터 작성에 소모되는 시간과 인력, 비용을 감소시키고, ②표준화된 설계절차에 의하여 정확한 입력데이터를 사용함으로써 구조물의 안정성을 도모하고, ③비슷한 유형의 프로젝트에 대해서는 과거 프로젝트 수행시 사용한 지식베이스를 이용할 수 있으

며, ④제한된 범위에서나마 비전문가도 전문가의 도움 없이 전문가와 같은 수준의 설계를 수행할 수 있고, ⑤도면이나 B/M 등을 자동으로 생성하여 엔지니어링 데이터베이스(EDB)를 구축하고, ⑥저장조 설계 전과정을 통합하는데 이 연구의 목적이 있다.

## II. 설계정보의 모델링 및 추론

저장조 설계를 위한 정보를 추출하는 과정에서 외부환경(DBMS, CAD등)과의 연계를 위해서는 효율적인 데이터모델링이 필요하며 사실의 객체모델링과 이들의 상호 관계성을 나타내는 지식모델링이 선행되어야 한다.

본 연구에서는 먼저 여러 종류의 저장조를 설계할 경우 필요한 구성요소들을 분석하고 해결해야 할 문제점들을 파악하여 데이터모델링을 하였다. 다음 단계에서는 데이터모델링된 저장조 구성요소들을 객체로 표현하였으며, 저장조 설계에 대한 정보를 조직화하기 위해서 전문가의 경험과 설계기준의 제반사항 등을 유효적절하게 지식으로 표현하였다. 또한 데이터, 객체, 지식의 표현방법으로는 객체지향 모델링 기법을 이용하였다.

### 2.1 문제영역

저장조는 주로 원료, 중간제품, 완제품, 부대설비의 물질을 기체나 액체로 대기압 또는 15PSI 이하의 상태로 저장하기 위하여 쓰이는 대형 용기로, 대개 원료유 저장조, 연료유 저장조 등등의 저장 유체명을 붙여 부르는 경우가 많다.

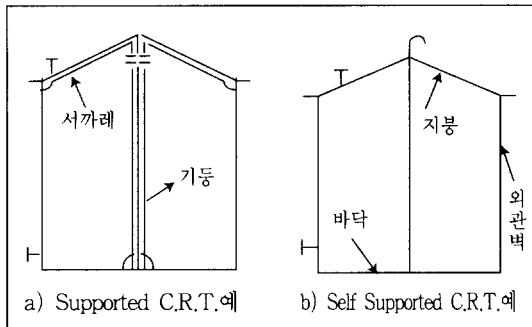
### 2.1.1 저장조 종류

저장조는 저장목적 및 저장조 형태에 따라 Cone Roof Tank (C.R.T.), Dome Roof Tank & Umbrella Tank, Floating Roof Tank (F.R.T.), Covered Floating Roof Tank (C.F.R.T.), Spherical Tank (Ball Tank), Gas Holder 등 다양한 종류로 구분된다[4].

본 연구에서는 가장 일반적으로 많이 사용되는 Cone Roof Tank (C.R.T.)의 일부와 Floating Roof Tank (F.R.T.) 두 종류의 저장조만을 다루었으며 종류별 자세한 특징은 아래와 같다.

#### 1) Cone Roof Tank (C.R.T.)

Cone Roof Tank는 유지관리가 쉽고 경제적이기 때문에 저장조의 대부분을 이루고 있으며 오일이나 화학용액 등의 저장을 위하여 많이 사용된다.



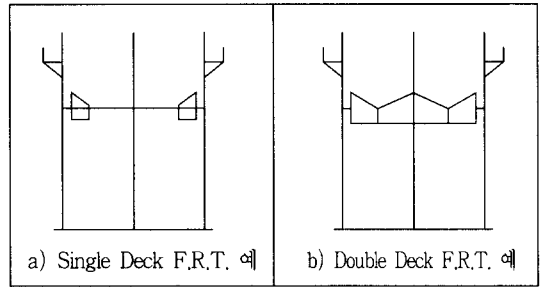
[그림 1] C.R.T. 종류의 예

Roof의 형상이 콘(Cone)을 이루고 있으며 크게 Supported Cone Roof Tank와 Self Supported Cone Roof Tank 두 종류로 구분된다.

#### 2) Floating Roof Tank (F.R.T.)

나프타나 가솔린과 같은 휘발성이 큰 유체의

증발을 최소로 하기 위해 고안된 저장조이다. 저장유체의 증감에 따라 액면에 떠있는 지붕이 상하로 움직이는 저장조이며, Floating Roof의 형상에 따라 Single Deck (Pontoon or Annular) Type 과 Double Deck Type으로 구분된다.



[그림 2] F.R.T. 종류의 예

### 2.1.2 간섭체크(Interference Check)

저장조의 각 구성요소(외관벽, Annular, 바닥, 지붕, Deck)는 크기가 방대하여 여러 장의 조각판들의 배열로써 이루어진다. 때문에 설계를 할 경우에는 저장조의 견고성 유지를 위해 각별히 유의하여야 한다.

본 연구에서는 견고성 유지를 위한 방법으로 저장조에 사용되는 구조재료를 API 650 코드집에 정의되어 있는 규격에 적합한 것, 또는 이와 동등 이상의 기계적 성질과 화학성분을 가진 것을 이용하였다. 또한 조각판과 조각판 또는 조각판과 노즐구멍 또는 배수구간에 최소거리 이상을 유지하는지 체크하는 '간섭체크'를 하였으며, 아래와 같은 여러 종류의 간섭체크를 위한 조건들을 규칙으로 표현하여 지식베이스화 하였다.

1) 외관벽 조각판들과 Annular 조각판들 또는 바닥 조각판들간의 간섭체크

2) 노즐구멍과 외관벽 조각판들 또는 바닥 조

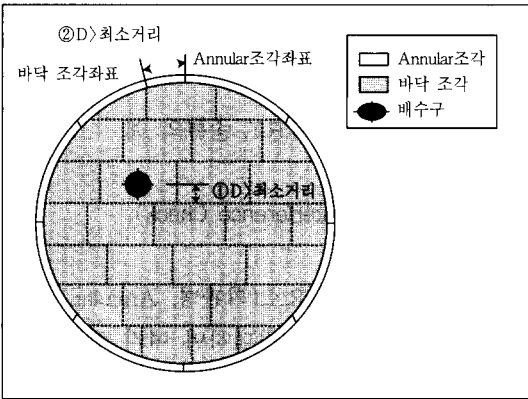
각판들간의 간섭체크

3) Annular 조각판들과 바닥 조각판들간의 간섭체크

4) 바닥 조각판들과 배수구간의 간섭체크

5) 외관벽 조각판들과 Deck 조각판들 또는 지붕 조각판들간의 간섭체크

[그림 3]은 여러 간섭체크 중 바닥의 간섭체크 예제이며, ①번은 Annular 조각판들과 바닥 조각판들 간의 간섭체크 조건이고 ②번은 바닥 조각판들과 배수구 간의 간섭체크 조건을 보여준다.



[그림 3] 바닥의 간섭체크

2.2 설계정보의 데이터모델링

구조해석과 설계에 있어서 가장 중요한 요소들 중의 한가지는 효율적인 데이터 관리이다. 그러나 최근에 강조되는 엔지니어링 데이터베이스(EDB)는 설계과정에서 여러 관점으로 추론되며, 계속적으로 변화하는 특징이 있으므로 데이터 관리에 상당한 어려움이 있다.

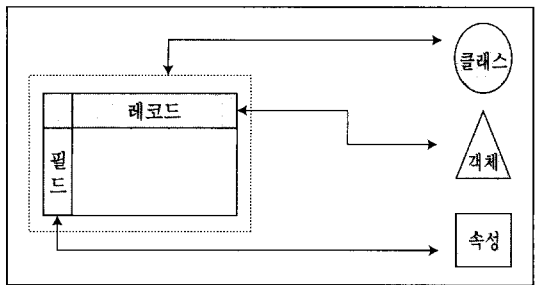
그러므로 프로젝트 진행과정을 통해 각 단계에서 대량의 이질성 데이터들을 생성, 유지, 교환하기 위한 데이터모델링은 필연적이다.

2.2.1 객체지향 데이터모델링 기법

실세계의 데이터들을 가장 유사하게 표현하는 객체지향 데이터베이스는 뛰어난 장점을 가지고 있으나 현재로서는 연구적인 단계에 머물러 있고, 계속적으로 관계형 데이터베이스에서 발전된 형태의 객체지향의 형태로 나가고 있어 아직은 완전한 객체지향 데이터베이스가 현실화되기에는 이른 감이 있다.

그러므로, 본 연구에서는 [그림 4]와 같이 데이터모델링은 객체지향으로 기법으로 설계하였으며, 데이터베이스는 객체지향 데이터베이스를 사용하지 못하고 전문가시스템 도구와의 호환성 문제를 고려하여 DBF화일과 평면파일(Flat-File)형태인 NXPDB를 사용하였다.

또한 데이터베이스 테이블로부터 데이터를 검색하고 필요한 정보들만을 얻기 위하여 SQL (Structured Query Language)문을 사용하였다[4].



[그림 4] 객체지향 데이터모델링 기법

[그림 4]는 객체지향 모델링 기법의 예제로써 데이터베이스 테이블은 클래스로 레코드는 객체로 필드는 속성으로 표현된 예이다.

2.2.2 설계 관련 데이터

저장소 설계와 관련하여 고려되어야 할 주위

환경데이터와 기본 설계데이터들은 아래와 같다 [4].

- 1) 주위 환경데이터
  - 최대풍속
  - 지진계수
  - 최대강우속도
  - 최고, 최저대기온도
  - 토질조사결과
  - Live Load (눈무게 포함)
  
- 2) 기본 설계데이터
  - 기기번호
  - 공칭용량
  - 저장 서비스 및 화학적 특성
  - 저장액의 충전속도 및 배출속도
  - 내구년한 (통상 20년)
  - 저장조 종류
  - 부식허용치
  - 운전조건(압력 및 온도)
  - 보온여부 및 사양
  
- 재질 및 배관재 규격 관련 적용코드
  - 공기(Target Delivery)
  - 페인트사양
  - Spare Parts 요구사항
  - 저장조 지름

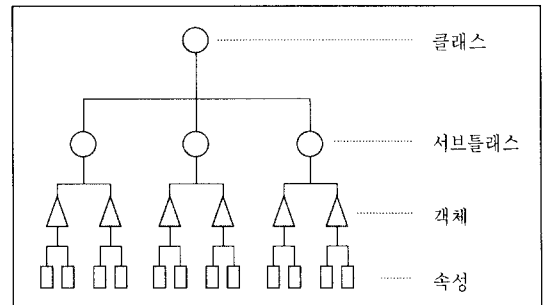
### 2.3 설계정보의 객체모델링

객체모델링은 저장조 설계에 필요한 데이터를 체계적으로 표현하여 설계정보에 대한 이해를 논리적이고 용이하게 한다. 따라서 시스템을 구축하기 위한 지식도 중요하겠지만 이들의 상호 관계를 연관시키기 위해서는 우선 객체모델링이 선행되어야 한다.

### 2.3.1 객체, 클래스, 속성의 정의

본 연구에서는 저장조 설계를 위한 지식모델링에 앞서 각 모델링에서 사용되어질 정보들을 객체지향 모델링 기법을 이용하여 객체(Object), 클래스(Class), 속성(Property)들로 구분하여 정의하였다.

객체는 정보의 가장 작은 단위로 지식을 표현 하는데 있어서 어떠한 정보의 단어를 설명하기 위한 클래스의 예(Instance)로 사용하였고, 서로 공통의 의미와 속성을 가진 객체를 조직화하고 객체들에 대한 관리를 쉽게 하기 위해 클래스로 묶어 정의하였다.



[그림 5] 클래스, 객체, 속성의 계층구조도

[그림 5]는 클래스, 객체, 속성의 계층구조도로 하나의 서브클래스는 상위 클래스의 데이터 구조와 오퍼레이션을 상속하게 된다. 단일상속은 하나의 클래스가 하나의 상위 클래스로부터 데이터 구조와 오퍼레이션을 상속 받고, 다중상속은 하나의 클래스가 하나 이상의 상위 클래스로부터 데이터 구조와 오퍼레이션을 상속 받는다. 상속되어지는 것은 속성에 대한 정의, 속성 값, 메타슬롯과 속성의 데이터 타당성(Data Validation) 등이 있다. 상속의 방향은 어떤 전략을 가지고 있느냐에 따라 상향, 혹은, 역방향일 수 있다.

```

-클래스

(@CLASS= Annular
  (@PUBLICPROPS=
    E_Angle
    Increment
    Plate_Name
    Plate_No
    S_Angle
  )
)
    
```

```

-객체

(@OBJECT= CD_Bottom_Plate_E_Angle
  (@PUBLICPROPS=
    Value @TYPE=Float:
  )
)
(@OBJECT= CD_Bottom_Plate_S_Angle
  (@PUBLICPROPS=
    Value @TYPE=Float:
  )
)
    
```

```

-속성

(@PROPERTY= Alpha @TYPE=Float:)
(@PROPERTY= B @TYPE=Float:)
(@PROPERTY= Beta @TYPE=Float:)
(@PROPERTY= BTM_Angle@TYPE=Float:)
(@PROPERTY= C_Arc @TYPE=Float:)
(@PROPERTY= C_Chld @TYPE=Float:)
(@PROPERTY= Cha @TYPE=Float:)
(@PROPERTY= D_Arc @TYPE=Float:)
(@PROPERTY= D_Chld @TYPE=Float:)
(@PROPERTY= E_Angle @TYPE=Float:)
(@PROPERTY= End_Position@TYPE=Float:)
(@PROPERTY= Find @TYPE=Integer:)
(@PROPERTY= Increment @TYPE=Float:)
    
```

### 2.3.2 객체, 클래스, 속성의 표현

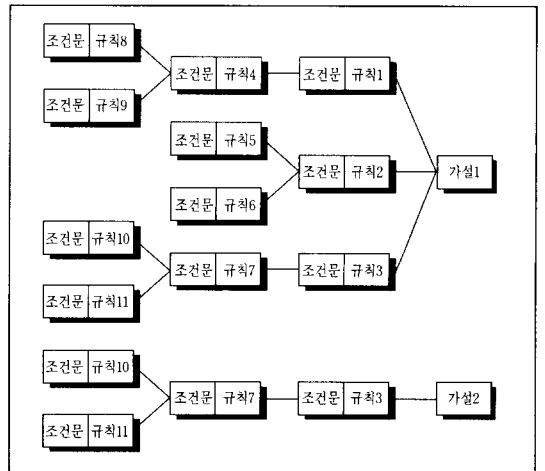
본 연구에서는 객체, 클래스, 속성을 표현하기 위해 먼저 문제해결에 관련된 지식들을 전문가의 도움을 받아 조사, 분석하여 각 지식들을 클래스, 객체 및 속성 등으로 구분하여 분류하였다.

분류된 내용을 바탕으로 전문가시스템 구현 도구에서 제공하는 클래스용 에디터를 이용하여 클래스 및 서브클래스를 생성하고, 각 클래스 및 서브클래스에 속한 속성을 정의하였다.

전문가시스템 도구에서 제공되는 클래스용 에디터 및 객체용 에디터를 이용하여 클래스, 객체를 생성하고 클래스와 객체에 대한 속성들을 정의하면, 그 내용들이 아래와 같이 지식베이스에 저장된다.

### 2.4 설계정보의 지식모델링

지식모델링은 객체모델링단계에서 생성된 객체들간의 상호관계로 전문가의 경험과 지식을 표현하는 것을 의미한다.



[그림 6] 가설을 이용한 규칙의 조합

본 연구에서는 지식모델링단계에서 지식을 표현하기 위한 방법으로 가장 기본적인 추론의 단위인 규칙(Rule)을 사용하였으며, 기억장소의 사용을 최소화하고, 추론속도를 빠르게 하며, 효율적인 지식의 관리를 위해 규칙들을 지식베이스라는 특정영역에 단위 기능별로 분류하여 독립적으로 구축하였다.

### 2.4.1 규칙의 분류

본 연구의 지식모델링 단계에서 생성된 규칙의 수는 총 682개이며 추론을 위해, 생성된 규칙을 조합하여 총 59개의 가설로 설정하였다.

[그림 6]에서 보듯이 규칙은 조건(Condition)문으로 구성되며, 이미 정의된 규칙들을 조합하여 가설(Hypothesis)을 설정하며 가설에서는 이미 다른 가설에서 사용된 규칙을 별도의 정의 없이 중복하여 사용할 수 있다.

<표 1> 저장조 정렬을 위한 지식의 분류

기능	지식베이스명	규칙수
외곽벽 정렬	Shell_1.KB	38
	Shell_2.KB	61
Annular	Annular.KB	25
바닥 정렬	Btm_60.KB	8
	Btm62.KB	105
	Btm63.KB	14
지붕 정렬	Roof_54.KB	97
	R_Ifc1.KB	37
	R_Ifc22.KB	16
Single Deck 정렬	Sdeck_1.KB	101
	Sdeck_2.KB	16
	Sdeck_3.KB	25
Double Deck 정렬	Deck_1.KB	104
	Deck_2.KB	8
	Deck_3.KB	27

### 2.4.2 규칙의 표현

본 연구에서는 규칙의 표현을 위해 전문가시스템 도구에서 제공하는 규칙용 에디터를 사용하였다. 규칙용 에디터는 객체모델링단계에서 생성한 클래스, 객체를 바탕으로 추론을 수행하여 원하는 최종 결과를 얻을 수 있도록 하며, 규칙의 작성을 용이하게 해준다. 규칙용 에디터를 이용하여 작성된 모든 규칙(Rule)들은 아래와 같은 특징들을 가지고 있다.

```

(@RULE= Rule_CleanOutDoor0
  @INFCAT=200:

  (@LHS=
    (= (SH_ARR.TYPE)
      ("EQUAL"))
    (Assign(FIND_FIRST_PLATE)
      (FIND_FIRST_PLATE))
  )
  (@HYPO= H2)

  (@RHS=
    (Assign((\CC7.NAME2\E_Angle-
      \CC7.NAME2\S_Angle)/2) (|Annular|,Increment))
    (Assign(<|Annular|>,S_Angle+
      |Annular|,Increment)
      (<|Annular|>,S_Angle))
    (Assign(<|Annular|>,E_Angle+
      |Annular|,Increment)
      (<|Annular|>,E_Angle))
  )
)
    
```

1) 하나의 규칙에는 반드시 하나 이상의 조건부를 가지고 있어야 한다.

2) 하나의 가설은 반드시 모든 규칙의 조건부를 만족하여야 참이 된다.

3) 조건부가 만족되어질 때 실행부에 실행시켜야 할 것이 있을 경우에만 실행이 되어지고 없는 경우에는 그 가설은 단지 참으로만 남아있게 된다.

전문가시스템 도구에서 제공되는 규칙용 에디터를 사용하여 표현한 규칙은 아래와 같이 지식 베이스에 저장되며, @RULE은 규칙의 이름을 의미하고, @LHS는 규칙의 조건부를 나타내고, @HYPO는 가설명을 의미하는 것으로써 규칙이 트리거 되었을 때 규칙의 진위 여부를 값으로 갖는다. @RHS는 규칙의 행동부를 나타내며 규칙의 조건부가 참일 때 수행된다.

## 2.5 규칙의 추론

추론이란 이미 구축되어 있는 규칙과 사실로부터 새로운 사실을 추리해 나가는 과정이며 전문가시스템의 효율성은 지식베이스에 저장되어 있는 지식을 이용하여 얼마나 효과적인 추론을 하는가에 달려 있다.

### 2.5.1 추론 방법

본 연구에서는 어떤 추론제어 기법을 이용하여 추론을 시작할 것인가에 대한 방법과 실행할 규칙들 사이에 충돌이 발생할 경우 해결방법을 결정하기 위하여 [그림 7]에서와 같이 메타슬롯(Meta-Slot)용 에디터를 사용하였으며, 자세한 추론방법은 아래와 같다.

1) 추론 중 발생할 규칙충돌 문제를 해결하기 위해 [그림 7]의 ①번과 같이 추론 우선순위와, 규칙 내의 슬롯 우선순위에 값을 할당하여 추론

메커니즘의 사용을 제한하였다.

2) 상속은 클래스와 서브클래스, 클래스와 객체간의 속성에 대해 [그림 7]의 ②번과 같이 아래로만 상속을 허용하도록 구성하여 추론 도중 필요한 슬롯의 정보만을 이용하도록 설계하였다.

3) 추론 중 슬롯의 상속에 대한 충돌문제를 해결하기 위한 탐색전략은 클래스우선/넓이우선, 클래스우선/깊이우선, 객체우선/넓이우선, 객체우선/깊이우선 탐색기법 등이 있으나 본 연구에서는 [그림 7]의 ③번과 같이 시작노드에서 가장 짧은 경로를 찾는 것을 보장하는 클래스우선/넓이우선 탐색기법을 이용하도록 설계하였다.

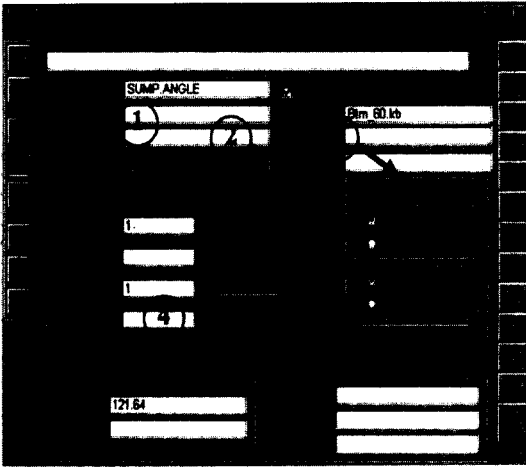
4) 추론 중 슬롯의 값이 없을 경우에 사용자에게 값을 묻지 않고 기본값으로 대치하기 위하여 [그림 7]의 ④번과 같이 초기값을 미리 지정하였다.

5) 추론 중 잘못된 데이터의 입력을 막기 위하여 [그림 7]의 ⑤번과 같이 입력 데이터에 대한 타당성을 체크하기 위한 조건을 추가하였다.

6) 저장조 설계를 위한 간섭체크를 위해 가설을 선정한 후, 지식베이스에서 그 가설과 관련된 규칙을 후향추론메커니즘으로 수행하고 선정된 가설을 중심으로 평가된 결과를 이용하여 원하는 결론에 도달할 때까지 추론을 계속하는 혼합형추론 기법을 이용하였다.

7) 전문가시스템 도구를 이용하여 지식베이스를 작성하였기 때문에 어떠한 추론방법에도 지식의 구조를 변경 없이 전향추론이나 후향추론 기법으로 용이하게 전문가시스템을 확장할 수 있다.



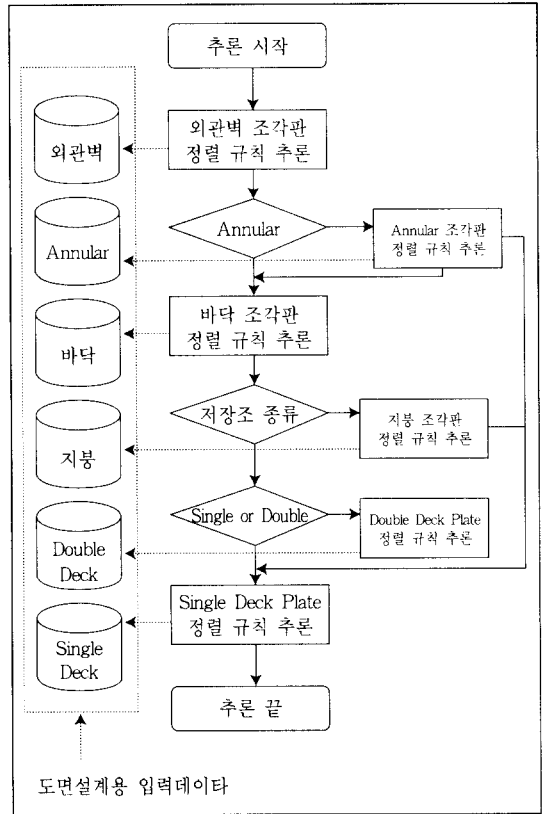


[그림 7] 메타슬롯 에디터

### 2.5.2 추론순서

도면설계용 입력데이터 생성을 위한 추론순서는 [그림8]에서 보듯이 먼저 외곽벽 조각판들 정렬을 위한 규칙들을 실행한 후 Annular의 유,무를 체크하였다. 체크 후 Annular가 있을 경우에는 Annular 조각판들 정렬을 위한 규칙들을 추론한 후 바닥 조각판들 정렬을 위한 규칙을 추론하고 Annular가 없을 경우에는 바로 바닥 조각판들 정렬을 위한 규칙들을 추론하였다.

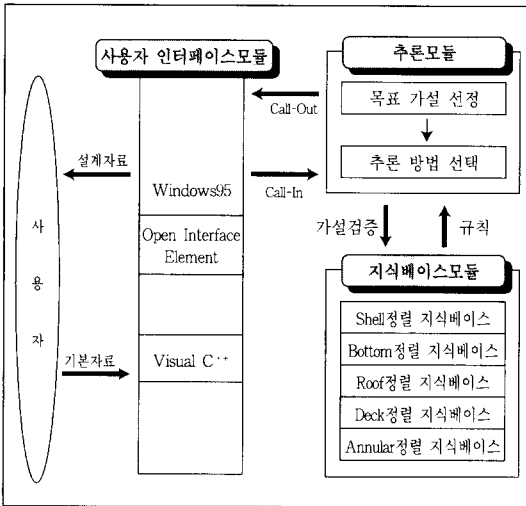
바닥 조각판들 정렬이 완료되면 저장조의 종류에 따라 C.R.T.인 경우에는 지붕 조각판들 정렬을 위한 규칙들을 추론하고 Single F.R.T.인 경우에는 Single Deck Plate 정렬을 위한 규칙을 추론하고 Double F.R.T.인 경우에는 Double Deck Plate정렬을 위한 규칙들을 추론한 후 추론을 마친다



[그림 8] 추론 순서도

## III. 시스템의 구현 및 실행예

전체 시스템은 [그림 9]에서 보듯이 Windows95환경하에서 그래픽 사용자 인터페이스를 중심으로 각 개별 시스템들을 하나로 묶고, 각 개별 시스템들은 상호 독립적인 시스템으로 구현하였다. 그 결과 시스템에 대한 관리가 용이해지게 되고, 시스템의 구성요소를 다른 시스템으로 대체시켰을 경우의 관리나 지식베이스의 부분수정이 간편하게 이루어질 수 있었다.



[그림 9] 전체 시스템 구성도

### 3.1 그래픽 사용자 인터페이스 개발

사용자와 시스템 개발자의 시스템에 대한 의견은 일치하지 않는 것이 대부분이지만 컴퓨터 기술의 발전에 따라서 개발자 중심의 하드 테크놀로지에서도, 사용자 중심의 소프트 테크놀로지로 나아가고 있어 그래픽 사용자 인터페이스(GUI : Graphical User Interface)의 중요성이 증대되고 있다[5].

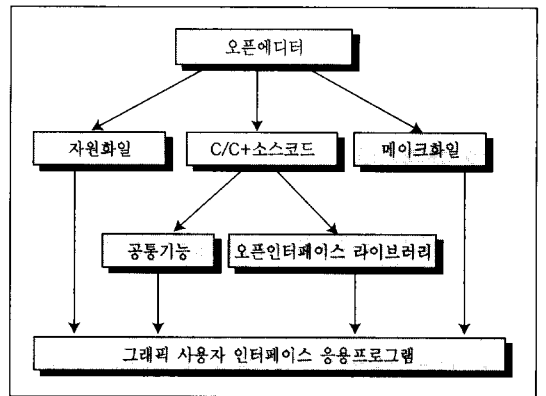
#### 3.1.1 그래픽 사용자 인터페이스 화면설계

본 연구에서는 효과적으로 그래픽 사용자 인터페이스 화면을 설계하기 위해 Element Environment /Open Interface Element에서 제공하는 오픈에디터를 사용하였으며, 오픈에디터는 플랫폼(platform)에 의존하지 않고 다른 기종들 간의 이식이 가능하고, 고도의 GUI도구를 보유하고 있고, 4GL 스크립트 언어를 지원하고, 외부환경과 인터페이스할 수 있는 C/C++라이브

러리 함수를 제공한다.

오픈에디터를 이용하여 화면을 디자인하면 [그림 10]과 같이 자원화일, C/C++ 소스코드, 메이크화일이 자동생성된다. C/C++ 소스코드는 C/C++에서 공통으로 사용되어지는 코드와 Open Interface Element에서만 제공하는 오픈인터페이스 라이브러리 함수로 나누어진다[3].

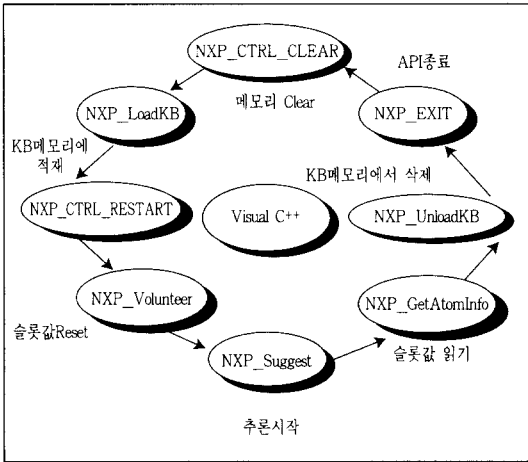
본 연구에서는 자동 생성된 C/C++ 소스코드 부분을 수정, 보완하여 그래픽 사용자 인터페이스 응용프로그램을 완성하였다.



[그림 10] Open Interface Element에 의한 화일 자동생성

### 3.2 지식베이스와 그래픽 인터페이스의 통합

전문가시스템 도구를 사용하여 구축한 지식베이스모듈과 디자인한 그래픽 사용자 인터페이스 화면은 Windows 95 환경하에서 Visual C++과 전문가시스템 도구에서 제공되는 API(Application Programming Interface)라이브러리 함수들을 이용하여 결합하였다. [그림 11]은 API함수를 이용하여 지식베이스를 실행하는 과정이다.



[그림 11] API함수를 이용한 지식베이스 실행과정

### 3.3 시스템 실행에

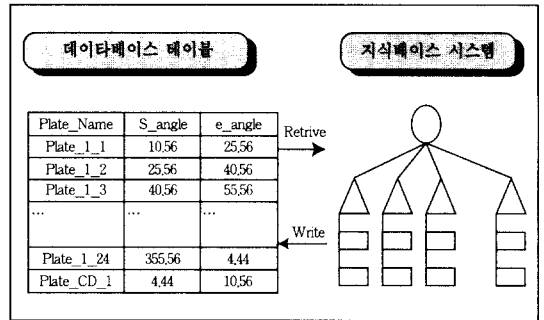
본 연구에서는 개발된 전문가시스템을 실행한 후 평가하기 위해, 선경건설에서 과거에 수행했던 저장조 설계 관련 프로젝트를 선택하여 대상으로 삼았다. 실행 및 평가 방법으로는 저장조 설계에 경험이 풍부한 전문가의 도움을 얻었으며, 전문가시스템 추론결과 생성된 설계도면과, 과거 프로젝트 수행 당시 출력되었던 도면을 비교함으로써 두 도면이 일치하는지 확인하였다.

#### 3.3.1 시스템 실행과정

구현된 전문가시스템의 실행과정은 아래와 같이 3단계로 나누어지며, 1단계에서는 주어진 기본 데이터베이스 테이블의 데이터들을 지식베이스 시스템으로 읽어 들이고, 2단계에서는 규칙들을 추론한 후 추론결과를 다시 데이터베이스 테이블에 저장하며, 3단계에서는 새롭게 생성된 데이터베이스 테이블의 데이터를 이용하여 설계도면을 생성한다

#### 1) 설계정보의 유출단계

저장조에 대한 설계데이터는 설계에 필요한 기본 데이터베이스 테이블로부터 필요한 정보만을 이에 해당하는 클래스에 객체로서 얻어진다. [그림 12]에서와 같이 데이터 유출과정에서는 테이블 레코드의 수 만큼 클래스 내의 동적 객체로 생성되며 필드의 수 만큼 동적객체들은 속성을 가지게 된다. 이 과정을 유출(Retrieve)라 한다.



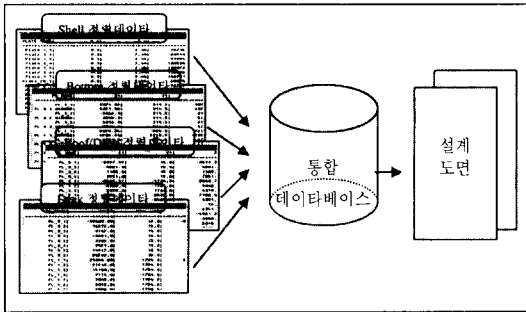
[그림 12] 데이터 유출 및 저장과정

#### 2) 설계정보 저장단계

지식베이스 내의 규칙을 추론한 결과 생성된 클래스들은 유출과정의 경로와 반대로 클래스 내의 동적객체는 데이터베이스 테이블의 레코드로, 동적객체들의 속성들은 레코드의 필드로 생성된다. 이 과정을 저장(Write)이라 한다.

#### 3) 설계도면 생성단계

설계정보의 유출단계와 저장단계를 거쳐 생성된 저장조 도면설계용 입력 데이터베이스 테이블(Shell, Bottom, Roof/Deck, Annular)들은 도면 설계를 위하여 하나의 데이터베이스로 통합되며, 통합된 데이터베이스를 입력으로 CAD시스템을 사용하여 저장조 설계도면을 완성한다.



[그림 13] 설계도면 생성과정

### 3.3.2 실행 결과

〈표 2〉는 과거에 수작업으로 설계했을 경우에 걸렸던 시간과 전문가시스템을 사용해 설계한 경우 시간을 비교한 예제이다. 표에서 보듯이 네가지 종류의 저장조에 적용해본 결과, 평균적으로 과거 수작업으로 수행했을 경우 소모되던 시간의 88%가 절감된 것을 알 수 있다.

〈표 2〉 전문가시스템 사용시 정량적 효과

저장조 종류	수작업 (시간)	전문가시스템 (시간)	절감비 (%)
Self Supported C.R.T.	460	50	89
Rafter Supported C.R.T.	594	10	91
Column Supported C.R.T.	910	110	87
Floating Roof Tank	1,200	150	87
합 계	3,119	360	88

## IV. 결론 및 향후 연구과제

### 4.1 결 론

본 연구에서는 저장조 설계에 필요한 도면설계용 입력데이터 생성을 위해 복잡하고 다양한

설계조건을 최근에 활발히 연구되고 있는 객체지향 모델링 기법을 이용하여 저장조 설계 전문가시스템을 개발함으로써 아래와 같은 결론을 얻을 수 있었다.

1) 저장조 설계를 위한 복잡한 문제를 해결하기 위해 전문가시스템을 이용함으로써 매우 제한된 범위에서나마 비전문가도 전문가의 도움 없이 전문가와 비슷한 수준의 저장조 설계를 수행할 수 있었다.

2) 전문가시스템 개발용 도구와 객체지향 기법 등을 이용해 기존의 프로그램 언어에서의 시스템 유지, 관리의 한계성을 극복하고, 현재 시스템상에 구축된 지식의 변경과 추가에 대한 대처가 용이하며 지속적인 개발이 가능해진다는 것을 알 수 있었다.

3) 본 연구에서 사용한 전문가시스템 구현 도구의 연산기능은 기본적인 사칙연산 정도이므로 복잡한 수치적 계산과정이 필요한 작업보다는 경험적인 지식을 필요로 하는 분야에 적합하다.

4) 다양한 요구와 변화를 가진 설계조건들에 대한 명확한 지식체계의 구현이 어렵고 지식 자체가 정형화되어 있지 못하므로 지식 수집에 많은 어려움을 겪었으며, 이를 위해 전문가의 적극적인 협조가 이루어져야 한다.

5) 표준화된 설계 절차대로 데이터를 입력함으로써 설계상의 오류를 감소시켜 같은 설계조건에서 서로 다른 제품을 만들어 내는 오류를 줄였으며, 수작업으로 설계하던 도면을 전문가시스템에 의해 자동으로 생성함으로써 설계에 드는 인력, 시간 등의 비용을 절감하였다.

### 4.2 향후 연구과제

본 연구에서는 여러 형태의 저장조 중 가장

많이 사용되는 일반적인 종류의 저장조만을 대상으로 다루었고, 실제로 저장조의 종류 및 저장조 설계에 필요한 조건들의 복잡성과 다양성은 무한하다. 그러므로 전문가의 판단과 경험에 의한 지식의 추가와 저장조 설계를 위한 지속적인 전문가시스템 개발의 노력이 있어야 하며 이를 위해서 아래와 같은 연구과제가 수행되어야 한다.

1) 저장조 종류 중 Cone Roof Tank의 일부인 Column Supported Cone Roof Tank, Rafter Supported Cone Roof Tank, Self Supported Cone Roof Tank와 Single Deck Floating Roof Tank, Double Deck Floating Roof Tank만을 다루었으므로 나머지 저장조에 대한 지식베이스의 구축이 이루어져야 한다.

2) 저장조 설계를 위한 단계별(Shell, Annular, Bottom, Roof, Deck) Plate들의 정렬과정을 그래픽 처리기법을 통해 실시간으로 전문가시스템 사용자가 시각적으로 확인할 수 있도록 그래픽 프로그램과의 인터페이스가 필요하다.

3) 전문가시스템의 가장 큰 약점은 기호(Symbol)를 다루는 언어로써 구현이 되어있기 때문에 계산(Algorithm)에 대해서는 대단히 약하다는 점이다. 저장조 설계는 복잡한 계산과정이 포함되어야 하는 작업이므로 이를 보완하기 위해서는 기존의 수치 계산적인 프로그램과의 인터페이스가 필요하다.

4) 현재 CAD시스템은 단지 도면작업으로만 끝나고 설계정보요소를 포함시킬 수가 없으므로, 설계정보요소의 관리, 유지를 위한 객체지향 데이터베이스(Object Oriented Database)와의 인터페이스가 가능한 인텔리전트(intelligent) CAD 시스템에 대한 지속적인 연구가 필요하다.

5) 지식베이스를 구현하는데 사용된 지식은

아주 제한된 범위에서만 적용된 것이기 때문에 전문가와의 지속적인 인터뷰를 통해 논리적으로 어려운 문제들의 해결 방안을 수집하고 모든 경우에 대해 범용성을 가질 수 있는 지식베이스로의 확장이 필요하다.

## 참고문헌

- [1] Donald A. Waterman "A Guide to Expert System", Addison-Wesley Publishing Company, 1985
- [2] Neuron Data Inc., "Open Interface Elements - API Reference Manual",
- [3] Neuron Data Inc., "SMART ELEMENTS Version 2.0 - Introduction Manual",
- [4] 김상철, "구조계획에서의 지식기반시스템 도입 연구", 한양대 석사논문, 1994.12
- [5] 김화수, 조용범, 최종욱, "전문가시스템", 집문당, 1995
- [6] 김화수, 정현욱, "TDX-10 전자식교환기 고장진단 전문가시스템 구축", 추계학술대회 논문집, 한국전문가시스템학회, 1994.11
- [7] 김재희, "인공지능의 기법과 응용", 교학사, 1998
- [8] 박광호, "동시 공학적 전문가시스템 개발방법론", 한국전문가시스템학회지, 한국전문가시스템학회, 1995.1
- [9] 박홍규, 박경원, 김화수, "위계인 지원용 내장형 전문가 시스템의 사후분석모듈개발", 춘계학술대회 논문집, 한국전문가시스템학회, 1997.6
- [10] 선경건설, "L/E 및 P/E 양성교육 교재 Volumel", 1997.6
- [11] 이민우, "기력 발전소 터빈의 운전지원 전문가시스템", 인하대 석사논문, 1995.8

- [12] 이희철, “전문가 시스템을 이용한 철골 구조물의 접합부 설계”, 한양대 석사논문, 1993.12
- [13] 조성진, 오준환, 남재중, 이재원, “Deep Drawing 공정 설계 전문가시스템 Dox의 개발에 관한 연구”, 한국전문가시스템학회지, 한국전문가시스템학회, 1996.12
- [14] 한국엑스퍼트(주), “Smart Elements Training”
- [15] 한국엑스퍼트(주), “Open Interface Element Training”
- [16] 한국과학기술원, “인공지능 프로그래밍과 전문가시스템의 개발실습”, 1989