韓國經營科學會誌
第18卷 第3號
1993年 12月

재사용 가능한 선형계획모형 요소의 인식과 추출에 관한 연구

박성주* · 권오병*

Identification and Extraction of Reusable Linear Programming Model Components

S.J. Park* and O.B. Kwon*

Abstract

This paper proposes an idea of *reverse modeling* that analyzes LP models and then converts them into an object-oriented model repository. The process of reverse modeling consists of (1) identifying and analyzing source models by *meta processor*, (2) *model decomposition and generalization* to scan the models and divide them into model components, and (3) deriving model selection rules from the components by *rule generator*. Through the process, we can extract reusable model components and build a model base with model selection rules. Examples with models created by SML and MODLER modeling languages are given to illustrate the methods. The model base management capabilities provided by reverse modeling can increase the reusability of current modeling tools.

1. 서 론

컴퓨터통합생산(computer integrated manufacturing)과 같은 통합시스템에서의 모형의 특징은 상황, 정책, 시스템내의 타 컴포넌트가 변화함에 따라 지속적인 변경, 수정을 하여야 하는 것이며 이때 상당 부분의 모형은 반복적으로 활용된다. 이러한 요구를 만족시키기 위해서 모형관리에 대한 연구에서는 모형을 재사용 가능하도록 고려하는 것을 중심적으로 시도하고 있다. 특히 최근에는 객체지향 패러다임을 기본으로 하는 객체지향 모형관리 [11,13,25,27,34] 와 모형의 모듈화(modularity)[5] 개념을 중심

* 한국과학기술원 경영과학과

으로 재사용성이 강조되고 있는데, 그 이유는 모형 작성을 하기 이전에 정보의 은닉이나 모듈화등의 충분한 고려를 한 후에 개발하면 사후의 모형 변경시의 어려움을 경감시킬 수 있다는 데 기인한다. 이들은 또한 모형의 복잡성이나 난해성 때문에 모형들이 실제적으로 잘 사용되지 않는다는 문제 [4] 에 대해서도 좋은 해결방안이 된다.

그러나 객체지향 개념에서 모형베이스를 구축할 때의 애로사항은 모형베이스에 등록될 내용의 상당부분이 이미 여러 형태의 모형들로 조직내에 존재하고 있으며, 또 이들을 무시하고 새로 모형베이스를 만든다는 것은 시간과 비용의 측면에서 효율적이지 못하다. 따라서 모형관리 시스템의 주요 기능의 하나로서 산재해 있는 모형들로부터 모형베이스에 등록될 가치가 있는 정보를 추출하는 것이 필요하다.

본 논문의 목적은 재사용에 대한 특별한 고려가 없이 수립된 선형계획 모형들로부터 재사용 가능한 부품을 추출하여 객체지향적으로 변형시켜주는 역모형화(Reverse Modeling) 방법론을 제시하는 것이다. 그리고 다른 형태의 모형으로의 확장 가능성에 대해서도 언급하였다.

2. 모형의 재사용성 (Model Reusability)

2.1. 재사용성(Reusability)

소프트웨어 재사용은 1969년에 McIlroy 의 모듈형태의 소프트웨어 단위(modular software units) 라는 개념으로부터 출발한 것으로 [21], 여러 분야에서 공통적으로 사용될 수 있는 요소들의 활용을 통해 소프트웨어 개발 비용을 감소시키는 것을 목적으로 하고 있다.

원천 코드수준의 재사용에 대한 실현전략은 시스템을 개발하기 전에 재사용 가능하도록 고려하는 전략과 [7], 이미 개발된 시스템들로부터 재사용 가능한 부품을 추출하는 전략 [2,15, 30,32] 으로 분리된다.

최근 들어서 소프트웨어의 재사용을 위한 적절한 방법중의 하나로 객체 지향성을 지적하고 있다. 그 이유는 객체 지향성의 본질인 상속성, 추상화, 정보은닉등의 개념이 재사용성을 가능하게 하는데 유리하게 작용하기 때문이다. 객체지향적인 패러다임은 소프트웨어의 개발 중심적 측면에서 재사용의 측면에 대한 새로운 기회들을 제공하고 있다 [22,33].

2.2. 모형의 재사용성

모형의 특수성을 감안하여 고려된 모형의 재사용성에 대한 연구들은 기본적으로 객체지향 패러다임에서 모형을 객체(혹은 복합객체)로 보고 있는데 이러한 시각은 모형의 통합이나 행위의 모형화, 복잡성의 완화, 그리고 모형의 확장성에도 유리하다 [11,13,17,24,27,34].

그런데 이러한 연구들은 앞에서 밝힌 재사용을 위한 두가지 방안 중 전자에 해당하는 것으로 이미 있는 모형들로부터 재사용 가능한 부분들을 추출하는 과정에 대한 언급은 구체적으로 되어 있지 않다. 이 경우 이미 조직내에 널리 퍼져서 축적되어 있는 모형들을 다시 재사용 가능하도록 해석, 분류하는 노력은 여전히 남게 된다. 이러한 이유로 단기적으로 볼 때 재사용 가능한 요소들을 결정하여 시스템을 개발하는

것은 아무 고려없이 개발한 것보다 더 많은 비용이 든다 [2].

결국 모형관리 시스템이 재사용 능력을 가지기 위해서 객체지향 개념을 도입하는 것 뿐 아니라 기존 모형을 해석하고 이를 객체지향 개념에 맞게 저장하도록 하는 방법론이 추가적으로 필요하다.

3. 역모형화 (Reverse Modeling)

3.1. 역공학과 역모형화

역공학(Reverse Engineering)은 시스템의 요소들과 그들 간의 상호관계성(interrelationship)을 파악하고 그 시스템을 다른 방식으로 표현하거나 보다 더 추상화된 형태로 전환시키는 과정이다. 역공학은 다른 형태의 표현으로 쉽게 변환가능하거나, 보다 추상화된 수준으로의 합성을 가능하게 하고, 부품들을 재사용 하도록 도와주며, 결국 시스템의 유지를 위한 비용을 절감하고 시스템 변화에 대한 적응을 쉽게 하는 것을 목적으로 한다 [3,9].

만약 역공학에서 언급한 시스템이 환경내에 존재하는 자원들의 하나인 수리 모형이라고 본다면 모형관리의 분야에서도 이러한 동기와 방법론이 적용가능하다. 본 논문에서는 역모형화 개념을 다음과 같이 정의한다.

[정의1] 역모형화(Reverse Modeling)란 모형에 내재된 요소들과 그들 간의 관계성을 파악하고 이 모형들을 모형화 도구에 독립적이고 재사용 가능한 형태로 전환시키는 과정이다.

역공학의 과정과 유사하게 역모형화 과정은, 「원천모형 → 분석기 → 모형베이스 → 모형합성 → 새로운 모형」이라는 단계를 거친다. 단, 모형베이스에 있는 수리적 모형의 요소들로부터 모형을 합성하는 후반부 과정은 Murphy [26] 의 개념을 참조할 수 있다. 그 이유는 첫째 Murphy 는 선형계획모형을 이루는 기본요소들이 완전히 수집되었다고 가정한 후에 이들을 통해 하나의 완전한 모형을 합성하는 부분을 다룬 반면 이 논문에서는 모형으로부터 모형의 기본요소들을 추출하는 것을 다루었기 때문에 각각 모형 사용의 후반부,전반부라고 칭할 수 있으며, 둘째로 그가 선형계획모형을 이루는 기본 요소로 제시한 의사결정변수(variables), 좌변계수(LHSCoefficients), 우변상수(RHSconstants), 및 인덱스(Indices)를 이 논문에서도 마찬가지 기본 요소로 인식하고 있기 때문이다. 더욱이 Murphy 의 연구는 아직 구현되지 않은 개념적인 연구이기 때문에 이 논문의 확장에 그의 연구를 사용할 수 있다고 본다. 따라서 본 논문에서는 원천모형으로부터 모형베이스의 구축에 이르는 전반부에 중점을 두고자 한다.

3.2. 선형계획모형의 분석

역모형화를 위한 첫 단계로서 우선 선형계획모형의 원천 모형에 대해서 살펴보고자 한다. 일례로 수송계획에 관한 수리 모형은 다음의 [그림 1]과 같이 표현될 수 있다.

선형계획모형은 외관상 크게 목적함수와 제약식으로 이루어져 있다. 그리고 그들은 의사결정변수와 좌변계수, 우변상수등의 모형 구성요소들의 수리적 관계로 구성되며, 이들 각각은 다시 인덱스를 보유할 수 있다. 그런데 이 요소들

```
        /* (STP) Simple Transportation Problem */
1: Min ∑i,j Cost{i,j}T{i,j} /* (CostFunction) Cost function */
2: s.t. ∑iT{i,j} >= Demand{j} /* (D{j} Required demand at {j} */
3:      ∑jT{i,j} <= Supply{i} /* (S{i}) Limits supply at {i} */
        T{i,j} >= 0 for all i,j.
   where Cost{i,j} : cost to transport 1 unit of products from {i} to {j}
        T{i,j} : product transports from {i} to {j}
        Demand{j} : required demand at {j}
        Supply{i} : available supply at {i}
```

[그림 1] 선형계획모형 예 : 수송모형

〈표 1〉 도형 요소

인덱스	변수/계수	목적함수/제약식
i, j	Cost	1
i, j	T	1, 2, 3
i	Demand	2
j	Supply	3

은 따로 분리되지 않고 하나의 모형의 내부에 혼합되어 있다[10].

이때 이 모형에 포함되어 있는 요소들을 〈표 1〉과 같이 정리할 수 있다.

이때 인덱스는 대상 영역의 개체에 관련되므로 가객체(PO : Pseudo-Object), 변수 및 각종 계수에 해당하는 부분은 실제적인 값을 가지는 부분들이므로 가속성(PP : Pseudo-Property), 목적함수 및 제약식은 함수 또는 제약의 수행(execution), 계산(computation) 혹은 가능해(feasible solution)점검에 관한 것이므로 가메소드(PM : Pseudo-Method)라고 명명하기로 한다. 이때 가(Pseudo)라고 하는 용어를 사용하는 이유는 인덱스를 변수/계수와 목적함수/제약식을 각각 속성 및 메소드로 인식하고 보유하는 객체로 표현할 수 있으며 이를 통해 인캡슐레이션, 상속성, 자료의 추상화등 객체지향개념의 특징중 일부를 사용할 수 있으나, 또다른 중요한 특징인 메세지전달(message passing)과 다형성(polymorphism)등의 개념을 사용하고 있지는 않기 때문에 완전한 객체지향성에서의 객체, 속성, 메소드로 보기 어렵기 때문이다.

한편 각각의 요소들을 모두 모아 놓은 것들을 각각 가속성집합(PPS : Pseudo-Property Set), 가객체집합(POS : Psuedo-Object Set), 가메소드집합(PMS : Pseudo-Method Set)이라고 하자. 이때 선형계획모형은 다음과 같이 단순화하여 정의된다.

[정의2] 선형계획모형(LP) 은 POS, PPS, 및 PMS 들의 집합이다. 즉,

LP = { POS, PPS, PMS },

단, POS = {∀k, POk},
PPS = {∀i, PPi},
PMS = {∀j, PMj} (1)

여기서 임의의 가메소드들은 어떤 둘 이상의 가속성들과의 관계식으로 표현되므로 가메소드는 가속성에 함수종속적이다. 예를 들어 ∑iT{i, j} >= Demand{j} 와 같은 제약식 D{j}가 있다면 D{j}는 가메소드가 되며, 가속성집합의 요소들 중 T{i,j}와 Demand{j}간의 관계식이므로 D{j} 는 T{i,j} 와 Demand{j} 에 함수종속적이라고 볼 수 있다. 즉 (D{j}) = f((T{i,j}, Demand{j})) 이며, 이를 일반화하면,

(a subset of PMS) = f((a subset of PPS)) (2)

단 f 는 가속성집합과 가메소드집합의 부분집합간의 대응관계
와 같다. 이 관계를 통해 다시 다음과 같은 PPS 과 PMS 의 부분집합과의 If-then 규칙을 추출해 낼 수 있다.

IF (a subset of PPS) THEN (a subset of PMS) (3)

또한 각 속성들은 하나 이상의 객체를 보유하며 객체에 의해서 규정되므로 마찬가지로,

(a subset of PPS) = g((a subset of POS)) (4)

단, g 는 가객체집합과 가속성집합의 부분집합간의 대응관계

IF (a subset of POS) THEN (a subset of PPS) (5)

가 된다.

한편 식 (1)은 식 (2) 와 (4)에 의해,

LP = { POS, PPS, PMS }
= { POS, PPS, f(PPS) } by 식 (2)
= { POS, g(POS), f(g(POS)) }
by 식 (4) (6)

와 같이도 표현된다.

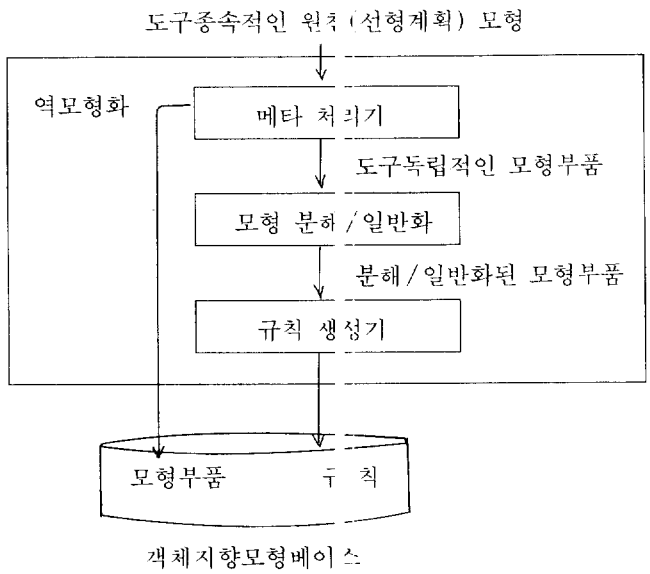
결국 선형계획모형으로부터 식 (1)-(5) 와 같은 공통의 재사용 요소들을 추출해 낼 수 있다.

4. 연구의 틀

역모형화의 관심은 MODLER, SML등의 복수의 형태로 작성된 도구종속적인 모형으로부터 단일의 표현형태를 가지는 모형베이스로의 이행에 있는데, 이는 구체적으로 다음의 네가지 요구사항들을 만족하기 위함이다.

(1) 선형계획 모형에 대한 일반화된 정의
(2) 복수의 도구종속적인 모형을 해석하는 기능
(3) 해석된 모형을 분해하여 모형베이스의 내용과 통합하는 기능
(4) 모형의 선정을 위한 규칙의 생성

첫째, 선형계획모형에 대한 일반화된 정의란 어떠한 도구종속적인 선형계획모형이라도 공통적으로 가지는 개체들과 그들 간의 관계에 대한 정의로서, 즉 선형계획모형에 대한 메타 명세를 말한다. 둘째, 메타 명세를 이용하여 각 형태의 도구종속적인 모형에 대한 명세를 할 수 있다. 위의 두가지 요구를 충족하기 위해 메타 시스템 접근 방법을 채택하였다. 메타 시스템이란 개발하고자 하는 개발도구의 정형화된 명세를 통해 개별 소프트웨어 도구를 자동생성하기 위한 시스템이다 [29]. 선형계획 모형에서의 메타 시스템의 접근방법의 근거는 선형계획 모형에



[그림 2] 역모형화의 전체적인 구조

관련된 모형화 도구들간에 구조적 유사성이 있으며, 도구들이 지원하는 모형들이 공통적인 요소들을 가지고 있다는 것이다. 그러므로 메타수준에서 정의된 선형계획 모형에 대한 메타 명세로 도구 수준의 모형들을 해석할 수 있는 해석기를 생성할 수 있다. 셋째, 메타 시스템에 의해 해석된 모형들을 모형베이스에 저장하기 위해서는 모형베이스에 이미 저장된 내용과 통합되어야 하는데 통합의 방법중에는 일반화에 의한 방법과 분해에 의한 방법이 있다[14]. 마지막으로 통합된 모형에 대한 내용들은 후의 사용을 위해 모형 선정을 위한 규칙을 가질 필요가 있다. 모형은 외부적으로는 모형의 실행에 필요한 지식이나 어떠한 상황에서 어떠한 모형이 선택되어야 하는지에 대한 정보를 가지고 있지 않다[10]. 모형 선정을 위한 규칙들을 외부 전문가에 의해서 일방적으로 입력되는 형태보다는 역모형화 과정에서 확보되는 지식을 활용하여 규칙들을 자동적으로 생성하는 방안을 강구하였다.

역모형화를 위해서 [그림 2]와 같은 전체적인 구조를 갖는다.

(1) 메타 처리기(Meta Processor)

메타 처리기란 메타 언어로 기술된 선형계획 모형에 대한 정의를 토대로 하여 각 도구 종속적인 모형들의 명세의 기술을 문법적으로 판독하고 점검하는 것으로 개체-관계 모형을 기반으로 한다. 이 과정에서 각 도구 종속적인 모형들을 해석하기 위한 해석기가 생성되며 이 해석기는 모형들로부터 도구 독립적인 형태의 모형 재사용 요소들을 추출한다.

(2) 모형분해/일반화

(Model Decomposition/Generalization)

메타처리기에서 유도된 재사용 요소들을 모형

베이스에 있는 내용들과 통합할 때, 통합 가능성을 점검하고 또한 요소들의 일반화와 분해과정을 통하여 통합하여 모형베이스에 저장한다.

(3) 규칙 생성기 (Rule Generator)

분해된 모형을 후에 재사용, 합성하기 위해서는 모형요소사이의 상호관계성에 대한 명료한 정보가 필요하다. 본 논문에서는 가객체, 가속성, 가메소드들간의 If-Then 규칙의 형태로 이러한 정보를 표현한다. 이때 규칙 생성기는 모형 선정을 위한 정보들을 추출하여 규칙의 형태로 생성시켜주는 역할을 한다. 이때 모형 선정을 위한 정보는 앞의 모형분해/일반화 과정에서 발생된 정보의 일부이다.

4.1. 메타 처리기 (Meta Processor)

선형계획모형은 식 (1), (2), (4)를 통하여 다음과 같은 객체유형과 관계유형을 정의한다.

(1) 객체 :

LP, POS, PPS, PMS, PO, PP, PM (식 (1) 참조)

(2) 관계 :

1) IsA

PO 들 사이의 계층적인 구조를 나타내는 관계이다. 이 관계에서는 하위 PO 의 속성들을 상속할 수 있는 관계를 설정해 주는 것으로 객체-지향 개념의 핵심적인 관계이다.

2) is part of

전체를 이루는 한 요소와 그것을 구성하는 하위 구조사이의 관계를 나타낸다. 하위 구조중 어느 하나가 불충분하여도 그것의 전체는 성립하지 않는다.

3) is_element_of

집합과 그것들의 원소들간의 관계를 나타내는 관계유형이다. is_part_of 와는 달리 원소들이 그것의 전체를 규정하지는 않는다.

4) contains

종속 변수와 독립 변수와의 대응 관계를 나타낸다.

위와 같은 객체와 관계를 이용해서 선형계획모형은 다음과 같이 정의될 수 있다. 단 괄호안은 대응관계를 나타낸다.

식 (1)에 의하여,

1) −POS is_part_of LP (1:1)
−PPS is_part_of LP (1:1)
−PMS is_part_of LP (1:1)
−PO is_element_of POS (m:1)
−PP is_element_of PPS (m:1)
−PM is_element_of PMS (m:1)

식 (2)에 의하여,

2) −PM contains PP (n:m)

식 (4)에 의하여,

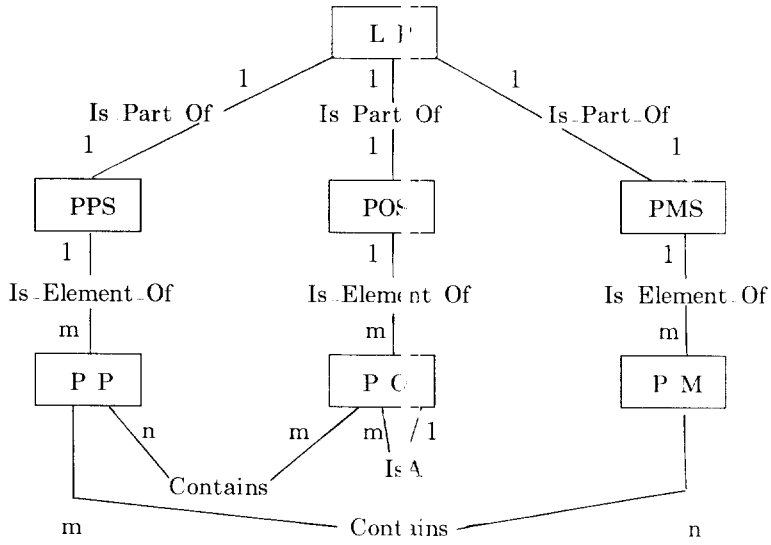
3) −PP contains PO (n:m)

또한,

4) −PO IsA PO (m:1)

와 같은 9가지 관계를 기술한다. 단, 4)는 어떤 가객체가 또다른 어떤 가객체의 클래스가 될 수 있음을 의미한다. 예를 들어 선형계획모형의 적용례의 하나인 경유수송문제(transshipment problem)에서 원천저장소와 중간저장소가 있다면 이들은 각각 가객체들로 인식되며 이들은 다시 저장소라고 하는 가객체를 그들의 클래스로 갖는다.

결국 선형계획모형의 일반적인 특성으로부터 앞에서 정의된 개념인 LP, POS, PPS, PMS, PO, PP, PM 를 개체로 하고, 위의 9가지를 관계로 하는 메타수준의 개체-관계 모형 (Entity-Relationship Model)을 [그림 3]과 같이 표현할 수 있다. 또한 이 그림에 표시되어 있는 개체-관계 모형은 다시 [그림 4]와 같이 개체-관계 모형을 기반으로 한 K-Meta 시스템 [29]의 메타 언어로 정의된다.



[그림 3] 선형계획모형에 대한 개체-관계 모형

```
ENTITY LP ;
ENTITY POS ;
ENTITY PPS ;
ENTITY PMS ;
ENTITY PO ;
ENTITY PP ;
ENTITY PM ;
RELATIONSHIP is_part_of rel1 ;
    PART part EN POS ;
        (partial : M)
    | aggregation_part EN LP ;
RELATIONSHIP is part of rel2 ;
    PART part EN PPS :
        (partial : M)
    | aggregation part EN LP :
RELATIONSHIP is part of rel3 :
    PART part EN PMS :
        (partial : M)
    | aggregation part EN LP :
RELATIONSHIP is element of rel1 :
    PART element part EN PO :
        (partial : M)
    | sets part EN POS :
RELATIONSHIP is element of rel2 :
```

```
    PART element_part EN PP ;
        (partial : M)
    | sets part EN PPS ;
RELATIONSHIP is_element_of_rel3 ;
    PART element_part EN PM ;
        (partial : M)
    | sets_part EN PM ;
RELATIONSHIP contains_rel1 ;
    PART contained_part EN PO ;
        (partial : M)
    | container part EN PP ;
RELATIONSHIP contains_rel2 ;
    PART contained_part EN PP ;
        (partial : M)
    | container_part EN PM ;
RELATIONSHIP IsA rel ;
    PART children_part EN PO ;
        (partial : M)
    | parent_part EN PO ;
```

[그림 4] 선형계획모형에 대한 메타 기술

이 메타기술을 근거로 하여 대상 모형화 도구로부터 재사용할 모형요소가 추출될 수 있는데, 예를 들어서 〈표 1〉의 수송모형의 요소들에 대해서 다음과 같은 대상 시스템에 대한 지식을 추출할 수 있다.

```
DEFINE ENTITY LP
    NAME-IS stp
    DESCRIPTION simple transportation
    problem

DEFINE ENTITY PO
    NAME IS i
    DESCRIPTION demand region
    ISA region
    IS_ELEMENT_OF POS

DEFINE ENTITY PO
    NAME IS j
    DESCRIPTION supply region
    ISA region
    IS_ELEMENT_OF POS

DEFINE ENTITY PP
    NAME IS t{i,j}
    DESCRIPTION product transports from
    {i} to {j}
    IS_ELEMENT_OF PPS
    CONTAINS i,j

DEFINE ENTITY PP
    NAME_IS cost{i,j}
    DESCRIPTION cost to transport 1 unit
    of products from {i} to {j}
    IS_ELEMENT OF PPS
    CONTAINS i,j

DEFINE ENTITY PP
    NAME_IS demand{j}
    DESCRIPTION required demand at {j}
    IS_ELEMENT OF PPS
    CONTAINS j

DEFINE ENTITY PP
    NAME_IS supply{i}
    DESCRIPTION available supply at {i}
    IS_ELEMENT_OF PPS
    CONTAINS i
```

DEFINE ENTITY PM
NAME_IS cost
DESCRIPTION cost function
IS ELEMENT OF PMS
CONTAINS null

DEFINE ENTITY PM
NAME_IS d{j}
DESCRIPTION required demand at {j}
IS ELEMENT OF PMS
CONTAINS t{i,j}, demand{j}

DEFINE ENTITY PM
NAME_IS s{i}
DESCRIPTION limits supply at {i}
IS ELEMENT OF PMS
CONTAINS t{i,j}, supply{i}

여기서 contains 관계는 식 (2), (4) 에서 나온 것이므로 다음과 같이 표현된다. 우선 POS, PPS 와의 관계에서,

(i,j) → (Cost{i,j})
(i,j) → (T{i,j})
(j) → (Demand{j})
(i) → (Supply{i})

이며 이들 각각을 일반화하여 표현하면,

(a subset of POS) → (a subset of PPS)

와 같다.

한편 PPS 와 PMS 의 관계에서도,

(Cost{i,j}, T{i,j}) → (Cost)
(T{i,j},Demand{j}) → (D{j})
(T{i,j},Supply{i}) → (S{i})

와 같으며, 이들 또한 각각

(a subset of PPS) → (a subset of PMS)

와 같이 일반화하여 표현할 수 있다.

4.2. 모형분해 / 일반화 (Model Decomposition / Generalization)

모형관리시스템의 목표중의 하나는 모형의 잠재력(potentiality)을 극대화하는 데 있다 [25]. 이에 따라 모형베이스가 데이타베이스와 마찬가지로 모형들간의 불일치성과 중복성을 최대한 방지하여 모듈화(modularity)와 유연성(flexibility)을 증진시키는 것을 목표로 한다고 볼 때 [6], 메타 처리기에 의해 해석된 모형들은 모형베이스에 등록되기 전에 통합능력과 사용능력을 확장하기 위한 단계를 거칠 필요가 있다. 모형의 통합을 위한 방법으로 일반적으로 사용되는 것이 분해에 의한 방법 [12,16,18,19] 과 일반화 [31,35] 에 의한 방법이며 여기서는 이들을 같이 사용한다.

4.2.1 일반화에 의한 통합(integration by generalization)

일반화에 대해 설명하기 전에 앞부분에서 언급한 바 있는 다음의 두가지 성질을 먼저 정리한다.

(1) PP 와 PM 은 모두 PO 를 가진다(식 (6)).
(2) PO는 IsA 관계를 가진다([그림 3]).

이 두가지 사실로부터 PO 는 일반화될 수 있고, 나아가서 PP 와 PM 도 일반화될 수 있다고 볼 수 있다. 일반화를 위해 기계학습의 유도(Induction) 방법중의 하나인 후보자제거알고리듬(Candidate Elimination Algorithm)[21] 을 응용하였다. 그 과정은 다음과 같다.

1. H 를 규칙들의 전체집합, G 는 공집합, 그리고 S 는 모든 가장 특정한 개념(most specific concept)을 가지는 집합으로 초기화한다.
2. 다음의 조치를 취한다.
   2.1 새로운 훈련 예(training instance) 로서 최초규칙을 받아들인다. 단, 최초규칙은 메타처리기에서 인식된 (a subset of POS) → (a subset of PPS) 또는 (a subset of PPS) → (a subset of PMS) 이다.
   2.2 G 로부터 새 예제를 설명하지 못하는 것들을 제거한다.
   2.3 S 를 가능한 한 최소범위에서 일반화한다.
3. G 와 S 가 동일하며 또한 그것이 유일한 해(singleton set)를 가질 때까지 2.의 과정을 반복한다. 이 조건이 만족되면 H 가 확정된다.
4. H 를 다음과 같은 형태로 출력한다.
   (a subset of PPS′) → (a subset of PMS′),
   (a subset of POS′) → (a subset of PPS′)

4.2.2 분해에 의한 통합(Integration by decomposition)

일반화 과정을 거친 f 와 g 는 다시 다른 규칙들의 혼합의 형태일 수 있다. 즉 어떤 f 는,

(a subset of PPS′) → (a subset of PMS′)
⇒ ((a subset of PPS′) →
(a subset of PMS′)),
.....
((a subset of PPS′) →
(a subset of PMS′))

이며 또한 g 는,

(a subset of POS′) → (a subset of PPS′)
⇒ ((a subset of POS′) →
(a subset of PPS′)),
.....
((a subset of POS′) →
(a subset of PPS′))

으로 분해될 수 있다.

위와 같은 분해는 갈로아 래티스(Galois lattice) [1] 구축 알고리듬을 사용하여 유도할 수 있다. 갈로아 래티스는 두 가지의 독립적인 래티스들로 이루어진 양방향 래티스로서 단일의 래티스로 두가지 형태의 정보구조를 표현할 수 있다는 특징이 있다. 이러한 특징은 한 래티스 셀안에 이질적인 요소들, 예를 들면 PPS′ 와 PMS′ 혹은 POS′ 와 PPS′의 관계를 표현할 수 있게 한다. 구축방법은 다음과 같다. 단 POS′, PPS′, PMS′ 는 각각 일반화 과정을 거친 가객체, 가속성, 가메소드들의 집합을 의미하며 f 와 g 는 식 (2), (4) 에서 소개된 대응관계이다.

1. 모든 일반화 규칙에 대해서 다음의 갈로아 래티스 구축방법을 적용한다.
   f: PPS′ → PMS′ 와 f′: PMS′ → PPS′ 와 같은 두 가지 대응규칙이 있다고 하자.
   그러면 PPS′의 멱집합의 모든 부분집합에 대해서,
   1.1 f(PPS′) ⊂ PMS′을 만족하는 PPS′을 찾는다.
   1.2 f′(f(PPS′)) = PPS′* ⊂ PPS′을 만족하는 PPS′*를 찾는다.
   1.3 f(PPS′*) = PMS′* ⊂ PMS′을 만족하는 PMS′*을 찾는다.
   1.4 (PPS′* ,PMS′*)을 얻는다.
2. 모든 가속성집합에 대해서 다음의 갈로아 래티스 구축방법을 적용한다.
   g:POS′ → PPS′와 g′:PPS′ → POS′와 같은 두 가지 대응규칙이 있다고 하자.

그러면 POS′의 멱집합의 모든 부분집합에 대해서,

2.1 $g(POS'_1) \subset PPS'$를 만족하는 PPS'_1을 찾는다.

2.2 $g'(g(POS')) = POS'^* \subset POS'$을 만족하는 POS'^*를 찾는다.

2.3 $g(POS'^*_1) = PPS'^*_1 \subset PPS'$을 만족하는 PPS'^*_1을 찾는다.

2.4 (POS'^*_1, PPS'^*_1)을 얻는다.

여기서 최종적으로 획득된 PPS'^*_1, PMS'^*_1과 POS'^*_1들은 각각 (1) 분해되고 일반화된 가속성 집합의 부분집합, (2) 분해되고 일반화된 가메소드집합의 부분집합, (3) 분해되고 일반화된 가객체집합의 부분집합이다. 위의 1.4 와 2.4 에서 얻어지는 결과물을 래티스 이론에서는 클리크쌍(clique couple)이라고 하며 그 의미는 다음과 같다. 예를 들어 단계 1.4 에서 획득된 (PPS'^*_1, PMS'^*_1) 중 하나가 ({DEM(DR),T(SR,DR)},{D(DR)}) 라면 여기서 다음과 같은 해석이 가능하다. 가속성과 가메소드는 "contains"라는 관계가 있는 것이 메타 기술에 의해 정의되었으므로 이들의 lattice cell 들은 "contains"관계를 나타낸다.

(1) DEM(DR) 과 T(SR,DR) 이라고 하는 가속성 요소들이 모두 포함되어 있는 가메소드는 D(DR) 이다.

(2) 역으로 D(DR) 라고 하는 가메소드가 포함하고 있는 가속성은 DEM(DR)과 T(SR,DR) 이다.

이들은 다음의 규칙 생성기에 의해서 모형의 선택을 위한 If−then 규칙으로 변환된다.

4.3. 규칙 생성기 (Rule Generator)

앞의 모형 분해/일반화 과정을 통하여 획득된 갈로아 래티스로부터 식3), 5)에 의거하여 모형 선택의 규칙을 유도할 수 있다. 먼저 앞에서 획득된 $(POS'^*_1, PPS'^*_1) = (\{PO'^*_{1,1}, \cdots\cdots, PO'^*_{1,s}\}, \{PP'^*_{1,1}, \cdots\cdots, PP'^*_{1,t}\})$ 부터의 규칙 변환 방법은 다음과 같다.

IF $PO'^*_{1,1} \wedge$
$PO'^*_{1,2} \wedge$
……
$PO'^*_{1,s}$
THEN $PP'^*_{1,1} \vee$
$PP'^*_{1,2} \vee$
……
$PP'^*_{1,t}$

이 규칙은 만약 모형화를 위해 $PO'^*_{1,1}$, …, $PO'^*_{1,s}$의 s개의 가객체가 선정되었다면 이들 가객체들을 모두 포함하는 가속성들은 $PP'^*_{1,1}$, …… , $PP'^*_{1,t}$ 의 t 가지가 있다는 것을 의미한다. 따라서 모형 작성자는 이들 가속성들 중에서 필요한 몇가지를 선별하여 모형화에 사용할 것이다. 그러므로 IF 부분은 $\wedge$ 관계로, THEN 부분은 $\vee$ 관계로 표현하였다.

한편 $(PPS'^*_1, PMS'^*_1) = (\{PP'^*_{1,1}, \cdots\cdots, PP'^*_{1,t}\}, \{PM'^*_{1,1}, \cdots\cdots, PM'^*_{1,v}\})$는 다음과 같이 변환된다.

IF $PP'^*_{1,1} \wedge$
$PP'^*_{1,2} \wedge$
……
$PP'^*_{1,t}$
THEN $PM'^*_{1,1} \vee$
$PM'^*_{1,2} \vee$
……
$PM'^*_{1,v}$

```
TYPE: (a subset of POS)
  SUPTYPE-OF : (a subset of POS')          // IsA(PO)
  ATTRIBUTES
    g(a subset of POS)                     // PP(s)
  OPERATIONS
    f(g(a subset of POS))                  // PM(s)
    g          // POS 의 부분집합에 대한 g 규칙(들)
    f          // POS 의 부분집합에 대한 f 규칙(들)
```

[그림 5] 객체지향적 정의

4.4 모형베이스로의 저장

앞의 세가지 과정을 통해서 추출되는 모형에 관한 정보를 정리하면 다음과 같다.

(1) 가객체집합
(2) 가속성집합
(3) 가메소드집합
(4) IF(가속성집합의 부분집합) THEN (가메소드집합의 부분집합)
(5) IF(가객체집합의 부분집합) THEN (가속성집합의 부분집합)

각 객체집합에 있는 모든 가객체(또는 가객체집합의 부분집합)에 대해서 객체지향 모형베이스에 저장될 수 있는 것은 (1) PO, (2) g(PO), (3) f(g(PO)), (4) f, (5) g 라고 볼 수 있으며, 이때 POS 의 각 부분집합에 대해서 [그림 5]와 같은 형태의 저장이 가능하다.

5. 예 제

앞에서 설명한 방법론을 설명하기 위하여 다음과 같은 예를 들고자 한다. 제조업체에서 제품을 생산하고 수요처까지 수송하는 것은 가장 기본적인 문제이며 시간의 흐름에 따라 반복적으로 고려되어야 하는 부분이다. 이 문제를 수리적으로 해결하기 위해 모형 전문가가 MODLER [8] 와 FW/SM [6] 이라고 하는 모형화 지원 도구를 사용하여 [그림 6]의 (a),(b)와 같이 각각 수송 모형과 생산 모형을 작성했다고 하자. [그림 6] (a)의 수송모형은 혼합물을 BSR이라고 하는 공급처와 BDR이라고 하는 수요처로 이동시키는 비용을 최소화하는 모형이며, 이때 SUP(BSR), DEM(BDR), COST(BSR, BDR)은 각각 공급량과 수요량, 그리고 단위당 수송비용을 의미한다. 한편 [그림 6] (b)의 생산계획모형은 제품(PROD)들을 해당 제작기계(MACH)와 특정 생산모드(MODE)에서 생산하는 것에 관계된 것이다. 이때 PCOST, X, T, AVAIL, Z, DEM 은 각각 생산단가, 생산량, 생산시간, 제작기계에 대한 가용시간, 판매량, 그리고 수요량을 의미한다. 이때 이 두가지 원천모형들로부터 모형 요소들을 추출하여 후의 계속적인 재사용을 위해 모형베이스에 저장하려고 한다. 본 예제에서는 역모형화를 통한 모형베이스의 구축 및 규칙의 생성과정을 모형화 도구에 대한 메타기술로부터 시작하여 원천

모형의 일반화 및 분해, 재사용을 위한 규칙의 생성에 이르는 5가지 단계를 소개한다.

```
SETS
  BSR blend supply region
  BDR blend demand region
TABLES
  SUP(BSR) available supply at (BSR) : 0/*
  DEM(BDR) required demand at (BDR) : 0/*
  COST(BSR,BDR) cost to transport 1 unit of blend from (BSR) to (BDR) : 0/*
ACTIVITIES
  T(BSR,BDR) blend transports from (BSR) to (BDR)
EQUATIONS
  S(BSR) limits supply at (BSR) = T(BSR,BDR) <= SUP(BSR)
  D(BDR) requires demand at (BDR) = T(BSR,BDR) >= DEM(BDR)
  COST total transportation cost = COST(BSR,BDR)*T(BSR,BDR)
ENDATA
```

(a) MODLER 형태의 수송모형

```
PRODj /pe/ There is a list of PRODUCTS.
&PRODUCTION PRODUCTION DATA
  MACHk /pe/ There is a list of MACHINES.
  MODEi /pe/ There is a list of production MODES.
  PCOST (MODEj, PRODj, MACHk) /a/ {MODE} * {PROD} × {MACH} production cost
  X (MODEi, PRODj, MACHk) /va/ {MODE} × {PROD} × {MACH} : REAL + production quantity
  T (MODEi, PRODj, MACHk) /a/ {MODE} × {PROD} × {MACH} : REAL + time standard
  AVAIL (MODEi, MACHk) /a/ {MODE} × {MACH} : REAL + time availability
  T:AVAIL (AVAILik) /t/ {MODE} × {MACH} ; @SUMj (Tijk * Xijk) <= AVAILik
&SALES SALES DATA
  Z (PRODj) /va/ {PROD} sales quantity
  DEM (PRODj) /a/ {PROD} : REAL+ demand (units)
  T:DEM (Zj, DEMj) /t/ {PROD} : Zj >= DEMj demand test
```

(b) SML 형태의 생산계획모형

[그림 6] 두 예제 모형에 대한 기술

5.1. 메타 처리기

모형 요소를 추출하기 전에 먼저 MODLER와 SML 에 대한 메타 정의가 준비되어야 하는데 여기서는 일단 MODLER 에 대한 정의만을 보인다.

5.1.1 MODLER 에 대한 기술

[그림 3]의 선형계획 모형에 대한 일반적인 메타기술을 MODLER 의 상황에 맞게 다시 메타 기술할 수 있는데 그 일부는 다음의 [그림 8]과 같다. 이는 [그림 3]의 선형계획 모형에 대한 메타기술에 [그림 7]과 같은 대응 규칙을 적용하여 나온 것으로 이를 토대로 하여 MODLER에 의해 작성되는 모형들을 해석하는 해석기를 생성한다.

```
MODLER ⇒ LP;
SETS ⇒ POS;
TABLES ⇒ PPS;
ACTIVITIES ⇒ PPS;
EQUATIONS ⇒ PMS;
SET ⇒ PO;
TABLE ⇒ PP;
ACTIVITY ⇒ PP;
EQUATION ⇒ PM;
```

[그림 7] MODLER 형태의 모형 해석을 위한 대응 규칙

```
ENTITY MODLER;
ENTITY SETS;
..... 중략 .....
RELATIONSHIP is part of rel1;
    PART part EN SETS;
        (partial : M)
    aggregation part EN MODLER;
RELATIONSHIP is part of rel2;
    PART part EN TABLES;
        (partial : M)
    aggregation part EN MODLER;
..... 중략 .....
RELATIONSHIP is a rel;
    PART children part EN SET;
        (partial : M)
    parent part EN SET;
```

[그림 8] Modler 에 대한 메타 기술

5.1.2 원천모형의 인식

5.1.1에 의해 정의된 MODLER 에 대한 메타기술을 통해서, [그림 6]의 (a)와 같은 수송모형(LP)으로부터 가객체집합, 가속성집합, 가메소드집합을 추출할 수 있다. 이때 [그림 8]의 메타기술에 의하여 SETS에 있는 BSR과 BDR은 POS 의 요소로 인식되며, TABLES 와 ACTIVITIES 의 supply(BSR), demand(BDR), cost(BSR,BDR), t(BSR,BDR) 는 PPS 의 요소로, s(BSR), d(BDR), cost 는 PMS 의 요소로 각각 결정된다. 이를 앞의 식 (1) 에 의거하여 표시하면,

$$\begin{aligned} &LP = \{ POS, PPS, PMS \} \\ &\text{단, } POS = \{ BSR, BDR \} \\ &\quad PPS = \{ SUP(BSR), DEM(BDR), \\ &\qquad COST(BSR,BDR), T(BSR,BDR) \}, \\ &\quad PMS = \{ S(BSR), D(BDR), COST \} \end{aligned} \tag{7}$$

와 같으며, 여기서 POS 와 PPS 는 메타기술의 contains 관계에 의해 다음을 유도할 수 있다.

a. (SUP(BSR), T(BSR,BDR)) → (S(BSR))
b. (DEM(BDR), T(BSR,BDR)) → (D(BDR))
c. (COST(BSR,BDR), T(BSR,BDR)) → (COST) (8)

한편 PPS 의 각 요소들과 POS 는 다음과 같은 관계를 가지고 있다.

a. (BSR) → (S(BSR), SUP(BSR))
b. (BDR) → (D(BDR), DEM(BDR))
c. (BSR, BDR) → (COST(BSR,BDR), T(BSR,BDR)) (9)

5.2 모형 분해 / 일반화

5.2.1 모형의 일반화

예를 들어 BDR 과 BSR 이 각각 DR 과 BR 로 일반화될 수 있었다고 하면, (8)의 세 규칙은 각각 (10)의 규칙 a,b,c 로 변하게 된다. 한편 [그림 6]의 (b)에 있는 SML 형식의 모형들에 대해서도 동일하게 적용을 하면 (10)의 규칙 d,e 가 유도된다.

a. (SUP(SR), T(SR,DR)) → (S(SR))
b. (DEM(DR), T(SR,DR)) → (D(DR))
c. (COST(SR,DR), T(SR,DR)) → (COST)
d. (TM(MODE,PROD,MACH), X(MODE, PROD, MACH), AVAIL(MODE,MACH)) → (T:AVAIL(MODE,MACH))
e. (Z(PROD), DEM(PROD)) → (T:DEM(PROD)) (10)

마찬가지로 (9) 에 대해서도 아래의 (11)과 같은 일반화가 이루어질 수 있다.

a. (SR) → (S(SR), SUP(SR))
b. (DR) → (D(DR), DEM(DR))
c. (SR, DR) → (COST(SR,DR), T(SR,DR))
d. (PROD) → (Z(PROD), DEM(PROD))
e. (MODE, MACH) → (AVAIL(MODE, MACH))
f. (MODE, PROD, MACH) → (TM (MODE,PROD,MACH), X(MODE, PROD,MACH)) (11)

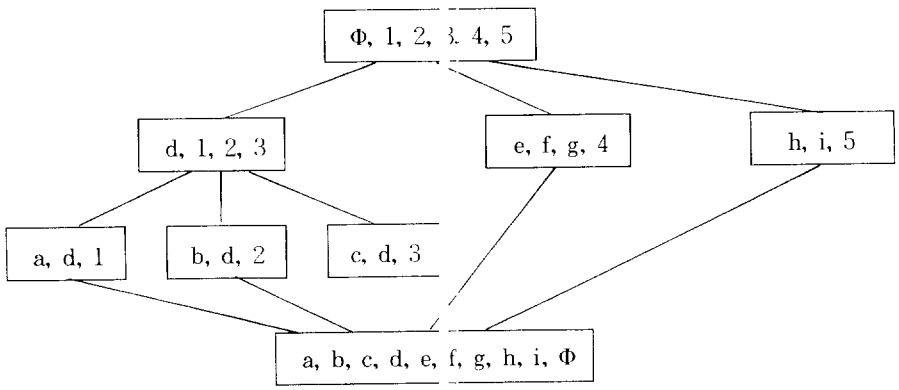
5.2.2 모형의 분해: 갈로아 래티스 구축

적절한 규칙들이 선정되었으면, 새 예제는 갈로아 래티스로 추가된다. [그림 9]에서 소개되

어있는 구축 알고리듬 과정을 거쳐서 갈로아 래티스의 형태를 결정짓게 된다. [그림 9]에서 획득된 (PPS_1^*, PMS_1^*)에 의해 [그림 10]과 같은 갈로아 래티스가 만들어진다.

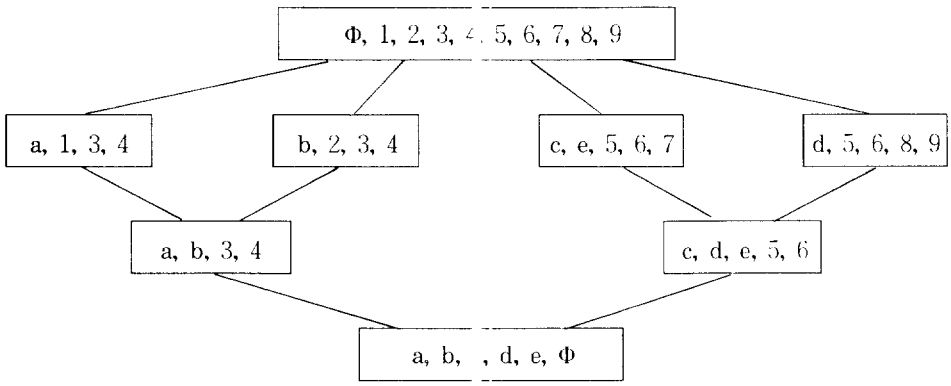
PPS1	f(PPS1)	$f'(f(PPS_1)) = PPS_1^*$	$f(PPS_1^* = PMS_1^*)$	(PPS_1^*, PMS_1^*)
Φ	1,2,3,4,5	Φ	1,2,3,4,5	(Φ,1,2,3,4,5)
a	1	a,d	1	(a,d,1)
b	2	b,d	2	(b,d,2)
c	3	c,d	3	(c,d,3)
d	1,2,3	d	1,2,3	(d,1,2,3)
e	4	e,f,g	4	(e,f,g,4)
f	4	e,f,g	–	–
g	4	e,f,g	–	–
h	5	h,i	5	(h,i,5)
i	5	h,i	–	–
a,d	1	a,d	–	–
a,b,……,i	Φ	a,b,……,i	Φ	(a,b,……,i,Φ)

[그림 9] 가속성-가메소드 갈로아 래티스의 구축 과정



a. SUP(SR) b. DEM(DR) c. COST(SR,DR)
d. T(SR,DR) e. X(MODE,PROD,MACH) f. TM(MODE,PROD,MACH)
g. AVAIL(MODE,MACH) h. Z(PROD) i. DEM(PROD)
1. S(SR) 2. (DR) 3. COST
4. T : AVAIL(MODE,MACH) 5. T : DEM(PROD)

[그림 10] 가속성-가메이드 갈로아 래티스



1. SUP(SR) 2. DEM(DR) 3. COST(SR,DR)
4. T(SR,DR) 5. X(MODE,PROD MACH) 6. TM(MODE,PROD,MACH)
7. AVAIL(MODE,MACH) 8. Z(PROD) 9. DEM(PROD)
a. SR b. DR c. PROP
d. MODE e. MACH

[그림 11] 가객체-가속성 갈로아 래티스

또한 5.2.1의 (11)의 가객체 집합과 가속성집합간의 일반화 규칙에 대해서도 동일한 적용이 가능한데 그 결과만 보이면 [그림 11]과 같다.

5.3. 모형 선정 규칙의 추출

일단 갈로아 래티스가 구축되었으면 모형을 선택할 수 있는 규칙을 자동적으로 생성할 수 있다. 이 예에서는 [그림 10, 11]의 래티스에 근거하여 다음과 같은 규칙들이 만들어진다.

(1) 가속성-가메소드 규칙 (f):

규칙1: IF (SUP(SR)) ∧ (T(SR,DR))
THEN (S(SR))

규칙2: IF (DEM(DR)) ∧ (T(SR, DR))
THEN (D(SR))

규칙3: IF ((COST(SR,DR)) ∧ (T(SR, DR))
THEN (COST)

..... 중략 .....

규칙12: IF (T(SR,DR))
THEN (S(SR)) ∨ (D(SR)) ∨ (COST)

(2) 가객체-가속성 규칙 (g):

규칙13: IF (SR)
THEN (SUP(SR)) ∨ (COST (SR,DR)) ∨ (T(SR,DR))

규칙14: IF (DR)
THEN (DEM(DR) ∨ (COST (SR,DR)) ∨ (T(SR,DR))

규칙15: IF (BSR)
THEN (SUP(BSR)) ∨

(COST(BSR,BDR)) ∨(T(BSR, BDR))

..... 중략 .....

규칙24: IF (MODE) ∧ (MACH) ∧ (PROD)
THEN (X(MODE,PROD, MACH)) ∨ (TM(MODE, PROD,MACH))

이상에 의해서 추출된 모형 요소들은 객체 지향적으로 표현되어 저장된다. [그림 12]는 하나의 예로서 BSR 에 대한 추출 지식을 보여주고 있다. 이는 BSR이라고 하는 객체가 SR이라고 하는 클래스를 가지며 그의 속성으로 SUP_BSR을 보유함을 보인다. 또한 BSR에 관계된 15번 및 4번 규칙과 공급량에 관련된 제약식이 메소드에 존재하고 있다. 이때 속성과 메소드는 모두 생성된 규칙에 의해 선정된다. 이러한 형태가 가색체집합의 모든 원소에 대해서 인식되며 이들은 모형베이스에 저장된다.

```
OBJECT BSR // Blend Supply Region
  CLASS : SR // Supply Region
  PROPERTIES
    SUP_BSR : integer // g(BSR)
  METHODS
    S_BSR( ): // f(g(BSR)) =
    f(SUP_BSR)
    Rule15( ):
    Rule4( ):
```

[그림 12] 모형 재사용 부분의 표현 예 (BSR)

6. 결 론

모형의 재사용성은 객체지향 모형관리시스템을 비롯한 여러 연구에서 제시되어 오고 있다. 그런데 이와 아울러 모형화 환경에서 모형베이스를 구축하고자 할 때 기존의 정보자원으로부터 재사용이 가능한 것들을 추출하도록 지원하여 구축과정에서 소요되는 노력을 경감시키는 것이 필요하다. 이 논문에서는 원천모형으로부터 모형 지식을 추출하여 모형베이스로 등록을 하는 방법에 대해 선형계획 모형에 국한하여 제시하였다. 이러한 방법은 첫째, 여러 가지 형태의 모형화 도구로부터 작성된 모형들을 단일의 표현방식으로 변환할 수 있다는 점과 둘째, 기존의 모형들을 객체 지향화하는 작업을 지원함으로써, 객체 지향 모형관리시스템의 구축에 장애 요소였던 최초의 객체 지향적 지식의 습득 노력을 경감시킬 수 있기 때문에 모형의 재사용성의 향상이라는 측면에서 의미가 있다.

이러한 의미를 가지고 본 논문에서는 현존하는 모형들을 재사용하기 위해 모형베이스와 모형 선정을 위한 규칙베이스의 구축과정을 지원하여 모형베이스 관리 기능을 구축하는 방법으로서의 역모형화에 대한 소개를 하는 것을 목적으로 하였으며, 우선 그 가능성을 보이기 위하여 선형계획모형만을 대상으로 하였다. 그러나 현재 사용가능한 모형화 도구들이 선형계획모형만을 대상으로 하지 않고 보다 일반적인 수리적 모형으로 확장되고 있을 뿐더러 이들에 대해서도 메타 정의가 가능하기 때문에 실제적인 구현상의 문제점을 극복한다면 정수계획법등 다른 형태의 모형으로도 확장 가능하다. 현재 다른 유형의 모형들로의 확장에 대한 연구와 아울러 메타 시스템인 K-META 시스템과 ONTOS 데이타베이스를 기반으로 하여 RMT(Reverse Modeling Tool)라고 하는 프로토타입 시스템을 개발중에 있다.

참 고 문 헌

[1] Birkhoff, G., *Lattice Theory*, American Mathematical Society, Rhode Island, 1967.

[2] Caldiera, G. and V.R. Basili, "Identifying and Qualifying Reusable Software Components," *IEEE Computer* (1991), pp. 61-70.

[3] Chikofsky, E.J., and J.H Cross Ⅱ, "Reverse Engineering and Design Recovery: A Taxonomy," *IEEE Software* (Jan., 1990), pp. 13-17.

[4] Prieto-Diaze, R. and P. Freeman, "Classifying Software for Reusability," *IEEE Software*, Vol. 4, No. 1 (1987), pp. 6-16.

[5] Franze, L.S., "Data Driven Modeling," *Decision Science*, Vol. 20, No.2 (1989), pp. 359-376.

[6] Geoffrion, A.M., "Reusing Structured Models via Model Integration," *Proceedings of the Twenty Second Annual Hawaii International Conference on System Sciences*, Vol. 3 (1989), pp. 601-611.

[7] Geoffrion, A.M., "FW/SM: A Prototype Structured Modeling Environment," *Management Science*, Vol. 37, No. 12 (1991), pp. 1513-1538.

[8] Goguen, J.A., "Reusing and Interconnecting Software Components," *Computer*, Vol. 19, No. 2 (1986), pp. 16-28.

[9] Greenberg, H.J., *A Primer for MODLER: Modeling by Object-Driven Linear Elemental Relations*, University of Colorado at Denver, Feb., 1991.

[10] Griswold, W.G., and D. Notkin, "Computer-Aided vs. Manual Program Restructuring," *Software Engineering Notes*, Vol. 17, No. 1 (1992), pp. 33-41.

[11] Hee, K.M., L.J. Somers, and M. Voorhoeve, "A Modeling Environment for Decision Support Systems," *Decision Support Systems*, No. 7 (1991), pp. 241-251.

[12] Huh, S.Y., "An Object-Oriented Model Management Framework for Decision Support Systems," Working Paper, Dept. of Information Systems, UCLA, 1991.

[13] Kim, T.G., C. Lee, E.R. Christensen, and B.P. Zeigler, "System Entity Structuring and Model Base Management," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 5 (1990), pp. 1013-1024.

[14] Kim, D.H and S.J. Park, "Model Management Using an Extended Object-Oriented Data Model," Working Paper, Dept. of Management Science, KAIST, Korea, 1992.

[15] Konsynski, B. and R.H. Sprague, "Future Research Directions in Model Management," *Decision Support Systems*, Vol. 2 (1986), pp. 103-109.

[16] Langerman, R.G., and C.A. Grasso, "Software Engineering with Reusable

Designs and Code," *IEEE Transactions on Software Engineering*, SE-10, No. 5 (1984), pp. 498-501.

[17] Lee, J.S., "A Model Base for Identifying Mathematical Programming Structures," *Decision Support Systems*, Vol. 7 (1991), pp. 99-105.

[18] Lenard, M.L., "An Object-oriented Approach to Model Management," *Proceeding of the Twentieth Annal Hawaii International Conferene on System Sciences*, Vol. 3 (1987), pp. 509-515.

[19] Liang, T.P., "Modeling By Analogy: A Case-Based Approach to Automated Linear Program Formulation," *25th HICSS*, 1991, pp. 276-283.

[20] Mannino, M.V., B.S. Greenberg, and S. N. Hong, "Model Libraries: Knowledge Representation and Reasoning," Working Paper, 1987.

[21] Mitchell, T.M., *Version Spaces: An Approach to Concept Learning*, Report No. STAN-CS-78-711, Computer Science Department, Stanford Univerity.

[22] McCullough, P., B. Atkinson, A. Goldberg, M. Griss, and J. Morrison, "Reuse: Truth or Fiction (PANEL)," OOPSLA'92, pp. 41-44.

[23] McIloy, "Mass Produced Software Components," *Proc. NATO Conf. Software Engineering*, New York, 1969, pp. 88-98.

[24] Meyer, B., "Reusability: The Case for Object-Oriented Dsign," *IEEE Software*, Vol. 4, No.1 (1987), pp. 50-64.

[25] Muhanna, W.A. and R.A. Pick, "*A Systems Framework for Model Management*," Working Paper, Dec., 1986.

[26] Muhanna, W.A., "An Object-Oriented Framework for Model Management and DSS Development," Working Paper, The Ohio State University, 1991.

[27] Murphy, F.H., E.A. Stohr, and P.C. Ma, "Composition Rules for Building Linear Programming Models from Component Models," *Management Science*, Vol. 38, No. 7 (1992), pp. 948-963.

[28] Page, T.W., S.E. Berson, W.C. Cheng, and R.R. Muntz, "An Object-Oriented Modeling Environment," *OOPSLA'89*, October, 1989.

[29] Park, S.J., "A Study on the Development of Software Tool Generation System," Dept. of Management Science, KAIST, 1991.

[30] Park, S.J., "A Study on the Development of Intelligent Integrated CASE Tool Generator," Dept. of Management Science, KAIST, 1992.

[31] Podgurski, A. and L. Pierce, "Behaviour Sampling: A Technique for Automated Retrieval of Reusable Components," *14th Conference on Software Engineering*, May, 1992, pp. 349-360.

[32] Shaw, M.J., P.L. Tu, and P. De, "Applying Machine Learning to Model Management in Decision Support Systems," *Decision Support Systems*, Vol. 4 (1988), pp. 285-305.

[33] Stacy, W., R. Helm, G.E. Kaiser, and B.

Meyer, "Ensuring [illegible] of Reusable Objects (PANEL)," *OOPLA'92* 1992, pp. [illegible].

[34] Tung, L., R.G. [illegible] and R.D. [illegible], "Model Integration [illegible] Object Oriented Model Management Systems," *The 24th HICSS*, 1991, pp. [illegible].

[35] Yasdi, R., "Learning classification rules from Database [illegible] Context of Knowledge Acquisition and Representation," *IEEE Transactions on Knowledge and Data Engineering* Vol. 3, No. 3 (1991), pp. [illegible].