

# 의미웹 환경에서 조건부함축 제약 지식표현을 위한 SWCL의 확장

김 학 진<sup>†</sup>  
연세대학교 경영대학

## An Extension of SWCL to Represent Logical Implication Knowledge under Semantic Web Environment

Hak-Jin Kim  
School of Business, Yonsei University

### ■ Abstract ■

By the publications of RDF and OWL, the Semantic Web is confirmed as a technology through which information in the Internet can be processed by machines. The focus of the Semantic Web study after then has moved to how to provide more useful information to users for their decision making beyond simple use of the structured data in ontologies. SWRL that makes logical inference possible by rules, and SWCL that formulates constraints under the Semantic Web environment are some of many efforts toward the achievement of that goal. Constraint represents a connection or a relationship between individual data in ontology. Based on SWCL, this paper tries to extend the language by adding one more type of constraint, implication constraint, in its repertoire. When users use binary variables to represent logical relationships in mathematical models, it requires and knowledge on the solver to solve the models. The use of implication constraint ease this difficulty. Its need, definition and relevant technical description is presented by the use of the optimal common attribute selection problem in product design.

Keywords : Semantic Web, SWCL, Constraints, Implication, Decision Making

## 1. 서 론

의미웹은 웹에 존재하는 정보를 기계가 처리할 수 있도록 하는 개념으로 제안되었다. 의미웹에 대한 연구는 주로 정보를 웹에 어떻게 저장하고 표현할 것인가에 대한 이슈를 중심으로 URI(Uniform Resource Identifier)[6]와 XML(Extensible Markup Language)[8]을 바탕으로 RDF(Resource Description Framework)[17]과 OWL(Web Ontology Language)[4]과 같은 표준이 권고안으로 발표를 통해 표준적인 기술로 정착되었다. 이후 연구는 저장된 정보를 바탕으로 이 정보를 어떻게 효과적으로 사용하여 사용자의 의사결정에 도움을 줄 수 있을지를 모색하는 방향으로 그 중심이 옮겨지고 있다[7]. 2004년 이와 같은 목적으로 SWRL(Semantic Web Rule Language)가 W3C에 의해 제안되었다[11]. 혼 논리(Horn Logic) 기반의 언어인 RuleML을 온톨로지 언어 OWL과 협력하여 정보간의 규칙을 표현하고 이를 정보처리 과정을 통하여 새로운 사실들을 표현하는 규칙을 도출하도록 하였다. 나아가 RIF(Rule Interchange Format)의 제안[12]을 통해서 데이터 간의 논리적 관계를 표현하는 규칙 데이터를 하나의 시스템에 머무르는 것이 아니라 다수의 규칙 시스템 간의 교환과 공유를 가능하게 하였다. 이 연구를 통해 웹상에서 사용자가 의도하는 것과 연관되어 있는 파편적인 데이터를 추출해내는데 그치지 않고, 의도와 관련된 규칙 데이터 역시 수집하고 이를 이용하여 수집된 데이터를 정보 처리할 수 있게 함으로써 보다 진보된 정보를 사용자로 하여금 웹 데이터로부터 얻어낼 수 있도록 한다.

하지만 현재까지 W3C에 의해 제안된 기술만으로는 웹 상에서 사용자의 진정한 의사결정을 위한 과정으로는 부족한 점이 있는 것이 사실이다. 더군다나 의사결정의 문제는 의미웹 밖의 여러 분야에서 역사적으로 많이 연구되어 왔던 주제이므로 그 축적된 연구를 지렛대 삼아 의미웹에서의 의사결정 문제를 보는 것이 더욱 타당하다고 할 수 있다. 이런 관점에서 모형 관리[15, 16, 19]와 같은 타 분야

에서 데이터 개체간의 관계를 표현하기 위해 사용하는 제약식 개념의 도입은 의미웹 연구의 발전에 큰 도움이 된다고 볼 수 있다. SWCL(Semantic Web Constraint Language)은 사용자의 의사결정 목표와 그 문제에 연관된 데이터의 관계를 제약식으로 기술하게 함으로써 수리적 모형을 구축할 수 있게 하는 RDF 기반의 언어 표현력을 제공한다[14]. 즉 온톨로지를 통해 선언된 데이터의 개념과 속성들과 함께 데이터 간의 관계를 표현한 제약식 또한 데이터로 표현되어, 사용자가 의사결정을 할 시에 OWL, SWRL 그리고 SWCL 기술을 사용하여 필요한 온톨로지와 규칙과 제약식 데이터를 모아 의사결정 모형을 구축하고 모형을 풀음으로써 사용자에게 유익하면서 원 데이터에서는 구할 수 없는 정보를 제공할 수 있도록 한다. 이 논문은 의미웹에서의 제약식 언어인 SWCL에 또 한 가지 구성요소를 추가함으로써 새로운 제약식을 표현할 수 있도록 함을 목적으로 한다. 구체적으로 부연하면 의사결정 모형으로서의 혼합정수계획 모형에서 일반적으로 이진 변수의 사용은 문제에서 나타나는 논리적인 관계를 대수적으로 묘사하기 위해 사용된다. 그리고 이와 같은 과정은 일반적으로 모형을 풀어주는 해결기의 구조 상 대수적인 알고리즘을 사용하고 모형의 대수적인 표현이 해결기의 이해에 부합하기 때문에 나타나는 현상이다. 하지만 대수적인 표현은 사용자에게 혼동을 주기 쉽고 내용을 이해하기 힘들게 한다는 어려움이 있다. 이런 이진 변수를 이용한 대수적인 표현 대신 논리적인 도구인 조건부함축(Implication)의 관계를 사용한다면 이 어려움을 훨씬 완화할 수 있다. 따라서 이 논문에서는 조건부함축을 포함하는 조건부함축 제약식을 SWCL에 도입할 것을 제안하고 시스템 상에 확장된 형태의 SWCL로 표현된 모형이 해결기와 연결 시 어떤 변환 과정을 거쳐야 하는지를 제품설계의 최적 공통속성 선택의 문제를 예로 설명하고 시스템의 구조를 제시한다.

이 논문은 다음과 같이 구성된다. 이 장 이후 제 2장에서 관련연구가 소개되고, 제 3장에서 조건부

합축 제약식과 이를 통한 SWCL의 확장에 대한 그 필요성, 정의 그리고 제품설계 최적 공통속성 선택 문제의 예를 통한 조건부합축 제약식의 표현에 대해 논한다. 제 4장에서는 표현된 조건부합축 제약식을 전체 시스템적으로 어떤 기술적인 과정을 통해 해결할 수 있는지를 논한다. 제 5장에서는 제기된 제품설계 최적 공통속성 선택 문제의 예에서 구체적인 데이터를 이용하여 제안된 시스템으로 문제 풀이가 이루어진 결과를 보여준다. 마지막으로 결론과 향후 연구 과제에 대해 논한다.

## 2. 관련 연구

최근 의미웹의 연구는 사용자의 의사결정을 위해 웹에 저장된 데이터를 어떻게 이용하는가의 이슈로 관심이 옮겨지고 있다. 논리와 규칙을 이용하고자 하는 시도로 SWRL이 제안되었고[11], 최근 제약식들의 표현을 이용해서 데이터를 추출하고 이로부터 저장된 웹 데이터를 필요한 응용문제들에 적용하는데 관심을 기울이게 되었다. 특히 제약식은 개체와 개체 간의 관계를 보는 패러다임으로서, 전통적으로 의사결정 분야에서 문제를 표현하는 데에도 강점을 지니고 있는 것으로 알려져 왔고 최근 의미웹에서도 관심이 부각되었다. SWRL 기반 위에 제약식을 표현하는 CIF/SWRL[10]의 제안에서 제약식 언어인 Colan[5]와 이 위에서 작동하는 데이터 모형을 제시하고 RDF 스키마를 통해 제약식 데이터 모형을 사용할 수 있는 인터페이스를 제시하였다. 또 다른 연구는 SWRL 위에서 이루어지고 있는 제약식의 표현상에 어떻게 소프트 제약식(Soft Constraint)을 구현할 수 있는지에 대해 논하였다[18]. 즉 문제에서 반드시 충족되어야 하는 제약식이 아니라 충족시키는 정도를 감안하여 모든 제약식에 대한 해가 존재하지 않을 경우 제약식 중에 중요성이 낮은 제약식을 제거함으로써 좀 더 약한 문제의 표현에서 만족하는 해를 찾을 수 있도록 제약식에 강제성의 정도가 부여한 개념을 적용한 것이다. 하지만 CIF는 데이터베이스의 개념에 바탕을 두고 있어 제약식으

로서의 표현에 한계가 있다. 즉 CIF는 제약식 문제를 표현하기 위해 FOL 기반의 논리적 연산자를 활용한 CSP(Constraint Satisfaction Problem)를 해결하는데 초점을 맞추고 있지만, 의미웹 환경에서 의사결정 분야에서 오랜 시간 다루어져 온 최적화 모형을 포함하여 다양한 형태의 의사결정 모형을 표현할 수가 없다.

이 한계를 극복하기 위해 Kim et al.[13]와 이명진 등[3]은 RDF를 기반으로 수리적 제약식 모형을 표현할 수 있는 SWCL을 제안하였다. 인터넷에서 일반 구매자의 구매 과정을 분석하고, 의미웹 환경에서 일반 구매자의 의사결정을 도울 수 있는 의사결정 에이전트를 구축할 수 있도록 인터넷 쇼핑물 예를 통하여 논하였다. 또한 Kim et al.[14]에서는 의미웹에서 RDF, OWL, SWRL과 같은 개념들과 함께 구매자의 의사결정 지원을 위해 온톨로지를 이용하여 구매자의 의사결정 목표를 수리 모형화하고, 이를 풀어 최적의 조합을 구매자에게 보여줄 수 있는 의사결정 지원 시스템 구축을 보여주었다. 즉 의미웹에서 OWL을 기반으로 제약식을 기술하기 위한 모형 언어를 제안하고, 이 언어를 통하여 기존의 OWL 상의 개념이나 속성과 같은 데이터들을 제약식과 어떻게 연결하여 표현할 수 있을 지를 제시하였다. 이와 같은 수리 모형에 대한 표준적인 언어를 이용함은 의미웹 환경에서 의사결정과 관련된 문제해결을 지원하기 위해 제약식을 표현하고 이해하는데 매우 유용함을 보였다. 이명진 등[2]은 SWCL의 모형 언어적인 측면에 주목하여 여기에 또 하나의 표현 요소를 더하였다. 즉 일반적으로 의사결정에 활용될 수 있는 여러 가지 형태의 비선형적 함수의 표현을 원활하게 하기 위하여 새로운 구성요소를 더한 SWCL의 확장을 제안하고 부분선형 형태의 제약식을 처리할 수 있도록 하였다.

이 연구는 SWCL의 연구에서 한 걸음 나아가 의사결정을 위해 사용자가 모형을 구축하는데 어려운 이진변수를 사용한 모형화 과정을 쉽게 하기 위해 SWCL에 새로운 제약식의 추가를 제안하고, 이를 통한 의사결정 모형화와 풀이를 위한 시스템적 틀을

제품설계 최적 공통속성 문제를 통해 그 활용을 보인다.

### 3. SWCL의 확장과 조건부합축 제약식

#### 3.1 조건부합축 제약식의 필요성

SWCL은 의미웹 기술을 확장하여 의미웹에서 저장하고 있는 구조화된 데이터를 가공함으로써 사용자에게 의미있는 정보를 제공하고자 하는 목적으로 만들어졌다. 특히 기존의 많은 의사결정에 관한 수리 모형들을 이루는 제약식들을 의미웹 상의 제약식 데이터로 구조화하여 의미웹 상의 데이터와 연결하여 시스템 상에 사용자의 필요에 따라 유연하고 중단없이 의사결정의 문제를 모형화하고 해결기와 연결하여 해법을 제시하는 방법을 제시한다. 다른 한편으로 많은 계량 의사결정 문제들은 수리계획적인 모형으로 표현되고 문제 해결기를 통해 문제의 해를 찾는다. 하지만 이론적으로 문제 해결기의 알고리즘적인 구조가 대수적인 기법에 의존하므로 의사결정 문제들의 표현은 문제 해결기로의 입력이 편리하도록 대수적인 표현으로 이루어져 왔고 이를 전제로 한다. 하지만 어떤 대수적인 표현들은 일반 사용자의 관점에서 사용에 혼란을 유도하고 정확한 표현을 이끌어 내는데 어려움을 갖게 한다. 예로서 고정비를 반영한 생산비의 문제에서 생산비는 다음과 같다.

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ F+cx & \text{if } x > 0 \end{cases} \quad (1)$$

고정비를 포함하는 생산비를 최소화하는 생산량  $x$ 를 결정하는 문제는  $\min\{f(x)|0 \leq x \leq M\}$ 와 같고 보통 이는 다음과 같은 혼합정수계획 모형을 통해 혼합정수계획 해결기로 해를 구한다.

$$\min \quad Fy+cx \quad (2)$$

$$\text{s.t.} \quad x \leq My \quad (3)$$

$$x \geq 0, y \in \{0,1\} \quad (4)$$

한편 식 (1)의 생산비는 다음과 같은 조건부합축 제약식으로 표현할 수 있다.

$$y=0 \rightarrow x=z=0 \quad (5)$$

$$y=1 \rightarrow x > 0, z = F+cx \quad (6)$$

이를 이용하면 최소 생산비 문제는 다음과 같은 동치의 수리계획 모형으로 표현될 수 있다.

$$\min \quad z \quad (7)$$

$$\text{s.t.} \quad y=0 \rightarrow x=z=0 \quad (8)$$

$$y=1 \rightarrow z = F+cx, x > 0 \quad (9)$$

$$0 \leq x \leq M \quad (10)$$

$$y \in \{0,1\} \quad (11)$$

식 (7)~식 (11)을 어떻게 푸는지에 대한 내용은 이 논문의 범위를 벗어나므로 논의에서 제외한다. 관심은 문제를 표현하는 식 (5)~식 (6)은 사용자의 관점에서 문제를 혼합정수계획으로 풀 때의 식 (2)~식 (4)의 모형에서처럼 이진변수를 이용한 대수적인 표현보다 훨씬 직관적이고 이해가 쉽다는 것이다. 즉 식 (8)~식 (9)에서 각각  $y$ 의 값이 0일 때는 생산량과 생산비용이 동시에 0가 됨을 얘기하고 있고, 값이 1일 때는 생산량은 양의 값을 갖고 생산비용은  $F+cx$  식에 의해 계산됨을 보여준다.

반면 식 (2)~식 (4)에서는  $x$ 와  $y$ 에 대한 네 개의 경우가 발생한다. 1)  $x=0, y=0$ , 2)  $x>0, y=1$ , 3)  $x>0, y=0$  그리고 4)  $x=0, y=1$ 이다. 1)과 2)는 원래 문제에서 원하는 바인 반면 3)과 4)는 이루어져서는 안되는 조합이다. 따라서 이를 배제하기 위해 식 (3)에서  $y$ 가 0일 때 식의 우변이 0이 되고 식 (4)에 의해 생산량  $x$ 가 0이 되게 되어 경우 3)이 배제된다. 하지만 경우 4)는 특별히 배제되지 않는데 이는 4)와 1)을 비교할 때 항상 4)의 경우 비용이 1)의 경우의 비용보다 크기 때문에( $F>0$ 을 전제로 하여) 4)의 경우를 최적해로 선택하지 않기 때문이다.

이와 같은 논의를 통해 알 수 있는 것은 식 (2)

~식 (4)와 같은 모형을 설정하기 위해서는 문제 자체의 논리 이외에도 모형을 풀기위한 문제 해결기의 성격과 모형의 형태와의 관계를 잘 알고 있어야 하는 부담이 모형을 설정하는 사용자에게 주어진다. 즉 일반적인 문제 해결기의 알고리즘이 대수적인 방법을 사용하기에 문제 해결기에 입력하기에 적합한 표현은 대수적인 표현이고 일반적으로 수리 계획 분야에서는 원 문제에서 논리적인 관계들을 대수적인 방식으로 표현하여 문제를 푼다. 하지만 이런 방식은 문제 해결기에 대한 지식을 전제로 하기에 그런 지식이 없는 사용자에게는 문제를 해결할 모형을 설정하는 것은 큰 장애가 된다. 만약 문제 해결기에 대한 지식 여부에 상관없이 원 문제의 논리적인 표현 자체로 모형이 설정될 수 있고 사용자와 문제 해결기 사이에 사용자의 모형 표현을 문제 해결기에 적합하게 변환시켜줄 수 있는 모형 변환 단계가 존재한다면 사용자는 훨씬 편하게 의사결정 모형에 대한 지식을 사용할 수 있다.

다음에서는 좀 더 복잡한 상황에서의 실제적인 문제에서 조건부합축 제약식의 표현이 이용되는 것을 볼 수 있다. 이 논문에서 조건부합축 제약식의 표현 정의 그리고 해법에 대한 논의를 이끌기 위해 다음의 제품설계 문제에서 최적의 공통속성을 선택하는 수리계획 모형[1]을 이용한다. 수리 모형에서 표현된 기호의 의미는 다음과 같다.

- $P_i$  시장 세그먼트  $i \in I$ 를 위해 제시된 제품의 가격
- $y_j$  속성  $j$ 의 공유를 나타내는 이진 의사결정 변수
- $w_{ij}$  시장  $i$ 의 소비자들이 속성  $j \in J$ 에서 느끼는 단위당 효용
- $n_i$  시장  $i$ 의 크기
- $c_j$  제품의 속성  $j$ 를 제공하기 위해 드는 단위당 비용
- $\alpha_j$  속성  $j$ 를 공유함으로써 얻는 비용 상의 할인을
- $x_{ijk}$  시장  $i$ 의 제품에서 속성  $j$ 가 레벨  $k \in K$ 를 가짐을 나타내는 이진변수
- $L_{jk}$  속성  $j$ 의  $k$  레벨에서 제공되는 속성의 양
- $M_{jk}$  속성  $j$ 의  $k$  레벨에서  $L_{jk}$ 만큼 제공하는데 드는 비용

$$\max \sum_{i \in I} n_i [P_i - \sum_{j \in J} \sum_{k \in K} M_{jk} x_{ijk} + \sum_{j \in J} \alpha_j \sum_{k \in K} M_{jk} x_{ijk} y_j] \quad (12)$$

$$\text{s.t.} \quad \sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x_{ijk} - P_i \geq \sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x_{i'jk} - P_{i'} \quad \forall i, i' (i \neq i') \quad (13)$$

$$\sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x_{ijk} \geq P_i \quad \forall i \quad (14)$$

$$\sum_{k \in K} x_{ijk} = 1 \quad \forall i, j \quad (15)$$

$$y_j = 1 \rightarrow x_{ijk} = x_{i'jk} \quad \forall i, i', j, k (i \neq i') \quad (16)$$

$$P_i \geq 0, x_{ijk}, y_j \in \{0,1\} \quad \forall i, j, k \quad (17)$$

이 모형에서 먼저  $x_{ijk}$ 는 앞의 정의에 따른 이진 변수로 식 (15)에서 각 제품  $i$ 와 속성  $j$ 에 대해 오직 하나의 레벨  $k$ 에 대해서만 1의 값을 갖는다. 이것이  $\sum_k L_{jk} x_{ijk}$ 의 식을 통해서 하나의  $L_{jk}$  값이 되므로 이것이 제품  $i$ 에 주어진 속성  $j$ 의 제공량이 된다. 마찬가지로  $\sum_k M_{jk} x_{ijk}$ 는 제품  $i$ 에 주어진 속성  $j$ 의 제공 비용이 된다.  $y_j$ 는 속성이 공통속성으로 선택되었음을 나타내는 이진변수이므로 이에 따라 해석하면 식 (12)은 먼저 각 시장  $i$ 별 수익을 계산하고 여기서 제공된 제품  $i$ 의 속성별 비용의 합을 빼주고 마지막으로 속성  $j$ 를 공통속성으로 선택했을 때의 비용에서 절감된 부분을  $\alpha_j$ 의 곱으로 계산해서 더해 줌으로써 총 이익을 계산한다. 식 (14)의 좌변은 시장  $i$ 에서 제공된 제품  $i$ 가 포함하는 속성들의 제공량에 그 효용의 곱으로 속성당 총 효용을 나타내고 이를 속성에 대한 합으로서 제공된 제품의 총 효용을 나타낸다. 즉 시장  $i$ 에 제품에 의해 제공된 총 효용은 소비자가 지불하는 가격 이상이 되어야 한다. 식 (13)에서는 시장  $i$ 에 제공된 제품의 총 효용은 이 제품이 다른 시장에 제공되었을 시의 총 효용이상이 되어야 함을 나타낸다. 조건부합축 식 (16)에서는 속성  $j$ 가 공통속성으로 선택이 되면 모든 제품들에 대해서 제공되는 속성의 레벨이 같아야 한다는 상황을 표현하고 있다. 마지막으로 식 (14)에서 가격  $P_i$ 는 연속변수이고  $x_{ijk}$ 와  $y_j$ 는 이진변수이다.

위 모형에서 속성이 공통속성으로 선택이 되면 제품에 상관없이 속성의 레벨이 다 동일한 값을 갖는다는 사실을 표현하기 위해 조건부함축 제약식을 사용하였다. 그리고 만약 조건부함축 제약식을 사용하지 않는다면 어떤 식으로 표현해야할지 직관적으로 알 수 있는 방법이 없다. 즉 모형화의 측면에서 사용자에게 조건부함축 제약식과 같은 직관적인 표현수단을 제공하고 세부적인 동치의 대수적 표현은 모형기의 기능에 구현하는 것이 시스템을 사용자가 보다 편리하게 이용할 수 있도록 한다.

### 3.2 조건부함축 제약식의 SWCL 정의

이 연구는 기본적으로 의미웹에서 제약식을 표현하는 수단으로서 SWCL을 이용한다. SWCL은 제약식에 대한 공리(Axiom)를 추가해서 OWL의 추상문법을 확장한다. 하나의 모형은 제약식의 집합과 그것을 통해 구하려는 측정하고자 하는 값을 나타내는 목적함수로 이루어져 있으며, 하나의 제약식은 두 가지 형태로 구분되는데 대수적인 제약식과 논리적인 제약식이다. 여기서 Kim et al.[13]에서와는 다르게 제약식의 구별이 추가되었다. 이는 일반적인 제약식을 대수적 제약식으로 하고 새롭게 도입하는 제약식을 논리적 제약식의 일종으로 하였다. 대수적인 제약식은 일반적인 대수적 문제 해결기에서 이해할 수 있는 제약식으로 한정자와 좌변, 우변, 그리고 좌우변 사이의 관계를 나타내는 연산자로 이루어진다. 우변과 좌변에서의 각 항은 집계 연산자와 매개변수, 인수, 그리고 부호로 구성된다. SWCL의 문법은 EBNF(Extended Backus-Naur Form) 형식으로 정의되어 다음과 같다[2].

```
axiom ::= constraint
constraint ::= [URIreference]{qualifier} algebraic |
              logical
algebraic ::= 'Algebraic('LHS operator RHS')'
LHS ::= 'LHS('termBlock | function('termBlock)
         ('function'))'
RHS ::= 'RHS('termBlock | function('termBlock)
```

```
{function})'
termBlock ::= 'TermBlock('sign[aggregateOperator]{parameter}(factor{factor}))'
function ::= 'Function('sign functionID{parameter}(p-property{p-property}))'
sign ::= '+' | '-'
aggregateOperator ::= 'Sigma' | 'Production'
factor ::= 'Factor('((classID datavaluedPropertyID)|(individualID datavaluedPropertyID))[function])'
parameter ::= 'Parameter('variable | description')'
functionID ::= URIreference
p-property ::= URIreference
```

LHS와 RHS는 연산의 좌변과 우변을 나타내며, 이는 하나의 항을 나타내는 termBlock으로 구성된다. 일반적인 termBlock은 부호와 집계 연산자, 그의 매개변수, 그리고 최소 하나 이상의 인자로 구성되지만, 일반적인 선형 이외의 함수를 표현하기 위해 인자를 대신하여 function이 사용될 수 있도록 설계되어 확장성을 고려하였다. 이는 SWRL의 builtin과 같이 내장된 함수의 형태로 그 모형을 기술하게 된다.

위에 주어진 SWCL의 정의에 조건부함축 제약식을 설정할 수 있도록 다음과 같은 정의를 추가한다.

```
logical ::= implication
implication ::= 'Implication('antecedent consequent')'
antecedent ::= 'Antecedent('{algebraic})'
consequent ::= 'Consequent('{algebraic})'
```

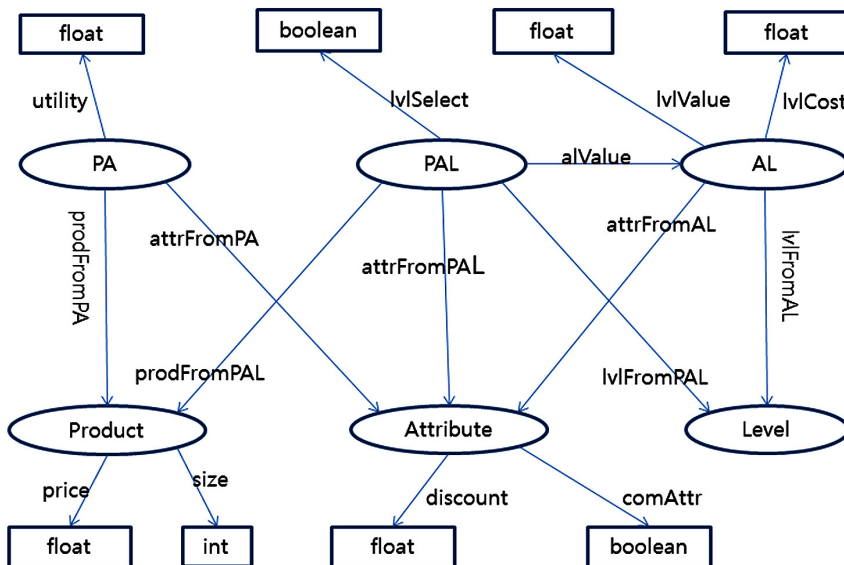
위 공리에서 보는 것처럼 논리적 제약식으로 조건부함축 제약식인 implication이 포함되어 있다. 조건부함축 제약식 이외의 특별한 논리 제약식을 정의하면 이 부분에 추가하게 된다. implication의 정의상 조건부함축 제약식은 조건(antecedent)과 결과(consequence)를 이루는 두 개의 요소로 이루어져 있고 이 각각은 키워드 'Antecedent'나 'Consequent'

와 함께 대수적인 제약식으로 이루어진다. 이때 대수적인 제약식은 하나도 없을 경우도 상정하는데 만약 조건에 제약식이 없을 경우는 논리식의 일반적인 정의에 따라 조건이 항상 참이므로 결과에 있는 대수적 제약식들을 추가하는 것과 같다. 즉 일반적인 대수적 제약식은 조건이 아무것도 없는 조건부합축 제약식으로 해석이 가능하다. 만약 결과에 아무 대수적 제약식이 없으면 이는 조건을 만족하지 않는 대수적 제약식의 추가를 의미한다. 하지만 이 두 경우는 대수적 제약식의 표현으로 가능하므로 굳이 조건부합축 제약식으로 표현할 이유는 없다. 또 조건과 결과에 하나가 아닌 여러 개의 대수적 제약식이 포함될 수도 있다.

조건부합축 제약식의 직접적인 모형이론적 의미(Direct Model-Theoretic Semantics)는 제약식의 모든 구성 요소들이 이미 SWCL의 추상문법 상에 정의된 요소들을 사용하므로 의미론 역시 이미 정의된 SWCL의 의미론을 그대로 사용하게 되는데 어느 해석이 조건을 구성하는 모든 구성요소를 만족할 때 결과의 모든 구성요소를 만족하게 되거나 조건을 구성하는 요소가 없으면 그 해석은 조건부합축 제약식을 만족하는 것이다.

### 3.3 제품설계 최적 공통속성 선택에서의 조건부합축 제약식 표현

앞에서 서술된 제약식의 정의에 따라 제품설계 최적 공통속성 선택에서의 조건부합축 제약식을 표현해보기로 한다. 이를 위해 먼저 제품설계 최적 공통속성 선택 문제의 데이터에 대한 온톨로지 스키마를 [그림 1]에 표시한다. [그림 1]은 6개의 클래스와 16개의 프로퍼티로 구성되어 있다. 그림에서 원형은 클래스를 나타내며 사각형은 데이터형을, 프로퍼티들은 화살표로 표현되었다. 각 화살표는 두 개의 원형을 연결하여 시작되는 원형을 도메인(domain)으로 끝나는 원형을 레인지(range)를 의미한다. 프로퍼티는 레인지가 원형이면 개체 프로퍼티를, 레인지가 사각형이면 데이터 프로퍼티를 나타내는 것으로 구분된다. 클래스 Product는 제품의 시장 세그먼트 크기  $n_i$ 를 나타내는 size라는 데이터 프로퍼티를 가진다. price는 시장에 출시될 제품의 가격을 보여주는 데이터 프로퍼티로 원 모형에서  $P_i$ 에 대응한다. 클래스 Attribute는 공통속성으로 제공될 때 비용의 할인율  $\alpha_j$ 를 보여주는 discount 데이터 프로퍼티의 도메인이다. 또 데이터 프로퍼티 comAttr는 해당 속



[그림 1] 제품설계 최적 공통속성 선택 문제의 온톨로지 스키마

성이 공통속성인지를 보여주는 프로퍼티로 원 모형의  $y_j$ 에 해당한다. 클래스 Level은 속성의 수준을 나타낸다. 원 문제의 데이터를 표현하기 위해 세 개의 클래스가 더 도입된다. 클래스 PA는 제품과 속성의 쌍을 표현하는 클래스이고, 클래스 AL은 속성과 수준의 쌍을 표현하는 클래스이고, 클래스 PAL은 제품-속성-수준의 세 조합을 나타내는 클래스이다.

PA 클래스를 도메인으로 하고 조합의 각 요소들을 값으로 하는 개체 프로퍼티들은 prodFromPA, attrFromPA가 있고, PAL을 도메인으로 하고 조합의 각 요소들을 값으로 하는 개체 프로퍼티로 prodFromPAL, attrFromPAL, lvlFromPAL가 있고, AL을 도메인으로 하며 조합의 각 요소들을 값으로 하는 개체 프로퍼티 attrFromAL과 lvlFromAL이 있다.

〈표 1〉 SWCL을 이용한 제약식 (16)의 표현 예

```

<swcl:Constraint rdf:ID="CondConstraint">
  <swcl:qualifier>
    <swcl:Variable rdf:ID="y">
      <swcl:bindingClass rdf:resource="#Attribute"/>
    </swcl:Variable>
    <swcl:Variable rdf:ID="x1">
      <swcl:bindingClass rdf:resource="#PAL"/>
    </swcl:Variable>
    <swcl:Variable rdf:ID="x2">
      <owl:equivalentClass>
        <owl:complementOf>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#prodFromPAL"/>
            <owl:hasValue rdf:resource="#x1"/>
          </owl:Restriction>
        </owl:complementOf>
      </owl:equivalentClass>
    </swcl:Variable>
  </swcl:qualifier>
  <swcl:Implication /* 조건부합축 제약식의 시작 */
  <swcl:antecedent /* 조건 부분의 시작 */
    <swcl:Algebraic>
      <swcl:hasLHS rdf:parseType="Collection">
        <swcl:termBlock>
          <swcl:sign rdf:resource="http://iwec.yonsei.ac.kr/swcl#plus"/>
          <swcl:factor rdf:parseType="Collection">
            <swcl:FactorAtom>
              <swcl:bindingClass rdf:resource="#y"/>
              <swcl:bindingDatatypeProperty rdf:resource="#comAttr"/> /* comAttr 프로퍼티 */
            </swcl:FactorAtom>
          </swcl:factor>
        </swcl:termBlock>
      </swcl:hasLHS>
      <swcl:hasOperator rdf:resource="http://iwec.yonsei.ac.kr/swcl#equal"/> /* 등식 */
      <swcl:hasRHS>
        <swcl:termBlock>
          <swcl:sign rdf:resource="http://iwec.yonsei.ac.kr/swcl#plus"/>
          <swcl:factor>
            <swcl:FactorAtom>
              <xsd:integer value="1"/> /* 값 1 */
            </swcl:FactorAtom>
          </swcl:factor>
        </swcl:termBlock>
      </swcl:hasRHS>
    </swcl:Algebraic>
  </swcl:antecedent /* 조건 부분의 종료 */

```



```

<swcl:consequent> /* 결과 부분의 시작 */
<swcl:Algebraic>
  <swcl:hasRHS rdf:parseType="Collection">
    <swcl:termBlock>
      <swcl:sign rdf:resource="http://iwec.yonsei.ac.kr/swcl#plus"/>
      <swcl:factor rdf:parseType="Collection">
        <swcl:FactorAtom>
          <swcl:bindingClass rdf:resource="#x1"/>
          <swcl:bindingDatatypeProperty rdf:resource="#lv1Select"/> /* lv1Select 프로퍼티 */
        </swcl:FactorAtom>
      </swcl:factor>
    </swcl:termBlock>
  </swcl:hasRHS>
  <swcl:hasOperator rdf:resource="http://iwec.yonsei.ac.kr/swcl#equal"/>
  <swcl:hasRHS rdf:parseType="Collection">
    <swcl:termBlock>
      <swcl:sign rdf:resource="http://iwec.yonsei.ac.kr/swcl#plus"/>
      <swcl:factor rdf:parseType="Collection">
        <swcl:FactorAtom>
          <swcl:bindingClass rdf:resource="#x2"/>
          <swcl:bindingDatatypeProperty rdf:resource="#lv1Select"/> /* lv1Select 프로퍼티 */
        </swcl:FactorAtom>
      </swcl:factor>
    </swcl:termBlock>
  </swcl:hasRHS>
</swcl:Algebraic>
</swcl:consequent> /* 결과 부분의 종료 */
</swcl:Implication> /* 조건부합축 제약식의 종료 */
</swcl:Constraint>
    
```

개체 프로퍼티  $alValue$ 은 도메인으로 PAL을 갖고 레인지로 AL을 갖게 되어 특정 PAL 개체에 대해서 해당하는 속성과 수준으로 이루어진 AL 개체를 값으로 주는 프로퍼티이다. PAL의 개체에 또 클래스 PA를 도메인으로 하는 데이터 프로퍼티  $utility$ 는 해당 제품과 속성에 대한 효용  $w_{ij}$ 을 값으로 하는 프로퍼티이다.  $lv1Select$ 는 제품-속성-수준의 조합에서 해당 수준이 해당 제품의 속성으로 선택되었는지를 나타내는 데이터 프로퍼티로  $x_{ijk}$ 에 해당한다. 도메인 AL을 갖는 데이터 프로퍼티  $lv1Value$ 는 도메인 개체에 해당하는 속성과 수준에 대한 제공양  $L_{jk}$ 을 나타내고 데이터 프로퍼티  $lv1Cost$ 는 속성과 수준을 제공하는데 드는 비용  $M_{jk}$ 를 나타낸다. 정의된 데이터 프로퍼티 중에 각 도메인의 개체별로 값을 특정 상수 값을 갖는 프로퍼티는 원 수리 모형의 파라미터를 지시하게 되고 값이 정해지지 않은 프로퍼티는 SWCL 표현에서 의사결정 변수의 역할을 하게 된다.

주어진 문제의 파라미터 데이터는 [그림 1]과 같

은 온톨로지 스키마의 구조에 따라 온톨로지 저장고에 저장된다. 그리고 이 온톨로지 스키마를 이용해 문제의 제약식은 SWCL로 표현되게 된다. 다음은 앞에서 주어진 수리모형 (12)~(17) 중에서 관심사인 조건부합축 제약식인 제약식 (16)을 SWCL에 의해 RDF 형식으로 표현한 것이다.

#### 4. 조건부합축 제약식의 해결을 위한 시스템 틀

제품설계 최적 공통속성 선택 모형 (12)~(17)는 상업적으로 가장 보편적인 해결기인 혼합정수선형 계획 해결기의 관점에서 보았을 때 받아들일 수 없는 두 가지 부분이 존재한다. 첫째는 식 (12)에 있는 비선형적인 표현이다. 이 부분은 원래 SWCL이 모든 다항식을 표현할 수 있는 표현력이 있기 때문에 문제가 되지 않는 부분이지만 해결기에서는 선형인 표현만 가능하다. 둘째로 식 (16)의 조건부합축 제약식 표현을 해결기로서는 이해하지 못한다는

것이다.

먼저 첫 장애를 해결하기 위해 다음과 같은 선형화 과정을 거친다. 먼저 식 (9)의 세 번째 항은  $\Sigma_k M_{jk} x_{ijk}$ 와  $y_j$ 와의 곱으로  $x_{ijk} y_j$ 이란 이차식을 갖는데, 새로운 이진 변수  $z_{ijk}$ 을 도입하여 다음과 같이 정의한다.

$$z_{ijk} = x_{ijk} y_j \quad \forall i, j, k \quad (18)$$

이를 이용하여 식 (12)의  $x_{ijk} y_j$ 를  $z_{ijk}$ 로 교체하고 식 (18)을 모형의 제약식으로 추가한다. 이로써 원 모형의 비선형 표현은 식 (18)로 이전되고 모형의 해법에서 중요한 것은 식 (18)을 어떻게 다루느냐에 있다. 식 (18)은 다음 사실을 이용해 선형 표현으로 변환될 수 있다. 다음의 선형 제약식들은 (18)과 동치이다.

$$x_{ijk} + y_j \leq 1 + z_{ijk} \quad \forall i, j, k \quad (19)$$

$$x_{ijk} + y_j \geq 2z_{ijk} \quad \forall i, j, k \quad (20)$$

이는 식 (18)에서  $x_{ijk}$ 와  $y_j$ 이 둘 다 1 값을 갖을 때만  $z_{ijk}$ 의 값이 1이 됨을 보여준다. 즉  $z_{ijk}=0$ 일 때는  $x_{ijk}$ 나  $y_j$  중 적어도 하나가 0이 되므로 식 (19)가 만족되고 식 (20)은  $x_{ijk}$ 와  $y_j$ 가 이진변수이므로 당연히 성립된다.  $z_{ijk}=1$ 인 경우  $x_{ijk}=y_j=1$ 이므로 식 (19)와 식 (20)은 등호로 자연스럽게 만족된다. 반대로 식 (19)~식 (20)에서  $z_{ijk}=1$ 는 이진변수  $x_{ijk}$ 와  $y_j$  합이 2임을 의미하고 이는 두 변수 모두 1이어야 한다. 마찬가지로  $z_{ijk}=0$ 는 식 (19)에서  $x_{ijk} + y_j \leq 1$ 이고 이는 둘 중 적어도 하나가 0임을 의미하며 따라서 식 (18)을 충족한다. 따라서 식 (18)에 따라 문제를 변형하였을 때 식 (19)와 식 (20)을 제약식으로 추가함으로써 혼합정수선형계획 해결기에서 다룰 수 있는 모형의 표현을 얻을 수 있게 된다.

둘째로 식 (16)의 조건부합축 제약식에 대한 대수적 동치를 다음의 제약식으로 얻는다.

$$x_{ijk} - x_{i'jk} \leq 1 - y_j \quad \forall i, i', j, k (i \neq i') \quad (21)$$

$$x_{i'jk} - x_{ijk} \leq 1 - y_j \quad \forall i, i', j, k (i \neq i') \quad (22)$$

여기서 우선 제약식 (16)은 다음과 동치이다.

$$|x_{ijk} - x_{i'jk}| \leq 1 - y_j \quad \forall i, i', j, k (i \neq i')$$

$x_{ijk}$ 와  $x_{i'jk}$ 가 이진값을 가지므로 위 식의 절대값은 이진값을 갖는다. 해  $(x_{ijk}, x_{i'jk}, y_j)$  값이 위 식을 만족한다고 가정할 때 식 (16)에서  $y_j=1$ 이 진리이면 위의 식은  $x_{ijk} = x_{i'jk}$ 를 강제하고 이는 식 (16)의 결론부를 만족한다. 그리고  $y_j=0$ 일 때는 절대값이 최대 1 이하이고 모든 두 변수의 값이 이를 만족하므로 두 변수의 값에 어떤 제약도 부여하지 않는다. 반대로 해의 값이 식 (16)을 만족하면  $y_j=1$ 일 때 결론부가 위 식의 절대값을 0이 되게 하므로 위 식을 만족하고  $y_j=0$ 일 때 위 식의 우편이 1이므로 절대값의 두 변수에 대한 어떤 제약도 가하지 않는다. 그리고 절대값의 식은 다음 식과 동치이다.

$$y_j - 1 \leq x_{ijk} - x_{i'jk} \leq 1 - y_j \quad \forall i, i', j, k (i \neq i')$$

그러므로 식 (21)과 식 (22)는 자연스럽게 얻어진다. 이를 통해 다음과 두 가지 제약식 변환 규칙을 마련할 수 있다.

### 규칙 1

- 1) 이진변수의 곱  $xy$ 를 발견한다.
- 2) 새로운 이진변수  $z$ 를 도입하여 곱  $xy$ 를 대체한다.
- 3)  $x + y \leq 1 + z$ ,  $x + y \geq 2z$ 의 제약식을 추가한다.

### 규칙 2

- 1) 서로 다른 이진변수  $x, x'$ 와  $y$ 에 대해  $y=1 \rightarrow x = x'$ 의 제약식을 발견한다.
- 2) 발견한 제약식을 제거하고  $x - x' \leq 1 - y$ 와  $x' - x \leq 1 - y$  제약식을 추가한다.

위의 두 가지 방법에 의해 앞의 제품설계 최적 공통속성 선택 문제가 혼합정수선형계획 해결기에서 이해할 수 있는 표현의 모형으로 변환이 되고 이를 정리하면 다음과 같다.

$$\max \sum_{i \in I} n_i [P_i - \sum_{j \in J, k \in K} M_{jk} x_{ijk} + \sum_{j \in J} \alpha_j \sum_{k \in K} M_{jk} z_{ijk}] \quad (23)$$

$$\text{s.t.} \quad \sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x_{ijk} - P_i \geq \sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x'_{ijk} - P_{i'} \quad \forall i, i' (i \neq i') \quad (24)$$

$$\sum_{j \in J} w_{ij} \sum_{k \in K} L_{jk} x_{ijk} \geq P_i \quad \forall i \quad (25)$$

$$\sum_k x_{ijk} = 1 \quad \forall i, j \quad (26)$$

$$x_{ijk} - x_{i'jk} \leq 1 - y_j \quad \forall i, i', j, k (i \neq i') \quad (27)$$

$$x'_{i'jk} - x_{ijk} \leq 1 - y_j \quad \forall i, i', j, k (i \neq i') \quad (28)$$

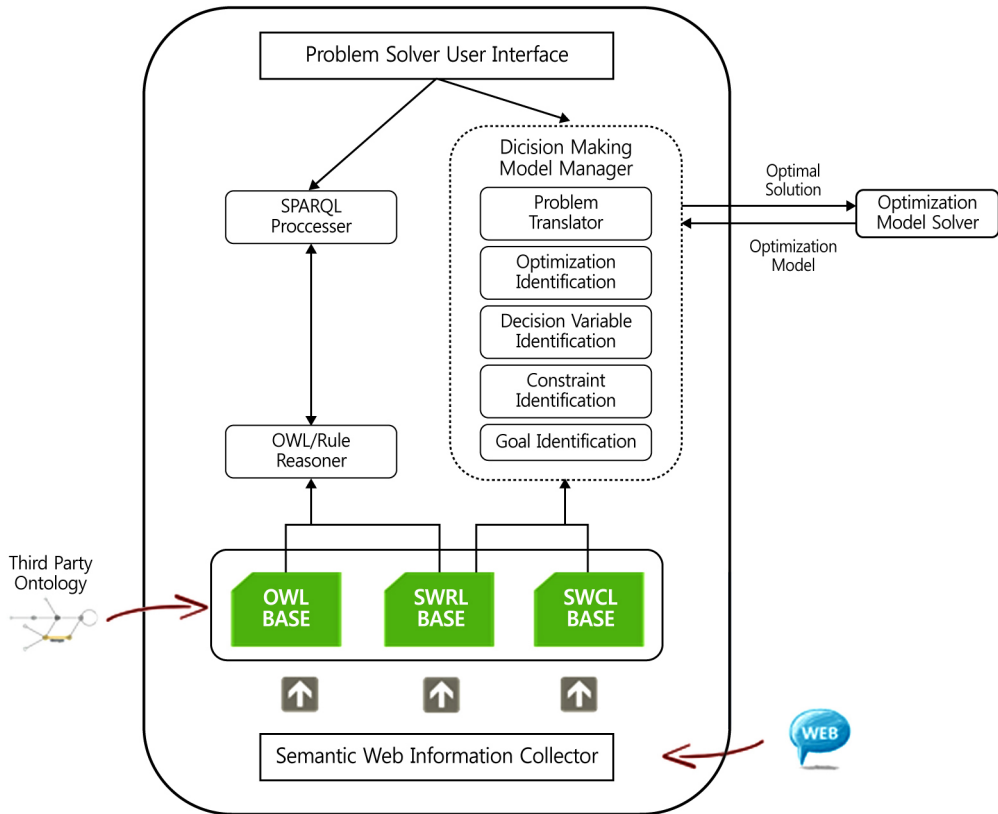
$$x_{ijk} + y_j \leq 1 + z_{ijk} \quad \forall i, j, k \quad (29)$$

$$x_{ijk} + y_j \geq 2z_{ijk} \quad \forall i, j, k \quad (30)$$

$$P_i \geq 0, x_{ijk}, y_j, z_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (31)$$

이 모형은  $P_i$ 의 연속형 변수와  $x_{ijk}, z_{ijk}$  그리고  $y_j$ 의 이진변수로 구성되어 모든 수식이 선형이므로 해결기에 의해 해를 구할 수 있게 된다. 위 모형을 SWCL로 전환하는 대신 여기서는 원 모형을 해결기가 받을 수 있는 OPL 형식의 문서로 전환을 할 때 앞에서 서술한 규칙을 이용하여 식 (23)~식 (31)과 같은 모형으로 전환한다. 이 전환하는 시스템의 프레임워크에 대한 내용을 [그림 2]에서 볼 수 있다.

[그림 2]에서 보는 것과 같이 문제풀이 프레임워크는 크게 세 부분으로 구성된다. 첫째로 모든 데이터는 OWL Base, SWRL Base, SWCL Base에 저장된다. 즉 모형의 파라미터를 포함한 모든 개별적인 데이터는 OWL Base에 온톨로지에 의해 구조화되어 저장되고 모든 규칙 데이터는 SWRL Base에 그리고 문제의 제약식을 표현한 데이터는 SWCL



[그림 2] 시스템의 틀

Base에 저장된다. 두 번째 부분은 SPARQL 처리기와 OWL/Rule 추론기로 정보 탐색을 위해 사용자의 인터페이스로부터 입력된 질의를 SPARQL로 변환 및 처리하고, 만약 추론이 필요하다면 추론기를 통해 새로운 지식을 생성 및 발견하게 된다. 세 번째로 모형 관리기(Decision Making Model Manager)이다. OWL과 SWRL, 그리고 SWCL로 표현된 온톨로지, 규칙, 그리고 제약식의 정보들을 읽어 사용자가 원하는 수리 모형을 선택 및 생성하고 이를 해결하여, 의사결정을 위한 모형의 최적 해를 도출한 후 사용자에게 제공하게 된다. 그림에서 보듯이 5개의 구성요소로 모형관리기가 하는 역할을 설명하고 있다. 목표 식별(Goal Identification), 의사결정 변수 식별(Decision Variable Identification), 제약식 식별(Constraint Identification), 최적화 식별(Optimization Identification) 그리고 문제 변환(Problem Translator)이다. 최적화 식별의 경우 주어진 문제에 특별한 목표가 별도로 주어지지 않았을 때 문제를 최적해가 아닌 단지 실현가능해만을 찾는 실현가능문제(Satisfiability)로 간주할 수 있다. 목표가 주어질 경우 모형관리기는 이를 최적화 문제로 인식하게 된다. 또한 이 과정에서 생성된 문제의 모형이 주어진 문제해결 도구인 솔버를 통해서 해찾기가 가능한지의 여부를 타진한다. 이것이 가능하다고 판단될 때에 문제 변환을 통해서 문제해결 도구가 원하는 데이터 형태로 모형을 변환한다.

모형관리기에서의 모형 구축 과정을 설명하면 다음과 같다. 먼저 사용자로부터 목표가 정해지면 이를 설명하는 SWCL 저장고로부터 제약식을 스캔한다. 이 제약식은 여러 개의 팩터로 이루어져 있고 각 팩터들은 온톨로지의 프로퍼티로 구성되어 있다. 온톨로지를 참고하여 각 프로퍼티가 데이터프로퍼티인지를 확인하고 온톨로지 상 레인지의 값이 존재하는지를 확인한다. 만약 값이 존재한다면 이는 모형에서의 파라미터로 정의되고 모형에서 계수 및 상수로 사용된다. 만약 값이 존재하지

않다면 이는 모형을 풀음으로써 구해야하는 값들로 이는 모형의 설정시 의사결정 변수로 선언된다. 제약식의 팩터를 스캔한 후에 얻어진 의사결정 변수들 중 하나를 선택하여 이를 포함하는 제약식들을 찾는다. 찾아진 각각의 제약식들에 대하여 마찬가지로의 과정으로 각 데이터프로퍼티들을 온톨로지로부터 확인하여 파라미터와 의사결정 변수를 확인한다. 의사결정 변수와 제약식간 번갈아가며 스캔을 하는 과정을 통해 모형의 구축에 필요한 파라미터의 값들과 의사결정 변수의 선언, 목적함수식, 그리고 모형을 이루는 제약식들을 모은다. 모형의 모든 구성요소들에 대한 데이터가 수집된 후 앞에서 설명한 제약식의 변환 과정을 수행한다. [그림 1]의 온톨로지 스키마에 따라 새로운 이진 의사결정 변수  $z$ 를 클래스 PAL의 각 개체  $a$ 마다 선언한다. 개체  $a$ 의 `lv1Select` 프로퍼티와 그 개체의 `attrFromPAL.comAttr` 프로퍼티의 곱으로 표현된 부분을 의사결정 변수  $z$ 로 대체한다. 또 각 PAL 개체마다 다음과 같은 제약식을 추가한다.

$$\begin{aligned} a.lv1Select + a.attrFromPAL.comAttr &\leq z \\ a.lv1Select + a.attrFromPAL.comAttr &\geq 2*z \end{aligned}$$

다음으로 두 개의 다른 클래스 PAL의 개체에 대해  $a.comAttr \rightarrow a.lv1Select = b.lv1Select$ 의 제약식을 다음과 같은 제약식으로 대체한다.

$$\begin{aligned} a.lv1Select - b.lv1Select &\leq 1 - a.attrFromPAL.comAttr \\ b.lv1Select - a.lv1Select &\leq 1 - a.attrFromPAL.comAttr \end{aligned}$$

이와 같은 변환이 문제 변환 과정에서 이루어진다. 변환 과정 후에 모든 정보가 OPL 표현의 문서로 변경된다. 이 논문에서는 모형의 해결기로 IBM ILOG CPLEX Optimization Studio[8]를 사용한다. 이 해결기는 모형의 입력으로 OPL이라는 언어로 표현해야 하므로 해결기로 모형을 풀기 위해 OPL로의 변환이 필요하다. <표 2>는 위의 과정을 거쳐 만들어진 모형의 OPL 표현의 일부를 보여준다.

〈표 2〉 제품설계 최적 공통속성 선택 모형의 OPL 표현의 일부

```

...
maximize /* 목적함수식 (23) */
sum(i in Products) Size[i]*Price[i] -
sum(i in Products, j in Attributes, k in Levels) Size[i]*Cost[j]*LvlCost[j][k]*LvlSelect[i][j][k] +
sum(i in Products, j in Attributes, k in Levels) Discount[j]*Cost[j]*LvlCost[j][k]*z[i][j][k];
subject to {
...
forall( j in Attributes ) forall( k in Levels )
forall( <i1, i2> in ProdPairs )
    LvlSelect[i1][j][k] - LvlSelect[i2][j][k] <= 1 - ComAttr[j]; /* 제약식 (27) */

forall( j in Attributes )
forall( k in Levels )
forall( <i1, i2> in ProdPairs )
    LvlSelect[i2][j][k] - LvlSelect[i1][j][k] <= 1 - ComAttr[j]; /* 제약식 (28) */
forall( i in Products )
forall( j in Attributes )
forall( k in Levels )
    LvlSelect[i][j][k] + ComAttr[j] - z[i][j][k] <= 1; /* 제약식 (29) */
forall( i in Products )
forall( j in Attributes )
forall( k in Levels )
    2 * z[i][j][k] - LvlSelect[i][j][k] - ComAttr[j] <= 0; /* 제약식 (30) */
};
    
```

### 5. 제품설계의 최적 공통속성 선택의 풀이 예

여기서는 앞에서 설명된 조건부합축 제약식을 이용하여 제품설계의 최적 공통속성 선택 문제를 풀어 본다. 먼저 시장에 4개의 세그먼트가 주어졌다고 가정하고 4가지의 다른 제품을 시장에 제공할 때 비용

을 최소화하도록 10가지의 속성을 어떻게 배합하고 어떤 속성을 공통속성으로 가져갈지에 대한 문제를 다루어 본다. 각 속성당 10개의 레벨이 있다고 가정하자. 각 세그먼트의 크기는 다음과 같다.

| 시장세그먼트     | 1  | 2  | 3  | 4  |
|------------|----|----|----|----|
| 크기(단위 : 천) | 10 | 10 | 20 | 20 |

10가지 속성별 비용의 할인율은 다음과 같다.

| 속성  | 1   | 2   | 3   | 4  | 5   | 6  | 7   | 8   | 9   | 10 |
|-----|-----|-----|-----|----|-----|----|-----|-----|-----|----|
| 할인률 | 10% | 20% | 30% | 0% | 30% | 0% | 30% | 10% | 20% | 0% |

각 속성이 각 레벨에서 제공되는 양  $L_{jk}$ 가 다음과 같다고 하자.

| 속성/레벨 | 1    | 2    | 3    | 4    | 5    | 6     | 7     | 8     | 9     | 10    |
|-------|------|------|------|------|------|-------|-------|-------|-------|-------|
| 1     | 1.00 | 1.10 | 1.40 | 1.90 | 2.60 | 3.50  | 4.60  | 5.90  | 7.40  | 9.10  |
| 2     | 1.20 | 1.41 | 2.04 | 3.09 | 4.56 | 6.45  | 8.76  | 11.49 | 14.64 | 18.21 |
| 3     | 1.00 | 1.18 | 1.72 | 2.62 | 3.88 | 5.50  | 7.48  | 9.82  | 12.52 | 15.58 |
| 4     | 0.80 | 0.93 | 1.32 | 1.97 | 2.88 | 4.05  | 5.48  | 7.17  | 9.12  | 11.33 |
| 5     | 1.00 | 1.23 | 1.92 | 3.07 | 4.68 | 6.75  | 9.28  | 12.27 | 15.72 | 19.63 |
| 6     | 0.10 | 0.46 | 1.81 | 4.06 | 7.21 | 11.26 | 16.21 | 22.06 | 28.81 | 36.46 |
| 7     | 1.30 | 1.62 | 2.58 | 4.18 | 6.42 | 9.30  | 12.82 | 16.98 | 21.78 | 27.22 |
| 8     | 0.50 | 0.93 | 2.22 | 4.37 | 7.38 | 11.25 | 15.98 | 21.57 | 28.02 | 35.33 |
| 9     | 0.82 | 0.95 | 1.34 | 1.99 | 2.90 | 4.07  | 5.50  | 7.19  | 9.14  | 11.35 |
| 10    | 0.21 | 0.41 | 1.01 | 2.01 | 3.41 | 5.21  | 7.41  | 10.01 | 13.01 | 16.41 |

각 속성의 각 레벨에서 제공될 때의 제공 비용  $M_{jk}$ 는 다음의 단위당 속성 비용에 제공량을 제공하여 얻어진다고 가정하자.

| 속성   | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   |
|------|------|------|------|------|------|------|------|------|------|------|
| 제공비용 | 0.88 | 0.94 | 0.98 | 0.20 | 0.11 | 0.29 | 0.69 | 0.09 | 0.61 | 0.87 |

마지막으로 각 속성별 각 시장 세그먼트에서 소비자가 느끼는 단위당 효용은 다음과 같이 주어진다.

| 시장/속성 | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   |
|-------|------|------|------|------|------|------|------|------|------|------|
| 1     | 1.39 | 1.47 | 1.62 | 1.06 | 0.99 | 0.71 | 0.84 | 0.89 | 1.22 | 1.39 |
| 2     | 0.91 | 1.52 | 1.48 | 0.82 | 0.98 | 0.59 | 1.05 | 0.79 | 1.54 | 1.85 |
| 3     | 0.72 | 1.31 | 1.24 | 0.74 | 0.86 | 0.62 | 1.18 | 0.77 | 1.32 | 0.54 |
| 4     | 0.45 | 1.02 | 1.37 | 0.43 | 1.26 | 1.04 | 0.95 | 0.69 | 1.08 | 0.36 |

이와 같은 데이터가 주어졌을 때 최적해를 구하면 10개의 속성 중 6개의 속성 1, 2, 3, 5, 7, 8이 공통속성으로 선택되고 4개 시장 세그먼트에 제공되는 4개 제품의 속성 구성을 보면 다음과 같이 설계된다.

| 제품/속성 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|---|---|---|---|---|---|---|---|----|
| 1     | 1 | 1 | 1 | 3 | 6 | 1 | 1 | 4 | 1 | 2  |
| 2     | 1 | 1 | 1 | 1 | 6 | 1 | 1 | 4 | 1 | 3  |
| 3     | 1 | 1 | 1 | 5 | 6 | 3 | 1 | 4 | 3 | 1  |
| 4     | 1 | 1 | 1 | 3 | 6 | 3 | 1 | 4 | 2 | 1  |

그리고 각 제품의 최적 판매 가격은 19.34, 19.34, 19.37, 19.35로 결정되고 총비용(단위 : 천)은 511.885으로 결정된다.

## 6. 결 론

이 논문에서는 의미웹 환경에서 개체간의 관계를 나타내는 제약식을 표현하는 SWCL의 확장을 모색한다. 특히 의사결정을 위해 제약식을 표현할 때 혼합정수선형계획 해결기에서 사용되는 이진 변수의 대수적인 표현이 갖는 이해의 어려움을 극복하기 위해 조건부합축을 포함하는 조건부합축 제약식을 SWCL 문법에 도입할 필요성을 논하였다. 그리고 이어 SWCL에서의 조건부합축 제약식의 정의를 보이고 제품설계 최적 공통속성 선택의 문제에서의 사용을 통해 이 제약식의 SWCL 문법을 이용한 예를 RDF 형식으로 보였다. 조건부합축 제약식의 해결 과정을 보이기 위

해 시스템 상에 확장된 형태의 SWCL로 표현된 모형이 해결기와 연결 시 어떤 변환 과정을 거쳐야 하는지를 제품설계의 최적 공통속성 선택의 문제를 예로 설명하고 시스템의 구조를 제시하였다. 이 연구는 두 가지 측면에서 의미가 있다고 본다. 첫째로 SWCL 사용자에게 의사결정 모형을 위한 또 하나의 제약식 구성요소를 제공하고 있어 그 표현력을 확장하였다. 둘째로 단순 대수적인 제약식을 벗어나 논리적인 제약식으로서의 확장 가능성을 보였다. 셋째로 조건부합축 제약식 해결을 위한 모형의 변환 과정 제시를 통해 제약식을 구축하는 사용자의 표현과 해결기에 입력되는 표현을 연결하는 변환 과정에서 단순한 표현의 번역 과정이 아닌 구조적인 변환의 과정 가능성을 보임으로써 사용자 위주의 제약식 표현에 가능성을 열었다고 본다.

이 연구를 통해 앞으로의 향후 다음과 같은 사항들을 연구 과제로 볼 수 있다. 우선 이 논문에서는

조건부합축 제약식을 도입하는 틀을 제시하였고 간단한 형태의 조건부합축 제약식을 이용하여 이의 유용성을 보였다. 향후의 연구로 보다 복잡한 상황의 복잡한 표현을 가진 조건부합축 제약식으로서의 확장은 실제 문제를 해결하는데 유용할 것으로 생각한다. 이를 위해서는 해결기 기술의 고려가 동반되어야 할 것으로 생각된다. 둘째 다른 형태의 논리적인 제약식의 사용에 대해서도 고려할 필요가 있다. 이를 위해서는 실제적으로 사용되는 많은 의사결정 모형에서의 표현에 대한 연구가 필요하다. 마지막으로 이 논문에서는 해결기로서 혼합정수선형계획의 해결기를 사용하였다. 다른 형태의 해결기에 대한 연구도 제안된 언어의 확장의 유용성을 확장할 수 있을 것으로 생각된다.

## 참 고 문 헌

- [1] 김학진, 이삼열, 윤소현, “논리기반 벤더스 분리법을 이용한 제품설계 공통속성 최적선택의 풀이”, 『한국생산관리학회지』, 제24권, 제2호(2013), pp.261-277.
- [2] 이명진, 김우주, 김학진, “시맨틱 웹 환경에서의 부분선형 제약지식 표현을 위한 SWCL의 확장”, 『한국경영과학회지』, 제37권, 제4호(2012), pp.19-35.
- [3] 이명진, 김학진, 김우주, “의미망 제약식언어를 기반으로 한 인터넷 쇼핑 의사결정 틀”, 『한국경영과학회지』, 제33권, 제3호(2008), pp.29-42.
- [4] Anoniou, G. and F. Harmelen, *Handbook on Ontologies in Information Systems*, Springer, 2003.
- [5] Bassiliades, N. and P.M.D. Gray, “Colan : A Functional Constraint Language and Its Implementation,” *Data and Knowledge Engineering*, Vol.14, No.3(1994), pp.203-249.
- [6] Berners-Lee, T., R. Fielding, and L. Masinter, “Uniform Resource Identifier(URI) : Generic Syntax,” RFC 3986/STD 66, 2005.
- [7] Bratt, S., “Semantic Web, and Other Technologies to Watch,” World Wide Web Consortium, 2007.
- [8] Bray, T., J. Aoli, C.M. Sperberg-McQueen, and E. Maler, “Extensible Markup Language (XML) 1.0(Fifth Edition),” W3C Recommendation, 2008.
- [9] Gerald, H. and M. Stefan, “ILOG OPL Studio,” *OR Spektrum : Organ der Deutschen Gesellschaft Fur Operations Research*, Vol.21, No. 4(1999), pp.419-427.
- [10] Gray, P., K. Hui, and A. Preece, “An Expressive Constraint Language for Semantic Web Applications,” IJCAI01 Workshop on e-business and the intelligent web, 2001.
- [11] Horrocks, I., P.F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean, “SWRL : A Semantic Web Rule Language Combining OWL and RuleML,” Acknowledged W3C Member Submission, NRC 48058, 2004.
- [12] Kifer, M., “Rule Interchange Format : The Framework,” *Lecture Notes in Computer Science*, Vol.5341(2008), pp.1-2.
- [13] Kim, H.J., W. Kim, and J. Kim, “An Intelligent Shopping Agent for optimal Purchasing Decision on the Semantic Web,” *Lecture Notes on Computer Science*, Vol.4623(2006), pp.192-201.
- [14] Kim, H.J., W. Kim, and M. Lee, “Semantic Web Constraint Language and its application to an intelligent shopping agent,” *Decision Support Systems*, Vol.46, No.4(2009), pp.882-894.
- [15] Krishnan, R., “Knowledge Based Aids for Model Construction,” Ph.D. Thesis, The University of Texas, Austin, 1987.
- [16] Lee, J.K. and M.Y. Kim, “Knowledge-assisted Optimization Model Formulation : UNIK-

- OPT,” *Decision Support Systems*, Vol.13, No.2(1995), pp.111-132.
- [17] Miller, E., “An Introduction to the Resource Description Framework,” *Bulletin of the American Society for Information Science*, Vol. 25, No.1(1998), pp.15-19.
- [18] Preece, A., S. Chalmers, C. Mckenzie, J.Z. Pan, and P. Gray, “Handling Soft Constraint in the Semantic Web Architecture,” In Proceeding Resoning on the Web Workshop (RoW) at WWW, Edinburgh, UK, (2006), pp.22-26.
- [19] Yeom, K. and J.K. Lee, “Logical Representation of Integer Programming,” *Decision Support Systems*, Vol.18, No.3/4(1996), pp.227-251.