

◆특집◆ IT 기반의 설계 및 생산

# PLM 지원을 위한 온톨로지 기반 지식 프레임워크

이재현\*, 서효원\*\*

## Ontology-Based Knowledge Framework for Product Lifecycle Management

Jae Hyun Lee\* and Hyo Won Suh\*\*

### ABSTRACT

This paper introduces an approach to an ontology-based knowledge framework for product lifecycle management (PLM). Participants in a product life cycle want to share comprehensive product knowledge without any ambiguity and heterogeneity. However, previous knowledge management approaches are limited in providing those aspects. Therefore, we suggest an ontology-based knowledge framework including knowledge maps, axioms and specific knowledge for domain. The bottom level, the axiom, specifies the semantics of concepts and relations of knowledge so that ambiguity of the semantics can be alleviated. The middle level is a product development knowledge map; it defines the concepts and the relations of the product domain common knowledge and guides engineers to process their engineering decisions. The middle level is then classified further into more detailed levels, such as generic product level, specific product level, product version level, and product item level for PLM. The top level is specialized knowledge for a specific domain that gives the solution of a specific task or problem. It is classified into three knowledge types: expert knowledge, engineering function knowledge, and data-analysis-based knowledge. This proposed framework is based on ontology to accommodate a comprehensive range of unambiguous knowledge for PLM and is represented with first-order logic to maintain a uniform representation.

**Key Words:** Ontology, First Order Logic, Knowledge Framework, Product Lifecycle Management

### 1. 서론

1980년대 초 지식관리 관련 연구들은 KBS (Knowledge-based System)의 개발을 사람의 지식을 지식베이스로 옮기는 프로세스로 보았다. 이는 KBS에 필요한 지식은 이미 존재하고 이를 수집하여 구현하면 된다는 가정을 기반으로 한다. 하

지만, 전문가의 문제해결 능력 같은 지식이 도메인 지식과 혼합되어 있기 때문에 규모가 큰 지식베이스 시스템에서는 이 가정이 적절하지 않다. 최근 지식관리 프로세스는 한 번에 지식베이스를 구축하는 것이 아니라, 지식베이스를 구축하는 프로세스 자체를 지식으로 모델링하는 활동으로 보고 계속적으로 전문가 지식을 모델링하며 개선해 나가는 방법을 취한다<sup>4</sup>.

PLM (Product lifecycle management)의 제품개발 지식은 매우 규모가 크고 복잡하기 때문에 이러한 점진적인 개선 접근 방법을 적용해야 한다. 이 접근 방법을 PLM 지식 모델링에 적용하기

\* 한국과학기술원 산업공학과

\*\* 한국과학기술원 산업공학과

Tel. 042-869-3123, Fax. 042-869-3110

Email hw\_suh@kaist.ac.kr

위해서는, 제품 지식을 모델링하기 위한 프레임워크를 명확히 정의하여 지식 엔지니어가 PLM 지식을 모델링 할 때 가이드 해주어야 한다. 이전에도 몇몇 지식 관리를 위한 프레임워크들이 제안되었지만 모호함과 이질감 없이 제품생명주기에 걸친 여러 유형의 포괄적 지식을 수용하는 데 한계가 있었다. 본 논문에서 제안하는 접근법은 새로운 기술인 온톨로지에 기반을 두고 있다. 자세한 내용은 다음 장에서 논의될 것이다.

2 장에서는 관련 연구들을 소개하고 3 장에서는 온톨로지 기반의 지식 프레임워크를 소개한다. 지식 프레임워크의 주요 세 레벨에 대한 설명이 4 장에서 논의되고, 5 장에서는 다중레벨이 소개된다. 6 장에서는 지식 프레임워크의 PLM 지원에 관하여 논의하고, 7 장에서는 제안된 프레임워크 기반의 지식 베이스의 라이프 사이클에 관하여 논의한다. 마지막으로 결론을 8 장에서 논의한다.

## 2. 관련연구

본 장에서는 온톨로지 기반 지식 프레임워크에 대한 기존 연구들을 살펴보면서 본 연구의 접근방법과 비교하겠다.

초기 온톨로지 관련 연구는 데이터 모델의 통합 문제에서 발생하였다. 서로 다른 두 데이터베이스의 통합을 위해서는 데이터 모델의 의미를 이해해야 했고, 이 의미를 표현하기 위해 온톨로지를 활용하였다<sup>5</sup>. Gruber<sup>6</sup>의 정의에 따르면 온톨로지는 ‘개념화에 대한 명확한 명세화’로써, 도메인의 개념과 관계들을 컴퓨터가 이해 가능하도록 논리적, 정형적으로 기술한 것이다. 또한 Bozsak et al.<sup>7</sup>는 온톨로지가 5 가지 구성요소로(개념, 관계, 계층 구조, 연계 함수, 공리)로 이루어진다고 하였다.

PLM(Product lifecycle management)을 위한 온톨로지는 ‘제품라이프사이클 전문가들 간에 합의된 지식 및 지식의 명세’로 여러 분야가 협업을 할 때 발생하는 정보 의미에 대한 이해를 돕고<sup>27</sup> 정보의 공유를 가능하게 한다<sup>28, 29</sup>. 이러한 기능 측면에서 온톨로지의 구성요소 중 개념, 관계, 계층 구조, 연계 함수는 합의된 지식의 구조를 정의하고, 공리는 이 요소들의 의미를

명세화 하는 역할을 맡는다. PACT<sup>9</sup>, SHADE<sup>10</sup> 연구는 온톨로지를 활용한 에이전트 협업 시스템을 개발하여, 서로 다른 시스템간의 정보 교환시 공유 온톨로지의 활용성을 보여주었다. Yoshioka<sup>8</sup>는 공리를 제외한 나머지 요소들로 구성된 합의된 지식을 ‘Meta-model’이라 정의하고, 이를 엔지니어링 분야에서 서로 다른 시스템들간 정보 교환 시 일관성 유지에 활용하였다. 이 연구는 개념들의 의미를 개념 사전 형식으로 정리함으로써 공리 정의의 복잡함을 피하고, 협업 체계의 일관성 유지를 피하였으나, 시스템 활용 과정에서 온톨로지의 수정/보완 시 발생하는 문제들에 대해서는 해결방법이 없다는 단점이 있다.

이 후 온톨로지의 PLM 분야 적용의 필요성이 커짐에 따라 많은 온톨로지 연구들이 있었는데, 그 중 대표적인 것으로 Lin et al.<sup>11</sup>은 고객의 요구사항들을 관리하기 위한 요구사항 온톨로지를 개발하였고, Kitamura et al.<sup>12</sup>은 제품의 기능을 표현하기 위한 기능 온톨로지를 개발하였다. Borst et al.<sup>13</sup>은 일반적인 물리적 시스템을 표현하는데 활용되는 PHYSYS 온톨로지를 개발하였다. Guarino<sup>14</sup>는 STEP 과 같은 제품형상 표준에 대해 근본적인 온톨로지 접근이 필요함을 주장하며 mereology, topology, morphology 이론을 통해 기하학적 특성과 형상을 정의할 수 있음을 보였다. PSL<sup>15</sup> 프로젝트는 제조분야의 프로세스 관련 지식을 정의하기 위한 프로세스 온톨로지를 정의하고 있다. 또한, Gruber 와 Olsen<sup>16</sup>은 엔지니어링 환경에서 공학적 계산을 위한 수학 온톨로지를 개발하였다. 이러한 기존연구들은 PLM 을 위한 온톨로지 연구의 기반으로 활용 될 수 있다.

PLM 분야에 KBS 의 활용을 높이기 위해서는 온톨로지 기반의 지식베이스를 구축해야 할 뿐만 아니라, 온톨로지 기반의 문제해결 지식들을 수집하고 관리해야 한다. 온톨로지가 전문가들 간에 합의된 공통 지식이라면, 실제 업무에서 발생하는 의사결정을 지원하는 지식은 문제를 해결하는 지식이다. 이러한 문제해결 지식들과 온톨로지는 KBS 에서 통합된 모습으로 관리되어야 한다. CommonKADS<sup>17</sup>, MIKE<sup>18</sup>, Protégé 2000<sup>19</sup> 과 같은 연구들은 이를 위하여 고유의 지식 모델링 프레임워크를 제안하였다.

한편, 온톨로지의 구성요소 중 공리는 지식을 명세화하는 역할을 하므로 매우 중요하지만 모델링이 어렵고, 많은 부분 온톨로지 구축자의 수작업(hand-coding)이 필요하다는 단점이 있다. 하지만, 그 필요성을 무시할 수 없기 때문에 공리를 독립적으로 모델링 하기 위한 연구들도 있었다. Ontolingua<sup>20</sup> 연구에서는 온톨로지 정의시 어떤 부분이 공리인지 명시 할 수 있도록 하여 공리를 부분적으로 분리하여 정의할 수 있도록 하였다. Staab 와 Maedche<sup>21</sup> 는 공리들을 분류하여 개념화하고 온톨로지에서 정의하는 공리들을 하나의 객체로 정의하는 접근방법을 제안하였다.

따라서, 본 연구에서 제안된 PLM 을 위한 제품 지식을 모델링 하기 위해 ‘문제해결 지식’, ‘도메인의 일반적 지식’과 ‘지식의 명세화’로 구분하여 모델링하는 지식 프레임워크를 제안한다. 이와 같은 지식의 구분은 각 지식 레벨의 역할을 명확히 하고, 상호간의 관계를 이해 하는데 도움을 준다.

### 3. 온톨로지 기반 지식 프레임워크

본 논문은 PLM 지원을 위한 온톨로지 기반의 지식 프레임워크를 제안한다. 프레임워크 접근법은 명확한 지식관리를 위해 온톨로지 기반의 지식 구조를 갖는다. 또한 PLM 의 포괄적인 지식관리를 위하여 함수, 규칙, 데이터 분석 기반 지식과 같은 전형적인 유형의 지식을 포함하며, 또한 일반제품, 특정제품, 제품버전, 제품아이템 등의 지식 및 연계 관계를 정의하여 PLM 을 통합적으로 관리 할 수 있게 해준다. 또한, 일관된 표현을 위해서 일차 논리 (FOL: First-Order-Logic)로 표현한다.

#### 3.1 지식 구조

이전 연구들<sup>22, 23</sup> 이 다양한 유형의 제품 관련 지식의 분류체계를 제안하였지만 분류의 기준과 분류된 카테고리 간의 관계성이 명확하게 제시되지 못했다. 본 연구는 지식의 분류를 위해 지식의 역할(role)과 출처(source)라는 두 가지 기준을 정의한다. 우선 지식을 역할에 따라 ‘문제 해결 지식’, ‘도메인의 일반적인 지식’, ‘지식의 명세화’ 세 가지로 분류한다. 또한, ‘문제 해결 지식’은 지식의 출처에 따라 ‘합수적 이론 지식’, ‘데이터 분석 기반 지식’, ‘규칙 기반 전문가 지식’의 타입으로 분

류한다. 기본 지식 구조에 대한 보다 자세한 설명은 4 장에서 기술된다.

#### 3.2 온톨로지 구조와 지식 구조

KAON 의 정의<sup>7</sup> 는 온톨로지의 일반적인 구조에 대해서 정의하고 있다. 이 정의를 바탕으로 온톨로지 구조는 개념, 개념 간의 관계, 계층 구조, 연계 함수, 공리는 지식 구조의 토대가 된다. 개념, 개념 간의 관계, 계층 구조, 연계 함수는 ‘도메인의 일반적 지식’의 기본 구조를 표현한다. 공리는 개념과 관계에 대한 의미를 구체화함으로써 ‘지식의 명세화’ 역할을 한다. 또한 ‘문제 해결 지식’은 온톨로지 개념들의 인스턴스 간의 관계성을 나타내기 때문에 온톨로지를 기반으로 정의된 지식이다.

#### 3.3 일관된 지식 표현

온톨로지의 개념들과 관계들을 상세하고 정확하며 모호성 없이 나타내기 위해서는 논리 기반의 표현으로 나타내는 것이 바람직하다<sup>24</sup>. 본 연구의 프레임워크에 사용될 논리 기반의 표현은 제품 개발 도메인의 복잡한 지식을 표현할 수 있어야 하며, 또한 적절한 추론 능력을 갖추어야 한다. Corcho 와 Perez<sup>25</sup> 은 여러 논리 기반의 언어들에 두고 표현력과 추론의 효율 면에서 비교해 보았다. 그 결과, 언어의 표현력과 추론의 효율 간에는 상충되는 관계가 있다고 하였다. 즉, 표현력이 강할수록 추론 능력은 떨어진다. 본 연구에서는 FOL 을 선택하였다. FOL 표현은 PLM 도메인의 공리는 물론, 개념과 그 관계를 표현하기에 충분하다. 또한 문제 해결 지식의 기반이 되는 추론 알고리즘과의 연계<sup>26</sup> 도 가능하게 한다

### 4. 세가지 주요 지식 레벨

본 논문에서는 앞서 언급한 온톨로지 기반의 PLM 지식인 ‘문제 해결 지식’, ‘도메인의 일반적인 지식’, ‘지식의 명세화’를 각각 도메인 특화 지식(Specialized Knowledge for Domain: SKD), 지식 맵(Knowledge Map: K-Map), 공리(Axiom)로 정의하고, SKD 를 다시 세 가지 타입; 엔지니어링 함수, 전문가 지식, 데이터 분석기반 지식으로 분류한다. 이와 같은 구조를 Fig. 1 과 같이 표현할

수 있다.

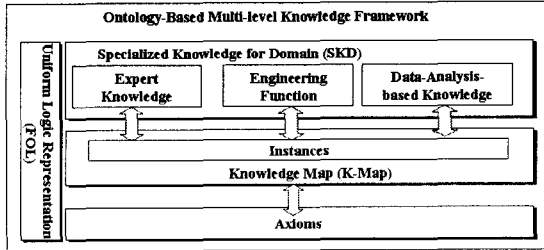


Fig. 1 Ontology-based knowledge framework

### 4.1 레벨 1: 공리

공리는 논리 표현을 통해 도메인에 존재하는 개념들과 관계들의 의미를 명세화하여 사람과 컴퓨터가 각각의 의미를 명확하게 이해할 수 있게 한다. 공리를 통해, 개념들과 관계들의 기본적인 특성과 정의를 기술할 수 있게 된다. 예를 들어, 'subPartOf' 라는 용어가 두 개의 부품 개념 간의 구조 관계를 표현한다고 가정하면 'subPartOf' 관계의 기본적인 특성들은 다음과 같다.

- 'subPartOf' 관계는 재귀적이지 않다; 한 부품은 그 자신의 하위 부품일 수 없다.
- 'subPartOf' 관계는 대칭적이지 않다; 부품 X 가 부품 Y 의 하위 부품이면, 부품 Y 는 부품 X 의 하위 부품일 수 없다.
- 'subPartOf' 관계는 추이적이다; 부품 X 가 부품 Y 의 하위 부품이고 부품 Y 가 부품 Z 의 하위 부품이면, 부품 X 는 부품 Z 의 하위부품이다.

개념들과 관계들의 정의 또한 논리 표현으로 기술될 수 있다. 간단한 예로, 'directSubPartOf'라는 용어가 두 개의 부품 개념 간의 직접적 구조관계를 표현한다면, 두 부품 사이에 다른 부품이 구조관계로 연결되지 않는 것이라고 하자. 그러면 'directSubPartOf' 관계는 다음과 같이 논리적으로 정의할 수 있다.

- 부품 X 가 부품 Y 의 직접적 하위 부품이라는 것은, 부품 X 가 부품 Y 의 하위 부품이고, 부품 Z 가 부품 Y 의 하위 부품이고 부품 X 가 부품 Z 의 하위 부품인

부품 Z 가 존재하지 않는다는 것이다.

우리는 이러한 공리들을 Fig. 2 와 같은 FOL 표현으로 변환할 수 있다.

공리	FOL 표현
1)'subPartOf' 관계는 재귀적이지 않다.	$(\forall p) \neg subPartOf(p, p)$
2)'subPartOf' 관계는 대칭적이지 않다.	$(\forall p1, p2) subPartOf(p1, p2) \Rightarrow \neg subPartOf(p2, p1)$
3)'subPartOf' 관계는 추이적이다.	$(\forall p1, p2, p3) subPartOf(p1, p2) \wedge subPartOf(p2, p3) \Rightarrow subPartOf(p1, p3)$
4)'directSubPartOf' 관계의 정의	$(\forall p1, p2) directSubPartOf(p1, p2) \Leftrightarrow subPartOf(p1, p2) \wedge \neg (\exists p3) \wedge subPartOf(p3, p2) \wedge subPartOf(p1, p3)$

Fig. 2 Example of axioms

### 4.2 레벨 2: 지식 맵 (K-Map)

지식 맵은 의미 네트워크와 같은 구조를 통해 도메인의 공통된 지식을 표현한다. 지식 맵의 구조는 개념, 관계, 연계 함수, 개념 간의 계층 구조, 관계 간의 계층 구조 등으로 이루어 지고, 이것은 다시 일반제품, 특정제품, 제품버전 및 제품아이템 레벨로 이루어 질 수 있다. 개념, 'Part'와 그 관계, 'subPartOf'는 실 세계의 개체와 그 관계를 표현한 말이다. 연계 함수 Rel 은 개념과 관계 간의 연계성을 정의한다. 'subPartOf' 관계가 두 개의 'Part' 개념과 관련되어 있다면, 이것을  $Rel(subPartOf) = (Part, Part)$ 와 같은 논리 표현으로 나타낼 수 있다. 개념 간의 계층 구조는 용어집(taxonomy)이라고 불리기도 하는데 개념 간의 계층 구조 함수,  $H^C$ ,를 통해 표현할 수 있다. 개념 'Assembly'가 개념 'Part'의 하위 개념이라면  $H^C(Assembly, Part)$ 와 같이 표현될 수 있다. 또한 관계 간의 계층 구조는 함수  $H^R$ 에 의해 표현될 수 있다. 예를 들어, 'directSubPartOf' 관계가 'subPartOf' 관계의 하위 관계라면,  $H^R(directSubPartOf, subPartOf)$ 과 같은 논리 표현으로 나타낼 수 있다. Fig.3 은 PLM 도메인의 지식 맵 예제를 나타낸다.

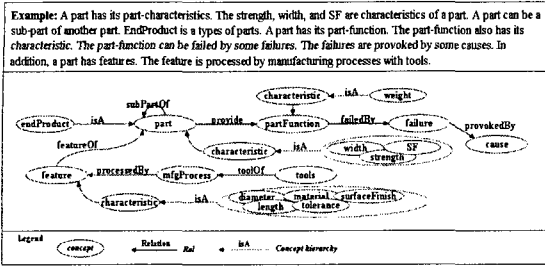


Fig. 3 Example of a K-Map

또한, 지식 맵은 FOL 형식으로 표현될 수 있다. 지식 맵은 도메인 전문가가 동의해야 하며 동의 후에 SKD의 베이스가 될 수 있다. 동의된 지식 맵은 도메인의 참조 모델로써 사용될 수 있다.

**4.3 레벨 3: 도메인 특화 지식(SKD)**

SKD는 사용자의 문제를 해결하는 역할을 수행하며 의사 결정을 돕는다. SKD는 특정 업무나 문제에 관한 것이기 때문에 SKD의 논리적 표현은 지식 맵의 개념과 그 관계 그리고 인스턴스를 통해 이루어진다. 따라서, SKD는 도메인의 지식 맵에 기초 하여 표현되고 지식 맵의 인스턴스 간의 관계에 대한 지식을 제공한다. SKD는 지식의 출처를 기준으로 엔지니어링 함수, 전문가 지식 그리고 데이터 분석 기반 지식의 세 가지 유형으로 분류될 수 있다.

첫번째, 엔지니어링 함수는 과학적 이론으로부터 나오고 수학적인 표현으로 나타내어진다. ‘Leg’ 부품의 너비와 다른 요인들로 이루어진 다음의 방정식이 좋은 예이다. Ws, X, St 그리고 SF가 지식 맵의 ‘개념’이다. 엔지니어링 함수는 지식 맵의 개념들 간의 엔지니어링 관계를 수식으로 표현한 것이다.

(SKD-1):  $X \geq (Ws / St * SF)^{1/2}$   
 X: width of a Leg part    Ws: student’s weight  
 St: material’s tensile strength    SF: Safety Factor

두번째, 전문가 지식은 전문가의 경험이나 직관으로부터 나온 지식을 나타내며 일반적으로 정성적 지식이 된다. 전문가 지식은 일반적인 규칙처럼 ‘IF.. THEN..’의 형태를 가진다. 다음의 논리 표현이 전문가 지식의 예이다.

(SKD-2):

- IF feature.type = Closed\_circular\_hole THEN mfgProcess= Drilling.
- IF mfgProcess=Drilling AND feature.material = Wood THEN tool = HandDrill.

예에서, ‘feature.type’, ‘part.material’ 그리고 ‘tool’은 지식 맵의 ‘개념’이고 ‘Closed\_circular\_hole’, ‘Wood’, ‘Drilling’, ‘HandDrill’은 그 개념들의 인스턴스이다.

마지막으로 데이터 분석기반 지식은, 데이터 분석을 통해 얻어지므로 수집된 데이터 및 데이터 분석 기법과 관련되어 있다. 예를 들어, ‘FailureChar’의 인스턴스인 ‘Slackness’와 ‘CauseChar’의 인스턴스들인 ‘Hole & Peg Tolerance’, ‘Relative Surface Friction’, ‘Peg strength’에 대한 여러 데이터가 수집되어 데이터 테이블이 구성되었다면 수집된 데이터를 바탕으로 각 속성들 간의 관계성을 분석하게 된다. 데이터 테이블에 대한 분석 결과로 ‘DataCubic’이 생성될 수 있고, 선형 회귀 분석외에 여러 데이터 마이닝 분석방법들을 통하여 명확한 지식을 정의할 수 있다.

SKD의 예는 FOL 표현을 이용하여 표현될 수도 있는데 Fig. 4는 각 유형의 SKD에 대한 FOL 표현 예를 보여주고 있다.

Type of SKD	FOL formulae of examples
엔지니어링 함수 (SKD-1):	$(\forall w1 w2 w3) Width(w1) \wedge Weight(w2) \wedge MaterialStrength(w3) \wedge (SF = 3) \wedge w1 \geq squar((\wedge / w2 SF) w3).$
전문가 지식 (SKD-2):	$(\forall x y z1 p) HasFeature(x y) \wedge Part(x) \wedge Feature(y) \wedge HasFeatureChar(y z1) \wedge FeatureType(z1) \wedge (= z1 ClosedCircularHole) \Rightarrow (= p Drilling)$ $(\forall y z2 p t) HasFeatureChar(y z2) \wedge Material(z2) \wedge (= z2 Wood) \wedge HasMfgProc(y p) \wedge MfgProc(p) \wedge ProcessedBy(p t) \wedge Tool(t) \wedge (= p Drilling) \Rightarrow (= t HandDrill)$
데이터 분석 기반 지식 (SKD-3):	$(\forall y x1 x2 x3) Slackness(y) \wedge HolePegTolerance(x1) \wedge RelativeSurfaceFriction(x2) \wedge PegStrength(x3) \wedge (= y (+ 5.12 (+ (* (-0.06) x1) (+ (* (-29.9) x2) (* 27.0 x3))))).$

Fig. 4 Examples of SKD in FOL

### 5. 다중 지식 레벨

4.2 절에서 K-Map 은 개념들, 개념들과 관련된 인스턴스들을 갖는다는 것을 보였다. 만약 개념들이 보편적 제품개념(예를 들어, ‘제품’, ‘부품’, 등) 들이라면, 그 인스턴스들은 특정 제품(예를 들어, ‘의자’, ‘자동차’, 등)이 된다. 또한, 개념들이 보편적 제품개념이라면, 그 인스턴스들은 특정 제품의 여러 설계 디자인 버전들로 볼 수 있다. 또한, 개념을 제품 디자인 버전으로 본다면, 그 인스턴스들은 이 버전을 통해 생산된 아이템이라고 볼 수 있다. 따라서, K-Map 은 제품 전수명주기를 아우르기 위해 제품개발 과정에서 발생하는 지식들을 제품개념에 대한 네 레벨로 분류할 수 있다.

- 일반제품 레벨: 특정 제품에 종속되지 않는 제품/부품 개념 (Generic product level)
- 특정제품 레벨: 참조 모델로서 제품/부품 개념 (Specific product level)
- 제품버전 레벨: 프로젝트 별 설계 제품/부품 개념 (Product version level)
- 제품아이템 레벨: 물리적 제품/부품 개념 (Product item level)

Fig. 5 는 ‘의자’에 대한 다중 레벨의 지식맵을 가시화한 것이다. 이 그림에서 각 레벨의 지식맵이 갖는 개념들과 관계들을 보이고 있으며, 네 레벨들간의 인스턴스 관계를 보여주고 있다.

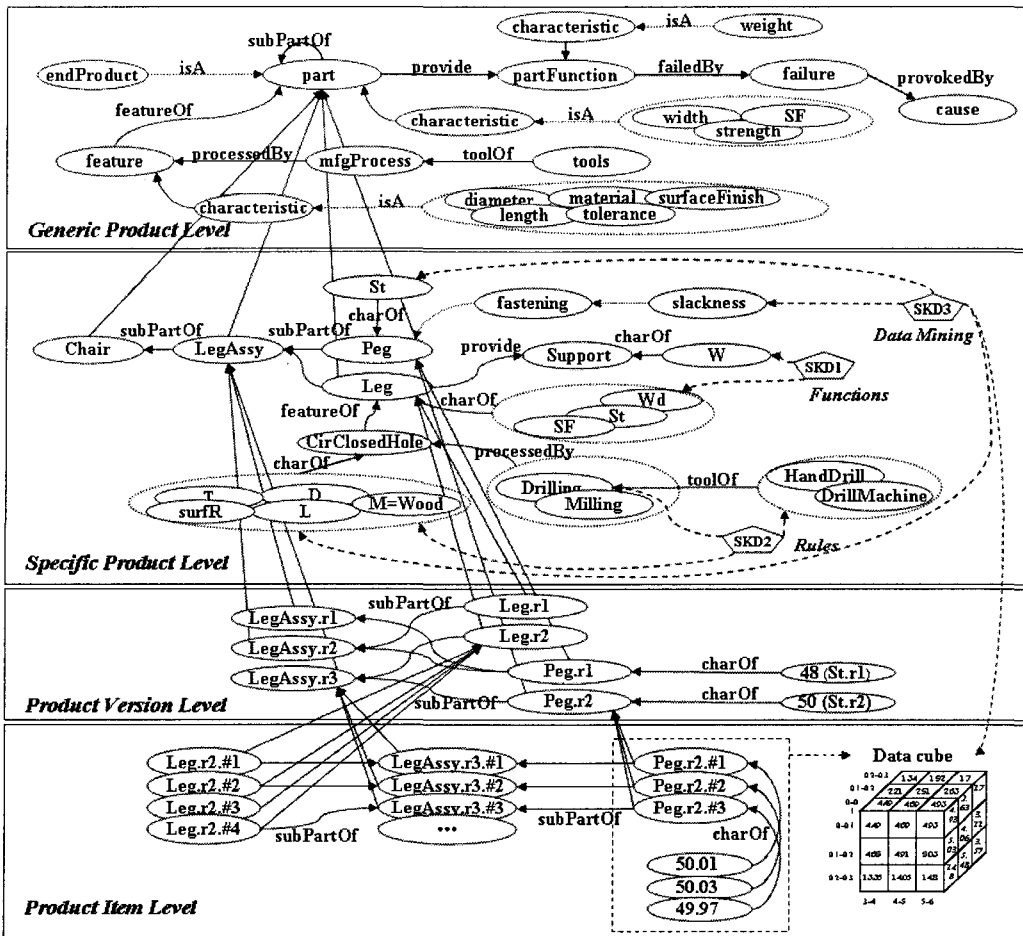


Fig. 5 Example of multi-level K-Maps for a ‘Chair’ product

각 레벨은 다른 레벨과 인스턴스화 관계를 갖는데, 인스턴스화 관계에서 높은 레벨은 낮은 레벨의 메타 모델이 된다. 이러한 인스턴스화 관계는 Guarino<sup>13</sup> 가 정의한 도메인 온톨로지와 애플리케이션 온톨로지 간의 관계와 유사하다. 공리와 SKD 는 네 레벨과 통합된다. 대부분 지식의 의미 정의는 일반제품 레벨과 특정제품 레벨의 지식맵에서 가장 필요하기 때문에, 공리는 이 레벨들에서 특히 중요하다. 전문가 지식과 엔지니어링 함수 지식은 보통 일반제품 레벨과 특정제품 레벨에서 정의되고, 제품버전 레벨과 제품아이템 레벨에서 활용된다. 데이터분석기반 지식 또한 일반제품 레벨 또는 특정제품 레벨에서 정의되지만, 이 지식이 개발되기 위해서는 제품버전 레벨 또는 제품아이템 레벨의 데이터들 분석을 통해 개발된다.

### 6. 다중 지식 레벨과 PLM 지원

PLM 의 각 단계들에서 참여자들은 모두 동일한 제품을 개발, 생산 및 공급 하기 위해 서로 다른 관점을 가지고 참여한다. 참여자들의 서로 다

른 관점은 각 관점 중심의 제품 지식을 만들어낸다. 이러한 제품 지식들은 다른 부문에서 필요로 하는 경우가 많다. 예를 들어, 설계자는 마케팅 부서에서 수행한 고객의 요구사항 결과를 파악하고자 한다. 그러나 표현된 지식만으로는 지식의 정확한 의미를 파악하기가 어렵다. 따라서 각 단계에서 정의되는 각 제품 관련지식들의 명세화 및 통합화가 필요하다.

본 연구는 서로 다른 관점의 제품 지식들을 명세화하고 통합하기 위해서 제품개발 지식 맵을 일반제품 레벨, 특정제품 레벨, 제품버전 레벨, 제품아이템 레벨 등 4 레벨로 정의하였다.

Fig. 6 은 PLM 단계에 따라 생성되는 대표적인 PLM 지식을 4 레벨에 따라 표시한 것이다. 각 단계에서 활용되는 PLM 지식은 상황에 따라 매우 다양하다.

PLM 의 각 단계는 4 레벨 관점에서 제품개발 지식을 생성하고 활용하고 있다. 일반제품 레벨의 지식 맵과 공리는 제품 라이프 사이클의 모든 단계에서 활용된다. 예를 들어, 온톨로지 레벨로 정의된 STEP 은 전 제품 라이프 사이클 단계에서 공통적으로 활용이 가능하다.

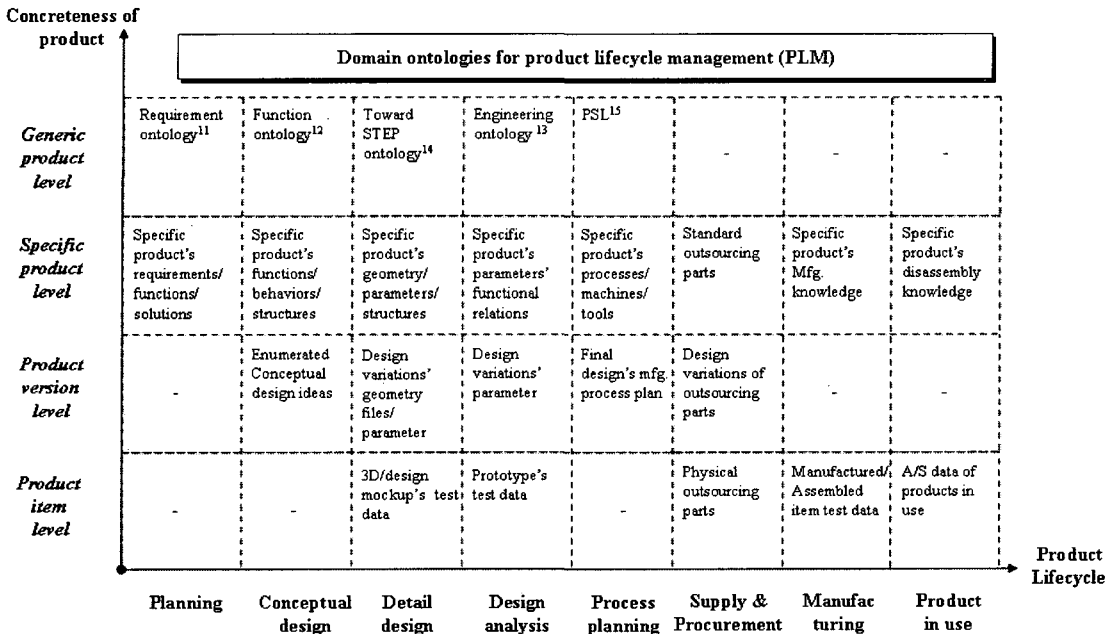


Fig. 6 PLM knowledge examples according to the 4-level viewpoint

특정제품 레벨의 제품개념에 대한 지식맵, 공리와 SKD 는 각 제품 라이프 사이클 각 단계에서 서로 다른 관점을 바탕으로 생성되고 활용된다. 예를 들어, 개념적 설계 단계에서는 특정 제품/부품들의 대표적인 기능/구조/행위들을 정의하고, 상세설계 단계에서는 특정 제품/부품들의 대표적인 형상/특성들을 정의한다. 특정제품 레벨에서 정의된 지식맵과 SKD 들은 참조 모델이 되어 제품버전 레벨의 제품/부품들에 적용된다. 제품버전 레벨의 제품개념에 대한 지식맵, SKD 와 공리는 특정 제품 버전(version)들에 관련된 지식을 표현한다. 제품 버전은 고유한 설계 사양을 갖고 있다. 제품아이템 레벨의 제품개념은 물리적 제품/부품에 대한 지식맵으로써, 제품버전 레벨의 설계 버전들을 바탕으로 만들어진다. 하지만, 제품버전 레벨과는 달리 주로 실측된 특성 값을 갖는다. 제품을 개발하는 과정에서 필요에 의해 만드는 여러 프로토타입들과 생산을 통해 만들어진 각 제품 및 소비자가 구매한 각 제품들 각각에 대한 지식이 제품아이템 레벨에 해당한다.

## 7. PLM 지원 지식 프레임워크 라이프 사이클

이 절에서 우리는 제안된 프레임워크의 생명주기를 논의하고자 한다. 즉, 제안된 프레임워크 기반의 지식베이스가 어떻게 구축되고, 어떻게 제품개발에 적용되며, 어떻게 지식베이스를 유지보수 및 개선하는지를 보인다.

### 7.1 구축 단계

구축단계에서 SKD, 지식맵, 공리를 포함하는 지식베이스를 구축한다. 이 단계는 도메인 전문가와 지식 엔지니어가 함께 수행한다. 구축 단계에서 우선 지식맵을 정의한다. 왜냐하면 지식맵은 주어진 도메인에 대한 공통적인 기본 지식이기 때문이다. 그 후 공리를 정의한다. 왜냐하면 공리를 통해 지식맵의 개념들과 관계들을 명확히 명세할 수 있다. 마지막으로 SKD 를 지식맵과 공리에서 정의한 개념들과 관계들을 바탕으로 정의한다. SKD 는 주어진 특정 문제를 위한 특화된 지식이고 이를 명세하기 위해서는 지식맵과 공리가 필요하기 때문이다. 또한, 구축 단계에서 일반제품 레벨과 특정제품

레벨 같은 지식맵의 두 레벨이 정의되어야 한다. 일반제품에 대한 지식맵은 특정제품에 대한 지식맵보다 먼저 생성되어야 한다. 특정제품 지식맵은 일반제품 지식맵으로부터 인스턴스 관계를 통해 구축되기 때문이다. 일반제품 또는 특정제품에 대한 지식맵, 공리, SKD 를 구축하는 것은 지식 엔지니어링 프로세스이다. 이 프로세스에서 지식엔지니어는 도메인 전문가의 암묵적 지식을 명시적 지식으로 도출한다. 따라서 우리의 지식 프레임워크는 명시적 지식을 활용하기 적절하게 구성하는 것뿐만 아니라 암묵적 지식을 수용하여 명시적 지식으로 도출할 수 있도록 도와준다.

### 7.2 적용 단계

지식 프레임워크를 제품개발에 적용한다는 것은 제품개발 엔지니어가 지식베이스, 특히 일반제품 또는 특정제품 레벨의 지식베이스, 지원을 받으며 각자의 프로젝트를 수행하는 것을 의미한다. 지식베이스를 활용하여 각자의 프로젝트를 수행하는 동안, 각 프로젝트의 산출물들은 제품버전 레벨과 제품아이템 레벨의 지식맵에 저장된다. 따라서, 제품버전 레벨과 제품아이템 레벨의 지식맵은 제품개발 프로젝트의 산출물을 모아 놓은 지식베이스이다.

구축단계에서 개발된 지식맵, 공리, SKD 들은 설계 엔지니어를 가이드하거나 제품 특성값을 결정하는데 활용된다. 제품맵은 개념들과 관계들을 의미 네트워크 형태로 가시화 하기 때문에, 지식맵은 엔지니어가 설계 특성들의 관계를 파악하고, 관계를 따라가 볼 수 있도록 가이드 한다. 각 지식맵의 공리들은 설계 엔지니어들에게 지식맵의 개념과 관계들의 의미에 대하여 알려줄 뿐만 아니라, 새로운 지식을 추론하거나 검증하는데 활용될 수 있다. SKD 는 문제에 특화된 지식이기 때문에, 설계 엔지니어가 부딪힌 문제를 해결하는데 직접적으로 활용된다. 엔지니어가 설계 특성들의 값을 결정하고자 할 때, 특성들과 관련된 SKD 가 있는지를 먼저 검토한다. 만약 관련된 SKD 가 있다면, 엔지니어는 그들이 정한 값이 적절한 값인지 SKD 를 통해 검토할 수 있고, 적절한 값을 제안 받을 수도 있다.



### 7.3 유지보수 단계

지식베이스의 유지보수 또는 업데이트는 도메인 전문가와 지식엔지니어가 수행한다. 지식베이스 업데이트는 일반제품 레벨 또는 특정제품 레벨과 주로 관련된다. 제품버전 레벨과 제품아이템 레벨의 정보들은 특정제품에 대한 여러 프로젝트 정보들을 축적하고 있다. 축적된 정보의 양이 특정 수준에 이르러 지식베이스의 업데이트가 필요하다면, 제품버전 레벨과 제품아이템 레벨의 정보들을 바탕으로 특정제품 레벨의 지식맵을 업데이트 한다. 지식엔지니어들은 새로운 지식을 정의하거나 새로운 지식을 수용할 수 있도록 지식맵을 수정할 수 있다. 또한 SKD 와 공리들도 새롭게 정의할 수 있다.

## 8. 결론

우리는 PLM 지원을 위한 엔지니어의 지식을 체계적으로 저장하고 활용하기 위한 온톨로지 기반 지식 프레임워크를 제안하였다. 프레임워크를 온톨로지 기반으로 하여 지식의 명세화, 통합화를 추구 하였고, 제품 지식의 4 가지 특성인, 일반제품, 특정제품, 제품버전 및 제품아이템에 대한 지식을 관리 할 수 있도록 하였다.

제안한 프레임워크가 도메인의 근본적인 개념 및 관계의 의미와 도메인의 지식들을 일관되게 표현할 수 있는 구조를 가지고 있더라도, 이를 상업적으로 실용성 있게 활용하기 위해서는 여전히 많은 연구가 필요하다. 통합 제품 온톨로지에 대한 연구와 이에 대한 성공적 적용사례들이 있다면 협업적 제품개발 환경을 위한 지식관리부분에 대한 다른 연구들도 더욱 발전할 수 있을 것이다.

## 참고문헌

1. Syan, C. S. and Menon, U., "Concurrent Engineering: Concepts, implementation and practice," Chapman & Hall, 1994.
2. Kuffner, T. A. and Ullman, D. G., "The information requests of mechanical design engineers," Design Studies, Vol.12, No.1, pp. 42-50, 1997.
3. Stauffer, L. A. and Ullman, D. G., "Fundamental process of mechanical designers based on empirical data," Journal of Engineering Design, Vol.2, pp.113-125, 1991.
4. Studer, R., Benjamins, V.R. and Fensel, D., "Knowledge Engineering: Principles and methods," Data & knowledge engineering, Vol. 25, No. 1/2, pp. 161-197, 1998.
5. Wache, H., Vogege, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hubner, S., "Ontology-Based Integration Information: A Survey of Existing Approaches," IJCAI Work-shop on Ontologies and Information Sharing 2001.
6. Gruber, T. R., "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition, Vol. 5, pp. 199-220, 1993.
7. Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S. and Stojanovic, L., "KAON - Towards a Large Scale Semantic Web," Lecture notes in computer science, No.2455, pp. 304-313, 2002.
8. Yoshioka, M. and Tomiyama, T., "Pluggable metamodel mechanism: A framework of an integrated design object modelling environment," Computer Aided Conceptual Design '97, Proceedings of the 1997 Lancaster International Workshop on Engineering Design CACD'97, pp. 57-70. Lancaster University, 1997.
9. Cutkosky, M. R., Engelmores, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M. and Weber, J. C., "PACT: an experiment in integrating concurrent engineering systems," Computer, Vol. 26, No. 1, pp. 28-37, 1993.
10. Kuokka, D. R., McGuire, J. G., Pelavin, R. N. and Weber, J. C., "SHADE: Technology for knowledge-based collaborative engineering," Artificial intelligence in collaborative design, pp. 245-262, 1994.
11. Lin, J., Fox, M. S. and Bilgic, T., "A Requirement Ontology for Engineering Design," Concurrent engineering, research and applications, Vol.4, No.3, pp. 279-292, 1996.
12. Kitamura, Y., Sano, T. and Mizoguchi, R.,

- “Functional Understanding Based on an Ontology of Functional Concepts,” Lecture notes in computer science, No.1886, pp. 723-733, 2000.
13. Borst, P., Akkermans, H. and Top, J., “Engineering ontologies,” International journal of human-computer studies, Vol.46, No.2/3, pp. 365-406, 1997.
  14. Guarino, N., Borgo, S. and Masolo, C., “Logical modelling of product knowledge: towards a well-founded semantics for STEP,” European conference on product data technology, sohpia antipolis, France, 1997.
  15. Gruninger, M. and Menzel, C., “The Process Specification Language (PSL): Theory and Applications,” AI magazine, Vol.24, No.4, pp.63-74, 2003.
  16. Gruber, T. R. and Olsen, G. R., “An Ontology for Engineering Mathematics,” Proceedings Fourth International Conference on Principles of knowledge representation and reasoning, pp. 258-269, 1994.
  17. Kingston, J. K. C., “Designing knowledge based system: the CommonKADS design model,” Knowledge-Based System, Vol. 11, pp. 311-319, 1998.
  18. Angele, J., Fensel, D., Landes, D. and Studer, R., “Developing knowledge-based systems with MIKE,” Automated Software Engineering, Vol. 5, pp. 389-418, 1998.
  19. Noy, N. F., Fergerson, R. W. and Musen, M. A., “The knowledge model of Protégé-2000: Combining interoperability and flexibility,” Knowledge engineering and knowledge management, pp. 17-32, 2000.
  20. Fikes, R., Farquhar, A. and Rice, J., “Tools for assembling modular ontologies in Ontolingua,” Artificial intelligence, pp. 436-441, 1997.
  21. Staab, S. and Maedche, A., “Axioms are objects, too: Ontology Engineering Beyond the Modeling of Concepts and Relations,” Workshop on Ontologies and Problem-Solving Methods, Berlin, 2000.
  22. Ferguson, E. S., “Engineering and the Mind’s Eye,” MIT Press, Cambridge, MA, 1992.
  23. Vincenti, W. G., “What engineers know and how they know it: analytical studies from aeronautical engineering,” John Hopkins University Press, 1990.
  24. Heflin, J., “OWL web ontology language use cases and requirements W3C Recommended, February,” <http://www.w3.org/TR/webont-req/>, 2004.
  25. Corcho, O. and Perez, A. G., “A Roadmap to Ontology Specification Languages,” Lecture notes in computer science, No.1937, pp. 80-96, 2000.
  26. Russel, S. and Norvig, P., “Artificial Intelligence, 2nd edition,” Prentice Hall, 272-315, 1995.
  27. Kim, K.Y. and Suh, H.W., “An Approach to Semantic Mapping using Product Ontology for CPC Environment,” Journal of the Korean society of CAD/CAM. Vol. 9, No. 3, pp.192-202, 2004.
  28. Kim, H., Kim, H.-S., Lee, J.-H., Jung, J.-M., Do, N.-C. and Lee, J. Y., “CPC Framework for Sharing Product Information Across Enterprises,” Journal of the Korean society of CAD/CAM., Vol. 8, No. 4, pp. 201-211, 2003.
  29. Lee, Y. S., Cheon, S.U. and Han, S.H., “Building Feature Ontology for CAD System Interoperability,” Journal of the Korean society of CAD/CAM., Vol. 9, No. 2, pp 167-174, 2004.