

## CPC 환경을 위한 Product 온톨로지 기반 의미 공유 접근법

김경영\*, 서효원\*\*

### An Approach to Semantic Mapping using Product Ontology for CPC Environment

Kim, K.-Y.\* and Suh, H.-W.\*\*

#### ABSTRACT

This paper introduces an approach to semantic mapping using Product ontology for CPC environment. In CPC environment, it is necessary that the participants in a product life cycle should share the same understanding about the semantic of product terms. For example, they should know that although 'COMPONENT' and 'ITEM' are different word-expressions, they could have the same meaning. In order to handle such terms in the information system, it is desirable that the system automatically recognizes that the terms have the same semantics. Serving this purpose, we described an ontology design methodology using first order logic, knowledge interchange format, and knowledge engineering process. In our approach, we investigated domain knowledge of the Bill Of Material, and then designed Product ontology of it. Based on the ontology, we described syntactic translation, semantic translation, and semantic mapping procedure with an example.

**Key words** : First order logic, KIF, Ontology, Semantic mapping, product ontology, Information sharing, CPC

#### 1. 서 론

제품설계, 엔지니어링, 생산, 구매, 판매, 마케팅, 현장 서비스 등의 제품의 라이프 사이클 단계들을 Web으로 결합시키는 CPC(Collaborative Product Commerce) 환경에서 제품, 프로세스, 조직 관련 정보의 공유 및 교환은 필수 요건이다<sup>1)</sup>. 이러한 정보들은 사용자들 간에 동일한 의미(semantic)로 이해되어야 하나, 동일한 의미를 다양한 어휘를 사용하여 나타내는 경우가 많다. 예를 들어, 기업A에서 원자재로부터 가공된 부품을 표현하기 위해 component를 사용할 때, 다른 기업에서는 동일한 부품에 대해 item이라는 어휘를 사용할 수 있다. 이 경우 제품 정보를 공유할 때 이들이 다른 어휘로 표현되어 있다는 이유로 동일한 부품으로 인식되지 못할 수 있다. 이러한 문제는 제품 라이프 사이클에 연관된 정보 시스템 간의 통합을 어렵게 만든다.

본 연구는 이러한 문제를 해결하기 위해 온톨로지를 이용한 접근방안을 제시한다. 즉, 부품 정보에 대한 온톨로지를 구성하여 이를 바탕으로 서로 다른 용어로 표현된 부품들을 semantic mapping을 통해 제품 정보 공유가 용이하도록 하는 방법이다.

Fig. 1은 본 연구의 접근 방안의 순차적 절차를 보여주고 있다. 제품 정보 공유를 위해서는 먼저 domain knowledge를 파악하여 knowledge engineering 과정에서 이를 논리 언어로 표현하여 온톨로지 디자인해야 한다. 이러한 온톨로지를 기반으로 semantic mapping을 통해 제품 정보 공유를 실현하게 되는 것이다.

이와 같은 절차에 따라, 이 논문에서는 온톨로지 디자인의 기본 요소인 first order logic, KIF(Knowledge Interchange Format) 등을 소개하고, 온톨로지 개요 및 knowledge engineering 과정을 기술한다. Knowledge engineering 과정에서는, 본 논문에서 선택한 BOM 구조자식을 KIF를 이용한 Product 온톨로지 디자인하는 절차를 기술한다. 나아가, Product 온톨로지를 바탕으로 semantic mapping과 정보 공유를 위한 예제를 서술한다.

\*학생회원, KAIST  
\*\*중신회원, KAIST  
- 논문투고일: 2003. 09. 24  
- 심사완료일: 2003. 12. 23

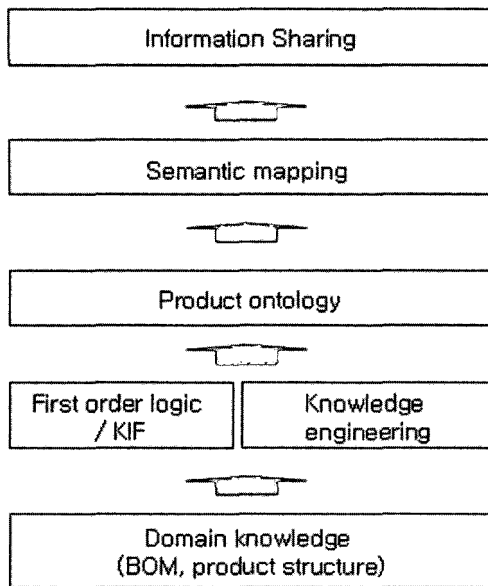


Fig. 1. Implementation procedure for the parts information sharing.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 기존 연구 내용을 서술한다. 3장에서는 온톨로지 디자인에 필요한 기본 요소와 절차, 방법을 기술하고, 이에 따라 Product 온톨로지를 디자인한다. 4장에서는 Product 온톨로지 기반의 semantic mapping의 방법을 제시한다. 5장에서는 semantic mapping 응용 예제를 서술한다.

## 2. 관련 연구

제품정보의 온톨로지 표현에 관한 연구로는 TWIST project 그룹의 연구<sup>[2]</sup>와 TOVE project<sup>[3]</sup>가 있고, semantic mapping에 관한 연구로는 PSL project<sup>[4]</sup>, Lukasz의 연구<sup>[5]</sup> 등이 있다. 본 논문은 위 관련 연구들에서 보완되고 해결되어야 할 부분에 주제의 초점을 맞추고 있다.

TWIST project 그룹 연구의 주제는, 자연과학과 공학에 관련된 애플리케이션을 개발하기 위한 기본 개념과 방법을 연구하는 것이다. 이 연구는 방대한 스케일의 온톨로지의 경우 그 실용적인 사용에 있어 실현된 점을 지적하고, 이에 대한 해결책을 제시하였다. 즉, 대규모 온톨로지를 효과적으로 구성하기 위하여, 도메인 지식을 작게 분해하여 이에 대응하는 온톨로지를 개발한 뒤 결합하는 모듈방식을 사용하였다. TWIST project 연구는 열전달 등의 physical system

을 표현할 수 있는 개념들을 온톨로지로 디자인하였으나, 상대적으로 BOM정보를 표현할 수 있는 개념과 어휘는 부족하다. 엄밀히 말하면, TWIST에서 정의한 온톨로지는 제품정보를 표현하기 위한 것은 아니며, 오히려 이를 참고하여 Product 온톨로지를 디자인할 수 있다고 생각된다.

TOVE(The Toronto Virtual Enterprise) project는 기업 모델링을 위한 통합된 온톨로지의 개발을 제안하였다. 즉, 온톨로지를 이용하여 기업 간에 보다 효과적인 의사소통과 비즈니스 통합을 가능하게 하는 term과 definition을 제공하고자 하였다. TOVE project 역시, BOM 등의 제품정보를 표현하기 위한 온톨로지를 정의하지 않았으며, 파트 간의 부모, 자식 관계의 간단한 구조만을 표현할 수 있다.

NIST(National Institute of Standards and Technology)가 권장하는 PSL project는 제조 도메인에서의 프로세스 정보 공유를 목적으로 한다. PSL은 ISO18629로서 표준화가 진행중이며 제조, 설계 프로세스를 묘사하기 위한 어휘를 온톨로지로 디자인하였다. 또한, NIST는 PSL 온톨로지를 이용하여 ProCAP process modeling tool과 ILOG Scheduler사이의 프로세스 정보 공유 사례를 제시하였다<sup>[6]</sup>. 그러나, PSL project의 경우 어휘 간의 의미 자동 매핑에 대한 방향성은 제시하고 있으나, 그 구체적인 접근법은 제시하고 있지 못하다.

한편, Lukasz 등은 도메인의 정보를 표현한 stand-alone XML문서 사이에서 XML 태그의 구조와 machine learning, XMapper, DataSqueezer 알고리즘을 이용하여 semantic mapping 방안을 제시하였다. 이 연구는, 구체적인 semantic mapping 방안이 언급되었으나 방법의 특성 상 완벽한 의미매핑을 보장하지 못한다. 온톨로지를 이용한다면 개념이 공통적으로 표현된 부분에 한해서 완벽한 의미매핑을 구현할 수 있다.

본 논문은 위에서 열거한 문제점들을 보완하여 Product 도메인 지식을 체계적인 방법론을 통하여 온톨로지로 구성하고 나아가 어휘 간의 자동 의미매핑에 대한 방안을 제안한다.

## 3. Product 온톨로지 디자인

### 3.1 Product 온톨로지 표현 언어

본 연구에서는 semantic mapping을 위한 Product 온톨로지 표현언어로 First order logic과 KIF를 사용하였다.

지식 표현 언어 중의 하나인 first order logic은 실세계를 object들의 집합체로 표현하는 방법이다. Object들을 구별 가능하게 하는 것은 property이며 first order logic은 실세계의 사실(fact)을 object와 property, 그리고 object간의 관계(relation)를 바탕으로 표현한다<sup>7)</sup>. 예를 들어, “Squares neighboring the wumpus are smelly”라는 사실은 wumpus, square라는 object들과 smelly라는 property, neighboring이라는 relation의 집합체로 표현된다. 실세계가 기본적으로 object와 relation으로 이루어졌다고 보는 것은 어떤 object에 대한 사실 표현과 그것에 대한 여러 가지 추론을 가능하게 한다.

‘고양이는 포유동물이다’란 사실을 first order logic으로 표현하면 다음과 같다.

$$\forall x \text{ Cat}(x) \Rightarrow \text{Mammal}(x)$$

$\forall$ 는 universal quantifier로서 변수 전체에 대해 지칭하여 ‘or all’의 의미를 갖는다. Cat은 predicate로서 variable x가 Cat이라는 의미를 부여한다. 일반적으로 predicate(a, b)는 “a is a ‘predicate’ of b”의 형태로 읽는다. Function은 어떤 하나의 object를 지칭하게 되는 return value가 있는 relation이다. 예를 들어, LeftLegOf(John)은 ‘John의 왼쪽다리’라는 object를 지칭하게 한다.  $\Rightarrow$ 는 imply의 의미이며 전체적으로 위 문장은 다음과 같이 자연어로 풀어쓸 수 있다.

*For all x, if x is a cat, then x is mammal.*

또 다른 지식 표현 수단인 KIF는 다른 기종의 프로그램 사이에서 지식 교환을 하기 위한 computer oriented language이다<sup>8)</sup>. KIF는 first order logic을 computation환경에서 사용 가능하도록 그 표현을 정형화시킨 것이다.

KIF는 first order logic에 대한 prefix notation으로서, KIF에서의 변수표현은 prefix인 ?로 시작한다. 또 sentence간의 연결은  $\Rightarrow$ ,  $\Leftarrow$ ,  $\Leftrightarrow$  operator로 표현한다. Term간의 동일성(equality)은 = relation으로 정의된다. Relation은 constant로 표시하는데 predicate symbol과 object들의 관계를 표시하는 term의 두 가지 역할을 한다. Function은 relation의 특수한 경우인데, N개의 argument를 가진 function은 N+1개의 argument를 가진 relation과 동일하다. 즉, relation에 있어서 N+1번째의 argument가 그 function의 함수값과 동일한 것이다. Definition은 function을 선언하는 deffunction, object를 선언하는 defobject, relation을 선언하는 defrelation 등의 operator를 통해 표현된다.

고양이가 포유동물이란 사실을 KIF로 표현하면 다음과 같다.

$$\begin{aligned} &forall (?x) \\ &(\Rightarrow (\text{cat } ?x) \\ &(\text{mammal } ?x)) \end{aligned}$$

### 3.2 온톨로지 개념

이 장에서 Product 온톨로지 디자인과 관련하여 온톨로지 기본 개념에 대해 정리한다. 일반적으로 온톨로지는 존재하는 것과 그것의 기본적인 범주를 연구하는 학문으로 정의한다. 온톨로지는 분야에 따라 조금씩 다르게 정의되는 데, 인공지능 및 지식표현 분야에서 널리 쓰이는 온톨로지의 정의는 Gruber<sup>9)</sup>가 제시한 것이다. 그의 정의에 따르면, 온톨로지는 개념화(conceptualization)에 대한 명시적인 명세사항(specification)이다. 즉, 암묵적인 지식을 명시화시켜 정형화된 표현으로 서술하고 있다는 것이다. 일반적으로, 철학적인 관점에서의 온톨로지는 세계의 특정 시각을 설명하는 분류/범주(categories) 시스템이며, AI 분야에서의 온톨로지는 특정 실체를 기술하기 위해 사용되는 어휘집과 어휘집의 단어의 의미에 관한 명시적인 가정으로 구성된 공학적 가공물이다.

온톨로지를 구성하는 기본 요소에는 다음과 같은 것들이 있다<sup>10)</sup>.

$$O := \{C, R, H, \text{rel}, A_o\}$$

C, R: concept, relation.

H: concept hierarchy 또는 taxonomy.  $H \subseteq C \times C$ .

$H(C1, C2)$ 는 C1이 C2의 subconcept임을 의미한다.

rel: function.  $R(C1, C2)$  등의 형태로 표현한다.

$A_o$ : axiom.

Fig. 2는 온톨로지 요소들의 구체적 의미를 설명하기 위한 간단한 예제이다. Fig. 2의 비정형적인 표현은 “Employee work at organization”이다.

$$C := \{x_1, x_2, x_3\}$$

$$R := \{x_4\}$$

$$H(x_2, x_1)$$

$$x_4(x_2, x_3)$$

$$L_o = \{“Person”, “Employee”, “Organization”\}$$

$$LR = \{“work at organization”\}$$

$$F(“Person”) = x_1$$

$$F(“Employee”) = x_2$$

$$F(“Organization”) = x_3$$

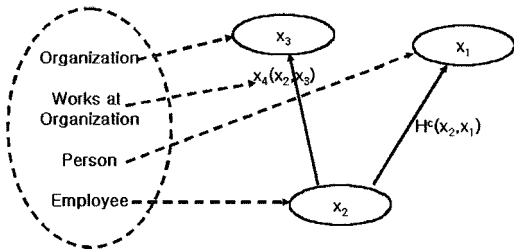


Fig. 2. Example: Instantiate Ontology Structure.

$$G(\text{"work at organization"}) = x_4$$

위에서 concept은  $x_1, x_2, x_3$ 이며, relation은  $x_4$ 이다. ' $x_2$ 가  $x_1$ 의 subconcept이다'라는 taxonomy가 설정되어 있으며, lexicon set(instance)은  $L_i$ 와  $L_R$ 이다. 또한, function  $F$ 와  $G$ 는 lexicon들을 온톨로지의 concept과 relation으로 매핑시키고 있다. 이를 통하여 'Employee는 Person의 subconcept', 'Employee work at organization'이라는 지식이 표현되고 있다. Axiom이란 자명한 것이라고 생각되는 명제, 즉 증명을 요하지 않는 명확한 명제로서 이 axiom을 바탕으로 다른 명제들이 도출된다. 이러한 추론에 사용되는 Axiom은 다음과 같이 first order logic으로 표현될 수 있다.

$$(\forall x) (Employee(x)) \Rightarrow (Person(x))$$

### 3.3 Knowledge engineering

Knowledge engineering은 도메인 지식을 정형화된 온톨로지 구성하는 과정이다. 본 논문에서는 Knowledge engineering을 온톨로지 디자인과 동의어로 사용한다. 지식 엔지니어는 온톨로지 디자인 시, 주요한 Object들과 Relation을 표현하기 위하여 도메인 지식을 충분히 이해하고 있어야 한다. 또한, knowledge를 표현할 수 있는 언어와 이에 따른 inference procedure를 정확히 이해하고 있어야 한다. 온톨로지를 구축하기 위한 하나의 방법론으로서 다음의 다섯 단계를 제시한다. 본 논문에서는 각 단계에서 BOM정보를 Product 온톨로지 디자인한다.

#### 3.3.1 도메인 지식 이해

이 단계는 지식을 표현하기 위하여 어떤 object들과 fact를 나누어야 하는가를 결정하는 과정이며 Knowledge engineering 작업의 성공여부를 결정짓는 중요한 단계이다. 본 연구에서 예제를 위하여 선택한 도메인 지식은 BOM에 관한 것이다. Fig. 3, 4의 BOM을 보면, end product, component part, material,

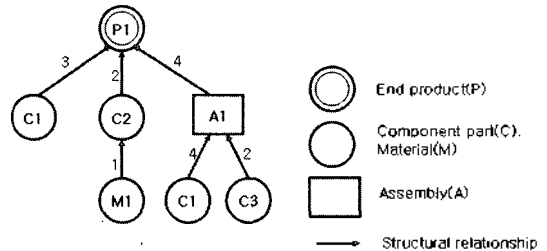


Fig. 3. Domain knowledge: Part type, structure, quantity.

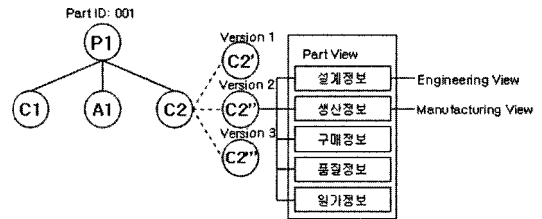


Fig. 4. Domain knowledge: Part version & Part View.

assembly 등이 part structure와 수량관계, version, view로 표현되어 있다.

온톨로지를 디자인하기 위하여 이해된 지식은 다음과 같다.

- End product, component, material, assembly는 part이다.
- End product는 component, assembly등의 결합체로서 최종적인 완성 part; component part는 raw material로부터 가공된 part; material은 component로 가공하기 위한 원자재; assembly는 component가 두 개 이상 결합된 part를 말한다.
- Part간에는 parent, child의 구조적 관계가 있다.
- 구조적 관계에는 수량정보가 포함된다.
- Component는 두 개 이상이 조립되어 assembly가 되거나, end product에 바로 포함될 수 있다.
- Part는 여러 version과 engineering view, manufacturing view의 view를 가진다.
- BOM 정보에서 하나의 part는 part id, version, view가 모두 고려되어 식별된다.

이 과정에서, Product 온톨로지 디자인을 위하여 'part', 'end product', 'component part', 'material', 'assembly', 'version', 'view' 등의 object를 결정하고, 지녀야 할 property를 파악하였다. 또한, part structure를 표현하기 위하여 'subPartOf'등의 relation을 고려하였다.

이처럼 Object, property, relation을 어휘로서 명시

화하는 과정은 도메인 지식에 관한 깊은 이해를 요구한다.

3.3.2 Predicate, function, constant 어휘 결정

이 단계는 위에서 선택된 어휘들을 온톨로지에서도 사용할 logic레벨의 어휘들로 변환하는 단계이다. 예를 들어 Product 온톨로지를 디자인함에 있어, 'part'라는 어휘가 constant인가, function인가, 혹은 relation인가를 결정하여야 한다. Knowledge engineering의 첫 단계와 이 단계를 합하여 ontological engineering이라 한다<sup>[6]</sup>.

Product 온톨로지에서도 사용될 vocabulary는 다음과

같다. 여기서 primitive lexicon은 온톨로지 디자인 시 정의되는 가장 기초적인 term들을 말하며, 의미를 부여하기 위하여 관련 axiom이 함께 선언되어야 한다. Defined lexicon은 primitive lexicon을 이용하여 정의된 어휘들을 의미한다.

<Primitive lexicon>

- relation
- (part ?p)
- (subPartOf ?p1 ?p2)
- (subPartQuantity ?p1 ?p2 integer)
- (hasVersionNo ?p float)

<Primitive lexicon>

① part  
Representation in KIF: (part ?p)

Relation	Arguments	Informal description
part	part	Part는 end product, assembly, component, raw material을 총칭하는 term이다. Product 온톨로지 전체에서 거의 모든 relation의 argument가 되는 가장 기초적인 term이다.

② subPartOf  
Representation in KIF: (subPartOf ?p1 ?p2)

Relation	Arguments	Informal description
subPartOf	part, part	subPartOf는 part의 source, child를 생성을 지칭한다.

Axiom 1  
(forall (?p1 ?p2 ?p3)  
=> (and (part ?p1)  
(part (subPartOf ?p1 ?p2))  
(part (subPartOf ?p2 ?p3)))  
=> (part (subPartOf ?p1 ?p3)))  
: "어떤 part는 그 child의 subPart가 되지 못한다."

Axiom 2  
(forall (?p1 ?p2 ?p3)  
=> (and (part ?p1)  
(part ?p2)  
(part (subPartOf ?p1 ?p2))  
(not (subPartOf ?p2 ?p1)))  
=> (part (subPartOf ?p1 ?p3)))  
: "Part의 구조관계는 순환적이지 않다."

Axiom 3  
(forall (?p1 ?p2 ?p3)  
=> (and (part ?p1)  
(part ?p2)  
(part (subPartOf ?p1 ?p2))  
(subPartOf ?p2 ?p3))  
=> (subPartOf ?p1 ?p3)))  
: "subPartOf는 transitive한 특성 값을 갖는다."

Axiom 4  
(forall (?p1 ?p2 ?p3)  
=> (and (part ?p1)  
(part ?p2)  
(part ?p3)  
(subPartOf ?p1 ?p2)  
(subPartOf ?p1 ?p3)  
(or (part (subPartOf ?p2 ?p3))  
(subPartOf ?p3 ?p2))))  
: "subPartOf는 transitive한 특성 값을 갖는다."

③ subPartQuantity  
Representation in KIF: (subPartQuantity ?p1 ?p2 integer)

Function	Arguments	Informal description
subPart-Quantity	part, part, integer	subPartQuantity는 child part가 parent part를 구성하는 part수의 하나 일때, 그 child part의 개수를 말한다.

BOM에서 수량정보를 지정하는 term이므로 axiom은 기술하지 않는다.

④ hasVersionNo  
Representation in KIF: (hasVersionNo ?p float)

Relation	Arguments	Informal description
has-versionNo	part, float	Part는 설계변경 등의 이유로 같은 part 모델이라도 여러 version이 있을 수 있으며 version 번호로 구별한다.

BOM 정보에서 part의 version을 지정하는 term이므로 axiom은 기술하지 않는다.

⑤ view  
Representation in KIF: (view ?v)

Relation	Arguments	Informal description
view	constant	관점에 따라 여러 형태의 BOM이 존재 가능하다. Engineering BOM은 설계관계에서 구성된 BOM이며, manufacturing BOM은 제조 관점에서 구성된 BOM이다. 이 밖에도 여러 종류의 view가 존재하나 위의 두 가지로 한정한다. Product 온톨로지에서는 engineeringView와 manufacturingView를 constant로 지정하였다.

Axiom 5  
(forall (?v)  
=> (view ?v)  
(and (not (part ?v))  
(=> (?v ?v engineeringView)  
(=> (?v ?v manufacturingView)  
(=> (?v ?v engineeringView)  
(=> (?v ?v manufacturingView))))))  
: "View는 engineeringView와 manufacturingView이다."

⑥ hasView  
Representation in KIF: (hasView ?p ?v)

Relation	Arguments	Informal description
hasView	part, view	BOM은 part가 가진 view에 따라 그 구조가 다르게 구성된다.

BOM 정보에서 part의 view 정보를 지정하는 term이므로 axiom은 기술하지 않는다.

⑦ characterizedPartOf  
Representation in KIF: (characterizedPartOf ?p1 ?p2 float ?v)

Relation	Arguments	Informal description
characterizedPartOf	part, part(mods), float, view	Part는 part id, version, view정보로서 서로 구별된다. 즉, 서로 같은 모델의 part도 version이나 어떤 view에서 이용되는가에 따라 서로 구별될 필요가 있는 것이다. characterizedPartOf는 BOM에서 part를 구별하기 위한 relation이다.

Axiom 6  
(forall (?p1 ?p2)  
=> (and (part ?p1)  
(hasVersionNo ?p1 float)  
(hasView ?p1 ?v1)  
(characterizedPartOf ?p1 ?p2 float ?v1)  
(part ?p2)))  
: "Part id와 version, view를 통하여 구별된 것은 part이다."

Axiom 7  
(forall (?p1 ?p2 ?p3)  
=> (and (part ?p1)  
(characterizedPartOf ?p1 ?p2 float ?v1)  
(characterizedPartOf ?p2 ?p3 float ?v2)  
(hasVersionNo ?p1 float)  
(hasVersionNo ?p2 float)  
(hasView ?p1 ?v1)  
(hasView ?p2 ?v2))  
=> (part (subPartOf ?p1 ?p3)))  
: "Part는 partID(?p)에 내장되는 값이 part ID라고 가정하고 view, version으로 구별된다."

<Defined lexicon>

① rawMaterial  
Representation in KIF: (rawMaterial ?p)

Relation	Arguments	Informal description
raw-Material	part	RawMaterial은 원재료는 in-house part를 제외하는 데 이용되는 관계에 관련된 RawMaterial의 집합이다 component이다. Part structure tree에서 가장 하위의 leaf이며 즉, sub part를 가지지 않는 part를 말한다.

Definition 1.  
(forall (?p)  
(definition rawMaterial(?p) :=  
(not (exists (?p1 ?p2)  
(and (part ?p1)  
(subPartOf ?p1 ?p2))))))

② endProduct  
Representation in KIF: (endProduct ?p)

Relation	Arguments	Informal description
endProduct	part	EndProduct는 더 이상 가공된 절차가 없는 최종 완성을 말한다. Part structure tree에서 가장 상위의 root이며 즉, parent part를 가지지 않는 part를 말한다.

Definition 2.  
(forall (?p)  
(definition endProduct(?p) :=  
(not (exists (?p1 ?p2)  
(and (part ?p1)  
(subPartOf ?p1 ?p2))))))

③ assembly  
Representation in KIF: (assembly ?p)

Relation	Arguments	Informal description
assembly	part	Assembly는 두 개 이상의 component로 구성된 part를 단하며 또 다른 assembly나 end product를 구성하는데 이용될 수 있다.

Definition 3.  
(forall (?p)  
(definition assembly(?p) :=  
(exists (?p1 ?p2)  
(and (part ?p1)  
(part ?p2)  
(subPartOf ?p1 ?p2)  
(subPartOf ?p2 ?p1)  
(= ?p1 ?p2))  
(not (endProduct ?p))))))

④ componentPart  
Representation in KIF: (componentPart ?p)

Relation	Arguments	Informal description
componentPart	part	ComponentPart는 RawMaterial보다 가공된 part를 말한다. Part structure tree에서 가장 하위의 leaf가 아닌 part는 모두 componentPart; 하위 또는 그 이상의 part의 집합체이다.

Definition 4.  
(forall (?p)  
(definition componentPart(?p) :=  
(and (part ?p)  
(not (rawMaterial ?p))  
(not (assembly ?p))  
(not (endProduct ?p))))))

⑤ hasPart  
Representation in KIF: (hasPart ?p1 ?p2)

Relation	Arguments	Informal description
hasPart	part, part	hasPart는 parent part가 child part가 sub part로 가지고 있을 수 있거나, hasPart는 subPartOf와 유사하다.

Definition 5.  
(forall (?p1 ?p2)  
(definition hasPart(?p1 ?p2) :=  
(subPartOf ?p2 ?p1)))

Fig. 5. Product Ontology: Primitive lexicons & its axioms, Defined Lexicons & its definitions.

(view ?v)  
 (hasView ?p ?v)  
 (characterizedPartOf ?p1 ?p float ?v)  
 • constant  
 engineeringView, manufacturingView  
 <Defined lexicon>  
 • relation  
 (rawMaterial ?p)  
 (componentPart ?p)  
 (assembly ?p)  
 (endProduct ?p)  
 (hasPart ?p1 ?p2)

### 3.3.3 결정된 어휘에 대한 axiom 부여

이제 위 단계에서 결정된 온톨로지의 어휘들에 대해 axiom을 기술해야 한다. 이는 두 가지 의미를 지닌다. 첫째, axiom에 의해 term들은 보다 정확한 의미를 지니게 되고, 이로 인해 사람은 그 의미해석을 일관되게 할 수 있다. 예를 들어 '베'라는 단어는 그 의미를 한정짓는 axiom에 의해 그것이 사람의 배인가 아니면, 교통수단으로서의 배인가를 판별할 수 있다. 둘째, axiom을 이용하여 추론을 할 수 있다.

Product 온톨로지에서 기술하는 primitive lexicon의 axiom과 defined lexicon의 definition들은 Fig. 5와 같다. 각 axiom과 definition은 KIF로 기술되었으며 아울러 어휘의 기술표기(예. part의 기술표기: (part ?p)와 argument(예. (part ?p)에서 argument는 ?p), informal description을 함께 정리하였다.

### 3.3.4 Knowledge base에 특정 지식 입력

위 단계까지 결정된 어휘들을 사용하여 KB (knowledge base)에 atomic sentence를 입력한다. Atomic sentence는 선언된 자체로 참값을 지닌다. Product 온톨로지에 관련하여 다음과 같은 atomic sentence들을 선언한다.

(part P1)  
 (part P2)  
 (part P3)  
 (subPartOf P2 P1)  
 (subPartOf P3 P2)

위는 세 개 part사이의 간단한 구조관계를 나타내고 있다. 그 의미는 'P1, P2, P3는 part이며, P2는 P1의 subpart이고, P3는 P2의 subpart이다'가 된다.

### 3.3.5 질의를 통한 지식 추론

우리가 알고자 하는 지식을 유도해내기 위하여 axiom과 위에서 입력된 사실들에 기반 하여 추론을 행하는 과정을 말한다. 예를 들어, P1이 P2라는 부품을 갖고 있는지, 그리고 P3가 P1이란 부품을 갖고 있는지를 알아내기 위해, 다음과 같이 product KB에 질의를 행하면 각각 yes, no라는 대답을 얻을 수 있다.

ASK(KB, hasPart(P1, P2)) : yes

ASK(KB, hasPart(P3, P1)) : no

## 4. Semantic mapping

### 4.1 Semantic mapping 개요

본 장에서는 온톨로지에 기반한 semantic mapping을 서술한다. 본 연구에서의 semantic mapping은 NIST에서 개발한 PSL 온톨로지의 방법을 기반으로 하고 있으며, Product 온톨로지를 이용하여 그 mapping과정을 보다 명확히 기술한다.

온톨로지 기반 semantic mapping의 기본 개념은 다음과 같다.

- ① 온톨로지 디자이너는 자사의 BOM 정보 표현 온톨로지를 표준 온톨로지의 primitive lexicon과 axiom을 사용하여 디자인한다. 경우에 따라서, primitive lexicon뿐만 아니라 defined lexicon까지 이용하여 디자인 가능하다. 또한, primitive lexicon을 활용하여 자사에 적합한 새로운 defined lexicon을 정의할 수도 있다.
- ② 표준 온톨로지를 바탕으로 정의된 임의의 두 회사의 BOM 온톨로지는 정형화된 의미구조(Product 온톨로지의 axiom과 definition을 의미)에 따라 의미매핑이 가능하다. 의미매핑은 각 회사에서 정의한 defined lexicon 레벨의 어휘쌍에서 이루어진다.

본 연구에서는 표준 온톨로지를 Product 온톨로지라고 가정한다. Product 정보가 semantic mapping되는 과정은 크게 semantic translation과 syntactic translation의 두 가지 절차로 나뉜다.

Fig. 6은 본 논문에서 선택한 semantic mapping 예제 어휘를 보여주고 있다. 즉, 예제에서 item 어휘는 표준온톨로지에서의 그 의미를 정의한 어휘인 componentPart를 거쳐 uniPart 어휘로 의미매핑된다.

Fig. 7이 의미하는 바는 애플리케이션 A의 고유 어휘가 Product 온톨로지에서의 정의한 표준 어휘로 변환된 후 다시 B의 어휘로 mapping되는 것이다. Fig. 7



(part Engine)  
 (part IgnitionCoil)  
 (part Copper)  
 (subPartOf IgnitionCoil Engine)  
 (subPartOf Copper IgnitionCoil)  
 (item IgnitionCoil)  
 (forall (?x) (= > (IgnitionCoil ?x) (item ?x)))

위는 A 애플리케이션의 정보가 KIF변환기에 의하여 syntactic translation을 거쳐 KIF로 표현된 형태이다. 이 정보는 Product 온톨로지로 전달되고, 동일한 의미구조를 통한 추론을 통해 item 어휘는 componentPart 어휘와 동일하다는 결론을 얻는다. 따라서 전달된 정보는 다음과 같이 변환된다.

(part Engine)  
 (part IgnitionCoil)  
 (part Copper)  
 (subPartOf IgnitionCoil Engine)  
 (subPartOf Copper IgnitionCoil)  
 (componentPart IgnitionCoil)  
 (forall (?x) (= > (IgnitionCoil ?x) (componentPart ?x)))

위 정보는 다시 B 온톨로지로 전달되고 추론을 통하여 componentPart는 uniPart와 같은 어휘라는 결론을 얻는다. 따라서 정보는 아래와 같이 바뀐다.

(part Engine)  
 (part IgnitionCoil)  
 (part Copper)  
 (subPartOf IgnitionCoil Engine)  
 (subPartOf Copper IgnitionCoil)  
 (uniPart IgnitionCoil)  
 (forall (?x) (= > (IgnitionCoil ?x) (uniPart ?x)))

위 정보를 B애플리케이션의 고유 문법으로 변환시켜주는 syntactic translation을 통해 B애플리케이션은 전달된 정보를 이용할 수 있게 된다.

Fig. 9는 semantic mapping 세부 절차를 나타낸 것이다. 이 절차는 B애플리케이션이 KIF형태의 A애플리케이션 BOM정보 파일을 받아 semantic mapping 처리를 통해 어휘변환을 하여 B애플리케이션 term의 KIF 파일을 생성하는 것을 나타내고 있다.

Fig. 9에서 Information file이란 BOM정보가 저장된 KIF-format-file을 말한다. Information file scanning은 information file을 한 줄씩 프로세서가 읽어들이는 것을 말한다. 한 줄씩 스캐닝하면서 file의

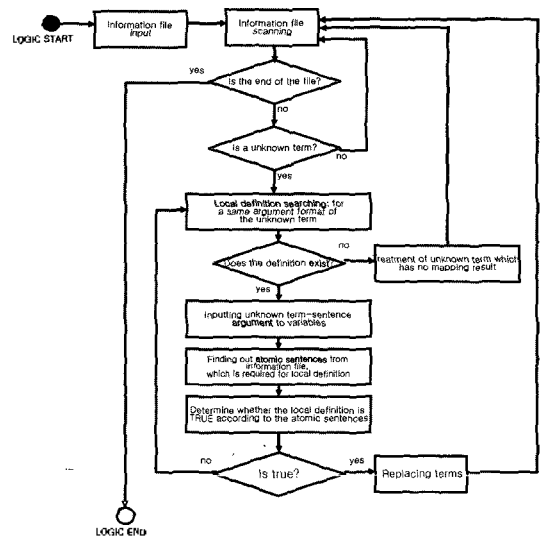


Fig. 9. Term replacing procedure.

끝이라면 실행 절차가 종료하게 된다. 끝이 아닐 경우, 현재 라인의 정보(BOM정보를 표현한 atomic sentence)가 unknown term을 포함하고 있으면, semantic mapping 처리를 하게 된다. 처리과정은, 읽어 들인 현재 라인의 atomic sentence의 argument수 (예. (part ?p)에서 argument수는 1)와 같은 local definition을 검색한다.

읽어 들인 atomic sentence의 argument를 검색된 local definition에 대입하면, local definition의 정의를 만족하는지를 판단하는 과정에서 필요한 추가적인 atomic sentence들을 information file로부터 찾는다. Argument와 검색된 추가적인 atomic sentence들을 바탕으로 local definition의 만족 여부를 판단하여 참일 경우 term의 대치처리를 통하여 semantic mapping이 성립된다.

## 5. Semantic mapping을 이용한 정보 공유 예제

본 장은 semantic mapping을 이용한 정보공유를 가상 시나리오를 통하여 서술한다.

### 5.1 BOM 정보 공유 예제 가정

BOM정보의 직관적 이해를 위하여 대상 도메인을 '가구제작'으로 정하였다. 예제의 가정은 다음과 같다.

- ① 대상 도메인 : 가구 제조 도메인, 생산계획
- ② 대상 기업 : A의자제작회사, B의자제작회사





A애플리케이션에서 표준 온톨로지 쪽으로 전달되는 정보는 위에서 Product 온톨로지의 기본 어휘들을 써서 정의한 BOM정보 전부이다. A애플리케이션의 rawMaterial은 어휘의 표현이나 의미가 표준 온톨로지와 완전히 동일하기 때문에 rawMaterial의 형태 그대로 B애플리케이션으로 전달된다. 즉, 위의 예제에서 정의한 BOM정보가 그대로 B애플리케이션으로 전달된다. 이러한 정보를 받아 B애플리케이션에서는 semantic mapping implementation 절차대로 information 정보의 한 줄씩 semantic mapping처리를 한다. 예를 들어 (part 의자)와 같은 atomic sentence의 경우, B온톨로지도 동일한 part 어휘를 사용하고 있기 때문에, 별다른 semantic mapping 처리 없이 그대로 B-term-KIF-format-file에 포함된다. 본 시나리오에서 semantic mapping 에시를 위하여 사용한 rawMaterial의 경우, 대응하는 어휘를 추론하기 위한 다음의 처리가 진행된다. 먼저,

(rawMaterial 오동나무)

를 읽어들이고, rawMaterial이 B온톨로지서 정의되지 않은 unknown term이라는 결과에 따라 인자가 1개인 B온톨로지의 definition들을 기준으로 semantic mapping작업을 수행한다. 이러한 definition중에서 rawMaterial에 대응하는 primitivePart의 의미가 B온톨로지에 다음과 같이 정의되어 있었다.

(forall (?p)  
 (defrelation primitivePart(?p) :=  
 (not ((exists (?p1)  
 (and ((part ?p)  
 (part ?p1)  
 (subPartOf ?p1 ?p))))))

(rawMaterial 오동나무) 문장으로부터 오동나무라는 인자가 변수에 다음과 같이 대입된다.

?p = 오동나무

또한, A애플리케이션으로부터 전달된 information file에 포함된 atomic sentence들로부터 primitivePart definition의 참/거짓 판단에 필요한 atomic sentence들을 검색하게 된다. BOM정보에서 다음과 같은 atomic sentence들이 있었다.

(part 의자)  
 (part 골격)  
 (part 바닥판)

(part 못)  
 (part 다리꾸)  
 (part 등받이부)  
 (part 연결지지대)  
 (part 다리)  
 (part 지지대)  
 (part 등판)  
 (part 등판지지대)  
 (part 오동나무)  
 (subPartOf 골격 의자)  
 (subPartOf 바닥판 의자)  
 (subPartOf 못 의자)  
 (subPartOf 등받이부 골격)  
 (subPartOf 다리꾸 골격)  
 (subPartOf 연결지지대 골격)  
 (subPartOf 못b 골격)  
 (subPartOf 다리 다리꾸)  
 (subPartOf 지지대 다리꾸)  
 (subPartOf 못c 다리꾸)  
 (subPartOf 등판 등받이부)  
 (subPartOf 등판지지대 등받이부)  
 (subPartOf 못d 등받이부)  
 (subPartOf 오동나무 등판)

위 atomic sentence들을 기준으로 ?p = 오동나무로 primitivePart의 definition과 대응시켜 생각해보면,

and ((part ?p)  
 (part ?p1)  
 (subPartOf ?p1 ?p))

의 경우 전체적으로 거짓이 된다. 왜냐하면 A애플리케이션으로부터 전달된 BOM정보 중에 오동나무의 경우 하위 파트가 없기 때문이다. and의 경우 모두 참을 만족하여야 전체적으로 참이 되는데, 여기서는 (subPartOf ?p1 ?p)을 참으로 만족하는 문장이 없으므로 다음의 문장은 거짓이 된다.

(exists (?p1)  
 (and ((part ?p)  
 (part ?p1)  
 (subPartOf ?p1 ?p))))

primitivePart의 정의에서 위 문장에 not을 붙여 참과 거짓을 바꾸고 있으므로 not이 붙여진 상태에서 위 문장은 참이 되며, 결과적으로 primitivePart의 정의를 참으로 만족하고 있는 것이 된다. 따라서 rawMaterial

= primitivePart와 같다는 semantic mapping이 도출된 것이며, term의 대치처리를 통하여 다음과 같은 atomic sentence가 B-term-KIF-format-file에 포함된다.

(primitivePart 오동나무)

## 6. 결 론

본 논문은 CPC환경에서 같은 의미이나 다르게 표기된 어휘들의 semantic mapping을 위하여 온톨로지를 이용한 접근 방안을 소개하고 온톨로지를 디자인하기 위한 요소로서 first order logic, KIF, knowledge engineering 과정, 추론 과정 등을 기술하였다. 이러한 것들은 BOM정보를 표현할 수 있는 'Product 온톨로지'의 디자인을 통해 그 흐름이 설명되었다. 또한, product 온톨로지를 이용한 semantic mapping에 기반한 정보공유의 예제가 제시되었다.

모든 PPO(Product/Process/Organization)정보가 온톨로지에 기반하여 semantic mapping과 정보 공유가 이루어지려면 각 PPO정보에 대한 온톨로지가 개발되어야 한다. 온톨로지 기반 semantic mapping접근은 방대한 양의 knowledge engineering 작업을 필요로 한다. 기업에서 사용되는 어휘의 의미가 정확히 정리되어야 하며, 또 이를 온톨로지로 표현하는 작업은 많은 시간과 노력을 필요로 한다. 그러나, 한번 구축이 된다면 기업 어휘의 거의 완벽한 의미적 매핑을 보장한다.

한편, Product 온톨로지의 실용성을 위해 part function decomposition에 관한 어휘나 STEP의 PDM 스키마에 구성되어 있는 BOM에 관한 어휘를 온톨로지로 정의하여 포함시키는 것도 의미있는 시도가 될 것이다.

## 참고문헌

1. Aberdeen Group, "Collaborative product commerce: delivering product innovations at internet speed," *Market View Point*, Vol. 12, No. 9, 1999.
2. Borst, W. N., "Construction of engineering ontologies for knowledge sharing and reuse," PhD thesis, University of Twente, Enschede, 1997.
3. Fox, M. S., "TOVE ontology project," <http://www.cil.utoronto.ca/enterprise-modelling/tove/index.html>, Enterprise Integration Laboratory, University of Toronto, 1992.
4. Michael Gruninger, "The Process Specification Language (PSL): Overview and Version 1.0 Specification," NISTIR 6459, National Institute of Standards and Technology, 2000.
5. Lukasz Kurgan, Waldemar Swiercz and Krzysztof J. Cios, "Semantic mapping of XML tags using inductive machine learning," Department of Computer Science and Engineering, University of Colorado at Denver, 2002.
6. Ken McKay, "Using PSL to exchange information between ProCAP and ILOG," NIST, <http://www.mel.nist.gov/psl/p4/demo/demo.html>, 1998.
7. Stuart Russell and Peter Norvig, "Artificial intelligence," Prentice Hall, pp. 185-216, 1995.
8. Michael R. Genesereth, "Knowledge interchange format Version 3.0 reference manual," Stanford University, Abstract, 1992.
9. Gruber, T. R., "What is an ontology?," <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>, Stanford University, 1993.
10. Alexander Maedche, "Ontology learning for the semantic web," Kluwer Academic Publishers, pp. 18-19, 2002.



김 경 영

2002년 한양대학교 산업공학과 학사  
2004년 KAIST 산업공학과 석사  
2004년 현 LG정유 근무



서 효 원

1981년 연세대학교 기계공학과 학사  
1983년 한국과학기술원 기계공학과 석사  
1991년 West Virginia University 산업공학과 박사  
1983년~1987년, 내우중공업(주) 중앙연구소 주임연구원  
1992년~1995년, 생산기술연구원 생산시스템센터 수석연구원  
1996년~현재, 한국과학기술원 산업공학과 교수  
관심분야: Product Development Process Modeling, Product Data Modeling, Concurrent Engineering Methodology, Computer-based Concurrent Engineering System