

이 희 정[†]

대구대학교 산업경영공학과

Product Family Design using Formal Concept Analysis and Ontology

Heejung Lee[†]

Department of Industrial and Management Engineering, Daegu University

A product family design has received much attention over the last several decades, since a product family-based development shortens lead-times and reduces cost, as well as increases efficiency and effectiveness of the product realization process. It is challenging work, however, to define the product family design in the heterogeneous product development environments, due to myriads of products related information described in different ways across products in any companies.

In this paper, we provided a way of defining product family design framework using formal concept analysis and ontology language. Based on this, the specific product family can be derived by ontological reasoning, and the new product concept can be also expanded in the framework. The proposed framework is formalized using *OWL* (Web Ontology Language) and implemented in *Protégé*. Actual product family design algorithm is carried out using *FaCT++* engine, a plug-in to *Protégé*, and the benefits of the proposed method are also demonstrated through a case study.

Keywords : Product Family Design, Formal Concept Analysis, Ontology

1. 서 론

표준화된 제품을 대량으로 제작하는 생산방식은 근대산업의 대표적인 특징이지만, 최근 정보기술과 생산기술이 비약적으로 발전한 디지털 환경에서는 고객의 다양한 요구를 충족시키면서도 높은 품질의 제품을 낮은 원가로 빠르게 대량생산하는 이른바 대량맞춤화(Mass Customization) 생산이 기업 경쟁력의 중요한 역량으로 부상하였다.

제품의 대량맞춤화를 효율적이고 효과적으로 수행하기 위해서는 고객의 다양한 요구를 제품에 신속하게 반영해야 한다. 이를 위해서는 고객이 요구하는 제품을 처음부

터 완전히 새로운 제품 개념으로 정의하는 것 보다는 이미 존재하는 제품들을 검토하고 새로운 제품 개념으로 확장하는 방식이 일반적으로 사용될 수 있다. 처음부터 완전히 새로운 제품을 개발하는 일은 대단히 복잡하고 많은 노력이 소모되는 작업이므로, 대부분의 기업은 이미 출시된 제품의 정보와 지식을 활용해서 시장의 다양한 요구를 반영한 제품을 신속하게 출시할 수가 있다.

따라서 이미 존재하는 제품들의 사양정보를 파악하고, 유사한 제품을 분류하여 정보를 서로 공유하는 작업이 필요하다. 즉, 제품사양, 설계지식, 설계프로세스, 제약조건 등을 포함한 제품정보들은 매우 다양한 형태로 생성되며

로, 서로 다른 제품 정보들을 모델링하고 공유할 수 있는 연구와 정보모델로부터 유사한 사양정보를 획득하고 제품의 공통적 특성을 파악하여 제품계열(Product Family)을 정의하는 연구가 매우 중요하다.

제품 정보를 모델링 하는 방법에 대해서 살펴보면 다음과 같다. 국제표준화기구(ISO)에서는 제조부문의 표준 정보모델을 구축하기 위하여 STEP으로 널리 알려진 ISO 10303을 제정하였으며[6], 인터넷 환경에서의 정보 교환 기술을 목적으로 한 XML(eXtensible Markup Language)가 주목을 받았다[5]. 학계 일부에서는 STEP과 XML을 기반으로 기계가 쉽게 이해할 수 있는 이른바 EXPRESS 데이터 모델링 언어 및 EXPRESS-X 데이터 매핑 언어를 개발하였다[14]. 하지만 EXPRESS-X와 같은 언어는 의미적으로 동일한 제조환경을 기반으로 하기 때문에 실질적인 제조환경에서 의미를 공유하기에는 한계가 존재한다[3]. 실질적인 제조 및 제품개발 환경에서는 정보의 의미, 해석과 활용목적이 서로 상이할 수 있는데, 이러한 정보 모델이 단순히 구조적으로만 공유된다면 정보의 일부가 왜곡되어 원하지 않은 결과를 얻을 가능성이 크다. 이러한 의미적인 불일치 가능성을 극복하기 위해서는, 정보 모델의 본질이 정확히 표현되고 기계를 통해 서로 공유되어야 하는 기술이 필요하다. 온톨로지 접근법은 실질적인 환경에서 의미적으로 일치하는 정보모델 체계를 제공해 줄 수 있는 해결책으로 많은 주목을 받고 있다. 온톨로지라는 용어는 본래 철학에서 사용되는 용어로서 ‘존재론’을 의미하며, 인공지능적 또는 컴퓨터 공학적으로 해석하면 ‘여러 사람들 간에 공유되고 합의되는 내용’으로 정의할 수 있다[10]. 또한 온톨로지는 특정영역의 지식을 모델링하고 처리하여 구성원 간의 지식 공유 및 재사용을 가능하게 하는 도구이다[1]. Cys[5]는 200,000개가 넘는 일반적 수준의 개념들을 기계가 이해할 수 있도록 정의한 상업적인 온톨로지이며, NIST(National Institute of Standards and Technology)는 Cyc를 기반으로 제품개발이라는 도메인에 활용될 수 있는 보다 구체적인 온톨로지 언어인 PSL(Process Specification Language)를 개발하였으며[15], PSL은 향후 제조부문 데이터의 범용적인 제품정보를 표현하는 데 널리 활용되었다. 한편 구체적인 지식을 표현하기 위한 온톨로지로서, 기업의 비즈니스에 초점을 둔 Enterprise Ontology[20]와 지식집약적인 특성이 강조되어 제품설계 등에 활용될 수 있는 TOVE(Toronto Virtual Enterprise)[7]가 등장하였다.

한편 제품계열을 정의하는 방법을 살펴보면, 일반적으로 제품계열은 하향식(top-down) 방법과 상향식(bottom-up) 방법으로 정의할 수 있다. 하향식 방법이란 서로 다른 제품들의 공통 특성인 이른바 플랫폼을 기반으로 전략적으로 제품들을 파생시켜 제품계열을 정의하는 방식이며,

상향식 방법이란 제품 표준화를 통한 규모의 경제를 달성하기 위하여 서로 다른 제품들을 재설계하고 통합해서 제품계열을 정의하는 방식이다[16]. 하향식 또는 상향식 관점과 별개로 제품계열들은 모듈기반 또는 스케일 기반으로 정의를 할 수 있다. 모듈기반의 제품계열 정의방법은 플랫폼으로부터 일부 특성들을 더하거나, 대체하거나 제거하는 방식이며[19], 스케일기반의 제품계열 정의방법은 시장의 요구사항에 대응하기 위하여 플랫폼을 수직적으로 늘이거나 줄이는 방식이다[8]. 한편, 실질적인 제품개발 환경을 지원하는 제품계열 정의방식으로는 온톨로지를 활용한 방법이 연구되고 있다[12]. 특히 Nanda [11]는 정형적 개념 분석(Formal Concept Analysis)과 시맨틱 기술을 활용하여 제품플랫폼 기반의 지식관리 방법을 제시하였다.

본 논문에서는 실질적인 제품개발 환경을 지원하기 위하여 시맨틱 웹상의 제품정보를 정형적 개념분석으로 모델링하고, 이후 제품계열을 정의하기 위해 온톨로지의 기술논리 추론을 활용하는 방법을 제시하고자 한다. 본문의 구성은 다음과 같다. 먼저 제 2장에서는 제품계열 정보의 기본구조 역할을 하는 정형적 개념 분석의 개념격자에 대해서 기술하였고, 제 3장에서는 실질적인 제품개발 환경에서의 제품정보 모델을 위한 기술논리 기반의 온톨로지에 대해서 설명하였다. 제 4장에서는 본 논문의 제안방법을 기술한 부분으로, 온톨로지를 활용하여 개념격자로부터 제품계열 정보를 도출하는 과정을 제시하였으며, 제 5장에서는 노트북 제품을 통해 제안된 방법의 효용성을 예제로서 검증하였다.

2. 정형적 개념 분석

정형적 개념 분석(Formal Concept Analysis, FCA)은 격자이론(lattice theory)과 순서이론(order theory)에 기반한 수학적 데이터 분석 방법이다[9]. FCA의 기본적인 개념을 살펴보면 다음과 같다.

(정의 1) 정형적 문맥(formal context), $K := (C, P, R)$, 단, 클래스 C는 객체(object)들의 유한 집합, 성질 P는 속성(property)들의 집합, 그리고 관계 R은 C와 P의 이항관계(binary relations)이다.

(정의 2) 관계(relation), $R \subseteq C \times P$. 만약 객체 c_1 이 속성 p_1 을 갖는다면(단, $c_1 \in C, p_1 \in P$), 관계는 $(c_1, p_1) \in R$ 로 표현된다.

(정의 3) 클래스 $C_1 \subseteq C$ 에 대해, 공통 성질(common

properties) $C_1' := \{p_1 \in P \mid \forall c_1 \in C_1 : (c_1, p_1) \in R\}$ 이며, 성질 $P_1 \subseteq P$ 에 대해, 공통 클래스(common classes) $P_1' := \{c_1 \in C \mid \forall p_1 \in P_1 : (c_1, p_1) \in R\}$ 이다.

(정의 4) 정형적 개념(formal concept), $B(K) := \{(C_1, P_1) \mid C_1 \subseteq C, P_1 \subseteq P, C_1' = P_1, P_1' = C_1\}$. 이때, C_1 을 정형적 개념 $B(K)$ 의 외연(extent)이라 하고, P_1 을 내연(intent)이라고 부른다.

(정의 5) 개념 격자(concept lattice), $B(K)$ 는 정형적 문맥 K 의 가능한 모든 개념들과 이들 간의 순서 관계이다.

(정의 6) (C_1, P_1) 과 (C_2, P_2) 가 하나의 정형적 문맥에 기초한 개념들이고, $C_1 \subseteq C_2$ 이거나 혹은 $P_2 \subseteq P_1$ 이면, (C_1, P_1) 은 (C_2, P_2) 의 하위 개념(sub-concept)이라 하고, $(C_1, P_1) \leq (C_2, P_2)$ 라고 표기한다.

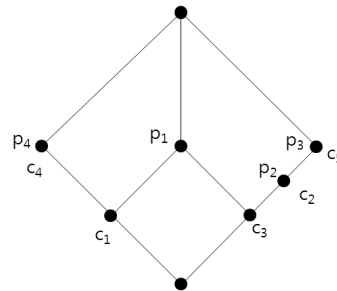
(정의 6) 클래스 C 와 성질 P 간의 관계를 나타내는 표를 십자표(cross table)라고 하며, 클래스는 십자표의 행에 나열하고, 성질은 십자표의 열에 나열한다. 또한 관계 R 은 <표 1>과 같이 십자표에 '×'로 표기한다.

<표 1> 십자표 예제

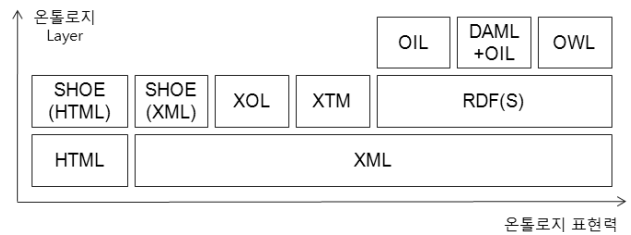
| | p_1 | p_2 | p_3 | p_4 |
|-------|-------|-------|-------|-------|
| c_1 | × | | | × |
| c_2 | | × | × | |
| c_3 | × | × | × | |
| c_4 | | | | × |
| c_5 | | | × | |

예를 들어, <표 1>로부터 객체 c_2 는 속성 p_2 와 p_3 을 갖고, 객체 c_3 는 객체 c_2 가 갖는 모든 속성을 동시에 갖고 있다. 이러한 객체들의 집합 $\{c_2, c_3\}$ 를 클래스 A라고 정의하고, 속성들의 집합 $\{p_2, p_3\}$ 를 성질 B라고 정의하자. 즉, 클래스 A는 성질 B의 모든 속성을 갖는 모든 객체들의 집합이며, 성질 B는 클래스 A의 모든 객체들에게 유효한 모든 속성들의 집합이다. 이때 (A, B) 쌍을 주어진 정형적 문맥 하에서의 정형적 개념이라고 부르며, A를 외연이라 하고 B를 내연이라 한다. 이와 같이 '정형적 개념'이라는 용어는 일반적으로 외연과 내연으로 정의될 수 있으며, 다양한 분야에 매우 유용하게 활용될 수 있다. 한편 $\{c_3\}$ 를 P라고 정의하고, $\{p_1, p_2, p_3\}$ 를 Q라고 정의하면, 외연 P는 외연 A에 포함되고, 내연 Q는 내연 B를 포함하게 되므로, 정형적 개념 (P, Q)는 정형적 개념 (A, B)의 하위 개념이 된다. 한편 위 십자표 예제는 하나의 정형적

문맥에 해당하며, 이를 도식적인 개념 격자로 표시하면 <그림 1>과 같이 나타낼 수 있다. 개념 격자는 주어진 정형적 문맥의 모든 객체와 속성, 그리고 선과 마디로 구성되며, 임의의 객체가 갖는 속성들은 해당되는 마디에서 상향 경로에 존재하는 모든 속성들을 찾아가면 된다. 예를 들어 객체 c_1 은 상향 경로에 존재하는 p_1 과 p_4 를 속성으로 갖는다.



<그림 1> 개념 격자 예제



<그림 2> 온톨로지 언어[1]

3. 서술논리 기반 온톨로지

<그림 2>에서 보는 바와 같이 최초의 온톨로지 언어인 SHOE(Simple HTML Ontology Extension)가 1996년에 발표된 이후, 2004년 W3C 권고안이 된 OWL(Web Ontology Language)이 현재까지 표현력이 가장 우수한 온톨로지 언어로 활용되고 있다[17]. OWL은 지원되는 표현력에 따라서 OWL Lite, OWL DL, OWL Full의 세 종류로 구분된다. 전자에서 후자로 갈수록 표현력은 향상되지만 추론기능에 대한 시스템 지원에 어려움이 있다. 따라서 표현력이 충분하면서도 추론기능이 원활하게 지원되는 OWL DL이 가장 널리 활용되고 있으며, 본 논문에서도 OWL DL을 사용하도록 한다.

OWL DL은 논리언어학적으로 기술논리(DL : Description Logic)를 기반으로 하는데, 기술논리란 Concept과 Role로 기본적인 개념을 표현하고, 개념들을 논리학의 Boolean 연산자와 각종 제약으로 연결하여 용어적 지식을 다루는 지식표현 체계이다[2]. <표 2>는 기술논리를 표현하는 문법

<표 2> 기술논리 문법

| 구성자 | Syntax | Semantic | 설명 | |
|------------|--------------|-------------------------------|---|--|
| Concept | C | {x C(x)} | x는 개념 C임 | |
| AND | $C \sqcap D$ | $\{x C(x) \wedge D(x)\}$ | x는 개념 C인 동시에 D임 | |
| OR | $C \sqcup D$ | $\{x C(x) \vee D(x)\}$ | x는 개념 C이거나 D임 | |
| NOT | $\neg C$ | $\{\sim x C(x)\}$ | x는 개념 C가 아님 | |
| 제한 | All | $\forall R.C$ | $\{x \forall y, R(x, y) \rightarrow C(y)\}$ | x와 R의 관계에 있으면서 C(y)를 동시에 만족하는 모든 y를 갖는 x를 의미 |
| | Some | $\exists R.C$ | $\{x \exists y, R(x, y) \rightarrow C(y)\}$ | x와 R의 관계에 있으면서 C(y)를 동시에 만족하는 어떤 y를 갖는 x를 의미 |
| | Number | $\geq n R$ | $\{x \{y R(x, y)\} \geq n\}$ | x와 R의 관계에 있는 y의 개수가 n 이상인 조건을 만족하는 x를 의미 |
| | | $= n R$ | $\{x \{y R(x, y)\} =n\}$ | x와 R의 관계에 있는 y의 개수가 n인 조건을 만족하는 x를 의미 |
| $\leq n R$ | | $\{x \{y R(x, y)\} \leq n\}$ | x와 R의 관계에 있는 y의 개수가 n 이하인 조건을 만족하는 x를 의미 | |

을 나타내며, Tbox(Terminological Box)와 ABox(Assertional Box) 구조를 활용하여 지식을 표현한다. Tbox에서는 개념에 대한 용어적(Terminological) 지식이 정의되고, ABox에서는 특정개념에 속하는 인스턴스에 대한 선언적(Assertional) 지식이 정의된다. 예를 들어 식 (1)은 ‘Part 라는 Concept은 모든 온톨로지의 최상위 개념인 Thing에 포함되는 동시에 Composite_Part 혹은 Primitive_Part에 포함된다’라는 개념을 의미하고, 식 (2)는 ‘Composite_Part라는 Concept은 Part인 동시에 하위 Part를 2개 이상 갖는다’라는 개념을 의미하며, 식 (3)은 ‘part1과 compositePart1은 각각 Part라는 Concept의 인스턴스이다’를 의미하고, 마지막으로 식 (4)는 ‘composite-Part1과 part1은 hasComponent 관계를 갖는다’를 의미한다. 본 예제에서 식 (1)과 식 (2)는 용어 정의에 해당하는 TBox 지식표현이며, 식 (3)과 식 (4)는 인스턴스에 대한 선언적 지식에 해당하는 ABox 지식표현이다.

- $Part \sqsubseteq Thing \sqcap (Composite_Part \sqcup Primitive_Part)$. (1)
- $Composite_Part \equiv Part \sqcap \geq 2 hasComponent$. (2)
- $part1 \in Part, compositePart1 \in Part$. (3)
- $(compositePart1, part1) \in hasComponent$. (4)

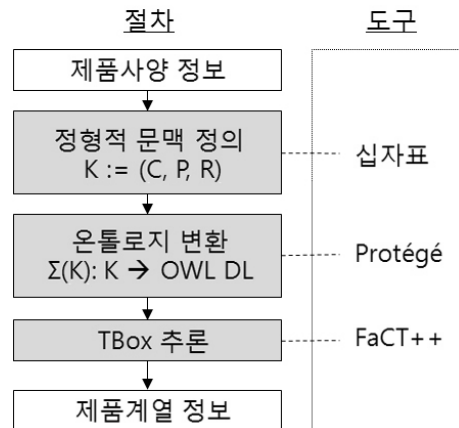
한편 서술논리 기반의 온톨로지는 TBox 추론 및 ABox 추론이 가능하다. TBox 추론은 Subsumption 관계를 추론하는 것으로, 상위 Concept과 하위Concept의 관계를 추론하게 된다. ABox 추론은 인스턴스 관계를 추론하는 것으로, 해당 인스턴스가 정의된 Concept에 포함되는지를 검사하는 것이다.

4. 제품계열 정보 정의

4.1 기본 절차 및 도구

본 논문에서 제시한 제품계열 정보 절차는 <그림 3>과

같다. 첫째, 제품사양의 정보를 입력받아 십자표를 활용하여 정형적 문맥을 정의한다. 둘째, 정의된 정형적 문맥을 온톨로지 개발도구 중 하나인 Protégé[13]를 이용하여 온톨로지 로 변환한다. 셋째, Protégé 추론엔진인 FaCT++을 활용하여 온톨로지 추론을 통해 제품계열 정보를 정의한다. 각 절차에 대해서는 다음과 같다.



<그림 3> 제품계열 정보 도출과정

4.1.1 정형적 문맥 정의

제품계열 정보의 대상이 되는 제품군을 선정하고, 제품별 사양을 취합한다. 제품 집합은 정형적 문맥의 클래스 C에 해당되고, 사양 집합은 정형적 문맥의 성질 P에 해당되며, 해당 제품과 사양간의 관계는 R에 대응된다. 취합된 C, P, R은 <표 3>과 같이 십자표를 활용하여 정형적 문맥 K를 정의한다.

<표 3> 정형적 문맥 정의

| 제품(C) | 사양(P) | | | |
|-------|-------|--|---|--|
| | | | × | |

4.1.2 온톨로지 변환

정형적 문맥 K 에 대하여 변환 식 $\Sigma(K)$ 를 통해 온톨로지 변환한다. 변환 식 $\Sigma(K)$ 는 다음과 같다.

$$C_k \in C, P_j \in P \text{에 대해, } \Sigma(K) := C_k \equiv (\bigcap_j (\exists \text{hasFeature}.P_j)), \text{ for } \forall_j : (C_k, P_j) \in R$$

클래스 C 와 성질 P 간의 관계 R 은 온톨로지 구성요소인 ‘hasFeature’를 통해 이항관계를 정의할 수 있다. 변환식 $\Sigma(K)$ 를 활용하여 제 2장에서 제시한 <표 1>의 정형적 문맥 예제를 온톨로지로 변환하면 아래와 같다.

- $C_1 \equiv (\exists \text{hasFeature}.P_1) \sqcap (\exists \text{hasFeature}.P_4)$.
- $C_2 \equiv (\exists \text{hasFeature}.P_2) \sqcap (\exists \text{hasFeature}.P_3)$.
- $C_3 \equiv (\exists \text{hasFeature}.P_1) \sqcap (\exists \text{hasFeature}.P_2) \sqcap (\exists \text{hasFeature}.P_3)$.
- $C_4 \equiv (\exists \text{hasFeature}.P_4)$.
- $C_5 \equiv (\exists \text{hasFeature}.P_3)$.

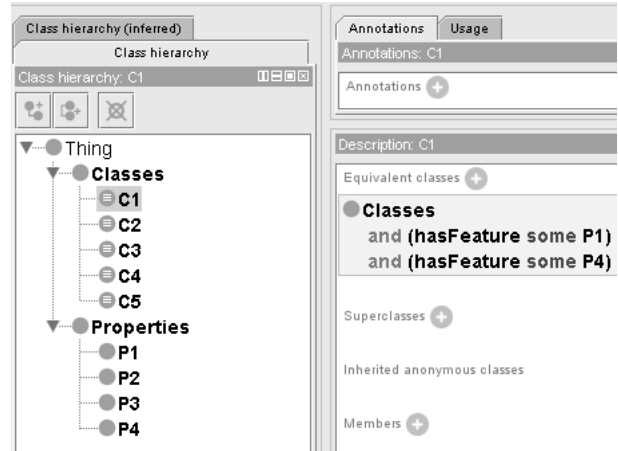
위 온톨로지의 변환 로직을 구현하기 위해서는 온톨로지 개발 도구의 선정이 필요하다. 본 논문에서는 온톨로지 개발 도구로 널리 사용되는 Protégé를 선택하였다. 더욱이 Protégé는 서술논리에 익숙하지 않아도 쉽게 이해할 수 있도록 <표 4>와 같이 기존의 서술논리를 쉽게 재해석한 Manchester OWL Syntax를 기반으로 하기 때문에 다루기가 용이하다. <그림 4>는 Protégé를 활용한 온톨로지 변환 예제를 나타낸다.

4.1.3 TBox 추론

서술논리 기반 온톨로지의 TBox 추론은 상위 개념과 하위 개념간의 관계를 추론하게 되며, 이는 이른바 정형적 문맥 하에서의 모든 개념들과 이들 간의 순서관계를 도출하는 과정과 일치한다. 정형적 개념분석에서는 이러한 순서관계를 도출하기 위하여 다양한 소프트웨어 도구들이

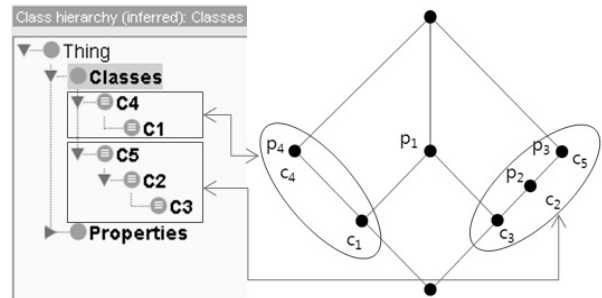
<표 4> Manchester OWL Syntax

| 서술논리(DL) | Manchester |
|--------------------------|------------|
| \exists (on concepts) | some |
| \forall | only |
| \exists (on instances) | value |
| \geq | min |
| $=$ | exactly |
| \leq | max |
| \sqcap | and |
| \sqcup | or |
| \neg | not |



<그림 4> Protégé를 활용한 온톨로지 변환 예제

개발되었으며[18], 본 논문에서는 Protégé의 TBox 추론으로도 동일한 결과를 얻을 수 있음을 보여주고 있다. 즉 Protégé에 플러그인(Plug-in) 되어 있는 FaCT++ 엔진을 활용하여 개념들 간의 순서관계를 예측할 수 있다. <그림 5>는 <그림 4>에서 정의한 온톨로지에 대하여 FaCT++을 활용한 TBox 추론의 결과를 나타낸다. 추론 결과 $C_1 \sqsubseteq C_4$ 및 $C_3 \sqsubseteq C_2 \sqsubseteq C_5$ 의 순서 관계를 얻었고, 이는 개념 격자에서의 관계와 정확히 일치한다.



<그림 5> TBox 추론 예제

4.2 계열 정보 확장

온톨로지는 이미 존재하는 개념을 기반으로 기본적인 기술논리의 연산에 의해 새로운 개념을 확장할 수 있다. 마찬가지로 이미 존재하는 정형적 개념을 활용하여, 새로운 개념을 확장하여 정의할 수 있다. 아래의 Case 1~Case 6은 기술논리의 기본적인 연산을 설명하고 있으며, 이들 연산자를 활용하여, 개념의 자유로운 확장이 가능하다. 또한 TBox 추론에 의하여 기존의 개념들과 새로 확장된 개념들의 순서관계는 다시 추론될 수 있다.

Case 1 : AND(Intersection) $C_1, C_2 \dots, C_k$ 의 특성을 모두 갖는 NewC를 정의하는 경우,

▪ $NewC \equiv C_1 \sqcap C_2 \sqcap \dots \sqcap C_k$

Case 2 : OR(Union) $C_1, C_2 \dots, C_k$ 중 적어도 하나 이상의 특성을 갖는 NewC를 정의하는 경우,

▪ $NewC \equiv C_1 \sqcup C_2 \sqcup \dots \sqcup C_k$

Case 3 : Not(Complement) C_k 의 특성을 안 갖는 NewC를 정의하는 경우,

▪ $NewC \equiv \neg C_k$

Case 4 : All(Universal restriction) 관계 R을 통해 오직 P_k 의 특성을 갖는 모든 인스턴트들의 집합 NewC를 정의하는 경우,

▪ $NewC \equiv \forall R.P_k$

Case 5 : Some(Existential restriction) 관계 R을 통해 적어도 하나 이상의 P_k 의 특성을 갖는 임의의 인스턴트들의 집합 NewC를 정의하는 경우,

▪ $NewC \equiv \exists R.P_k$

Case 6. Number(Cardinality restriction) 관계 R을 통해 n개 이상의, n개의, 또는 n개 이하의 P_k 의 특성을 갖는 임의의 인스턴트들의 집합 NewC를 정의하는 경우,

▪ $NewC \equiv \geq n R.P_k.$

▪ $NewC \equiv = n R.P_k.$

▪ $NewC \equiv \leq n R.P_k.$

5. 예 제

본 장에서는 실제 제품정보의 속성을 기반으로 제품 계열 정보를 정의하는 사례를 소개하도록 한다. 본 사례에서 적용된 제품은 국내 S기업에서 생산하는 노트북을 대상으로 진행하였으며, 필요한 속성 정보는 해당 S기업에서 운영하는 쇼핑몰에 등록된 제품사양 정보를 참조하였다.

(1) 제품계열 정보 도출

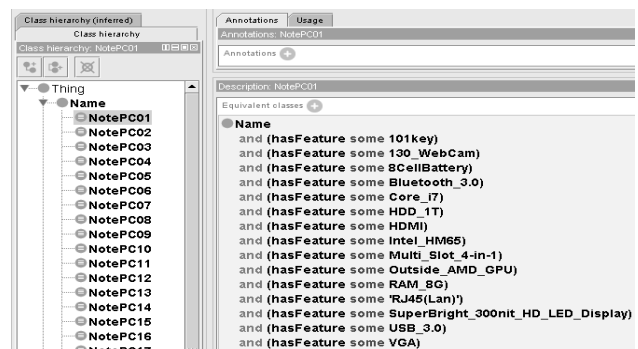
본 사례에서는 Intel 계열의 NotePC 30대를 선정하여 <표 5>와 같이 속성을 도출하였고, 해당 노트북에 속성들을 매핑하는 십자표를 활용하여 <그림 6>과 같이 정형적 문맥 K를 정의하고, NotePC의 정형적 문맥 K에 대하여 변환 식 $\Sigma(K)$ 를 통해 온톨로지 변환하였다. <그림 7>은 Protégé를 활용하여 NotePC01에 대해 온톨로지 정의한 예제이다.

<표 5> NotePC의 제품속성

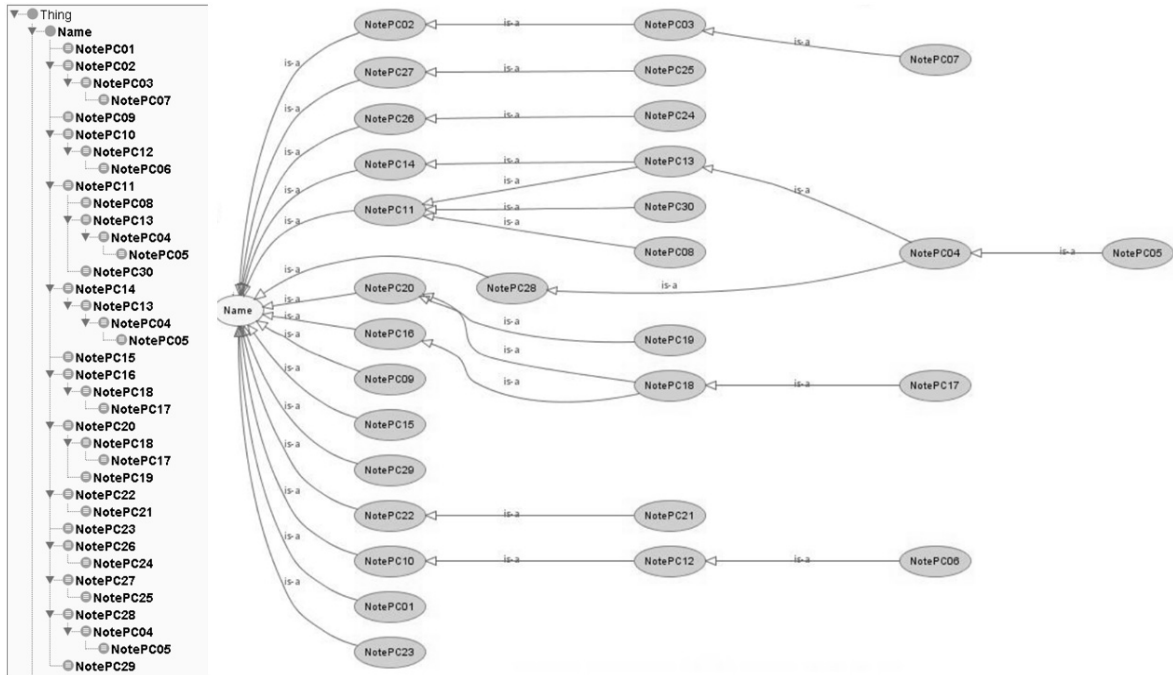
| 상위 속성 | 하위 속성 |
|----------|---|
| CPU | 코어i3, 코어i5, 코어i7 |
| Display | HD LED, Super Bright 300nit HD LED, SuperBright 400nit HD LED |
| GPU | AMD(ATI) 외장/내장, NVidia, Intel 외장/내장 |
| RAM | 2MB, 4MB, 6MB, 8MB+ |
| DISK | HDD 500G/750G/1T+, SSD 128G/256G+ |
| Chipset | AMD A60M, Intel HM65 |
| Camara | WebCam, WebCam 130/200 |
| I/O | 블루투스3.0, VGA, HDMI, USB3.0 |
| Card | Micro SD, 4-in-1, 7-in-1 |
| Internet | RJ45, Wimax |
| Keyboard | 80, 81, 86, 87, 101, 103 Keys |
| Battery | 4, 6, 8 Cells |

| Name | CPU | | | Display | | | GPU | | | | |
|----------|-------|-------|-------|---------|---------------------------|---------------------------|----------|---|--------|-------|---|
| | 코어 i3 | 코어 i5 | 코어 i7 | HD LED | SuperBright 300nit HD LED | SuperBright 400nit HD LED | AMD(ATI) | | Nvidia | Intel | |
| NotePC01 | | | o | | o | | | o | | | |
| NotePC02 | | o | | o | | | | | | | o |
| NotePC03 | | o | | o | | | | | | | o |
| NotePC04 | | o | | o | | | | | o | | |
| NotePC05 | | o | | o | | | | | o | | |
| NotePC06 | o | | | o | | | | | | | o |
| NotePC07 | | o | | o | | | | | | | o |
| NotePC08 | | o | | o | | | | | o | | |
| NotePC09 | o | | | o | | | | | o | | |
| NotePC10 | o | | | o | | | | | | | o |
| NotePC11 | | o | | o | | | | | o | | |
| NotePC12 | o | | | o | | | | | | | o |
| NotePC13 | | o | | o | | | | | o | | |
| NotePC14 | | o | | o | | | | | o | | |
| NotePC15 | o | | | o | | | | o | | | |
| NotePC16 | | o | | o | | | | o | | | |
| NotePC17 | o | | | o | | | | o | | | |
| NotePC18 | o | | | o | | | | o | | | |
| NotePC19 | o | | | o | | | | o | | | |
| NotePC20 | o | | | o | | | | o | | | |
| NotePC21 | o | | | o | | o | | | | | |
| NotePC22 | o | | | o | | o | | | | | |
| NotePC23 | | o | | o | | o | | | | | o |
| NotePC24 | o | | | o | | o | | | | | o |
| NotePC25 | | o | | o | | o | | | | | o |
| NotePC26 | o | | | o | | o | | | | | o |
| NotePC27 | | o | | o | | o | | | | | o |
| NotePC28 | o | | | o | | o | | | o | | |
| NotePC29 | | o | | o | | o | | | o | | |
| NotePC30 | o | | | o | | o | | | o | | |

<그림 6> NotePC의 정형적 문맥



<그림 7> NotePC의 온톨로지 변환 사례



<그림 8> NotePC의 제품계열 정보

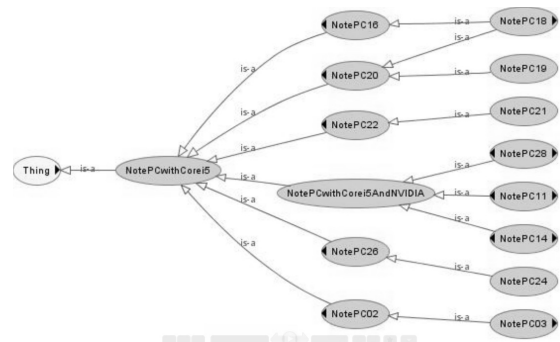
NotePC01부터 NotePC30까지 온톨로지를 모두 변환하고, Protégé에서 제공하는 FaCT++ 엔진을 활용하여 TBox 추론을 실행하면, <그림 8>과 같이 제품계열 기본정보를 얻을 수 있다.

(2) 계열 정보 확장

NotePC의 제품계열 기본정보를 바탕으로 제 4.2절에서 제시한 기본적인 기술논리(Case 1~Case 6)를 적용하면, 새로운 제품군을 정의하고 추가적으로 확장된 계열정보를 얻을 수 있다. 예를 들어 CPU가 코어i5인 제품은 'NotePCwithCorei5 ≡ hasFeature some core_i5'와 같이 정의할 수 있고, CPU가 코어i5인 동시에 GPU가 NVIDIA인 제품은 'NotePCwithCorei5AndNVIDIA ≡ hasFeature some core_i5 AND hasFeature some NVIDIA_GPU'와 같이 정의할 수 있다. 이후 TBox 추론을 실행하면, <그림 9>와 같이 기존에 정의된 제품계열 기본정보들(NotePC01~NotePC30)은 새로 정의된 가상제품 NotePCwithCorei5와 NotePCwithCorei5 AndNVIDIA를 기준으로 제품계열 정보들이 새로 확장됨을 확인할 수 있다.

6. 결론

본 논문에서는 디지털 환경에서 다양한 고객 요구에 신속하게 대응하기위한 방법 중 하나인 대량맞춤화 역량



<그림 9> NotePC의 제품계열 정보 확장

을 높일 수 있는 제품계열 정보 정의방법을 제시하였다. 이를 위해 제품의 정형적 개념 분석을 활용하여 제품의 속성을 정의하였고, 정형적 문맥을 기반으로 제품 계열의 기본 정보를 유추할 수 있음을 보였다. 또한 다양하고 이질적인 개발환경을 지원하기 위해 온톨로지를 활용한 제품정보 모델링방법을 제시하였고, 제품개념간 포함관계를 추론하는 온톨로지 TBox 알고리즘을 적용하여, 온톨로지 기반의 제품계열 정보를 정의하는 새로운 방법을 제시하였다.

본 논문에서 제시한 방법은 향후 제품 플랫폼을 정의하는 연구에 활용될 수가 있을 뿐 아니라, 시맨틱 웹 환경에 산재된 정보들의 관계를 추론하여 유사한 개념들을 발견하는데 사용할 수 있다.

참고문헌

- [1] 노상규, 박진수, 인터넷 진화의 열쇠-온톨로지, 가즈토이, 2007.
- [2] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F.; *The Description Logic Handbook : Theory, Implementation and Applications*, 2nd Edition, Cambridge, 2007.
- [3] Bakis, N., Aouad, G., and Kagioglou, M.; "Towards distributed product data sharing environments-Progress so far and future challenges," *Automation in Construction*, 16(5) : 586-595, 2007.
- [4] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., Yergeau, F., and Cowan, J.; "Extensible Markup Language (XML) 1.1," <http://www.w3.org/TR/xml11/#sec-xml11>, 2006.
- [5] Cycorp. Inc, <http://www.cyc.com/cyc>, 2002.
- [6] Fowler, J.; STEP for Data Management, Exchange and Sharing, *Technology Appraisals*, 1995.
- [7] Fox, M. S.; "The Tove Project-Towards a Common-Sense Model of the Enterprise," *Lecture Notes in Artificial Intelligence*, 604 : 25-34, 1992.
- [8] Fujita, K. and Yoshida, H.; "Product variety optimization : simultaneous optimization of module combination and module attributes," *ASME Design Engineering Technical Conferences*, DETC2001/DAC-21058, 2001.
- [9] Ganter, B. and Wille, R.; "Formal Concept Analysis : Mathematical Foundations," *First, Springer-Verlag*, Heidelberg, Germany, 1999.
- [10] Gruber, T. R.; "A translation approach to portable ontology specification," *International Journal of Knowledge Acquisition for Knowledge-Based Systems*, 5(2) : 199-220, 1993.
- [11] Nanda, J.; "Product Platform Knowledge Management Using the Semantic Web Paradigm," VDM, Germany, 2009.
- [12] Nanda, J., Thevenot, H. J., Simpson, W., and Kumara, S. R.; "Exploring Semantic Web Technologies for Product Family Modeling," *Proceedings of ASME/DETC*, 2004.
- [13] *Protégé*, <http://protege.stanford.edu/>, 2007.
- [14] Schenck, A. D.; *Information Modeling : the EXPRESS Way*, Oxford University, New York, USA, 1994.
- [15] Schlenoff, I. R., Libes, D., Denno, P., and Szykman S.; "An analysis of existing ontological systems for applications in manufacturing and healthcare," *National Institute of Standards and Technology*, 1999.
- [16] Simpson, T. W., Maier, J. R., and Mistree, F.; "Product platform design : method and application," *Research in Engineering Design*, 13(1) : 2-22, 2001.
- [17] Smith, M. K., Welty, C., and McGuinness, D. L.; "OWL Web ontology language guide, W3C recommendation," <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, 2004.
- [18] Tilley, T.; "Tool Support for FCA, Concept Lattice," Second International Conference on Formal Concept Analysis (ICFCA), Sydney, Australia, 104-111, 2004.
- [19] Ulrich, K. T. and Eppinger, S. D.; "Product Design and Development," *McGraw-Hill/Irwin*, New York, USA, 2004.
- [20] Uschold, M., Moralee, S., and Zorgios, Y.; "The enterprise ontology," *Knowledge Engineering Review*, 1998.