

다중 레이어 기반 제품 지식 모델

이재현*(한국과학기술원 산업공학과), 서효원(한국과학기술원 산업공학과)

MULTI-LAYERED PRODUCT KNOWLEDGE MODEL

J. H. Lee(Industrial Eng. Dept. KAIST), H. W. Suh (Industrial Eng. Dept., KAIST)

ABSTRACT

This paper introduces an approach to multi-layered product knowledge model for collaborative engineering environment. The participants in collaborative engineering want to share and reason product knowledge through internet without any heterogeneity and ambiguity. However the previous knowledge models are limited in providing those aspects. In this paper, the collaborative engineering domain is analyzed and then the product knowledge is organized into four levels such as product context model, product specific model, product design model and product manufacturing model. The four levels are represented by first-order logic in layered fashion. The concepts and the instances of a formal ontology are used for recursive representation of the four levels. The instances of the concepts of an upper level like product context model are considered as the concepts of an adjacent lower level like product specific model, and this mechanism is applied to the other levels. These logic representations are integrated with the schema and the instances of a relational database. OWL representation of the four levels is defined through the integration of the logic representation and OWL primitives. The four product knowledge models have their major representation according to the characteristics of each model. This approach enables engineer to share product knowledge through internet without any ambiguity and utilize it as basis for additional reasoning.

1. 서론

협업적 엔지니어링 과정의 참여자들은 인터넷을 이용하여 의미의 모호함 없이 제품 지식을 공유하고, 추론하기를 원한다. 하지만 기존의 지식 모델들은 명확한 의미 정의 측면에서 한계를 가지고 있다. 기존 지식모델들의 표현 문법들은 상이하기 때문에 제품 지식 공유 시 문법 변환을 필요로 한다. 또한 도메인 의미를 명세하는 방법도 명확하지 않다. 서로 다른 도메인에서 같은 개념들이 종종 다른 용어로 표현 되기 때문에 개념의 공유를 위해서는 개념들 간의 의미 맵핑이 필요하다. 게다가 이전의 모델들은 엔티티들과 클래스들 간의 관계에 대한 풍부한 의미들을 가지고 있지 않기 때문에 관계형 데이터베이스 기반 제품 모델들은 추론의 근거를 제한적으로 밖에 제공할 수 없다.

최근 들어 정보와 지식을 포괄적으로 표현할 수 있는 온톨로지가 주목받고 있다. 온톨로지의 기본 개념은 “공유된 개념화에 대한 형식적이고, 명백한 설명”이다[4]. Bozsak et al.[2]는 6 개의 요소들; 개념, 관계, 개념 구조(Concept Hierarchy), 관계 구조

(Relation Hierarchy), 관계함수, 공리, 인스턴스로 온톨로지를 정의했다. 이 온톨로지 요소들은 제품 지식 모델의 기본 요소가 될 뿐만 아니라 협업적 엔지니어링 추론의 근거로도 사용될 수 있다. 개념과 관계들은 추론의 토대가 되는 사실들을 정의하는데 사용될 수 있다. 공리들은 논리적 표현을 사용하여 개념과 관계들, 그리고 이들의 제약조건까지 설명하므로 추론 과정에 사용될 수 있다. 또한, OWL 기반 표현은 인터넷 상에서 온톨로지의 일관된 표현 문법을 제공하기 때문에 협업적 엔지니어링 환경하의 문법 변환을 피할 수 있다.

이 논문은 일차 논리(First Order Logic), 관계 데이터베이스, OWL 을 이용하여 제품 지식 모델을 제안한다. 제품 지식모델은 4 레이어의 계층적 구조를 갖는다. 이 접근 방법은 인터넷을 통하여 엔지니어가 이질적인 의미와 모호함 없이 제품 지식을 공유할 수 있도록 해주며, 추가적인 추론에도 이용할 수 있게 도와준다. 이에 대한 상세한 설명은 이후 섹션들에서 구체적으로 논의할 것이다.

논문의 구성은 다음과 같다. 2 장에서 우리가 제안하는 접근방법인 OWL 기반 하이브리드 제품 지

식 모델(OBPKM)을 서술한다. 3장에서 OBPKM 4 레이어의 세부 사항은 다루고, 4 장은 표현방법의 이슈들을 논의한다. 마지막으로 5 장에서는 본 연구의 결론과 향후 연구 방향에 대해서 서술한다.

2. 기존 연구

제품 개발 환경에서의 제품 지식에 초점을 맞춘 기존 연구들 [5, 1] 이 있다. Borst 와 Akkermans[5]은 제품 지식을 표현하는 엔지니어링 온톨로지를 더욱 구체적이고 명확하게 발전시켰다. 엔지니어링 온톨로지는 구성 요소, 프로세스, 엔지니어링 수학, 이렇게 세 부분으로 구성되어 있다. Lin and et al.[1]은 고객의 요구사항과 제품의 상세 내역을 관리하기 위한 요구사항 온톨로지를 제안했다. 비록 두 개의 연구들이 온톨로지를 이용한 동일한 방법으로 제품 지식 모델을 설계했다더라도 일차 논리를 제외한 다른 표현을 제공하지는 않는다.

Rezayat, M.[6]은 지식 공유를 위한 공유 용어집을 제안했다. 이 용어집에 정의된 용어들은 ‘주요 특성치’라 불리는 중요한 제품 특성들이다. 그는 어떤 지식이 중요하고, RDF 문법으로 그것들을 어떻게 사용하는가에 대해 제안했지만 공유 용어집의 데이터 구조를 너무 단순하게 정의했다. CONGEN[3] 시스템은 독자적인 데이터 구조에 기초하여 개발되었다. 이 시스템은 설계자의 의도나 요구사항 같은 제품 지식을 관리한다. 두 연구 모두 고유의 제품 지식 모델 데이터 구조를 개발했지만 의미의 이질성이나 모호함을 해결할 수 있는 ‘형식을 갖춘 공리’는 정의하지 않았다.

위에서 설명한 제품 지식 모델들은 웹 상에서 이질적이고 모호한 의미없이 제품 지식을 공유하고 추론하는데 있어서 한계를 가지고 있다.

3. OWL 기반 하이브리드 제품 지식 모델

본 장에서는 협업적 엔지니어링 환경에서의 OWL 기반 하이브리드 제품 지식 모델(OWL based Hybrid Product Knowledge Model: OBPKM)을 제안한다. 협업적 환경에서 제품 지식은 포괄적이며 명백하고 일관된 지식 공유를 위한 구조와 표현을 필요로 한다. 이것을 위한 제품 지식 모델은 4 레이어로 구성되며, 그것들은 형식 온톨로지에 따른 일차 논리, 관계 데이터베이스, OWL 에 의해 종합적으로 표현된다.

제품 지식 모델: 기존의 연구들은 제품 지식 모델의 다양한 양식을 제안한다. 그러나 제품 설계와 제조를 포함한 제품 지식의 포괄적인 통합 모델은

완벽하게 논의되지 못하였다. 우리는 제품 지식을 제품 배경 모델(PCM), 특정 제품 모델(PSM), 제품 설계 모델(PDM), 제품 제조 모델(PMM), 이렇게 4 레이어로 정의한다. PCM 은 특정 제품에 대한 고려 없이 제조 산업의 제품 배경 정보를 표현한다. PCM 은 기능, 부품, 외형, 구조, 특성 등과 같은 개념들을 포함한다. 또한 IS-A, HAS-A 등과 같은 개념 사이의 관계 술어들을 포함한다. 이 모델은 PSM 의 메타 모델이다. PSM 은 자동차, 휴대폰, 책상 등과 같은 특정 제품의 온톨로지를 표현한다. PSM 모델은 또한 제품 설계 정보, 특히 설계 변경 정보를 포함하는 설계 정보의 온톨로지를 표현하는 PDM 의 메타 모델이다. PMM 은 상세 설계 버전의 개별 제조 항목의 온톨로지를 표현하며, PDM 의 인스턴스 레이어이다. 그림 1 은 제품 지식 모델의 4 레이어를 보여준다.

온톨로지 기반: 온톨로지 요소들은 제품 지식 모델의 기초로써 이용될 수 있다. 일반적으로 개념과 관계들은 제품 지식의 기본 구조를 표현한다. 그러므로 PCM 은 개념 체계, 관계 체계, 관계함수를 포함하는 개념과 관계로 표현될 수 있다. PCM 개념의 인스턴스들은 PSM 같이 인접한 하위 레이어의 개념을 표현하는데 사용된다. 이와 같이 제품 지식의 네 레이어는 통합적으로 표현된다. 반면, 공리들은 개념과 관계들을 명확하게 설명하기 때문에 제품 지식의 모호함을 완화시킬 수 있다. 또한 공리들은 논리적 언어로 표현되는 개념과 관계들의 관계적인 제약조건들을 포함하기 때문에 부가적인 추론의 토대가 될 수 있다.

모델의 표현: 제품 모델의 4 레이어는 일차 논리, 관계형 데이터베이스, OWL 에 의해서 표현된다. 개념, 관계, 공리 등과 같은 PSM 의 온톨로지 요소들의 대부분은 일차 논리에 의해 표현된다. 인접한 하위 레벨인 PDM 의 개념인 PSM 인스턴스들은 관계형 데이터베이스의 인스턴스들에 의해 통합된다. 일차 논리에 의해 표현되는 온톨로지 요소들은 OWL 로도 표현될 수 있다. BUBO 접근 방법[8]은 OWL 표현을 일차논리로 변환하는데 이용된다. 마찬가지로 다른 레이어에서도 일차 논리, 관계형 데이터 베이스, OWL 에 의한 표현 방법과 변환 방법을 반복적으로 적용할 수 있다. 이런 접근 방법으로 대부분의 데이터들은 관계형 데이터베이스에 의해 효율적으로 관리될 수 있으며, PCM 과 PSM 레이어에서의 논리적 표현에 의해 효율적으로 추론할 수 있다. 그리고 인터넷을 통한 정보와 지식의 공유는 OWL 표현으로 실현된다.

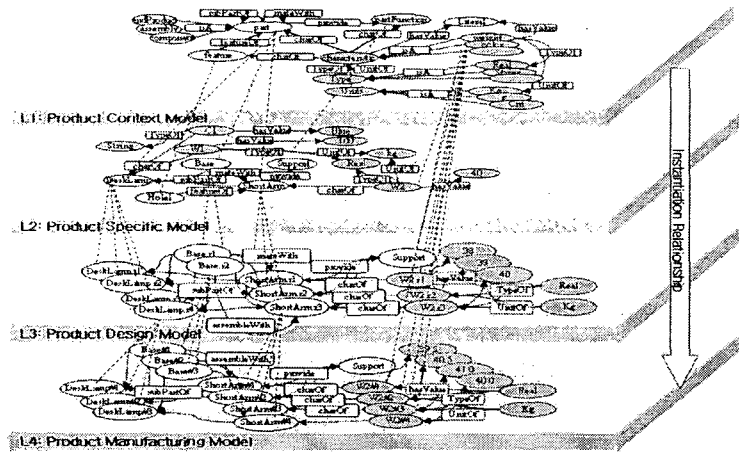


그림 1 : Desk-Lamp example for 4-layers Product Knowledge Model

4. 제품 지식 모델 : 4 레이어

제품 지식 모델의 4 레이어는 온톨로지 요소에 근거하여 설계되었다. 그렇기 때문에 각 레이어는 개념, 관계, 개념 구조, 관계 구조, 관계함수, 공리로 구성된 동일한 구조를 가진다. 또한 상위 레이어와 하위 레이어의 관계는 인스턴스화 관계이다. 예를 들면, 'DeskLamp', 'ShortArm', 'Base'들과 같은 PSM의 특정 부품 개념들은 그림 1에서 보여주듯이 PCM에서의 부품 개념 인스턴스들이다. 'DeskLamp.r1', 'DeskLamp.r2', 'DeskLamp.r3'와 같은 PDM에서의 설계 변동 개념은 PSM의 특정 부품 개념인 'DeskLamp'의 인스턴스들이다. 그리고 'DeskLamp.r3.#1', 'DeskLamp.r3.#2', 'DeskLamp.r3.#3'과 같이 PMM에서 물리적으로 제조된 항목은 PDM의 설계 변동 개념인 'DeskLamp.r3'의 인스턴스들이다. 마지막으로 각 레이어들은 고유의 특성을 가지고 있고, 각 레이어의 주요 표현방법은 이 특성에 따라 결정된다.

4.1 제품 배경 모델 (PCM)

온톨로지 요소: PCM은 특정 제품에 대한 고려 없이 제조 산업에 적용될 수 있는 일반적인 제품 배경 온톨로지를 표현한다. PCM은 제품 지식을 표현하는 기본 개념과 관계들을 정의한다. 'IS-A' 관계는 온톨로지 요소들의 개념 구조를 표현한다. 각 관계들은 도메인(domain)과 범주(range)를 정의하는 관계함수를 가지고 있다. 관계함수는 관계에 의해 연결된 개념들을 명시한다. PCM에서는 인스턴스들이 온톨로지 요소일지라도, PSM 레이어에서는 개념으로서 정의된다.

가장 중요한 온톨로지 요소들 중 하나인, PCM

레이어의 공리는 두가지 종류로 분류된다. 하나는 용어에 대한 공리이고, 다른 하나는 도메인 제약조건에 대한 공리이다. 엄밀히 말하면, 용어에 대한 공리는 용어 형태에 따라 다시 두가지 종류로 분류된다. 하나는 기초 용어들(primitive terms)에 대한 공리이고, 다른 하나는 정의된 용어들(defined terms)에 대한 공리이다. PCM의 두가지 용어들 모두 기본적인 속성들을 가지고있기 때문에, 공리에 대한 논리적 표현은 추론의 근거로 사용될 수 있다. 기초 용어들에 대한 공리는 개념과 관계들의 기본 속성을 정의한다. 정의된 용어에 대한 공리는 기초 용어를 기반으로 정의된 용어 정의를 포함한다.

도메인 제약 조건에 대한 공리도 역시 개념과 관계의 기본 의미들로 정의할 수 있다. 하지만 그 의미들은 양적인 값과 더 깊은 관련을 가지고 있다. 예를 들면, 'featureOf' 관계의 카디널리티(cardinality)는 N:1이 되어야 한다. 부품은 여러 형상을 가질 수 있지만 형상은 오로지 하나의 부품에만 속해야 하기 때문이다. 다른 예를 들면, 'weight'는 0 이상이어야 한다. 이런 논리들이 제품 배경에는 자연스럽게기 때문에 그것들을 공리로서 정의해야 한다.

4.2 특정 제품 모델 (PSM)

온톨로지 요소: PSM은 자동차, 휴대폰, 책상등과 같은 특정 제품에 대한 온톨로지이며, PCM 레이어의 인스턴스이다. 대부분의 PSM 개념들은 PCM의 인스턴스들이며, PSM 인스턴스들 간의 관계는 PCM에서 정의되어있다. 다른 온톨로지 요소들 역시 PCM 레이어에서 했던 것처럼 정의할 수 있다.

PSM의 공리들은 PCM의 공리들과는 다르다.

PSM의 공리들은 특정 제품의 요구사항이나 속성의 구속과 같은 특정 제품의 제약조건을 고려한다.

4.3 제품 설계 모델 (PDM)

온톨로지 요소: PDM은 PSM에서 정의한 특정 제품의 설계 변동에 대한 온톨로지이다. 일반적으로 각각의 설계 변동은 값을 제외한 속성들과 관계들이 비슷한 유형으로 나타난다. 마찬가지로 PDM의 개념과 관계는 PSM의 그것들과 비슷한 유형을 가지고 있다.

필요한 경우에는 다른 관계들을 정의할 수 있다. PDM의 대부분의 개념과 관계들이 PSM의 개념과 관계들로부터 인스턴스화되었을지라도 인스턴스화된 개념들 간에 새로운 관계가 요구된다면, PDM에서 관계를 새롭게 정의할 수 있다.

PDM의 공리들은 설계 변동에 대한 제약조건과 관계가 있다. 공리들은 특정 설계 변동에 적용될 수 있어야 한다.

4.4 제품 제조 모델 (PMM)

온톨로지 요소: PMM의 개념들은 실제로 존재하고, 제조되는 물리적인 아이템들을 지칭한다. 일반적으로 각각의 제조 아이템은 속성 값들을 제외하면 비슷한 정보를 가지고 있다. PMM의 관계들은 PDM의 관계들로부터 인스턴스화된다. 개념/관계 구조와 관계함수는 이 레이어에서 정의될 필요가 없다. PMM에서의 공리들은 제조 이슈들의 제약조건으로서 고려될 수 있다.

5. 모델의 표현

제품 지식 모델의 4 레이어는 일차 논리(FOL), 관계형 데이터베이스(RDB), OWL로 표현된다. 개념, 관계, 공리 등과 같은 각 레이어의 온톨로지 요소

들은 대부분 FOL로 표현되며, OWL 용어으로도 표현된다. 인스턴스 요소들은 관계형 데이터베이스로 표현된다.

FOL, RDB과 OWL, 이 세가지 표현 방법은 OBPKM에서 서로 다른 역할을 하고 있다. 먼저 FOL은 충분한 논리 표현을 가지고 있어서 온톨로지의 개념과 관계 뿐만 아니라 공리들까지도 효율적으로 표현할 수 있다. 따라서 이 표현 방법은 추론의 근거로 사용될 수 있다[7]. 둘째로 RDB는 많은 양의 데이터들을 강력하게 관리할 수 있다. FOL은 데이터들을 표현할 수 있을 지라도 많은 데이터들을 관리하는 데에는 적합하지 않다. 온톨로지 요소들에 대한 많은 양의 인스턴스들을 다루어야 한다면, 그것들을 관리하기 위한 데이터베이스를 사용하는 것이 좋다. 마지막으로 OWL은 웹 상에서 OBPKM을 나타내기 위한 의미와 XML 문법을 제공한다. OWL 덕분에 인터넷을 통해 제품 지식 모델을 공유하고 교환할 수 있다.

3 장에서는 OBPKM의 4 레이어를 정의하였으며, 이제 제품 지식 모델의 각 레이어에 대한 적절한 표현 방법을 정의할 필요가 있다. 세가지 표현을 이용한 접근 방법으로 제품 지식의 4 레이어를 표현할 수 있지만, 특정 표현 방법은 특정 레이어에 알맞은 것이다. 일반적으로 PCM과 PSM는 제품 배경과 특정 제품의 의미를 묘사하므로 용어에 대한 논리적인 정의와 개념과 관계에 대한 관계형 제약 조건을 가지고 있어야 한다. 따라서 PCM과 PSM의 주 표현 방법으로 FOL이 추천된다. 웹 상에서 지식 공유가 요구될 때 마다 OWL 표현 방법이 사용될 뿐만 아니라 PCM과 PSM의 알맞은 표현 방법으로도 사용될 수 있다. 그림 2은 PCM/PSM에서의 OWL 표현 방법을 보여준다.

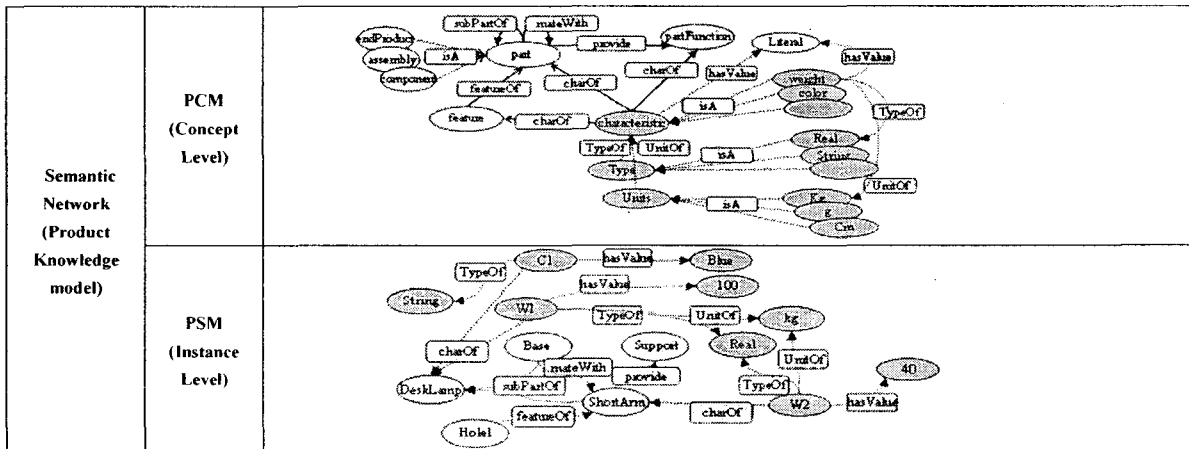


그림 2: Three Representations of PCM/PSM level (Continued)

| FOL | PCM | <p>Concept part(p) feature(f) partFunction(r) characteristics(c) color(c) weight(c) Type(t) Unit(u) String(s) </p> | <p>Concept Hierarchy endProduct(p) ⇒ part(p) assembly(p) ⇒ part(p) </p> <p>Relation provide(p, r) subPartOf(p1, p2) mateWith(p1, p2) featureOf(f, p) TypeOf(c, Type) UnitOf(c, Units) hasValue(c, String)</p> | <p>Axioms (term's specification) Primitive term's axioms $(\forall p1, p2) \text{ subPartOf}(p1, p2) \Rightarrow \neg \text{subPartOf}(p2, p1)$ $(\forall p1, p2, p3) \text{ subPartOf}(p1, p2) \wedge \text{subPartOf}(p2, p3) \Rightarrow \text{subPartOf}(p1, p3)$ $\neg (\exists p) \text{ endProduct}(p) \wedge \text{assembly}(p)$ $\neg (\exists p) \text{ assembly}(p) \wedge \text{component}(p)$ $\neg (\exists p) \text{ component}(p) \wedge \text{endProduct}(p)$ Defined term's axioms $(\forall p1, p2) \text{ directSubPartOf}(p1, p2) \Leftrightarrow \text{subPartOf}(p1, p2) \wedge \neg (\exists p3) \wedge \text{subPartOf}(p3, p2) \wedge \text{subPartOf}(p1, p3)$ Axioms (domain constraints) $(\forall f) \text{ feature}(f) \rightarrow (\exists^{\geq 1} p) \text{ featureOf}(f, p)$ $(\forall c, x) \text{ weight}(c) \wedge \text{hasValue}(c, x) \wedge (x > 0)$</p> | | | | | | | | | | | |
|---------------------------|-----------------|---|--|--|------|----------------|----|--------------|----|------|----|---------|----|-----------------|-----|
| | PSM | <p>Instance part(DeskLamp) part(ShortArm) part(Base) feature(Hole1) partFunction(Support) weight(W1) weight(W2) color(C1)</p> | <p>Instances' Relation provide(ShortArm, Support) subPartOf(ShortArm, DeskLamp) mateWith(Base, ShortArm) featureOf(Hole1, ShortArm) charOf(W2, ShortArm) charOf(W1, DeskLamp) charOf(C1, DeskLamp) hasValue(W2, 40) hasValue(C1, Blue)</p> | | | | | | | | | | | | |
| OWL & Relational Database | | OWL | | Relational Database | | | | | | | | | | | |
| | PCM | <pre> <?xml version="1.0"?> <rdf:RDF ...> <owl:Class rdf:ID="endProduct"> <rdfs:subClassOf> <owl:Class rdf:ID="part"> <rdfs:subClassOf> <owl:disjointWith> <owl:Class rdf:ID="component"/> </owl:disjointWith> <owl:disjointWith> <owl:Class rdf:ID="assembly"/> </owl:disjointWith> </owl:disjointWith> </owl:Class> ... <owl:ObjectProperty rdf:ID="provide"> <rdfs:range rdf:resource="#partFunction"/> <rdfs:domain rdf:resource="#part"/> </owl:ObjectProperty> ... <owl:TransitiveProperty rdf:ID="subPartOf"> <rdfs:type rdf:resource="#ObjectProperty"/> <rdfs:range rdf:resource="#part"/> <rdfs:domain rdf:resource="#part"/> </owl:TransitiveProperty> ... <owl:ObjectProperty rdf:about="#hasType"> <owl:inverseOf> <owl:ObjectProperty rdf:ID="TypeOf"> </owl:inverseOf> <rdfs:domain rdf:resource="#characteristic"/> <rdfs:range rdf:resource="#Class"/> <protege:allowedParent rdf:resource="#Type"/> </owl:ObjectProperty> </owl:inverseOf> ... </pre> | | <p>FOL: Concepts (ConceptList) Concepts({part, feature, partFunction, characteristics, ...})</p> <p><i>(Relation Database - FOL Translator)</i></p> <p>SQL: SELECT Name FROM Concept_Table WHERE All</p> <p style="text-align: center;">Concept_Table</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>T-No</th> <th>Name (concept)</th> </tr> </thead> <tbody> <tr> <td>T1</td> <td>partFunction</td> </tr> <tr> <td>T2</td> <td>part</td> </tr> <tr> <td>T3</td> <td>feature</td> </tr> <tr> <td>T4</td> <td>characteristics</td> </tr> <tr> <td>...</td> <td>...</td> </tr> </tbody> </table> | T-No | Name (concept) | T1 | partFunction | T2 | part | T3 | feature | T4 | characteristics | ... |
| T-No | Name (concept) | | | | | | | | | | | | | | |
| T1 | partFunction | | | | | | | | | | | | | | |
| T2 | part | | | | | | | | | | | | | | |
| T3 | feature | | | | | | | | | | | | | | |
| T4 | characteristics | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | |

그림 2: Three Representations of PCM/PSM level (Continued)

| | OWL | Relational Database | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---|--|--------|--------|--------|-------|--------|----|----------|-----|------|--|----|----------|----|-----|----|----|------|----|--------|----|------|------|--------|-------|--------|----|----------|-----|------|--|------|------|------|--------|----|--------|----|----|----|--------|----|----|----|----------|----|----|
| PSM | <pre> ... <part rdf:ID="ShortArm"> <subPartOf> <part rdf:ID="DeskLamp"/> </subPartOf> <provide> <partFunction rdf:ID="Support"/> </provide> </part> ... <feature rdf:ID="Hole1"> <featureOf rdf:resource="#ShortArm"/> </feature> ... <weight rdf:ID="W2"> <charOf rdf:resource="#ShortArm"/> <hasValue rdf:datatype="#string">40</hasValue> <hasType rdf:resource="#Real"/> <hasUnit rdf:resource="#Kg"/> </weight> </pre> | <p>FOL: part(DeskLamp),part(ShortArm),part(Base), weight(W1), weight(W2), weight(W3), charOf(W1,DeskLamp),charOf(W2,ShortArm), charOf(W3, Base),hasValue(W1, 100), hasValue(W2, 40), hasValue(W3, 60), ...</p> <p>SQL: SELECT P-No, Name, Weight, Color FROM PART WHERE All</p> <p>Part Table</p> <table border="1"> <thead> <tr> <th>P-No</th> <th>Name</th> <th>Weight</th> <th>Color</th> <th>P-Part</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>DeskLamp</td> <td>100</td> <td>Blue</td> <td></td> </tr> <tr> <td>P2</td> <td>ShortArm</td> <td>40</td> <td>Red</td> <td>P1</td> </tr> <tr> <td>P3</td> <td>Base</td> <td>60</td> <td>Yellow</td> <td>P1</td> </tr> </tbody> </table> <p>endProduct Table</p> <table border="1"> <thead> <tr> <th>P-No</th> <th>Name</th> <th>Weight</th> <th>Color</th> <th>P-Part</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>DeskLamp</td> <td>100</td> <td>Blue</td> <td></td> </tr> </tbody> </table> <p>Feature Table</p> <table border="1"> <thead> <tr> <th>F-No</th> <th>Name</th> <th>Part</th> <th>Length</th> </tr> </thead> <tbody> <tr> <td>F1</td> <td>Hole 1</td> <td>P2</td> <td>10</td> </tr> <tr> <td>F2</td> <td>Hole 2</td> <td>P2</td> <td>15</td> </tr> <tr> <td>F3</td> <td>CounSink</td> <td>P2</td> <td>20</td> </tr> </tbody> </table> | P-No | Name | Weight | Color | P-Part | P1 | DeskLamp | 100 | Blue | | P2 | ShortArm | 40 | Red | P1 | P3 | Base | 60 | Yellow | P1 | P-No | Name | Weight | Color | P-Part | P1 | DeskLamp | 100 | Blue | | F-No | Name | Part | Length | F1 | Hole 1 | P2 | 10 | F2 | Hole 2 | P2 | 15 | F3 | CounSink | P2 | 20 |
| P-No | Name | Weight | Color | P-Part | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P1 | DeskLamp | 100 | Blue | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P2 | ShortArm | 40 | Red | P1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P3 | Base | 60 | Yellow | P1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P-No | Name | Weight | Color | P-Part | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P1 | DeskLamp | 100 | Blue | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F-No | Name | Part | Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F1 | Hole 1 | P2 | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F2 | Hole 2 | P2 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F3 | CounSink | P2 | 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

그림 2: Three Representations of PCM/PSM level

6. 결론

우리는 협업적 엔지니어링 환경 하의 포괄적이고 명확한 지식 공유를 위한 OWL 기반 하이브리드 제품 지식 모델 (OBPKM) 을 제안했다. 제품 지식은 제품 배경 모델, 특정 제품 모델, 제품 설계 모델, 제품 제조 모델, 이렇게 4 레이어로 구성된다. 4 레이어의 모델은 계층적 방식의 일차 논리로 표현된다. 이 논리 표현 방법은 관계형 데이터 베이스로 통합되고, OWL 기반 제품 지식으로 발전된다.

제품 지식 모델을 위한 OBPKM 이 제안되었지만, 이것을 실용적이고 상업적으로 만들기 위해서는 더 많은 연구가 필요하다. 무엇보다도 OBPKM 은 실용적인 응용으로의 확장과 지능형 결정을 위한 문제 해결 에이전트와의 통합이 필요하다. 각 단계에서 온톨로지 요소들의 변화와 확장은 일차 논리, 관계형 데이터 베이스, 그리고 OWL 의 표현 방법에 의해 통합적이고 일관적으로 관리되어야 한다. 특히 표현 방법들 간에 온톨로지 요소의 공리들을 공유하는 것은 다양한 응용에 필수적이다.

참고문헌

- Borst, P. and Akkermans, H., 1997, "Engineering ontologies." International Journal of Human-Computer Studies, Vol. 46, pp. 365-406.
- Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S. and Stojanovic, L., 2002. "KAON - Towards a Large Scale Semantic Web," Lecture notes in computer science, No. 2455, pp.304-313.
- Gorti, S.R. and Sriram, R.D., 1996, "From symbol to form: a framework for conceptual design," Computer-Aided Design, Vol. 28, No. 11, pp.853-870.
- Gruber, T. R., 1993, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition, Vol. 5, pp. 199-220.
- Olsen, G. R., Cutkosky, M., Tenenbaum, J. M. and Gruber, T. R., 1994, "Collaborative engineering based on knowledge sharing agreements," Engineering data management: integrating the engineering enterprise, 1994, pp.1-12.
- Rezayat, M., 2000, "Knowledge-based product development using XML and KCs," Computer-Aided Design, Vol. 32, pp. 299-309.
- Russell, S., and Norvig, P., 1995, *Artificial Intelligence - A Modern Approach*, Prentice Hall, pp. 240-271.
- Volz, R., Decker, S., and Oberle, D., 2003, "Bubo - implementing owl in rule-based systems," *WWW2003*, Budapest, Hungary.