

수명주기가 긴 제품의 설계정보관리를 위한 다층 제품정보 모델링 방안

이재현^{1*} · 서효원²

¹미국표준기술연구소 시스템통합부, ²한국과학기술원 산업공학과

Multi-level Product Information Modeling for Managing Long-term Life-cycle Product Information

Jaehyun Lee^{1*} and Hyowon Suh²

¹System Integration Division, National Institute of Standards and Technology

²Industrial Engineering, Korea Advanced Institute of Science and Technology

Received 25 January 2012; received in revised form 9 May 2012; accepted 11 May 2012

ABSTRACT

This paper proposes a multi-level product modeling framework for long-term lifecycle products. The framework can help engineers to define product models and relate them to physical instances. The framework is defined in three levels; data, design model, modeling language. The data level represents real-world products, The model level describes design models of real-world products. The modeling language level defines concepts and relationships to describe product design models. The concepts and relationships in the modeling language level enable engineers to express the semantics of product models in an engineering-friendly way. The interactions between these three levels are explained to show how the framework can manage long-term lifecycle product information. A prototype system is provided for further understanding of the framework.

Key words: Multi-level modeling framework, Long-term product information, OWL, Semantic modeling

1. 서 론

자동차, 비행기, 선박 또는 건물과 같은 제품의 경우 수명주기가 짧게는 5년에서 길게는 20년이 된다. 이처럼 수명주기가 긴 제품의 경우 제품의 유지 및 보수관리를 목적으로 또는 법률적 규제에 따라 설계 모델 정보를 장기간 보관해야 할 필요

가 있다.

제품 설계 정보들의 특성들 중 다음과 같은 특성들은 제품설계 정보들을 장기간 보관하는데 어려움들을 일으킨다^[1].

- 제품 설계 정보는 CAD(Computer Aided Design), PDM (Product data management) 시스템 및 사무용 소프트웨어 등 여러 다양한 어플리케이션들로 만들어진다.
- 제품 개발 프로세스를 진행함에 따라 설계 정보의 양이 기하급수적으로 증가한다.

*Corresponding Author, jaehyun.lee@nist.gov
©2012 Society of CAD/CAM Engineers

- 제품 설계 정보의 작은 오류가 후속 제품 개발 프로세스에 큰 영향을 미치기 때문에, 제품 정보의 보관 기간 동안 높은 수준의 정확도가 유지되어야 한다.
- 정보 시스템이나 정보 기술의 발전 속도가 빠르기 때문에, 기존 시스템 또는 기존 기술이 사장 되더라도 제품 설계 정보는 재활용이 가능해야만 한다.
- 장기간 보관되는 제품 설계 정보를 잘 검색하고 활용하기 위해서는 각 제품 정보들에 대해 충분한 의미 기술들이 필요하다.

미국 표준과학기술연구원(NIST)에서는 몇 차례의 워크숍들^[2,3]을 통해 디지털 설계 정보를 장기간 보관하고 재활용 하기 위한 제안들을 하였다. 그 중 몇 가지 제안들은 제품정보 메타모델에 대한 설계에 초점을 맞추었다. 디지털 제품설계정보를 저장할 때 향후 사용자의 정보활용활동을 고려하여 제품정보를 저장할 것을 제안하며, 제품정보 메타모델이 저장정보에 대한 설명을 제공하도록 하였다.

일반적으로 엔지니어들은 기존의 제품 정보들을 단순히 참조하거나, 더 나아가 다른 어플리케이션을 이용해 재활용하거나, 제품 정보들을 설계한 배경에 대한 추론활동을 수행한다. NIST의 워크숍 보고서^[1]에 따르면 제품정보 보관시스템의 제품정보 메타모델을 개발하기 위해서는 1) 제품과 프로세스 정보를 위한 새로운 표현방법 개발, 2) Open Archival Information System (OAIS) 참조모델^[4]을 제품 도메인에 확장하여 정의, 3) 보관된 디지털 정보관리 측면에서 미래 제품정보 접근 요구사항에 대한 연구가 필요하다 하였다.

이와 같은 연구 활동들은 기존의 제품정보 관리 방법들을 장기간 보관이라는 새로운 관점에서 바라보기 위한 시도들이다. 기존의 데이터베이스 중심의 제품 정보 관리 시스템들은 각각의 설계 정보들과 그들 간의 복잡한 관계를 체계적으로 저장하는 것을 가장 중요하게 생각하였다. 하지만, 제품 엔지니어링 정보들이 장기간 보관될 필요가 있는 상황에서는 주어진 정보의 단순 저장보다는 추후에 그 정보들을 사용하는 엔지니어들의 요구사항을 고려한 보관이 필요하다.

본 논문에서는 다음과 같은 엔지니어의 요구사항들을 가정하였다. 첫째, 수명주기가 긴 제품들

의 경우에는 과거의 설계정보들을 참조하거나 재 활용하고 싶더라도 파일 형식의 불일치, 과거 어플리케이션의 부재 등으로 문제가 발생할 수 있다. 이를 위해서는 제품 정보에 대한 중립적 표현 방법이 필요하다. 둘째, 제품정보 메타모델도 장기간의 시간이 지남에 따라 수정이 필요할 수도 있다. 따라서, 제품정보 메타모델은 데이터베이스 스키마보다 유연한 구조를 필요로 하며, 수정된 메타모델이 기존의 메타모델과 모순을 일으키는지 확인할 수 있는 추론방법이 필요하다. 셋째, 설계자가 검색을 하고 싶을 때, 과거 자료에 충분한 설명이 없거나, 적절한 검색어를 모르거나, 검색한 결과에 대한 확신이 없을 수도 있다. 이를 위해서는 보관된 제품정보의 의미를 기술할 수 있는 제품정보 메타모델이 필요하다. 넷째, 수명 주기가 긴 제품의 경우 시간이 지날수록 설계정보와 실제 사용정보가 다를 수 있다. 따라서, 설계모델과 실제 데이터간의 관계를 관리할 수 있는 정보 모델링들이 필요하다.

본 논문에서 제안하는 다층 제품정보 모델링 방안은 정보모델과 실측데이터 간의 관계를 논리적으로 관리할 수 있는 정보 틀을 제공함으로써 앞서 정의한 요구사항들을 만족시킨다. 제안하는 정보 틀은 제품정보의 중립적 표현방법으로써 OWL(Web Ontology Language)^[4]을 활용하되, 엔지니어에게는 OWL 문법을 숨기고, 엔지니어 친화적인 정보모델링 환경을 제공한다. 그리고, 제품정보를 1) 메타모델 레벨, 2) 설계모델 레벨, 3) 실측데이터 레벨로 구성하고 각 레벨에 정의되는 정보모델 간의 관계를 논리적으로 관리할 수 있는 방법을 제공한다. 제품 메타모델은 제품 개발관련 일반적 개념과 관계들을 온톨로지로 제공한다. 본 논문에서는 메타모델로 저자들이 개발한 SPMM (Semantic Product Meta-Model)^[5]을 활용하였다. 설계모델 레벨은 엔지니어가 SPMM 개념과 관계들을 이용하여 정의하는 제품설계모델이다. SPMM과 제품설계모델은 OWL로 표현된 후, OWL 추론을 통해 모델의 오류를 검토하거나 의미기반 검색에 활용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 제품정보 모델링 프레임워크 연구들을 검토한다. 3장에서는 본 논문에서 제안하는 다층 제품정보 모델링 프레임워크를 소개한다. 4장에서는 프레임워크의 가장 상위 레벨에 정의된 SPMM 클레

스와 관계들을 간략히 소개한다. 5장에서는 엔지니어가 SPMM 클래스와 관계들을 이용하여 어떻게 설계모델을 만들 수 있으며, 수명 주기가 긴 제품의 경우 실측 데이터와 설계모델 간의 관계를 어떻게 관리할 수 있는지 설명한다. 6장은 제안한 프레임워크를 구현한 프로토타입의 아키텍처와 구현 예제를 설명하고, 마지막으로 7장에서 본 논문 요약과 추후 연구 과제들을 기술한다.

2. 기존 연구

다층 정보 모델링은 정보모델의 재활용성을 최대화 시키기 위한 방법이다. 정보모델은 일반적으로 모델링 대상이 바뀔 때마다 새롭게 만들어야 하지만, 각 정보모델의 일반적인 개념과 관계들을 모아서 만든 상위 정보모델은 재활용이 가능하다. 다층 정보모델링 구조는 과거에도 많이 제안되었었다.

Guarino^[6]는 온톨로지를 상위 온톨로지, 도메인 온톨로지, 응용 온톨로지 등으로 구분하여 온톨로지의 재활용성을 높였다. 상위 온톨로지는 추상화 수준이 높은 개념과 관계들이 정의되는데, 예를 들어 SUMO(Suggested upper merged ontology)^[7]와 같은 상위 온톨로지들이 개발되었다.

Model-Driven Architecture(MDA)^[8]에서도 다층 모델링 구조가 제안되었는데, 각 층을 데이터, 모델, 모델링 언어, 메타 모델링 언어로 구분하였다. Lee *et al.*^[9]은 MDA를 협력적 제품개발환경에 적용한 온톨로지 구조를 제안하였다. 이 구조는 제품정보 표현언어와 모델을 서로 다른 층으로 구분하고, 제품정보 모델은 다시 Guarino의 온톨로지 구분처럼 상위 온톨로지 모델, 도메인 종속적 모델, 특정 어플리케이션 모델로 구분함으로써 상위 모델의 재활용성을 추구하였다.

Core Product Model 2(CPM2)^[10]는 제품정보 모델링을 위한 상위 개념과 관계들을 정의하였다. CPM2는 UML(Unified modeling language)로 클래스들과 관계들을 정의하였고, Java 클래스를 이용하여 구현 가능성을 보였다. CPM2의 정보모델은 클래스와 인스턴스 층으로 구분된다. 하지만, CPM2를 산업 현장에 적용할 때 제품 정보의 추상화 수준이 다양할 경우, CPM2는 ‘중간 모델링 층’(intermediate level)을 기존 클래스와 인스턴스 층 사이에 추가하는 것을 제안하였다.

Ontological Product Modeling(OPM)^[11] 연구는 CPM2와는 다르게, 제품정보 모델링 상위 개념과 관계들을 모델링 언어 수준에서 정의하였다. 상위 개념과 관계들이 언어 수준에 정의되어 있으므로, 엔지니어가 설계정보를 모델링 할 때 이들을 직접적으로 이용할 수 있다는 장점이 있다.

SPMM^[5] 연구는 OPM에서 제안한 다층 정보 모델링 프레임워크를 활용하고, OPM와 CPM2의 제품 정보 모델링 개념과 관계들의 부족한 점들을 보완하여 통합된 OWL기반의 제품정보 메타모델을 제안하였다. 또한, SPMM에서는 서로 다른 층에 정의된 정보들 간의 관계를 체계적으로 관리하기 위한 접근방법이 구체적으로 제안되었다.

ISO-15926 공정플랜트 정보모델 표준안^[12]도 OWL을 이용한 다층 정보 모델링 구조를 제안하였다. 이 표준은 플랜트 내 부품의 유지보수정보를 관리하기 위하여 크게 메타 모델, 모델, 데이터 층을 정의하였다. 그리고, 플랜트 정보를 모델링 할 때 엔지니어의 편의를 위하여 지식 입력 양식과 입력된 양식을 OWL로 변환하는 규칙들을 정의할 수 있는 표준 양식을 제안하였다. 이 표준은 국내 공정플랜트에도 적용되어 활용되고 있다^[13].

본 논문에서는 수명주기가 긴 제품정보를 관리하기 위한 요구사항들을 만족시키기 위해서 SPMM 연구에서 제안한 다층 정보모델링 프레임워크를 이용한다. 그리고, ISO-15926에서 제안한 지식 입력 양식을 이용하여 엔지니어가 수명주기가 긴 제품 정보모델을 정의하도록 하고, 실측데이터와 정보모델 간의 관계를 관리하는 방법을 제안한다.

3. 다층 정보모델링 프레임워크

다층 정보모델링 프레임워크는 크게 세 층으로 구성된다. 최상위 층에는 제품정보 모델링언어(M2)가 정의되고, 중간 층에는 그 모델링언어로 기술된 제품 설계모델(M1)이 정의되고, 최하위 층에는 설계 모델에 따라 만들어진 제품의 실측데이터(M0)가 기술된다. 각 층의 명확한 구분을 위하여 본 연구에서는 M2, M1, M0라는 표기를 각 모델링 층에 추가하였다. Fig. 1은 다층 정보모델링 프레임워크의 각 층을 보여준다.

제품정보 모델링언어(M2)는 제품 설계 영역의 일반적 개념과 관계들을 정의한 제품 온톨로지이다. 이 언어는 OWL을 상속받아 만든 언어이며,

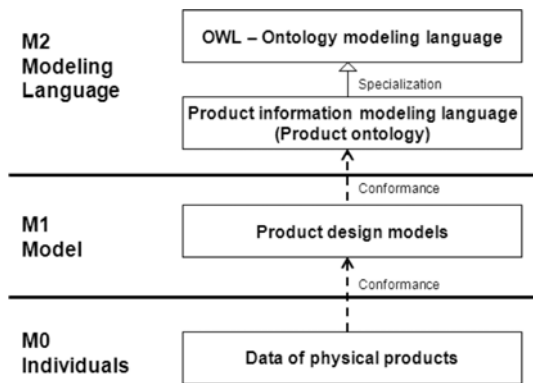


Fig. 1 Multi-level information modeling framework

동시에 OWL로 각 개념과 관계들의 의미들(공리들)이 기술된다. 제품정보 모델링언어는 엔지니어에게 친숙한 용어들을 제공해 줌으로써 엔지니어가 제품 설계정보를 모델링을 할 때 OWL의 언어 요소들을 배워야 하는 어려움을 덜어준다. 그리고, 정의된 개념과 관계들에 대한 공리들은 엔지니어가 정보를 모델링을 할 때 부족한 정보가 없는지 또는 잘못 정의된 정보가 없는지 여부를 자동 확인할 수 있는 규칙들이 된다. 제품정보 모델링언어에 정의된 개념들과 관계들은 4장에서 자세히 소개한다.

제품 설계모델(M1)은 엔지니어가 관심있는 제품 또는 부품에 대한 정보모델이다. 이 정보모델은 제품정보 모델링언어(M2)를 이용하여 정의된다.

제품 실측 데이터(M0)는 제품 설계모델에 따라 만들어진 제품의 성능을 측정된 데이터이다. 제품 수명주기가 긴 제품의 경우 제품의 성능 측정값이 시간에 따라 변할 수 있다. OWL로 표현된 제품 설계모델(M1)은 각각의 측정값들(M0)이 제품 설

계모델의 허용치에 부합하는지 여부를 추론을 통해 확인할 수 있다. 만약 측정값들이 설계 모델에 부합하지 않을 경우, 적절한 조치를 취하기 위한 규칙들을 별도로 정의할 필요가 있다. 이러한 규칙을 정의하는데 설계모델에 정의된 개념과 관계들이 활용된다.

제품정보 모델링언어(M2)는 엔지니어에게 제공되는 제품 온톨로지인 반면에, 제품 설계모델(M1)과 실측 데이터(M0)은 엔지니어가 관심있는 제품의 정보를 직접 정의한 것이다. 엔지니어가 두 층의 정보들을 정의하기 위해 OWL과 같은 온톨로지 언어까지 이해하는 것은 엔지니어에게 큰 부담이기 때문에, 엔지니어에게 편리한 정보 정의 인터페이스 개발이 필요하다. 또한, 엔지니어가 인터페이스를 통해 정의한 제품 모델과 실측 정보들을 OWL 언어로 변환하여 시스템에 저장하기 위해 정보 변환 규칙들도 개발이 필요하다. 5장에서 엔지니어가 다층 정보모델링 프레임워크를 사용하기 위한 인터페이스와 정보 변환 규칙들을 어떻게 설계해야 하는가에 대하여 논의한다.

4. 제품정보 모델링언어(M2)

제품정보 모델링언어 층은 OWL 언어와 제품 온톨로지로 구성된다. 본 연구는 제품 온톨로지 SPMM^[5]을 활용한다. SPMM은 제품의 요구사항 분석, 기능 및 형상 설계, 설계성능 평가 정보를 모델링하기 위한 개념과 관계들을 정의하고 있다. 본 장에서는 SPMM에서 정의한 개념과 관계들을 요약하여 소개한다.

owl:Class, owl:ObjectProperty, owl:Datatype Property들은 OWL 구성 요소인데, SPMM은 이

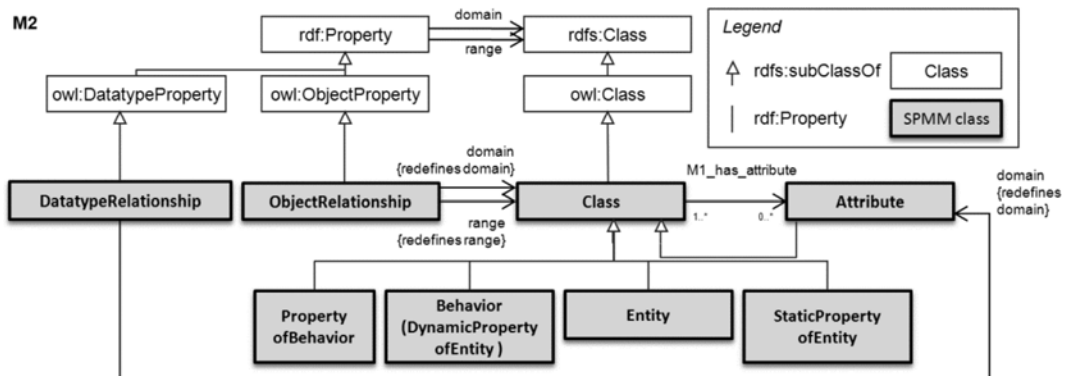


Fig. 2 Relationships between OWL classes and SPMM classes

요소들로부터 제품 개발 영역에 특화시킨 클래스들을 정의하고 있다. Fig. 2는 OWL 클래스들과 제품 온톨로지의 상위 개념들 간의 관계를 표현하고 있다. 본 논문에서는 OWL 클래스와 관계들을 도식적으로 표현하기 위하여 UML^[14] 표기 방법을 활용하였다.

‘개체’(Entity) 클래스는 엔지니어가 표현하고자 하는 제품 및 제품 사용 환경을 포함하여 식별자를 가진 것으로 정적 특성과 동적 특성들을 가진다. ‘개체 정적 특성’(StaticPropertyofEntity) 클래스는 시간에 따라 변하지 않는 특성으로 형상 정보나 재질 정보와 같은 특성을 포함한다. ‘개체 동적 특성’(DynamicPropertyofEntity) 클래스는 시간에 따라 변하는 특성으로 개체의 행위(Behavior)와 같은 의미를 갖는다. 예를 들어, 모터라는 개체는 ‘회전한다’, ‘정지한다’와 같은 행위를 갖는다. ‘행위특성’(PropertyOfBehavior) 클래스는 행위들 간의 관계를 정의하기 위해 특별히 정의된 클래스이다.

SPMM의 주요 클래스들은 각각 하위 클래스들이 존재하고, 서로 간에 관계들을 맺고 있다. 그리고, 클래스와 관계들에 공리들이 정의되어 있다. Fig. 3은 SPMM 주요 클래스들의 하위 클래스들과 관계를 UML 클래스 다이어그램으로 표현한 것이다.

‘개체’ 클래스는 ‘외부개체’(ExternalEntity), ‘설

계대상개체’(SpecifiedEntity), ‘특징’(Feature) 클래스로 구분된다. ‘설계대상개체’ 클래스는 엔지니어의 설계 대상들을 대표하고, ‘외부개체’ 클래스는 그 설계 대상과 상호작용을 하는 환경요소들을 대표한다. 예를 들어, 항공기 착륙기어 시스템이 설계대상이라면, 착륙기어 시스템과 상호작용을 하는 활주로가 환경요소이다. ‘특징’ 클래스는 개체의 일부 특성을 식별하기 위한 개체다. 예를 들어, 착륙기어 시스템에 피스톤과 실린더 부품이 있을 때 실린더의 구멍은 하나의 ‘특징’ 클래스이다. 실린더 구멍 크기가 전체 시스템의 성능에 영향을 주기 때문에 엔지니어가 특별히 식별 관리하고자 하기 때문이다. 설계 대상이 되는 부품과 그것의 특징을 구분하기 위하여 ‘설계대상개체’ 클래스는 ‘인공물’(Artifact)과 ‘규정된 특징’(SpecifiedFeature) 클래스를 하위 클래스로 갖는다.

‘모양’(Form) 클래스는 ‘형상’(Geometry) 클래스와 ‘재료’(Material) 클래스로 구분된다. ‘형상’ 클래스는 개체에 대한 점, 선, 면, 공간체에 대한 정보를 표현한다. ‘재료’ 클래스는 개체를 구성하는 재료에 대한 정보를 표현한다. ‘모양’ 클래스는 설계하기 전 요구되는 형상과 재료 정보를 표현할 수도 있고, 설계 후 형상과 재료에 대한 정보를 표현할 수도 있다. 예를 들어, 착륙기어를 수납하는 장소의 크기 제약으로 인해 특정 길이를 넘지 않는 착륙기어를 설계해야 한다면, 수납장소를 표현

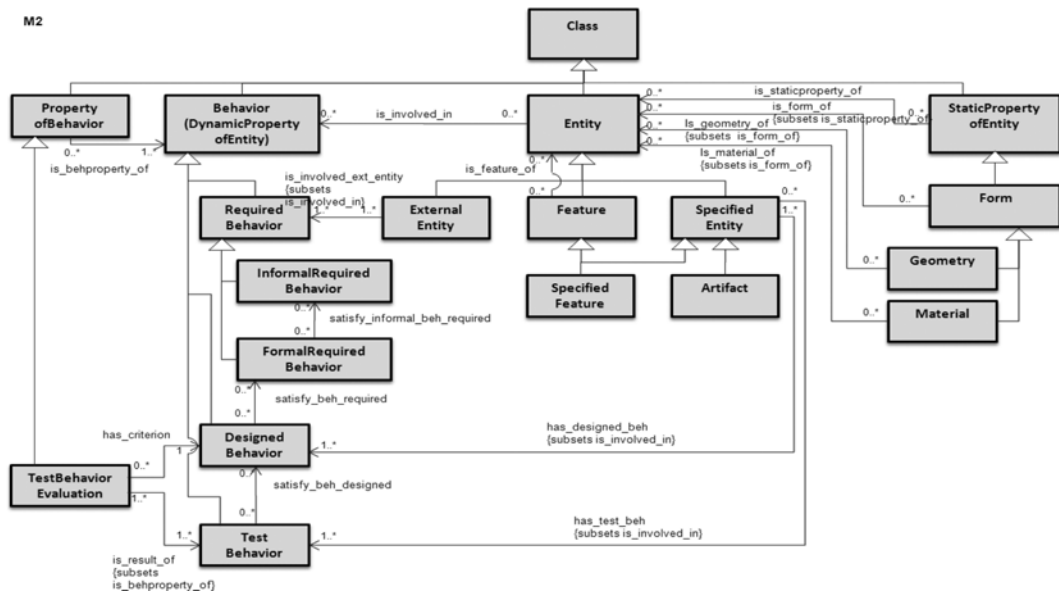


Fig. 3 SPMM class diagram

한 ‘형상’ 클래스는 요구되는 특정 크기 값들을 정의한다. 또한, 착륙기어 설계의 ‘형상’ 클래스는 특정 높이보다 낮아야 한다는 제약 정보가 정의된다. 이처럼 설계 환경요소와 설계대상의 정보들이 정의되고, 상호간의 관계가 명시되면, 나중에 설계 환경요소가 변화했을 때, 어떤 설계 요구사항들을 바꿔야 하는지, 그리고 그에 따라 어떤 설계 변경이 가능한지 추적할 수 있게 된다.

‘행위’(Behavior) 클래스는 개체들 간의 상호작용을 표현한다. ‘행위’ 클래스는 크게 세 가지로 분류되는데, 설계 환경요소와 설계대상 간의 관계를 명시하는 ‘요구된 행위’(RequiredBehavior) 클래스가 있으며, 설계 대상의 부품들 또는 특징 간의 관계를 명시하는 ‘설계된 행위’(DesignedBehavior) 클래스가 존재한다. 그리고, 설계모델과 만들어진 제품의 실측데이터 간의 관계를 명시하는 ‘실험행위’(TestBehavior) 클래스가 있다.

‘요구된 행위’ 클래스는 외부 환경 중심에서 외부 환경 요소들이 설계 대상과 갖는 상호작용들을 표현한 클래스이다. ‘요구된 행위’ 클래스는 설계대상에 대한 정보를 제공하지 않고, 오직 외부 환경 요소들이 설계대상에 요구하는 행위들을 기술한다. 예를 들어, 항공기가 착륙기어를 사용하는 용도인 ‘이륙하다,’ ‘착륙하다,’ ‘주행하다,’ ‘정지하다’ 등이 요구된 행위로 정의된다. 이때 어느 정도의 하중이 착륙기어에 주어지는지, 어느 정도의 충격이 가해지는지 등 각 요구된 행위들의 수치적 특성들이 추가적으로 클래스에 정의된다.

‘설계된 행위’ 클래스는 설계 대상 중심에서 다른 개체들에게 어떤 상호작용들을 제공할 수 있는가를 표현한 클래스이다. ‘설계된 행위’ 클래스 정보는 설계 대상의 형상과 재료 정보에 따라 바뀔 수도 있고, 반대로 ‘설계된 행위’ 클래스 정보에 따라 형상과 재료정보가 바뀔 수도 있다. 예를 들어, 엔지니어가 착륙기어를 설계할 때 이착륙 충격량을 견딜 수 있도록 부품 형상과 재질을 설계할 것이다. 그리고, 필요에 따라 형상과 재질이 변경된다면, 설계 변경에 따라 착륙기어가 견딜 수 있는 충격량이 달라질 것이다.

엔지니어가 설계한 제품의 형상과 재료에 따라 제품을 만들었을 때, 그 제품이 ‘설계된 행위’ 클래스의 정보와 다르게 작동할 수도 있다. ‘실험행위’(TestBehavior) 클래스는 ‘설계된 행위’ 클래스

정보와 실측 행위 정보(M0)를 비교하기 위해 실측 행위 정보들을 대표하는 클래스이다. ‘실험행위’ 클래스는 실측 데이터를 수집할 때 어떤 정보들을 어떻게 수집해야 하는지를 정의한다.

‘설계된 행위’ 클래스 정보와 실측 행위 정보를 비교한 후, 그 결과를 표현하기 위하여 ‘실험행위 평가’(TestBehaviorEvaluation) 클래스를 정의하였다. 이 클래스는 ‘설계된 행위’ 클래스와 ‘실험행위’ 클래스 양쪽과 모두 관계를 가지고 있음으로써, 실측 데이터를 검토할 때 어떤 ‘설계된 행위’ 클래스 정보를 기준으로 실측 데이터를 검토해야 하는지 알려준다.

5. 설계모델(M1) 및 실측데이터(M0) 개발

본 장에서는 앞서 정의한 SPMM 클래스들을 활용하여 엔지니어가 제품 설계정보를 어떻게 모델링하며(M2-M1 상호작용), 설계 모델들에 의미를 어떻게 추가 할 수 있는지(M1-M1 상호작용), 마지막으로 수명주기가 긴 제품들의 실측 데이터를 설계 정보와 어떻게 비교할 수 있는지(M1-M0 상호작용) 설명한다.

5.1 제품 설계모델 개발(M2-M1 상호작용)

엔지니어가 SPMM 클래스들을 이용하여 설계모델(M1)을 개발하기 위해서는 엔지니어에게 SPMM에 특화된 설계모델링 시스템을 지원해야 한다. 이 시스템은 엔지니어가 SPMM 클래스의 의미를 완벽히 알지 못하더라도 각 클래스들의 의미에 맞게 설계모델을 개발할 수 있도록 도와주는 기능을 제공해야 한다.

이러한 설계모델링 지원 기능은 SPMM 클래스에 정의된 공리를 이용하여 구현이 가능하다. 예를 들어, SPMM의 ‘외부개체’ 클래스는 외부 개체가 존재하기 위해서는 반드시 하나 이상의 ‘요구된 행위’ 클래스와 관계가 존재해야 한다는 공리를 가지고 있다. 이러한 공리들은 시스템이 설계모델을 검사할 때 빠진 정보가 없는지 확인할 수 있는 규칙들이 된다. 또한, 엔지니어가 ‘외부개체’를 정의하였을 때, 시스템은 반드시 이와 관계를 갖는 ‘요구된 행위’를 정의하도록 엔지니어를 가이드 할 수도 있다.

엔지니어는 시멘틱 네트워크^[15]와 같은 직관적

인 문법을 이용하여 설계모델을 표현한다. 시멘틱 네트워크는 개념과 관계를 각각 노드와 아크로 표현한 것으로 지식 표현에 적합한 방법이다. 엔지니어는 SPMM에서 정의한 SPMM 클래스와 관계들을 이용하여 설계모델의 각 노드와 아크 타입을 정의할 수 있다.

Fig. 4는 엔지니어가 설계모델을 정의하는 예제와 엔지니어의 입력에 대해 개발환경 시스템이 반응해야 하는 모습을 도식적으로 표현하고 있다. 그림에서 엔지니어가 ‘인공물’(Artifact) 타입의 착륙기어(LandingGear) 클래스를 정의하였을 때, 시스템은 SPMM 공리 정보를 바탕으로 어떤 SPMM 클래스들이 추가로 정의되어야 하는지 엔지니어에게 보여주어 올바른 제품 모델을 만들 수 있도록 도와준다.

5.2 제품 모델에 의미 추가(M1-M1 상호작용)

설계모델링 시스템은 엔지니어가 정의한 설계모델을 OWL표현으로 변환하는 기능을 제공해야 한다. OWL로 변환된 정보모델은 지식추론과 시스템간 정보교환에 활용될 수 있기 때문이다.

OWL은 시멘틱 네트워크로 표현된 모든 지식들을 표현할 수 있다. OWL의 개발 역사에 시멘틱 네트워크도 포함되기 때문이다. 하지만, 시멘틱 네트워크의 표현력이 OWL에 비해 부족하기 때문에, 엔지니어가 시멘틱 네트워크로 모든 지식을 표현하지 못하는 경우가 있다. 예를 들면, 엔지니어가 설계하는 착륙기어는 네 개의 바퀴로 구성된다는 지식은 OWL에서 쉽게 표현이 되지만, 시멘틱 네트워크로는 표현이 복잡해진다. 또한, 바퀴의 유효수명은 3년 미만이거나 착륙회수가 1000회 미만이다라는 논리적 표현이 필요한 지식도 시

멘틱 네트워크로는 표현이 복잡해진다.

설계모델링 시스템은 엔지니어의 복잡한 지식 입력을 돕기 위해서 지식 입력들을 제공할 필요가 있다. 지식 입력틀이란 엔지니어가 입력하고자 하는 지식의 특징을 사전에 분석하여 만들어 낸 입력 양식이다. 각 입력 양식마다 엔지니어가 그 양식을 채우면 OWL 표현으로 변환하는 규칙들이 필요하다. 지식 입력틀 접근방법은 과거 Lee^[16]와 Liang^[17] 연구들에서 XSLT와 Java를 이용하여 제안하였으며, 최근에는 ISO-15926 표준^[18]에서 지식 입력틀 접근방법을 표준의 일부로 정의하였다.

Fig. 5는 착륙기어의 바퀴 수를 정의하는 지식 입력틀과 입력결과를 OWL로 변환하는 규칙 예제를 보여준다. OWL로 변환하는 규칙은 예제처럼 일반적인 규칙들도 있지만, 설계모델에 따라 복잡한 규칙을 정의해야 할 수도 있다.

5.3 설계모델과 실측데이터 비교(M1-M0 상호작용)

수명주기가 긴 제품의 경우 설계모델에 따라 제

```

Knowledge Template #01
<X:Artifact> must have <N:number> <Y:sub-Artifact>.

Rule for the template #01
IF X, N, Y exist in the template #01,
THEN define the following OWL sentence.
<SubClassOf>
  <spmm:Artifact IRI="#X"/>
  <ObjectExactCardinality cardinality="N">
    <ObjectProperty IRI="spmm:has_subArtifact"/>
    <spmm:Artifact IRI="#Y"/>
  </ObjectExactCardinality>
</SubClassOf>
    
```

Fig. 5 An example of knowledge template and OWL conversion rule

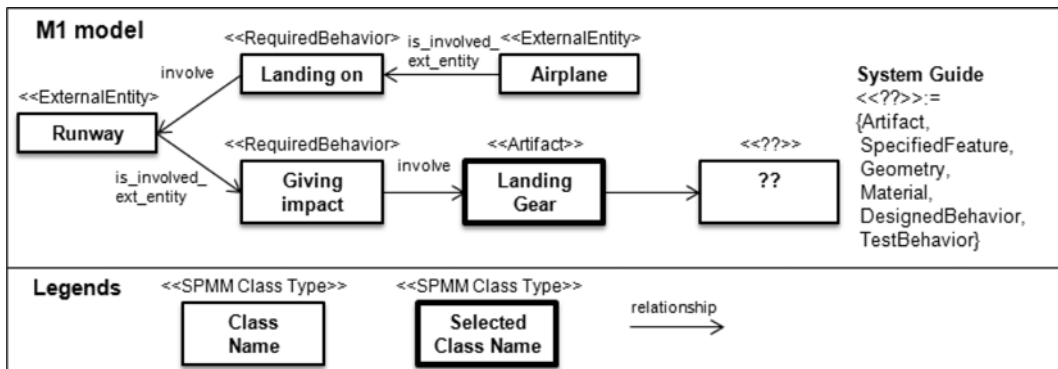


Fig. 4 An example of design information modeling guide

품을 만들었더라도 시간이 지남에 따라 유지보수를 하게 되고, 유지보수에 따른 실측데이터들이 수집된다. 수집된 실측데이터들을 통해 제품의 성능 변화를 측정하고, 제 기능을 유지할 수 있도록 관리를 받게 된다. 모든 부품이 제품의 수명주기와 같지 않기 때문에, 유지보수 중 부품을 교체하게 되는데 부품 교체정보 관리 또한 유지보수 정보관리의 중요한 데이터가 된다.

본 논문에서 제안하는 다층 정보모델링 프레임워크는 제품의 실측데이터를 추가할 때 각 제품의 설계모델과 비교가 가능할 수 있도록, 서로 다른 층의 제품 정보 관계를 명시적으로 표현한다는 장점이 있다. 앞서 5.2장에서 설명한 설계모델의 OWL 표현은 설계모델을 OWL클래스로 표현하고, 실측데이터들은 이들의 개체(individuals)로 표현한다. OWL 추론은 OWL 클래스에 정의된 공리를 바탕으로 개체들이 이를 만족하는지 검증하게 해준다.

설계모델과 실측데이터를 비교할 때, 단순히 실측 값 하나를 클래스 공리와 비교하는 것은 일반 데이터베이스 어플리케이션도 할 수 있는 작업이다. 다층 정보모델링 프레임워크는 실측 값들의 모음이 어떠한 경향이 있을 때나, 비교하고자 하는 여러 속성들 간의 관계를 고려한 의사 결정을 해야 할 때, 논리적 규칙을 정의할 수 있는 기반을 제공한다. 예를 들어, 어떤 비행기의 착륙기어의 타이어를 사용한 기간이 3년을 넘지 않았지만, 교체하지 않은 채 1000회 이상 이착륙을 하였다면, 5.2장에서 예로 들었던 타이어 클래스의 수명주기 공리에 따라 그 해당 타이어를 교체해야 함을 자동으로 검토할 수 있다.

6. 프로토타입 구현

본 장에서는 다층 정보모델링 프레임워크를 구현한 프로토타입 시스템을 소개한다. 이 시스템은 엔지니어가 제품 설계정보를 SPMM 클래스들을 이용하여 모델링 할 수 있는 환경을 제공한다. 그리고, 각 설계모델들에 해당하는 실측데이터들도 함께 볼 수 있는 환경을 제공한다. 엔지니어가 정의한 설계모델들은 OWL로 변환됨으로써 타 시스템과 설계모델 정보 공유가 가능하다.

Fig. 6은 구현한 프로토타입의 모듈 아키텍처를 보여주고 있다. 관계형 데이터베이스로 MS-Access를 이용하였고, OWL 파일들은 사전에 정의된 폴더에 저장된다. 각 모듈들은 Microsoft .Net 프레임워크에서 구동되며, C# 언어를 이용하여 모듈들과 사용자 환경은 개발하였다. 외부 컴포넌트는 크게 두 가지를 사용하였는데, OWL 파일을 만드는 .NET OWL API 컴포넌트가 활용되었고, 사용자 화면의 그래픽 모델링을 위해서 Addflow.NET 컴포넌트가 활용되었다.

프로토타입은 제품 설계 도메인의 전문가가 사전에 정의한 SPMM OWL 파일과, 설계용어 온톨로지, 지식 입력 틀과 각 입력 틀의 OWL 변환 규칙들이 사전에 정의되어 제공되어야 한다. 엔지니어는 프로토타입 시스템에서 제공된 SPMM 클래스와 온톨로지, 지식 입력 틀을 이용하여 설계모델을 개발하고, 실측 데이터를 입력하게 된다.

실측데이터를 저장하는 DB의 스키마는 엔지니어가 정의한 설계모델에 따라 스키마가 바뀌어져야 한다. 실측데이터 DB 관리 모듈은 설계모델에 따라 실측 데이터 DB 스키마를 수정하는 역할을

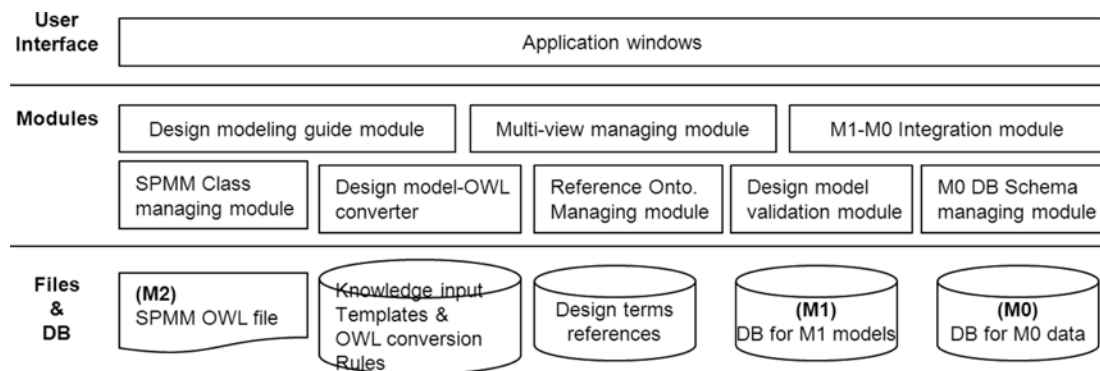


Fig. 6 Prototype system architecture

한다. 설계모델 유효성 관리 모듈은 엔지니어가 만든 설계모델이 SPMM에 정의한 공리를 만족하는지 확인하고 그 결과를 알려준다. SPMM 클래스 관리 모듈, 설계모델-OWL 변환 모듈, 참조 온톨로지 관리 모듈은 각 정보들을 추가, 수정, 삭제, 보기 등을 할 수 있는 기능을 제공한다. 설계 모델링 가이드 모듈은 엔지니어가 입력한 정보를 SPMM의 공리들과 비교하여 엔지니어가 다음에 해야 할 작업을 알려준다. 예를 들어 엔지니어가 특정 클래스를 선택하였을 때, 어떤 관계들을 추가할 수 있는지 화면에 색다르게 표시하는 기능을 수행한다. 설계모델 다중관점 관리모듈은 엔지니어가 하나의 제품에 대하여 서로 다른 관점에서 설계 모델들을 만들었을 때, 각 모델들의 정보를 하나로 통합하여 보여주는 기능을 한다. M1-M0 레벨 통합관리 모듈은 엔지니어가 M1 설계모델을 수정하거나, M0 실측데이터를 수정하였을 때, 사용자 화면에 수정이 필요한 정보들을 갱신하는 기능을 한다.

프로토타입 구현 예제를 이용하면 프로토타입의 모듈들의 기능에 대하여 바탕으로 좀 더 상세히 설명할 수 있다. Fig. 7은 구현한 프로토타입 화면 예제를 보여주고 있다. Fig. 7의 좌측 화면 내 우측의 트리 구조(SPMM 탭)는 SPMM 클래스와 관계들을 보여준다(SPMM Class managing module). 설계자는 이 트리 노드를 끌어다가 중앙 모델링 화면에 놓음으로써 설계모델의 클래스와 관계들을 정의하게 된다. 좌측 상단의 트리 구조는 엔지

니어가 만든 클래스와 관계들을 보여주며, 부품의 구조 관계, 속성의 상속 관계 등을 보여주기 위한 탭들이 존재한다.

엔지니어는 중앙의 모델링 화면 탭을 필요에 따라 여러 개 만들 수 있는데, 각 모델링 탭에서 정의된 클래스와 관계들은 좌측 트리 구조에 모두 열거되고 재활용될 수 있다(Multi-view managing module).

엔지니어가 화면의 가운데에서 설계 모델을 만들 때 각 노드를 선택하면, 물음표 표시가 나오는데, 이를 클릭하면 엔지니어가 다음에 해야 할 작업이 무엇인지 메뉴를 보여준다(Design modeling guide module, Design model validation module).

가운데 하단의 여러 탭들은 실측 데이터를 입력할 수 있는 테이블들을 가지고 있다. 각 테이블은 설계모델의 클래스마다 하나씩 가질 수 있고, 테이블의 열들은 설계 모델의 클래스 속성이 정의됨에 따라 동적으로 바뀌게 된다(M1-M0 integration module, M0 Schema managing module).

Fig. 7의 우측 화면은 엔지니어가 정의한 설계모델 클래스의 속성을 정의하는 화면이다. 엔지니어는 필요에 따라 각 설계모델 클래스에 원하는 만큼의 속성들을 정의할 수 있다. 각 클래스의 속성 이름은 앞서 설명한 실측 데이터 테이블의 열 이름이 된다.

Fig. 7의 우측 화면과 같은 정보모델링 화면은 엔지니어가 시멘틱 네트워크 형태로 정의하기 어

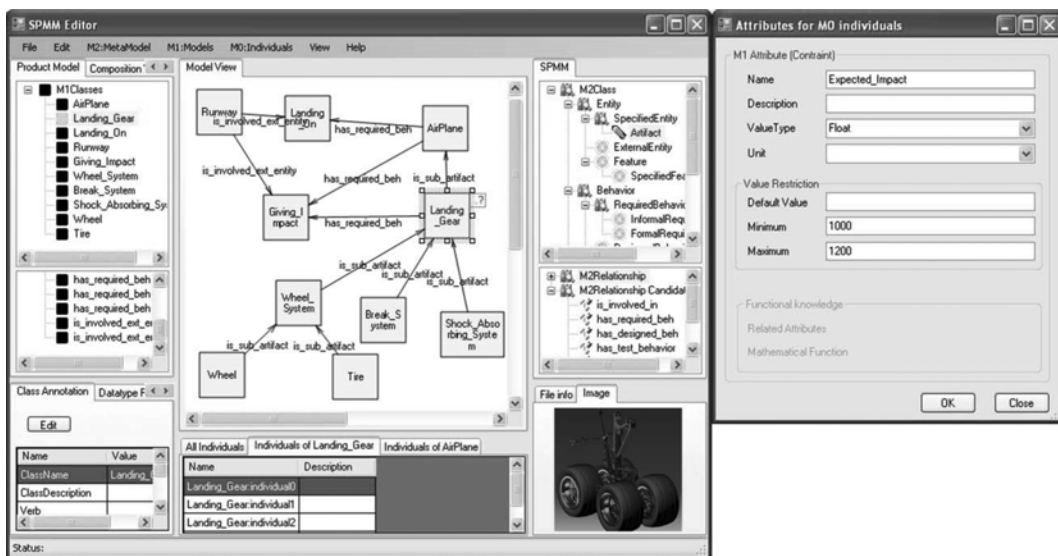


Fig. 7 A prototype screen-shot

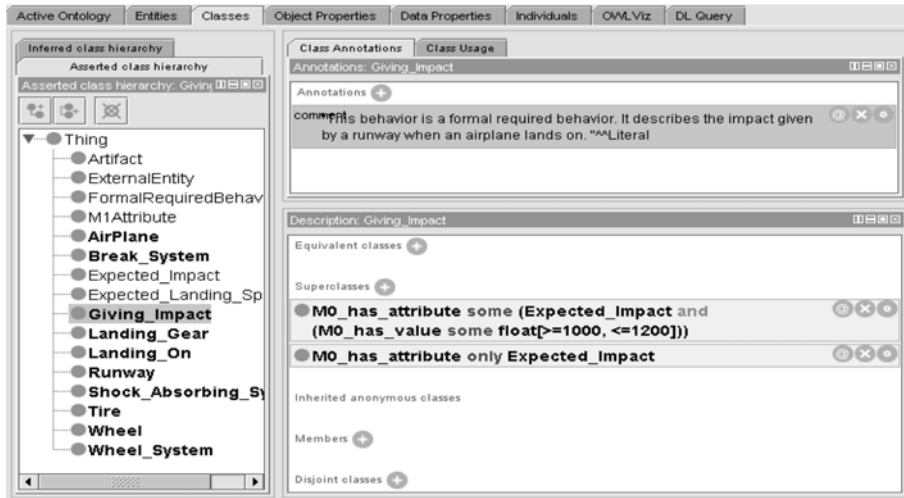


Fig. 8 OWL representation in Protege for the SPMM design model example

려운 정보들을 정의할 수 있는 정보 입력틀이 된다. 이렇게 입력된 설계모델 정보는 사전에 정의된 OWL 변환 규칙(OWL conversion rules)에 따라 OWL 표현으로 변환된다(Design model - OWL converter).

프로토타입 시스템은 OWL 변환에 필요한 규칙들 중 일부를 사전에 정의하여 C# class로 구현하였다. OWL 변환 규칙을 엔지니어가 정의하기 위한 사용자 화면은 추후 개발 과제이다.

Fig. 8은 프로토타입 시스템에서 정의한 설계모델을 OWL로 변환한 후, 변환된 OWL 파일을 OWL 편집기인 Protege(버전 4.0.2)에서 읽은 화면이다. OWL로 표현된 'Giving_Impact' 클래스의 공리를 보면 'Expected_Impact' 속성을 하나 가져야만 하고, 그 값이 1000과 1200 사이의 값이어야 함을 정의하고 있다. 이 공리는 엔지니어가 'Giving_Impact' 클래스의 속성과 속성값을 Fig. 7의 화면에서 정의하였을 때, 사전에 정의한 OWL 변환 규칙에 따라 만들어진 것이다.

제조 기업에 보편적인 제품 설계 모델링 시스템은 CAD와 PDM이다. CAD 시스템은 제품의 형상정보를 모델링 하는데 활용되고, PDM 시스템은 CAD 파일 관리뿐만 아니라 제품 개발에 관련된 정보들을 관리한다. CAD와 PDM 시스템은 제품에 따라 크게 변하지 않는 정형화된 정보들을 관리하는데 유용하다. 하지만, 이전 제품에 없는 새로운 정보를 정의하기 위해서는 별도의 파일을 만들어 관리하거나 데이터베이스의 스키마를 변

경하여야 한다.

본 논문에서 제안하는 프로토타입은 엔지니어의 필요에 따라 추가적인 설계 정보 정의가 가능하며, 이를 위해 데이터베이스 변경과 같은 큰 노력이 불필요하고, SPMM 클래스를 이용한 구조적인 제품정보 관리가 가능하다는 장점이 있다. 따라서, 이 프로토타입 시스템은 형상정보 이외의 부가적 제품설계정보들을 관리하는 CAD나 PDM의 추가 모듈로써 활용되거나 또는 독립적인 프로그램으로써 활용될 수 있다.

7. 결 론

다층 제품정보 모델링 프레임워크는 설계모델과 실측데이터 간의 관계를 논리적으로 관리할 수 있는 장점을 갖는다. 그리고, 이는 설계모델과 유지보수에 따른 실측 데이터간의 관계 관리가 중요한 수명주기가 긴 제품의 경우 유용하게 활용될 수 있다.

본 논문은 다층 모델링 프레임워크에서 엔지니어가 설계모델을 개발하기 위한 환경 요구사항을 정의하고, 설계모델과 실측데이터 간의 관계 관리 방법을 제안하였다. 그리고, SPMM에서 제안한 메타 모델을 이용하여 엔지니어가 설계모델을 개발할 수 있는 방법을 제안하였고, 시스템 프로토타입을 구현하여 그 활용 가능성을 보였다.

SPMM은 제품의 요구사항, 기능, 구조, 행위 등 일반적인 설계 개념의 클래스들을 제공한다. 제안

한 시스템 프로토타입이 특정 도메인에서 엔지니어가 유용하게 사용하기 위해서는 도메인에 특화된 참조 클래스 라이브러리와 온톨로지 제공이 필수적이다.

제품정보 모델링 관점에서 향후 연구문제는 엔지니어가 정의한 설계모델이 추가됨에 따라 OWL로 변환하는 규칙의 일반화와 다양성을 확보하는 것이고, M1와 M0 층의 정보간 일관성을 유지하기 위한 규칙 개발이 필요하다. 향후 시스템 개발 방향은 제안한 프로토타입 시스템을 CAD 또는 PDM 시스템과 직접 연동하도록 구현하고, 현장 응용사례와 객관적 성능 검증을 받아 접근방법의 실효성을 인정 받을 필요가 있다.

감사의 글

이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 [NRF-2009-352-D00324].

참고문헌

- Lubell, J., Kassel, B. and Rachuri, S., 2009, Descriptive Metadata Requirements for Long-term Archival of Digital Product Models, *Proceedings of the Indo-US Workshop on International Trends in Digital Preservation*, Pune, India, pp. 39-45.
- Lubell, J., Mani, M., Subrahmanian, E. and Rachuri, S., 2007, *Long Term Knowledge Retention Workshop Summary*, NIST interagency/internal report (NISTIR)-7386, National Institute of Standards and Technology, Gaithersburg, USA.
- Lubell, J., Mani, M., Subrahmanian, E. and Rachuri, S., 2008, *Long Term Sustainment Workshop Report*, NIST interagency/internal report (NISTIR)-7496, National Institute of Standards and Technology, Gaithersburg, USA.
- McGuinness, D.L. and Harmelen, F.V., 2004, *OWL Web Ontology Language Overview W3C Recommendation 10 February 2004*, <http://www.w3.org/TR/owl-features/>.
- Lee, J.H., Fenves, S.J., Bock, C., Suh, H.-W., Rachuri, S., Fiorentini, X. and Sriram, R.D., 2012, A Semantic Product Modeling Framework and Its Application to Behavior Evaluation, *IEEE Transactions on Automation Science and Engineering*, 9(1), pp. 110-123.
- Guarino, N., 1998, Formal Ontology and Information Systems, *Proceedings of FOIS'98*, Trento, Italy, pp. 3-15.
- Pease, A., Niles, I. and Li, J., 2002, The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and Its Applications, *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, Edmonton, Canada.
- OMG, Model-Driven Architecture, <http://www.omg.org/mda>.
- Lee, J., Chae, H., Kim, C.H. and Kim, K., 2008, Design of Product Ontology Architecture for Collaborative Enterprises, *Expert Systems with Applications*, 36(2), pp. 2300-2309.
- Fenves, S.J., Fofou, S., Bock, C., Rachuri, S., Bouillon, N. and Sriram, R.D., 2005, *CPM 2: A Revised Core Product Model for Representing Design Information*, NIST interagency/internal report (NISTIR)-7185, National Institute of Standards and Technology, Gaithersburg, MD, USA.
- Bock, C., Zha, X.F., Suh, H.W. and Lee, J.H., 2009, *Ontological Product Modeling for Collaborative Design*, NIST Interagency/Internal Report (NISTIR)-7643, National Institute of Standards and Technology, Gaithersburg, MD, USA.
- ISO 15926-1, 2004, *Industrial Automation Systems and Integration—Integration of Life-cycle Data for Process Plants Including Oil and Gas Production Facilities—Part 1: Introduction*, International Organization for Standardization (ISO), Geneva, Switzerland.
- Moon, D.H., Kim, B.C. and Han, S.H., 2009, Representing Process Plant Equipments Using Ontology and ISO 15926, *Transactions of the Society of CAD/CAM Engineers*, 14(1), pp. 1-9.
- OMG, Unified Modeling Language Specification v1.4., <http://www.omg.org/cgi-bin/doc?formal/01-09-67>.
- Deliyanni, A. and Kowalski, R.A., 1979, Logic and Semantic Networks, *Communications of the ACM*, 22(3), pp. 184-192.
- Lee, J.H. and Suh, H.W., 2007, OWL-based Product Ontology (POWL) Architecture and Representation for Sharing Product Knowledge on a Web, *ASME 2007 International Design Engineering Technical Conferences Computers and Information in Engineering Conference (IDETC/CIE)*, Las Vegas, NV, USA.
- Liang, V.C., Bock, C. and Zha, X.F., 2008, *An Ontological Modeling Platform*, NIST Interagency/Internal Report (NISTIR)-7509, National Institute of Standards and Technology, Gaithersburg, MD, USA.

18. ISO 15926-7, 2011, *Industrial Automation Systems and Integration—Integration of Life-cycle Data for Process Plants Including Oil and Gas*

Production Facilities—Part 7: Template Methodology, International Organization for Standardization (ISO), Geneva, Switzerland, 2011.



Jae Hyun Lee

received the M.S. and Ph.D. degrees in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 2001 and 2008, respectively. He has been an Associate Researcher since 2008 in the Systems Integration Division, Engineering Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD. He has developed product information models for sustainable product design and manufacturing. Currently, he is involved in developing material information standards to support sustainable design and manufacturing.



Hyo-Won Suh

received the B.S. degree in mechanical engineering from Yonsei University, Seoul, Korea, in 1981, the M.S. degree in mechanical engineering from the Korea Institute of Science and Technology (KAIST), Seoul, in 1983, and the Ph.D. degree in industrial engineering from West Virginia University, Morgantown, in 1991. Currently, he is a Professor with the Department of Industrial Engineering, KAIST, Daejeon, Korea. His research interests are product lifecycle management (PLM), ontology application to engineering and he has published several papers in these areas.
