

# 온톨로지 기반 설계 프로세스 모델링에 관한 연구

김정기\*(성균관대 대학원 기계공학과), 강무진(성균관대 기계공학부)

## A Study on the Ontology-based Design Process Modeling

J. K. Kim(Graduated School, SKKU), M. Kang(School of Mechanical Eng., SKKU)

### ABSTRACT

Design process model represents how a design project proceeds. It encompasses the individual activities of design, their precedence relationships, and the relevant information related to each activity. In contrast to the conventional visual representation methods, ontology-based process model is machine-readable, and therefore it can be implemented in a software system without repeating the whole steps of coding, compiling and link. This paper proposes a framework for design process ontology that defines the relevant objects and attributes in the design process as well as the relationships between them. An example for injection mold design process is shown to explain the substance of the design process model.

**Key Words** : Design process(설계 프로세스), Ontology(온톨로지), Process ontology(프로세스 온톨로지)

### 1. 서론

현대의 제품 설계 과정은 복잡해지고 있어 시간이 많이 소요되고 비용도 늘어나게 된다. 설계 프로세스 모델의 이용은 설계 실행에 앞서 각 설계 업무 사이의 관계를 설계자에게 제시함으로써 설계 업무의 반복을 방지하고 설계 업무의 효율적 진행을 가능하게 한다.<sup>1</sup> 하지만 설계 프로세스는 설계자의 머리 속에 남아 있는 경우가 많아 관리하기 어렵고 검증하기도 어렵다. 설계 프로세스를 표현하는 데에는 일반적으로 도식적 표현 방법들이 많이 사용되어 왔고, 이러한 표현은 프로세스의 전체 구조를 시각적, 직관적으로 표현하는 데에 편리한 장점을 가지고 있다. 그러나 도식적으로 표현된 프로세스 모델은 기계가 해석하지 못하므로 새로운 프로세스 기반의 시스템 구현을 위해서는 컴퓨터 언어로 다시 구현하는 작업이 필요하게 된다. 이것은 프로세스가 변경될 때마다, 시스템을 새로 수정해야 하는 문제점을 유발한다. 반면, 온톨로지 기반 설계 프로세스 모델은 기계가 해석 할 수 있어 추론을 통해 제시된 프로세스 모델에 따라 워크플로우(Workflow)를 시스템에 구현할 수 있다.

본 논문에서는 OWL 과 Protégé 를 이용한 설계 프로세스 온톨로지의 모델링 예를 통하여, 설계 프

로세스 및 설계 프로세스의 구성 요소를 정의하고 각 요소들간의 관계를 명시적으로 표현함으로써 설계 프로세스 온톨로지가 설계 시스템의 작업 흐름 구현에 이용될 수 있음을 보이고자 한다.

### 2. 설계 프로세스 모델링

설계 프로세스는 설계 과정에 일어나는 업무의 흐름을 표현하는 것으로 활동, 활동들 간의 시간적 선·후 행 관계, 각 활동의 입력, 출력, 활용자원, 제한 등의 정보를 표현한다. Fig.1 은 사출 금형 설계

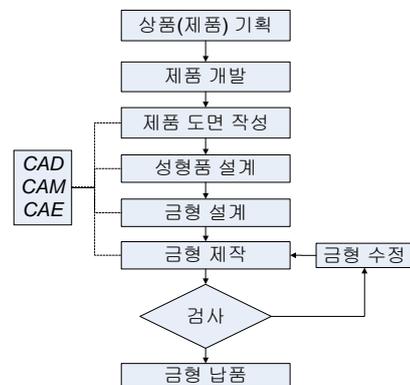


Fig. 1 Typical injection mold design process

프로세스 모델 예를 보여준다.

설계 프로세스를 모델링하는 데에는 Simple representation, 네트워크 표현(Network representation), IDEF0(Integration DEFinition), DFD(Data Flow Diagram) 과 같은 도식적 표현들이 많이 사용되고 있다.

Simple representation 은 노드(Node)와 화살표(Arrow)를 포함하는 간단한 다이어그램을 사용하여 설계 프로세스를 표현한다.(Fig.2) 노드는 사각형 박스, 둥근 사각형 박스, 원 등으로 표현되고 이산적인 활동, 프로젝트 상태, 심리적인 상태 등과 같은 여러 의미를 나타낸다. 화살표는 활동 순서를 지시하거나 상태 변화 등을 의미한다. 이 표현은 프로세스 표현에 있어 유연하기 때문에 프로세스 표현이 쉽다는 장점이 있는 반면, 표현 가능한 정보가 제한적이라는 단점이 있다.

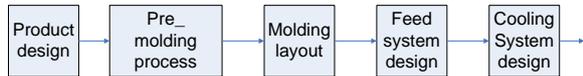


Fig. 2 Example of simple representation

네트워크 표현은 노드와 링크(Link)를 이용하여 설계 프로세스를 표현한다. 네트워크 표현에는 활동을 링크에 표시하고 활동 사이를 노드로 연결하는 Activity-on-arrow 타입과 노드에 활동을 표시하고 노드를 링크로 연결하는 Activity-on-node 타입이 있다. 네트워크 표현의 사용은 복잡한 프로젝트에서 일정 관리를 할 수 있는 반면, 프로세스에서 여러 번 나타나는 활동들은 개별 활동으로 네트워크에 표현해야 하는 단점이 있다. 이 점은 설계 활동의 주요한 특성인 반복 행위의 모델링을 명시적으로 하기 어렵다는 것을 의미한다.<sup>2</sup>

DFD 는 시스템의 활동과 활동들 간의 흐름 관계를 나타내기 위한 방법이다.(Fig.3) DFD 는 원 (Bubble) 또는 둥근 사각형으로 표현하여 입력/출력 데이터의 변환 과정을 나타내는 활동 또는 프로세스, 화살표로 표현하고 구성 요소들간의 연결 관계를 나타내는 데이터 흐름, 평행선으로 표현하고 단순한 데이터의 저장을 나타내는 데이터 저장소, 정사각형이나 직사각형으로 표시하여 프로세스의 시작 및 종료를 나타내는 외부 엔터티로 구성된다. DFD 는 기능별 분할, 하향식 전개 특징을 가지며, 사용자의 업무 및 요구사항을 표현하는데 있어 자유스럽고 제약이 별로 없어 작성하기 쉬운 장점을 가진다. 그러나 자료 흐름을 기준으로 표현하기 때문에 설계 활동에서 발생하는 정보나 이를 통제하는 제약 등을 표현하는데에는 어려움이 있다.<sup>3</sup>

IDEF0 는 박스와 화살표로 설계 프로세스에서 수행되는 활동 및 활동 간의 관계를 표현하는 방법이다.<sup>4</sup> 화살표는 활동이 수행되는데 필요한 개념들,

혹은 활동간에 관여된 개념들인 ICOM(Input, Output, Control, Mechanism) 을 표현한다. 입력(Input)은 박스의 왼쪽으로 들어가는 화살표로 표시하고 활동을 수행하는데 필요한 개체 혹은 데이터를 의미한다. 출력(Output)은 박스의 오른쪽으로 나오는 화살표로 표현되며, 활동의 결과로 산출되는 산출물을 뜻한다. 제어(Control)는 박스 위쪽에서 들어가는 형태로 표현되며 활동을 통제, 제어하는 제약조건을 의미한다. 메커니즘(Mechanism)은 박스 아래쪽에서 들어가는 형태로 표현되며, 무엇에 의해 그 활동이 수행되는가 혹은 활동의 수행에 어떠한 자원이 이용되는가를 나타내는 개체를 의미한다. IDEF0 는 설계 과정에 행해지는 업무의 흐름을 명확하게 표현하지만 모델링 규칙이 매우 엄격하고 쉽게 수정하지 못하는 단점이 있어 모델링의 범위가 매우 커지면 전체 시스템에 대한 구조를 이해하기 어려워진다.

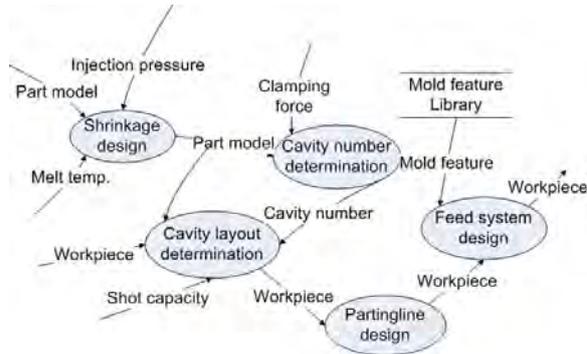


Fig. 3 Example of data flow diagram

이상의 제시한 도식적인 설계 프로세스 표현들은 프로세스의 전체 구조를 직관적이고 전체적으로 볼 수 있고, 프로세스 모델링과 관련된 사람들 간의 의사소통을 원활하게 할 수 있는 장점이 있다. 그러나 이 표현들은 기계가 해석하지 못하여 초기 설계 프로세스나 변경된 설계 프로세스에 대한 시스템 구현시 개발자가 Coding, Compiling, Link, Test 와 같은 작업을 반복해야 하는 것이 큰 문제점이다. 온톨로지를 이용한 설계 프로세스 모델링은 도식적 설계 표현의 한계를 극복하고 추론을 통하여 시스템 내의 프로세스 구현에 직접 연계할 수 있는 대안을 제시한다.

### 3. 설계 프로세스 온톨로지

온톨로지는 어떤 개념 체계에 대한 명시적 규정으로서<sup>5</sup>, 인간 또는 소프트웨어 에이전트 간에 정보의 구조에 대한 이해를 공유하기 위한 목적으로 사용된다. 프로세스 온톨로지는 일반적인 프로세스를 표현하는 개념 체계로, 프로세스와 관련된 입력, 출

력, 제한, 관계, 순서적인 정보를 표현한다. 본 논문에서 제시하는 설계 프로세스 온톨로지 프레임워크(Framework)는 설계 프로세스의 주요 개념들 간의 택소노미(Taxonomy)와 연관 관계를 포함한다.

Fig. 4 는 설계 명세(Specification), 태스크(Task), 인공물(Artifact)의 세가지 구성 요소를 이용하여 설계 프로세스의 온톨로지 구조를 모델링한 것을 보여준다. 인공물은 설계 프로세스의 실제적인 결과물로서 설계 프로세스에 *has\_a* 관계로 표현되고 태스크와는 *is\_related\_to* 관계로 표현한다.

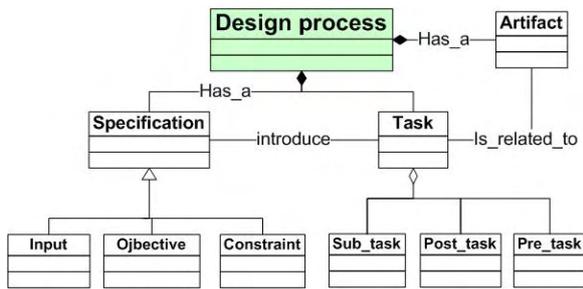


Fig. 4 Design process ontology framework

설계 명세는 사용자의 입력을 정의하고 Input, Constraint, Objective 로 분류 할 수 있다. Input 은 태스크를 수행하는데 필요한 객체나 데이터, Constraint 는 태스크를 제어하는 제약조건이나 통제하기 위해 필요한 조건, Objective 는 설계시 만족해야 하는 인공물의 특성으로 사용자가 충족되기를 바라는 것으로 정의할 수 있다. 설계 명세는 설계 프로세스와 태스크에 각각 *has\_a* 와 *introduce* 관계로 표현한다. 태스크는 설계 프로세스에서 행해지는 업무로 Pre\_task, Post\_task, Sub\_task 의 세가지 요소로 구성되고, 각각 *has\_a* 관계로 표현한다. Pre\_task 와 Post\_task 는 태스크의 선·후 관계를 표현하여 순서 정보를 나타내고, Sub\_task 는 태스크를 전개하여 상세하게 설명하기 위한 엔터티이다.

#### 4. 사례 연구 : 사출 금형 설계 프로세스

Fig. 5 는 IDEF0 로 표현된 사출 사출 금형 설계 프로세스의 한 단편을 보여준다. 캐비티 설계 프로세스는 성형품의 형상 정보인 Part Model 과 Core/Cavity Plate 를 입력(Input)으로 하고 사출 성형 조건(Injection condition), 성형품의 특성(Moldings), 성형품의 품질(Molding quality), 금형 구조(Mold structure)등을 제어 정보로 고려하며, CAD/CAE 툴을 자원으로 활용하여 Core/Cavity Plate 를 결과물로 출력한다.

Fig. 6 은 사출 금형 설계 프로세스에 대한 온톨로지 모델을 보여준다. 사출 금형 설계 프로세스에

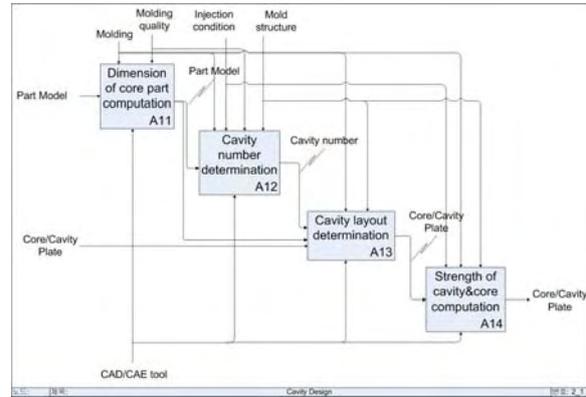


Fig. 5 Cavity design process

서 인공물은 금형이며, 설계 명세 중 Input 은 성형품 재료, 성형품 형상 모델, 성형품 중량과 같은 성형품 정보, Objective 는 주문량, 납기일, Constraint 는 사출 압, 사출 속도, 형체압(Clamping force), 타이바(Tie bar) 간격, 가스화율(Plasticating rate), 노즐 홀의 지름, 로케이팅 홀의 지름, 이젝팅 로드 홀의 지름과 같은 성형기의 사양등이다. 또한, 설계 프로세스는 캐비티 설계, 파팅 라인(Parting line) 설계, 몰드베이스(Moldbase) 선정, 유동 시스템(Feed system) 설계, 이젝터(Ejector) 설계, 벤트(Vent) 설계 등의 Sub\_task 로 구성되는 Task 를 갖는다. 캐비티 설계 Task 는 성형부 치수 결정, 캐비티 수 결정, 캐비티 레이아웃 결정, 캐비티&코어 강도 계산의 Sub\_task 를 갖고, Post\_task 는 파팅 라인 설계이다. 설계 명세로서, Input 은 성형품 재료, 성형품 형상 모델, 성형품 중량, Objective 는 주문량과 납기일, Constraint 는 사출기 사양 중 사출 속도, 사출량, 형체압, 사출압, 타이바(Tie bar) 간격등이다. 주어진 설계 명세를 수용하여 금형의 구성 요소 중 캐비티 부(Cavity\_section)의 캐비티와 코어, 몰드베이스 Upper\_section 의 고정측형판(Cavity\_retainer\_plate)와 Lower\_section 의 가동측형판(Core\_retainer\_plate)에 대해 설계한다. Fig. 7 은 OWL 과 Protégé 를 이용하여 온톨로지를 모델링한 예를 보여준다. Protégé 에서 owl:Thing 은 모든 개체를 담고 있는 집합을 표현한다. 모든 클래스는 owl:Thing 의 하위 클래스이므로 설계 프로세스(Design process), 태스크(Task), 설계 명세(Specification), 인공물(Artifact)는 owl:Thing 의 하위 클래스이고 각각의 클래스들은 계층 구조로 하위 클래스를 갖는다. 클래스 요소들 간의 관계는 *has\_task*, *has\_artifact*, *has\_specification*, *has\_post\_task*, *has\_pre\_task*, *has\_sub\_task*, *introduce*, *is related to*, *create* 와 같은 객체형 속성(ObjectProperty)으로 정의되고, 클래스 구성원에서 속성값을 취하는 제약사항은 *allValuesFrom*, *someValuesFrom* 를 이용하여 표현한다. Design process



는 구성 요소인 Task, Specification, Artifact 를 *has\_task*, *has\_specification*, *has\_artifact* 로 연결하고, 구성 요소 중 Task 를 Specification 과 Artifact 에 각각 *introduce* 와 *is\_related\_to* 로 연결하여 Design process 를 표현한다. Injection mold design process 는 Design process 의 하위 클래스로 속성들을 상속받고, Artifact 의 하위 클래스인 Mold 에 연결된 속성에 대해 *allValuesFrom*, Task 의 하위 클래스인 Cavity design, Parting line design, Moldbase selection, Feed system design, Ejector design, Vent design 과 Specification 의 하위 클래스인 Machine spec. , Moldings\_information, Delivery date, Order size 에 연결된 속성들에 대해 *someValuesFrom* 를 사용하여 Injection mold design process 를 표현한다. Cavity design 은 *has\_post\_task* 로 Parting line design 을 연결하여 순서를 표현하고, *has\_sub\_task* 와 *someValuesFrom* 으로 Cavity layout determination, Cavity number determination, Computation of cavity and core strength, Determination of core part dimension 을 연결하여 Sub\_task 를 표현한다. 또한 *is relate to* 속성의 하위 속성인 *create* 와 *someValuesFrom* 으로 Cavity, Core, Cavity retainer plate, Core retainer plate 을 연결하고, *introduce* 와 *someValueFrom* 으로 Clamping force, Distance between tie bars, Shot capacity, Shot pressure, Shot velocity, Moldings geometric model, Moldings material, Moldings weight, Delivery date, Order size 를 연결하여 Cavity design 의 Artifact 와 Specification 을 표현한다.

## 5. 결론

기존의 도식적 설계 프로세스 모델에 비하여 온톨로지 기반 설계 프로세스 모델은 OWL 과 같은 사람과 컴퓨터가 모두 이해할 수 있는 형태로 표현되기 때문에 설계 프로세스 구성 요소와 요소들간의 관계에 대해 명시적으로 표현할 수 있을 뿐만 아니라 추론을 통하여 제시된 프로세스를 시스템 내에 직접 연계할 수 있는 대안을 제시한다.

## 후 기

본 연구는 산업자원부 핵심연구개발사업 “미세형상 설계 지능화 기술 개발” 과제 수행의 일환으로 이루어진 것임을 밝히며, 지원 기관에 감사 드립니다.

## 참고문헌

1. 박지형, 천준원, 이광락, “제품의 기능 복잡도를 고려한 설계 초기 단계에서의 설계 프로세스 계획 방법: 오토레버 설계 적용 연구,” 한국

CAD/CAM 학회 학술발표회 논문집, pp.759-766, 2005

2. M. Cumming, “Constructing Process Models form Distributed Design Activity,” Ph. D. Dissertation. Faculty of Architecture. Delft University of Technology, 2004
3. 류성열, 김진수, (새로운 방식에 의한) 구조적 시스템 분석, 1994
4. <http://www.idef.co.kr/>
5. T. R. Gruber, “A translation approach to portable ontologies,” Knowledge Acquisition, Vol.5, No.2, 1993