

건축물 정보표현 및 처리를 위한 온톨로지 적용에 관한 연구^o

An Ontology-Based Representation and Processing of Building Information

김 성 아*

Kim, Sung-Ah

Abstract

The building data modeling is always at the core of design computation in every phases of building design and construction. As it is not easy to reconcile material and construction viewpoints, however, existing building data models could not easily apply these concepts within the CAD system. On the other hand, such models need more flexibility and complexity in order to handle associated meanings, activities, and design information when processed in various applications. Due to the recent advances in semantic web research, ontological approach seems to promise a flexible building information modeling. Ontology is an specification of a conceptualization, and provides a vehicle to construct a network of concepts for a semantically rich knowledge representation. In addition to its possibility of becoming a standard for knowledge sharing, the ontological method is worth to be investigated in the domain of architecture. The author has investigated the nature of building systems in order to build its formal descriptions, and a general ontological model for typical building has been developed using OWL and Protégé. Some important elements of building data model are examined in ontological sense. Ontology modeling issues and findings are discussed, and a possible development directions are presented in this paper.

키워드 : 온톨로지, 웹온톨로지언어, 건물데이터모델, 프로테제
Keywords : Ontology, OWL, Building Data Modeling, Protégé

1. 서 론

1.1 연구의 배경과 목적

본 연구의 목적은 건축물의 정보를 표현함에 있어서 온톨로지(Ontology) 엔지니어링의 지식표현방법론을 적용함으로써 에이전트¹⁾가 물리적 환경이 가지는 복잡한 의미체계를 추론하고 지능적인 의사결정을 할 수 있는 정보모델을 구축함에 있다. 온톨로지는 건축분야에서는 다소 생소한 개념이지만 최근 인터넷 혁명이 가져다준 온라인 정보환경에서의 지능적인 정보처리를 목적으로 구체화되고 있는 시맨틱웹(Semantic Web)의 근간이 되는 방법론이라고 할 수 있다. 이러한 연구진행과 평행선상에서 그 개념과 성과를 건축분야에 접목하여 건물공간 정보의 자동화된 처리(Space Ontology), 건설전자상거래의 지능화(Building Product Ontology), 더 나아가 건축설계 프로세스의 모델링(Design Ontology)에 있어서 기존의 객체지향모델링 등에서 지원하지 못하던 복잡한 의미체

계를 구축할 수 있게 하고자 하는 것이 본 연구의 배경이 될 것이다.

1.2 연구의 방법과 범위

본 연구의 발단은 지능형 로봇이 인간의 요구에 능동적으로 대응하기위하여, 센서데이터로부터 환경과 상황의 의미를 추출하여 판단을 하기위한 지식표현 프레임워크의 구현이었다(Wang et al. 2004). 따라서 에이전트가 활동하는 환경, 물체, 그리고 사람의 행위라는 도메인을 구성하는 개념들(Concepts) 및 개념들 간의 관계(Properties or Relations)를 정형화된 방법으로 표현하기위하여 온톨로지를 방법론으로 채택하였다. 온톨로지는 웹 관련 연구와 AI 분야에서 최근 활발하게 연구되고 있지만 디자인, 특히 건축분야와 관련한 온톨로지의 연구는 매우 드물다고 할 수 있다. 다만 최근의 건설전자상거래 등을 위한 프로세스 표현방법론으로(Staub-French et al.2002) 초기 연구 성과를 내고 있을 뿐이다. 본 연구의 궁극적인 목표는 디자인 온톨로지의 구현이지만, 과거의 디자인 프로세스에 대한 연구가 온톨로지를 적용한다고 해서 해결되는 것도 아니며, 마찬가지로 많은 기본적인 연구를 필요로

* 성균관대 건축학과 부교수, 건축학박사

^o 본 연구는 한국과학재단의 창의적연구진흥사업의 지원을 받는 성균관대학교 CREDITS 연구센터의 연구결과물이다.

1) 여기서 에이전트는 로봇, 또는 소프트웨어 에이전트를 포함한다.

한다. 특히 건축분야에서 생소한 온톨로지의 개념확립도 중요한 과제일 것이다. 따라서 본 연구는 우선 시맨틱웹의 개념과 온톨로지의 역할에 대한 기초적인 연구를 행하였다. 다음으로 온톨로지의 요소들을 파악하여 온톨로지 구축을 위한 일반적인 방법론을 따르면서 건축물 온톨로지 구축에 있어서 고려해야할 이슈들을 논의하였다.

디자인 온톨로지의 기초적 단계로서 건축물의 물리적 특성을 온톨로지화하는 프로세스를 통해서 온톨로지의 구현 방법론을 시도하였다. 연구의 내용을 정리하면 다음과 같다.

1. 시맨틱웹의 개념 및 온톨로지의 기본적인 이해
2. 온톨로지 개념에 의한 건축물 정보 표현의 개념화
3. 온톨로지에 의한 건축물 표현방법 시도 및 검증
4. 응용방법의 탐구와 예시

4번 단계에서는 대표적인 온톨로지 편집기인 프로테제(Protégé)²⁾와 추론엔진인 레이서(Racer)³⁾를 이용하여 구현추론 테스트를 행하였고, 온토비즈(Ontioviz)⁴⁾를 통해서 연결관계를 가시화하여 검증하는 방법을 이용하였다.

본 연구는 구체적인 기능유형에 따른 건물 정보의 정의보다는 일반적인 건축물의 기능적 공간을 분석하고 이를 온톨로지 모델로 변경하는 문제에 초점을 맞춘다.

2. 온톨로지의 의의

2.1 시맨틱웹

시맨틱웹(Semantic Web)은 웹에 편재한 정보에 잘 정의된 의미를 부여함으로써 사람뿐만 아니라 컴퓨터도 쉽게 문서의 의미를 해석할 수 있도록 하여 컴퓨터를 이용한 정보의 검색 및 해석, 통합 등의 업무를 자동화하기 위한 목적으로 제안되었다(최중민 2003).

시맨틱웹의 문서는 자연어 위주의 기존 웹 문서와는 달리 컴퓨터가 해석하기 쉽도록 부여한 의미를 가지고 있기 때문에 자동화된 에이전트나 정교한 검색엔진들이 부여된 의미를 이용하여 고수준의 자동화와 지능화를 이룰 수 있게 된다. 시맨틱웹은 문서의 각 부분을 컴퓨터가 이해할 수 있는 형식으로 기술(description)할 수만 있다면 복잡하게 얽혀져 있는 정보 리소스들 사이의 의미적 연관성으로 인해 웹을 통해 다양한 정보를 보다 효과적으로 활용할 수 있게 해준다. 시맨틱웹은 온라인 리소스 간의 의미관계를 정의하기 위하여 RDF(Resource Definition Framework)와 같은 기술을 사용한다. RDF는 개체(object)-속성(attribute)-값(value)의 구조를 갖고 있는데, A(O,V)로 표현된다⁵⁾.

2.2 온톨로지의 정의

시맨틱 웹의 이상을 실현하기 위해서 반드시 선행되어야 할 기술 기반이 RDF층 상단에 위치하는 온톨로지(Ontology)이다⁶⁾.

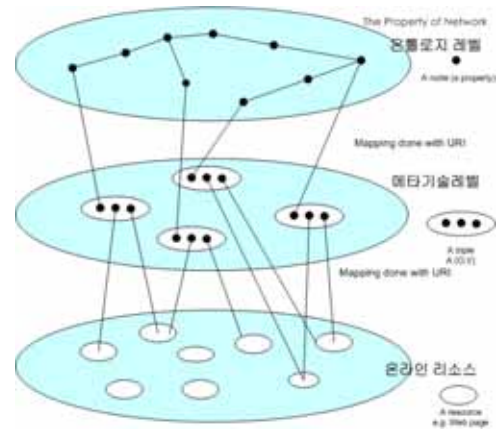


그림 1. Ontology와 RDF, Web resource의 관계

철학적인 개념으로써 존재의 본질과 유형에 관한 이론을 의미하는 온톨로지는 인공지능이나 웹 관련 연구에서는 기본적으로 개념 규정을 말하며 일반적으로 용어와 용어들 간의 관계를 표현하는 컴퓨터 판독이 가능한 공식 규정을 의미한다.

온톨로지 개발의 가장 큰 목적은 인간 또는 소프트웨어 에이전트 간에 정보의 구조에 대한 이해를 공유하는 것이다 (Musen 1992; Gruber 1993). Gruber(1992)에 의하면 온톨로지는 어떤 개념체계에 대한 명시적 규정(explicit specification of a conceptualization)이다. 어떤 도메인의 지식이 선언적 형식성으로 표현될 때 표현 가능한 일단의 오브젝트들, 그리고 그들 간의 기술 가능한 관계들이 표현어휘들에 반영되고 지식기반프로그램은 이러한 어휘들을 가지고 지식을 표현한다.

하나의 온톨로지는 담론의 도메인에서 사용되는 개념들에 대한 정형화되고 명시적인 기술, 즉 클래스(Class) 또는 개념(Concepts)과, 그러한 개념들의 특성이나 속성을 기술하는 프로퍼티들(Properties, Slots, Roles), 그리고 그 프로퍼티에 대한 제약조건(Facets, Role restrictions)들로 정의된다(Noy & McGuinness 2001).

온톨로지의 개발은 반복순환적인 프로세스이기 마련이다. 온톨로지의 클래스는 관심대상의 도메인에 있는 실제 또는 논리적 오브젝트, 그리고 관계들에 근접해야한다. Noy와 McGuinness(2001)는 온톨로지 개발의 7단계를 다음과 같이 설명하고 있다.

1. 온톨로지의 도메인과 범위를 설정한다.
2. 기존 온톨로지의 재사용을 고려한다.
3. 온톨로지에 있어서 중요한 명칭들을 열거해본다.
4. 클래스와 클래스위계를 정의한다.

6) 그림 1의 온톨로지 레벨에 해당

2) Protégé 프로젝트에 관해서는 <http://protege.stanford.edu> 참조 (2005년 1월 31일 현재)

3) Racer 프로젝트에 관해서는 <http://www.cs.concordia.ca/~haarslev/racer/> 참조 (2005년 1월 31일 현재)

4) Ontoviz에 관해서는 <http://protege.stanford.edu/plugins/ontoviz/ontoviz.html> 참조 (2005년 1월 31일 현재)

5) 그림 1의 메타기술레벨에 해당

5. 클래스의 속성들을 정의한다.
6. 속성값을 정의한다.
7. 인스턴스를 생성한다.

2.3 온톨로지에 의한 건축정보표현의 의의

온톨로지적 접근방법이 설계 지식을 정형화하는 유일한 방법은 아니다. 그럼에도 불구하고 온톨로지 기술에 의해 제공되는 원리와 수단은 실무에 직접 응용될 수 있다. 이 접근방식은 기존디자인에 대한 보다 지식집약적(knowledge-intensive)지원을 할 수 있도록 해준다.

온톨로지 구축은 우선 시맨틱웹 또는 웹비즈니스의 관점에서 보았을 때 분산된 데이터를 활용할 수 있게 해준다. 이는 상거래의 상당부분이 웹을 통해 이루어지고 있을 때 사이트간의 공통된 개념에 대한 기술과 명칭 이질성을 해결(heterogeneity resolution)해줌으로써 단순질의와 인텍싱에 의해서 불가능한 고수준의 검색 및 처리를 가능하게 함을 뜻한다. 따라서 온톨로지는 지식공유(knowledge sharing)의 표준이라는 점에서 건축분야 적용의 가능성을 가지는 외에 건설전자상거래에서 자연스럽게 중요한 연구개발 대상이 된다. 즉, 불완전하며, 분산되고, 이질적인 데이터로부터 검색, 추론이 가능하며, ‘가격-단품’ 형식의 일대일 인텍싱이 아니라 디자인 의도(design rationale)와 상황정보의 의미망을 구축함으로써 고수준의 프로덕트 유사성(product similarity) 추론이 가능하기 때문이다. 예를 들어, 현재의 물량산출 소프트웨어의 경우 시공비용에 영향을 주는 컴포넌트의 특성을 표현하는 가격정보와 자재와의 관계만을 표현한다. 이러한 표현은 컴포넌트 특성이 어떻게 특정 가격 정보에 영향을 미치는지에 대한 평가자의 의도 및 기준을 표현하지 못하며, 중요한 컴포넌트의 특성, 그리고 그 특성들이 어떻게 컴포넌트의 가격에 영향을 미치는지를 표현하지 못한다.

온톨로지는 중요한 건물 프로덕트 모델의 특성(feature), 그 특성들이 언제 중요하고 시공 액티비티에 어떻게 영향을 미치는 지, 그리고 공사비를 계산하기 위한 리소스를 표현함으로써 자재와 가격정보 간의 관계표현을 풍부하게 해준다(Staub-French et al.2002). 또한 기존의 건축물사례베이스(case-base)의 구현에도 새로운 차원을 부여할 수 있다. 본 연구의 궁극적인 결과물은 결국 디자인 온톨로지일 것이다. 디자인 온톨로지의 목적은 암묵적인 디자인 지식을 전달하기 위한 지식표현체계를 구축하고, 건축물과 관련된 다양한 개념들의 관계망을 통해서 가능한 디자인 방법을 명시적으로 표현하여 디자인 가이드를 구성할 수 있고, 디자인 프로세스 모델을 정교하고 풍성하게 할 수 있다는 데 있다.

이러한 온톨로지는 다음과 같은 기본적인 요구조건을 가지고 있다.

- 특정영역이나 목적에 초점을 맞추어야 한다.
- 해당전문가들이 동의하고 합의되어 수용되어야 한다.

- 실세계를 모델링 할 수 있고, 컴퓨터가 읽고 처리 가능해야 한다.
- 형식을 갖춘 의미와 구조로 표현되어야 한다.

OWL은 온톨로지 구현의 이러한 조건들에 대응하기 위하여 개발된 표준온톨로지 언어 중의 하나이다. 표 1은 OWL 구문의 사례로서 연구 초기단계에서 잠정적으로 상정되었던 클래스들 중 일부에 관한 OWL 표현이다. 여기에서 건물(Building), 컴포넌트(Component), 그리고 공간(Space)을 클래스로 정의하고 있으며, 또한 벽체(Wall)는 컴포넌트의 서브클래스임을 표현하고 있다.

표 1. OWL을 이용한 클래스 텍소노미 표현의 예

```
<owl:Class rdf:ID="Building"/>
<owl:Class rdf:ID="Component"/>
<owl:Class rdf:ID="Space"/>
<owl:Class rdf:ID="Wall">
  <rdfs:subClassOf rdf:resource="#Component">
</owl:Class>
```

표 2는 구축된 온톨로지에서 정의하는 *hasOpening* 관계속성에 대한 OWL 표현의 예이다. 이는 건물공간의 온톨로지 모델링에 있어서 동일한 개념의 관계가 대상에 따라서 다른 제약을 가지는 것을 표현한다. 예를 들어 건물에서 문으로 인식되는 개구부(Opening)나 층간 슬래브에 존재하여 수직연결을 가능하게 해주는 수직개구부(VertHole)는 모두 연결자(Connector)이다. *hasOpening*은 구체적으로는 벽체(Wall)에 있어서는 개구부를 값으로 가지지만, 슬래브에 대하여 사용된 경우(즉, 도메인이 슬래브인 경우), 값의 범위(range)는 수직개구부에 한정된다.

표 2. OWL에서의 프로퍼티 표현의 예

```
<owl:ObjectProperty rdf:ID="hasOpening">
  <rdfs:domain rdf:resource="#Slab"/>
  <rdfs:range rdf:resource="#VerhHole"/>
</owl:ObjectProperty>
```

3. 건축물 온톨로지

3.1 건축물의 구성체계에 대한 고찰

일반적으로 건물의 데이터모델링은 건물을 공간(Space)과 구조체(Structure) 또는 컴포넌트(Component)의 위계적 구성체로 취급하고 있다(Björk 1992; Eastman 1995).

공간 중심의 관점에서 보면 하나의 건물은 건물 전체(Building)로부터 층(story), 그리고 각층의 개개의 실(Room)에 이르기까지 모든 개념을 공간(a kind of space)으로 보고, 건물은 이들 공간의 위계적인 추상 체계(abstraction hierarchy)로 볼 수도 있다. 이보다는 복합적으로 건물을 공간과 구조체의 위계적인 추상체계로 정의할 수도 있다. 후자의 경우 일반적인 벽식 아파트와 같은 건물은 하나 이상의 층(공간)과 외벽(구조체)로 구성되고

층은 하나 이상의 실(공간) 또는 호(공간)과 호간벽체(구조체)로 조직되는 구성체계를 가진다.

이러한 기존의 모델은 건축물을 공간 또는 컴포넌트의 위계로 분해하는 객체지향적 방법을 사용하고 있으나 온톨로지 기반 모델링은 개체들 간의 관계에 중점을 두며 더 나아가 여기에 의미구조와 관계의 제약조건을 기술함으로써 보다 복잡한 의미의 프로세싱이 가능하도록 한다.

3.2 서브시스템의 집합체로서의 건물

우리가 일상적으로 경험하는 대부분의 건물은 여러 개의 층으로 나뉘어져 있고 각 층은 방과 복도로 나뉘어져 있다. 또한 각층은 계단과 같은 수직동선체계에 의해서 연결된다. 그 외에 방과 방을 나누는 벽체, 외피시스템, 기둥과 같은 구조체로 이루어진다. 본 연구에서는 건물을 우선 서브시스템의 집합체(aggregation)로 정의하고 공간이라는 개념은 층(Story)의 하위구성요소로서 비로소 도입된다. 서브시스템이라 함은 단순히 컴퍼넌트로 취급하기에는 복잡한 구성체로 인식되는 건물의 구성요소로서 여기서 층(Story)이나 수직연결요소(Vertical Link: Vert Link)를 서브시스템으로 분류하였다(그림 2). 물론 이러한 구성위계는 단일공간이 두 개 이상의 층에 걸쳐있는 경우(예를 들면 두세 개의 층에 각각 출입구를 가지는 강당)이나, 최근의 현대건축경향에서 나타나는 비정형의 바닥과 상호관입된 공간의 개념을 모델링 할 수는 없다. 따라서 본 연구에서 제시하는 건축물은 일반적인 벽식건물 또는 도미노 구조의 건물로 제한한다.

층(Story)을 서브시스템으로 분류하는 데는 직관적으로 문제가 없어 보이나 수직연결요소의 경우에는 데이터모델링의 관점에 따라서 이견이 있을 수 있는 부분이다. 여기서 수직연결요소의 종류로는 승강기(Elevator)나 계단(Stair)을 들고 있다(그림 2). 두 요소 모두 단순히 하나의 컴포넌트로 취급하기에는 다양한 컴포넌트들로 구성되어 있고(예를 들어 승강기의 경우 승강기 캐빈(Cabin) 뿐만 아니라 이양기와 컨트롤러(Controller), 그리고 레일링(Railing)을 포함하는 서브시스템으로 보는 것이 타당하다. 계단(Stair) 역시 그러한 특성을 가지는 요소로 보았다. 이러한 분류체계는 반드시 최적의 분류가 아니라, 단지 공간 온톨로지를 구현함에 있어서 애플리케이션의 용도에 따라 얼마든지 다른 개념체계(conceptualization)에 따라 구성될 수 있음을 가정하고 있다.

3.3 서브시스템(SubSystem)의 온톨로지적 개념화

그림 2에서 보듯이 하나의 건축물은 여러 개의 서브시스템으로 구성된다고 설정하였다. 따라서 '서브시스템을 가진다'라는 관계에 의해서 *hasSubSystem*(Building, SubSystem)이라는 명제가 성립한다. '서브시스템이다'라는 관계에 의해서 서브시스템은 어떤 빌딩의 서브시스템이다 *isSubSystemOf*(SubSystem, Building)이라는 명제가

성립한다. 이러한 관계는 서로 역관계(inverse relation)이며 객체지향시스템에서 말하는 일종의 *part-whole* 관계로 연결된다.

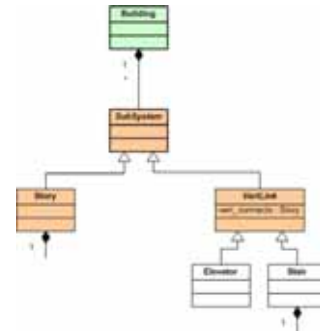


그림2. 서브시스템과 건물과의 관계 UML 다이어그램

수직연결요소는 층과 층을 연결한다. 즉 *connects* (VertLink, Story)로 표현된다. Elevator의 경우 2개 이상 복수 층을 연결하며(cardinality = n), 계단의 경우 2개의 층을 연결한다(cardinality = 2). Elevator는 인스턴스 레벨에서는 특정 층만을 운행할 경우 cardinality가 달라질 수 있다.

3.4 컴포넌트

건축물과 서브시스템은 모두 컴포넌트에 의해서 구성된다. 컴포넌트는 그 자체로서 다른 컴포넌트(sub-component)들로 구성되는 것으로 정의할 수 있지만 본

연구에서는 개념 명칭의 편의상 기둥(Column), 개구부(Opening), 창호(Window)와 같이 건축적인 오브젝트(즉, AEC Object)의 레벨에서 다루어지는(인지되는) 요소들을 컴포넌트로 정의하고 그보다 하위의 컴포넌트는 파트(Part)라는 클래스에 의해 별도로 정의하였다. 예를 들어 문(Door)이라는 요소는 컴포넌트에 속하지만 도어손잡이(Knob)나 목재프레임은 같은 레벨에서 문을 구성하는 파트에 속하며 각각의 파트는 필요에 따라 위계적(recursive)으로 다른 파트를 포함하거나 다른 파트의 서브파트가 될 수 있도록 개념화하였다(그림 3).

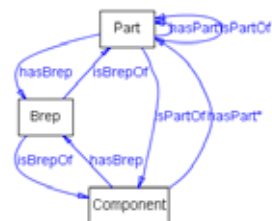


그림3. 컴포넌트와 파트의 관계 다이어그램

파트는 AEC 오브젝트(즉, 컴포넌트)를 구성하는 인공물이다. 파트의 구조는 파트를 구성하는 부품파트들에 의

해 정의된다. 어떤 파트와 그것을 구성하는 부품 파트들과의 관계는 *isPartOf*로 표현된다. 따라서 두개의 파트 간에 *isPartOf*(p, p')는 p가 p'의 서브파트임을 의미한다. 예를 들어 문이란 문짝과 문틀로 구성된 체계로 본다면 문짝(doorPanel)은 문(door)과 *isPartOf* 관계이며 손잡이(DoorKnob)는 문짝과 *isPartOf* 관계이다. 여기서 *isPartOf* 관계는 이른 바 추이적 관계(transitive relation)이다. 즉, 위의 관계들에서 손잡이도 문의 파트 *isPartOf*(DoorKnob, Door)라고 추론할 수 있다.

추이적 관계는, 어떤 파트 A가 다른 파트 B의 파트이고 B는 C라는 파트의 파트이면 A는 C의 파트라는 명제가 성립한다. 또한 레벨이 같은 파트 간에는, 대칭적(symmetric) 관계가 성립한다. 즉, *isPartOf*(p1, p')이고 *isPartOf*(p2, p')이 성립하면 *sibling*(p1, p2)이며 동시에 *sibling*(p2, p1)이 성립한다.

3.5 컴포넌트의 형상정보

컴포넌트는 공간 개념을 제외한 대부분의 건물 온톨로지 클래스가 상속을 받는 클래스이다. 컴포넌트는 형상정보표현(Shape representation)을 위한 정보와 위치정보(Placement)을 위한 정보를 가진다. 형상정보표현은 전형적인 상자(box)형 형상을 가진 컴포넌트 (예: 벽체의 경우) 벽체단면 프로파일의 스위핑(sweeping) 형상으로 표현하거나, 이러한 s방식에 의해 표현이 불가능한 복잡한 형상의 경우 경계면표현(boundary representation)을 취한다. 또한 작동성능(performance)이나 LOD(Level of Detail) 조절을 위한 추상적인 표현을 위하여 바운딩박스표현(bounding box representation)을 취할 수 있다. 위치 정보는 Position과 Orientation의 조합으로 나타낸다. Position은 바운딩박스의 3차원(x, y, z) 좌표로서 이는 일반적으로 절대좌표이기 보다는 상대좌표이다. 즉, 컴포넌트인 문의 경우, 그것이 속한 벽체의 position을 원점(0, 0, 0)으로 한 문 바운딩박스의 상대좌표이다. 예를 들어 벽체 W의 절대좌표 위치가 2000, 2000, 2000 이고 W에 속한 문의 Position 값이 0, 500, 0이라고 한다면 문의 절대위치 값은 2000, 2500, 2000으로 계산된다. 한편 Orientation은 바운딩박스의 3차원 회전값을 나타낸다. 이는 x_a, y_a, z_a 라는 벡터에 의해 3차원 회전축을 표현하고 그 회전축을 중심으로 한 회전각도값을 부가적으로 표현하는 모두 4개의 요소로 구성되는 데이터타입이다. 그림 3의 경우 벽체의 위치는 벽체의 상위 어셈블리 오브젝트의 x, y, z 좌표값에 대한 상대좌표인 x', y', z' 이며 회전값은 0, 0, 1 즉 z 축을 중심으로 a 만큼 회전한 형상이 된다. 물론 이 회전값 역시 상위 어셈블리의 회전값에 누적하여 위계적(recursive)으로 계산된다. 이러한 위치정보 표현방법은 실제로 IFC 모델의 그것과 유사하며 (IFC 2002), 장면그래프(Scene Graph) 중심의 3차원 표현에서 일반적으로 채택되고 있는 방법이다(Foley et al. 1990).

3.6 컴포넌트의 온톨로지 추론

프로덕트 모델링 수준에서의 온톨로지에 의한 추론의 예는 다음과 같다. 예를 들어 컴포넌트를 구성하는 파트에는 단말오브젝트(terminal object)와 복합오브젝트(composite object)가 있다. 단말오브젝트는 더 이상 서브파트로 분해될 수 없는 파트이다. 이러한 종류의 파트는 프로젝트해체위계(decomposition hierarchy)에서 최하위레벨에 있다. 예를 들어 단말오브젝트는 서브파트를 가질 수 없는 파트로서 $(\forall p) \text{ terminalObject}(p) \equiv \neg(\exists p') \text{ isPartOf}(p', p)$ 라는 명제로 표현될 수 있다. 즉, 모든 p는 p가 만약 terminalObject라면 p와 *isPartOf* 관계로 연결되는 어떠한 오브젝트 (즉, p')도 존재하지 않는다고 정의할 수 있다.

프로덕트 모델링의 관점이 아닌 디자인 프로세스의 관점에서 볼 때 이 단말오브젝트는 디자인 단계와 시공단계의 연결을 할 수 있다. 예를 들어 단말오브젝트는 필요에 따라 벽체, 또는 벽체를 구성하는 재료 등으로 세분화될 수 있다. 또는 디자인 단계에서 LOD의 적절한 범위(granularity)를 제공하는 부분이다. 즉, 디자인의 각 단계에서 요구되는 추상성에 따라 단말오브젝트의 범위는 조정될 수 있다. 디자인 지원 시스템에서 디자인의 상황적 콘텍스트와 관련된 LOD를 온톨로지화 할 수 있다(그림 5).

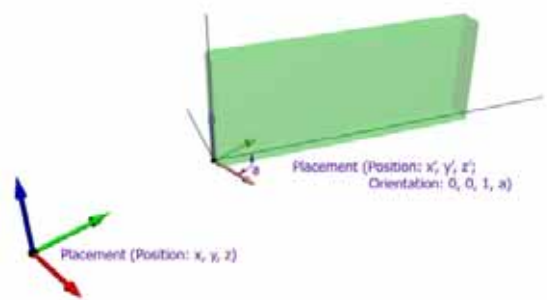


그림 4. Placement의 개념

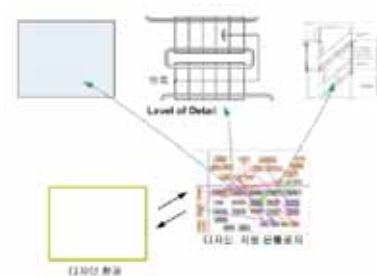


그림 5. LOD와 온톨로지의 관계

온톨로지의 논리적 기술에 따라 복합오브젝트의 경우 다음과 같이 정의할 수 있다. 복합오브젝트는 단말오브젝트가 될 수 없다. 즉, $(\forall p) \text{ composite}(p) \equiv \neg \text{primitive}(p)$ 로 표현가능하며 그 의미는 단말오브젝트가

아닌 것은 복합오브젝트라는 것이다. 복합오브젝트 내에서 어떤 파트의 직접적인 파트는 다음과 같이 추론할 수 있다. 이는 *isPartOf* 관계로 구성되는 *part-whole* 위계에서 두 파트 사이에 어떤 중간 파트도 없는 경우에 해당된다. 즉, $(\forall p1, p2) \text{ direct_part_of}(p1, p2) \equiv \text{isPartOf}(p1, p2) \wedge \neg(\exists p') \text{ isPartOf}(p1, p') \wedge \text{isPartOf}(p', p2)$ 로 표현될 수 있으며 그 의미는 p1이 p2의 직접적인 파트라는 것은 p1이 p'의 파트이고 p'가 p2의 파트가 되는 그런 p'는 존재하지 않는 것을 말한다.

온톨로지의 가능성은 이렇게 다양한 지식을 구축하기 위하여 해당하는 관계속성을 모두 만들기 보다는 서술논리(descriptive logic)을 이용한 관계추론에 의하여 확장가능하다는 점에 있다.

3.7 분리자와 연결자(Divider and Connector)

벽체(Wall)와 슬래브(Slab)는 분리자(Divider)라는 클래스의 서브클래스가 되도록 정의한다(그림 13). 객체지향적 모델링을 할 경우 이들 클래스는 컴포넌트와 분리자 모두를 다중상속 받는 개념으로 정의하는 것이 당연하지만 OWL에서는 다중상속이 인스턴스 레벨에서 문제를 일으키는 경우가 많으므로 분리자는 컴포넌트의 서브클래스이고 벽체와 슬래브를 분리자의 서브클래스로 정의하였다. 상위 클래스인 컴포넌트에서 정의한 형상정보표현 방식에 따라서 단면(polygon profile)을 스윕한 것으로 형상을 정의하고 복잡한 형상의 경우 경계면표현으로 표현한다.

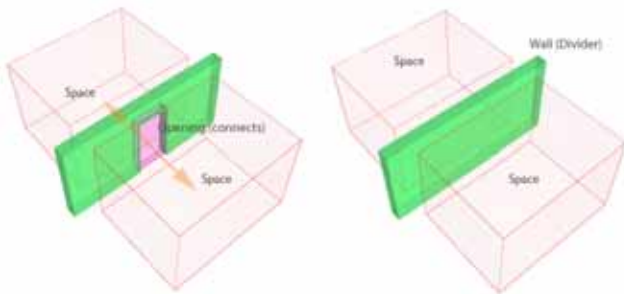


그림 6. 연결자(Connector)와 분리자(Divider)

분리자는 두 개의 공간(Space)을 수평적으로 분리하거나(즉, 벽체), 두 개의 층을 분리하는(즉, 슬래브) 컴포넌트로 정의된다(그림 6). 슬래브와 벽체는 모두 연결자를 가질 수 있다. 연결자(Connector) 클래스는 컴포넌트의 서브클래스로서, 벽체가 공간을 분리하는 분리자의 속성을 가지는 반면 개구부(Opening)는 공간과 공간을 연결하는 연결자(Connector)의 서브클래스이다. 즉 공간(Space)과 공간을 connects의 관계로 연결한다(그림 6). 따라서 OWL의 프로퍼티인 connects 라는 속성을 통해서 모든 연결자는 항상 두 개의 공간을 연결한다는 명제를

7) 또는 분리자를 interface로 정의하여 벽체 레벨에서 implement하도록 하는 방법이 일반적이다.

정의할 수 있다. 역으로 말하면 공간을 연결하는 것은 모두 연결자의 필요충분조건이 된다. 단 슬래브가 가질 수 있는 연결자는 수직개구부(VertHole)만 가능하고 벽체가 가질 수 있는 연결자는 개구부(Opening)에만 한정된다고 의미를 세분할 수 있다. 수직개구부는 슬래브와 관계되는 요소로서 존재하는 것으로 대개의 경우 계단실과 계단실을 연결하는 것이며 슬래브에 의해 *hasOpening* 관계로 연결된다. 연결자는 컴포넌트의 서브클래스로서 정의되는 형상정보와 위치정보를 가진다.

3.8 개구부(Opening)와 창호(Window)

창호는 공간과 공간(대개의 경우 내부공간과 외부공간)을 가시성관계(visibility)로 연결하고, 어떤 공간에 환기와 채광을 제공하는 등의 기능과 관련된 속성이 부여될 수 있다. 예를 들어 *ventilates*(Window, Space)로 표현할 수 있다. 창호는 벽체의 파트로서 연결되는 컴포넌트로 *hasWindow*(Wall, Window)의 관계에 의해 연결된다. 또한 창호의 관점에서 보았을 때 *isWindowOf*(Window, Wall)의 관계로 연결된다. 하나의 벽체는 0 또는 하나 이상의 창호를 가질 수 있다. 창호는 반드시 하나의 벽체에 속한다. 두 벽체가 접하는 모서리에 속한 창호는 두 개로 분할하는 것으로 정의하였으며, 이는 IFC 모델의 스펙과 유사하다(IAI 1999). 그림 6에서 벽체 eWall_01의 경우 window_01과 window_03을 *hasOpening* 관계로 연결하고 있다.

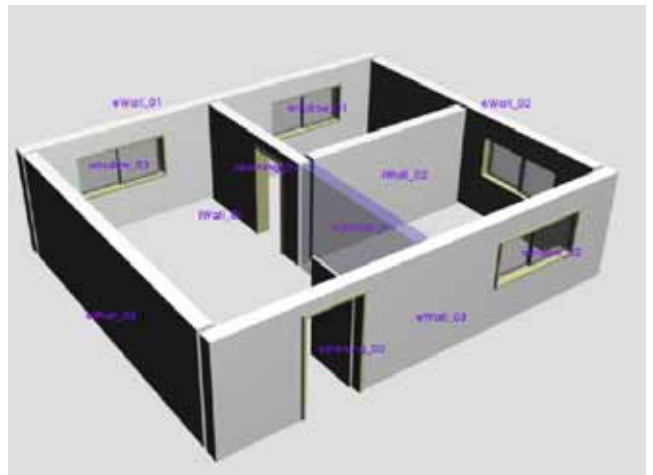


그림 7. 벽체와 창호, 개구부의 관계

창호는 컴포넌트의 직접적인 서브클래스이고 개구부는 컴포넌트의 서브클래스인 연결자(Connector)의 서브클래스이지만 두 클래스는 공통적으로 Filler의 개념을 가진다. 물리적으로는 개구부나 창호모두 벽체의 파트와 같이 취급할 수 있지만(그림 8), 추론에 보다 효율성을 제공하고 온톨로지 모델의 유연성을 제공하기 위하여 개구부와 창호는 벽체와는 컴포넌트를 상속받는 별개의 물체로 정의하였다. 따라서 이 온톨로지를 사용하는 애플리케이션(예를 들어 CAD 프로그램)에서 온톨로지 정보로부터 3

차원 모델의 생성이 요구된다면 벽체로부터 개구부나 창호의 형상을 제거(boolean operation의 subtract에 해당하며, Filler 오브젝트는 void object가 된다)하는 것으로 하였다. 이때 각 클래스는 WindowFiller나 OpenignFiller가 없거나(default), 하나를 가질 수 있다. 전자의 경우는 벽체에 구멍(hole)이 난 상태의 창호/개구부이며 후자의 경우는 일반적인 창호나 문짝이 있는 경우이다(그림 9의 개념도 및 그림 13의 UML 다이어그램 참조). 즉 개구부(Opening)는 OpeningFiller 클래스를 part-whole 관계의 일종인 *hasOpeningFiller* 관계로 연결하며, 창호(Window)는 WindowFiller를 *hasWindowFiller* 관계로 연결한다. 실제로 이러한 방식은 IFC 모델에서도 적용되고 있는 방식이다(IAI 1999).



그림 8. 벽체에 중속된 Opening의 개념

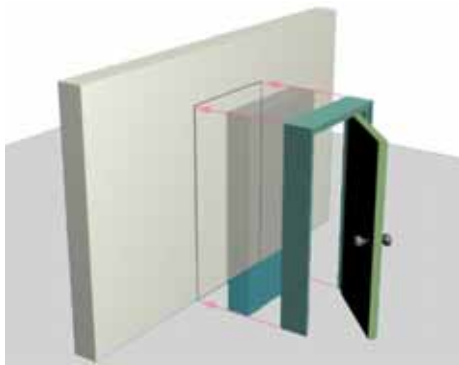


그림 9. 벽체에 대한 Opening과 OpeningFiller의 개념

4. 모델의 제안

4.1. 온톨로지 구현에서 발견된 이슈들

3장에서 건물 온톨로지의 구현을 위한 개념화 및 온톨로지적 사고를 통해서 다음과 같은 이슈들이 제기되었다. 첫째 온톨로지는 아무리 작은 도메인이라 할지라도 그 범위가 매우 커지는 경향이 있으며, 다양한 전문가 및 사용자 그룹의 의견수렴, 그리고 검증은 거쳐야 한다. 또한 작업의 성격상 분산된 온톨로지 구현과 병합이 요구된다. 그러나 온톨로지 구현 작업에서 병합(merging)이 결코 단순한 작업이 아니라는 점을 확인하였다. 따라서 초기

개념화 및 모듈화에 있어서 전략적인 접근이 필요하다는 것을 알 수 있었다. 두 번째의 이슈는 기존 표준과의 연계이다. 앞서 OeningFiller 개념에서처럼 이러한 개념은 온톨로지 구현작업에 있어서 다양한 측면을 고려하여 여러 차례의 다른 방식을 시도한 끝에 선택한 결론이지만, IFC 모델에서도 유사한 방식으로 해결법을 제시하고 있으며, 실제로 Filler라는 용어도 결국에는 IFC 모델의 그것을 차용하도록 하였다. 이는 2.2절에서 소개한 온톨로지 개발 단계에서 2단계인 기존 온톨로지 사용 고려항목에 해당하는 것으로서 결국 본격적인 온톨로지 개발에 있어서는 기존 빌딩데이터모델 등을 수용함으로써 효율적이며, 추후 호환성 측면에서 유리한 온톨로지 개발을 할 수 있을 것이다. 세 번째의 이슈는 역시 온톨로지 구현은 근본적으로 반복적인(iterative) 프로세스라는 점이다. 온톨로지 구현을 위해서 도메인 전문가들의 의견이 수렴되어야하고 구현과 검증의 사이클을 거치면서 업데이트 되어야 한다. 또한 구현과정 중에서 반복적으로 사용되는 개념들은 공통 온톨로지화 하여 재사용하는 것이 필요하다. 마지막으로 경량화된(stripped-down version) 온톨로지의 고려이다. 실제 현재 사용 중인 많은 온톨로지 기반 애플리케이션에서는 OWL과 같은 풍부한 온톨로지 표현수단보다는 경량화된 버전 또는 RDF와 같이 표현범위가 OWL보다 풍부하지는 않지만 보다 확립된 인프라를 가진 구현수단을 사용하는 경우가 많다.

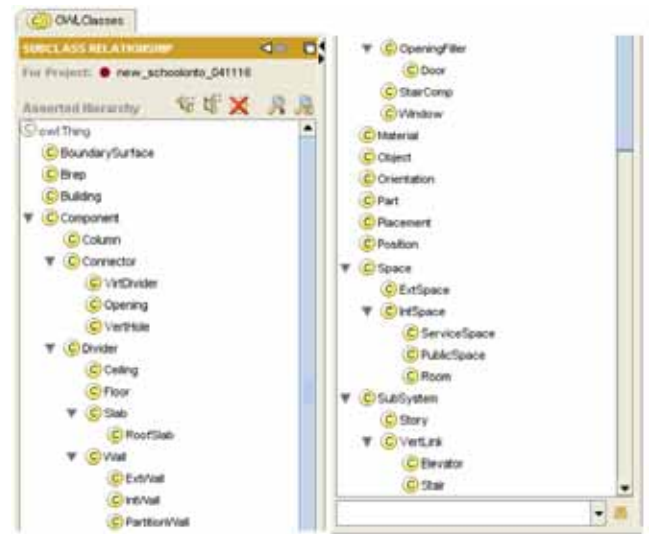


그림 10. Protégé를 이용한 온톨로지 구축(클래스 위계)

4.2. 온톨로지 구현

그림 10은 본 연구의 Protégé를 이용한 온톨로지 구현 사례를 보여준다. Protégé는 OWL을 지원하는 대표적인 온톨로지 편집기로서 활용되고 있다. 그림 10은 그중에서도 클래스의 위계관계(taxonomy)를 보여준다. 예를 들어 슬래브(Slab)나 벽체(Wall)는 분리자(Divider)의 서브클래스이고 지붕슬래브(RoofSlab)는 슬래브의 특수한 경우임을 나타낸다.

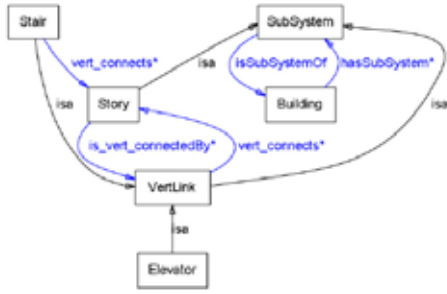


그림 11. 서브시스템 중심의 온톨로지 다이어그램

이러한 온톨로지 구현은 단순히 편집기를 이용한 입력 보다는 UML과 같은 기존의 정형화된 표현방법을 병행하면서 진행되었다. 그림 13의 UML 다이어그램은 일반적인 학교건물을 대상으로 한 건물 온톨로지의 UML 표현이다. UML 표현은 성격상 온톨로지의 풍부한 관계속성을 가시적으로 표현하기 어려우므로 보다 설명적인 그래프 다이어그램을 제공하는 Ontoviz를 이용하여 온톨로지 구현과정 매단계마다 온톨로지 표현 그래프를 생성하여 비교, 검토하였다. 그림 11의 경우 서브시스템을 중심으로 한 관계를 보여주고 있으며 그림 12의 경우 연결자(Connector)와 Filler를 중심으로 한 관계를 보여준다.

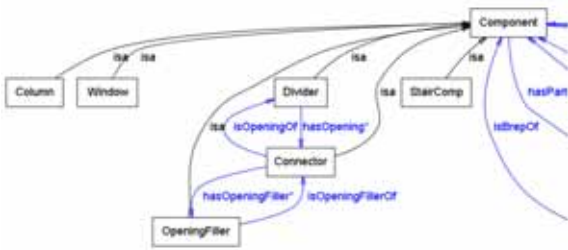


그림 12. Filler 관계

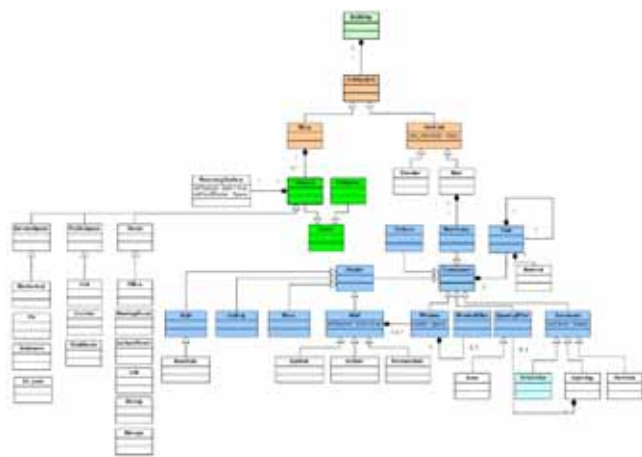


그림 13. 건물 온톨로지의 UML 다이어그램

5. 결론

본 연구는 건물공간구조와 관련된 의미를 온톨로지화 하였고, 이를 위하여 건축물의 주요 요소들에 대한 구조적 의미와 상호관련성을 온톨로지적 관점에서 고찰하였

다. 건물공간이 벽체 등의 건물구성요소와 가지는 상호관계를 유연성과 효율성을 고려하여 온톨로지화 하였으며, 온톨로지 개발과정에서 제시된 이슈들을 제시하였다. 본 연구는 추후 행위 및 오브젝트 온톨로지와의 통합을 통한 추가적인 구현을 통해서 계속 보완될 것이며, 궁극적으로는 디자인 온톨로지의 구현을 위한 기본적인 연구가 될 수 있을 것이다.

참고문헌

1. Björk-C., "A Conceptual Model of Spaces, Space Boundaries and Enclosing Structures", Automation in Construction, 1(3), pp. 193-214, 1992
2. Eastman C. M. & Siabiris A., "A generic building product model incorporating building type information", Automation in Construction, 3(4), pp. 283-304, 1995
3. Gruber, T.R., A translation approach to portable ontology specifications (Technical Report KSL 92-71), Knowledge Systems Laboratory, Stanford University, Stanford, CA, 1992
4. Gruber, T.R., "A Translation Approach to Portable Ontology Specification", Knowledge Acquisition 5, pp.199-220, 1993
5. IAI, IFC Release 2.0 Specifications. International Alliance for Interoperability, 1999
6. Musen, M.A., "Dimensions of knowledge sharing and reuse", Computers and Biomedical Research, 25, pp.435-467, 1992
7. Noy, N. & McGuinness, D.L. (2001). Ontology development 101: A guide to creating your first ontology. Retrieved December 23, 2004 from http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
8. Staube-French, S. et al., An Ontology for Relating Features of Building Product Models with Construction Activities to Support Cost Estimating, CIFE Working Paper#70, Stanford University, Stanford, CA. 2002
9. Foley, J. D., van Dam, A., Feiner, S. K, and Hughes, J. F., Computer Graphics: Principles and Practice, second edition, Addison-Wesley, Reading:MA, 1990
9. Wang, E., Kim, S., & Kim, Y.S., "A Rule Editing Tool with Support for Non-Programmers in an Ontology-Based Intelligent Tutoring System", SWEL Workshop, ISCW Conference, Hiroshima, Japan, Nov. 7-11, 2004
10. 김성아, 건물공간정보의 전자적 교환과 시각화에 관한 연구, 대한건축학회논문집, 19권 5호, 2003
11. 이규철, 시맨틱 웹 서비스, SIGDB 2002 Tutorial 세미나/XML·시맨틱 웹·웹 서비스와 데이터베이스 기술, pp. 147~184, 2002
11. 최중민, 시맨틱 웹의 개요와 연구동향, 정보과학회지, 제21권 제 3호, pp. 4~17, 2003

(接受: 2005. 2. 2)