

OWL을 이용한 공간 온톨로지 구현방법론의 기초적 연구^o

A Basic Study on Constructing a Space Ontology Using OWL

김 성 아*

Kim, Sung-Ah

Abstract

Information about space and activity is of interest in every phases of building design and construction. As it is not easy to reconcile material and construction viewpoints, however, existing data models could not easily apply these concepts within the CAD system. Due to the recent advances in semantic web research, ontological approach seems to promise a flexible building information modeling. More precisely, the ontology approach enables an explicit specification of conceptualizing building information so that higher levels of reasoning is possible both by human and machine agents. The author has investigated the spatial nature of buildings in order to build its formal description to be used by agents, and a general ontological model for typical building has been developed using OWL and Protégé. The OWL is introduced as a main vehicle to encode the information and knowledge about the building structures and spaces. Protégé is employed as a modeling environment in which a test case of a anonymous building is analyzed and modeled in an ontology modeling manner. Ontology modeling issues and findings are discussed, and a possible application is presented in this paper.

키워드 : 온톨로지, 웹온톨로지언어, 공간, 건물데이터모델, 프로테제

Keywords : Ontology, OWL, Space, Building Data Modeling, Protégé

1. 서 론

1.1 연구의 배경과 목적

본 연구의 목적은 건축물의 공간과 관련된 정보를 표현함에 있어서 온톨로지 엔지니어링의 지식표현방법론을 적용함으로써 에이전트가¹⁾ 건물 내외부의 특정 장소로 이동하거나, 센서를 통해 획득한 데이터로부터 공간의 용도를 파악하는 등의 다양한 작업(task)을 수행하는 과정에서 물리적 환경의 의미를 추론하여 지능적인 의사결정을 할 수 있는 정보환경을 구축함에 있다.

이러한 연구는 발단은 기존의 건물공간정보 표현의 문제점을 보완하기 위하여, 시맨틱웹(Semantic Web) 기술 발전의 근간이 되는 온톨로지(Ontology)를 수용함으로써, 에이전트가 다루는 환경의 구조와 의미를 보다 유연하게 기술하고 이를 응용할 수 있게 되었다는 기술적인 배경에서이다. 특히 지능형 에이전트가 인간의 일상적인 명령

으로부터 구체적인 작업을 계획하고 수행하기 위해서는 건물공간의 구조에 대한 추론은 물론, 인간의 행위와 건축물이 상호작용하는 지식 네트워크의 추론이 요구된다. 온톨로지는 건물환경의 물리적 특성 외에 인간과 기계가 해석할 수 있는 환경의 의미론적 구조를 정형화되고 명시적인 방법으로 기술하는 체계를 제공한다. 물론 과거 인공지능 분야의 연구에서 온톨로지적 접근이 이루어져 왔으나 최근 시맨틱웹의 연구에서 제시하는 다양한 표준 기술과 응용방법은 기존 데이터 모델링의 문제점 극복에 대한 실마리를 제공하고 있다. 시맨틱웹은 정보통신분야 외에도 의학(Musen 1992, Hahn & Schultz 2004), e-러닝(Brase & Nejd 2004), 전자상거래(Ding *et al.* 2004), 등에서 활용되고 있으나 건축분야에서는 비교적 새로운 방법론이어서, 본 연구는 건축분야에 온톨로지 방법론 도입의 기초적 연구를 제공한다는 의의도 가지고 있다고 할 수 있다.

1.2 연구의 방법과 범위

온톨로지를 이용한 공간정보 기술방법론은 물리적 환경과 거기에 존재하는 오브젝트들이 인간의 행위와 사물에 대한 개념이 맞물려서 구성되는 의미의 연결관계를

* 성균관대 건축학과 부교수, 건축학박사

^o 본 연구는 과학기술부의 창의적연구진흥사업의 지원을 받는 성균관대학교 창의적설계추진 지적교육시스템(CREDITS) 연구센터의 연구결과물이다.

1) 여기서 에이전트(agent)는 물리적 환경을 이용하는 로봇, 또는 데이터모델의 인스턴스를 이용하는 소프트웨어 에이전트 모두를 포함한다.

정의함으로써 가능하다. 본 논문은 공간 온톨로지 구현에 있어서 기본적인 건축공간의 구조와 상호관계를 정의하는 데 초점을 맞추며 다음과 같은 단계로 진행된다.

- i. 건물공간 모델링의 개념과 요구에 대한 고찰
- ii. 온톨로지의 개념과 구현방법론에 대한 고찰, 특히 웹온톨로지언어를 이용한 의미기술에 대한 기초적 연구
- iii. 대상 건축물의 내부공간과 구성체계에 대한 온톨로지 디자인
- iv. Protégé 를 이용한 온톨로지 구축과 JESS를 이용한 활용방안 제시

본 연구는 구체적인 기능유형에 따른 건물공간구조의 정의보다는 일반적인 건축물의 기능적 공간을 분석하고 이를 온톨로지 모델로 구축하는 문제에 초점을 맞춘다. 따라서 임의의 건축물을 온톨로지 구현 대상으로 하였고 공간의 기능은 학교건물에서 나타나는 일반적인 실들을 사례로 하였다.

2. 건물공간의 개념화

공간은 우리가 물질세계를 지칭할 때 사용하는 기본적인 개념들 중의 하나이다. 우리가 일상적으로 이해하는 공간이라는 개념은 매우 자명해 보이지만 일단 이를 정형화 하려고 하면 많은 논란의 소지를 가진다(Ekholm & Fridqvist 1997). 공간은 어떤 물체들 사이의 비어있음(emptiness)으로서, 그 물체들의 존재 여부에 종속적일 수 있지만 한편으로는 어떤 물체의 내재된 속성(예: 속이 비어있는 구체)이기도 하다. 건축공간은 물리적 실체를 가지지 않는 개념으로서 솔리드(solid)한 컴포넌트, 예를 들어 벽체와 슬래브에 의해 둘러싸여(enclosed) 형성되는 종속적인(파생적인) 요소(dependent or derivative concept)이다.

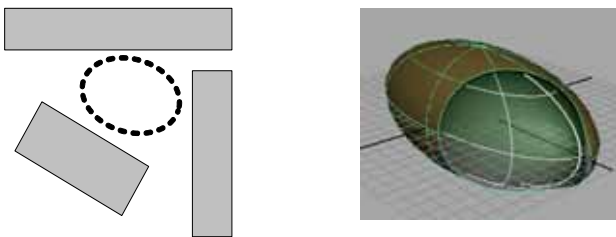


그림 1. 종속적 성격과 내재된 속성으로서의 공간

그럼에도 불구하고 건축설계의 본질은 공간의 구축에 있으므로 물리적 실체보다 더 중요한 실체로서 인지되는 경우가 많다. 건축설계는 많은 경우 구체적인 벽체의 형상이나 재질과 관계없이 공간의 구성으로 시작하여, 어떤 시점에서 공간의 경계선(boundary)이 자연스럽게 솔리드(solid)로 변환되는 디자인 프로세스를 밟는다. 따라서 공간을 단순히 벽체들의 상호관계에 의해 파생되는 종속적

인 개념으로 모델링하는 방식은 특히 개념설계를 제대로 지원하기 위한 건축 CAD 시스템과 같은 경우 큰 문제점을 내포한다. 즉, 공간은 존재론적인 관점에서 파악되는 추상적인 개념인 동시에 건축물을 구성하는 단위이자 물리적인 건축요소들에 의해 형성되는 실체이다. 이러한 복잡성으로 인해 건축분야에서 엔지니어링적인 접근만으로 공간정보를 다루려는 시도는 본능적인 저항에 부딪쳐왔다고 할 수 있다. 일반적인 건물 데이터 모델(building data model)은 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 요소를 공간으로 보고, 건물은 이들 공간의 위계적인 추상체계(abstraction system)로 볼 수도 있다. 이보다는 복합적으로 공간과 구조체의 위계적인 추상체계로 정의할 수도 있다. 후자의 경우 하나의 건물은 하나 이상의 층(공간)과 외벽(구조체)로 구성되고 층은 하나 이상의 실(공간) 또는 호(공간)와 호간 벽체(구조체)로 조직되는 구성체계를 가진다고 볼 수 있다.

이러한 기존의 모델(Björk 1992; 김억 외 1999)은 건축물을 공간 또는 컴포넌트의 위계로 분해하는 객체제항적 방법을 사용하고 있으나 온톨로지 기반 모델링은 개체들 간의 관계에 중점을 두며 더 나아가 여기에 의미구조와 관계의 제약조건을 기술함으로써 보다 복잡한 의미의 프로세싱이 가능하다.

3. 온톨로지의 도입

3.1 온톨로지의 정의

온톨로지는 어떤 개념체계에 대한 정형화된 명세(explicit specification)이다(Gruber 1993). 보다 구체적으로 온톨로지는 어떤 도메인을 구성하는 개념들과 그 개념들이 서로 가지는 관계를 인간은 물론 기계(즉 컴퓨터)가 이해하고 처리할 수 있는 정형화된 수단으로 표현한 것이다. 이를 이용하여 어떤 추론기(reasoner)가 개념들의 관계를 파악하고 지능적인 판단과 처리를 할 수 있다. 온톨로지 방법론이 기존의 AI와 구별되는 점은 온톨로지 발전의 배경에 웹(world wide web)의 발전이 있기 때문이다. 온톨로지는 웹이 궁극적으로 지향하는 semantic web을 구현하는 수단이 된다(Berners-Lee et al. 2001). 하나의 온톨로지는 논의대상의 도메인에서 사용되는 개념들에 대한 정형화되고 명시적인 기술, 즉 클래스(Class) 또는 개념(Concept)과 그러한 개념들의 특성이나 속성을 기술하는 프로퍼티들(Properties, Slots, Roles), 그리고 그 프로퍼티에 대한 제약조건(Facets, Role Restrictions)들로 정의된다(Noy & McGuinness 2001). 온톨로지 구축을 위하여 다양한 도구언어들이 제

시되어왔으나 본 연구에서는 현재 W3C의 표준으로 자리 잡고 있는 웹온톨로지언어(OWL)을 이용한다.

3.2 웹온톨로지언어(OWL)의 정의

웹온톨로지언어(이하 OWL)는 DAML+OIL¹⁾에 기반한 차세대 온톨로지 구현 표준언어이다. DAML + OIL은 기존의 RDF와 RDF 스키마²⁾가 역관계, union, intersection 등의 주요관계를 지원하는 표현력을 결합함에 따른 모델링요소 확장·강화 요구에서 개발되었다. 이 DAML + OIL을 기반으로 클래스(class)와 프로퍼티(property)의 개념 및 그들 사이의 관계가 보다 명료하게 정의되도록 정리한 온톨로지 언어가 OWL이다.

OWL은 OWL-Lite, OWL-DL 그리고 OWL-Full의 세 가지 종류로 나뉜다. OWL-Lite는 단순한 클래스 위계와 제약조건(constraint) 표현에 이용되는 반면, OWL-Full은 언어 자체의 결정가능성이나 전산적 완전성(computational integrity)을 보장하기보다는 언어의 표현력에 중점을 두는 경우를 의도한 것이다. OWL-DL은 표현력에 있어서 두 언어의 중간쯤에 해당되지만, 가장 구별되는 특성은 서술논리(Description Logic)에 기반했다는 점이다. 서술논리는 자동추론을 가능하게 한다. 즉, 클래스 위계를 자동적으로 계산하고 온톨로지가 OWL-DL의 규칙을 준수하는 지를 체크할 수 있다³⁾. 본 연구에서는 OWL-DL을 사용하였다.

3.3 OWL의 구성체계

OWL 명제는 클래스의 구성원(members)들에 관한 사실과 구성원들 간의 관계를 기술하며, 이러한 명제의 집합으로 이루어진 온톨로지는 구문적으로 정의되지 않은 사실의 논리적 유추를 가능하게 한다⁴⁾. OWL 온톨로지는 클래스(classes), 프로퍼티(properties), 그리고 개체(individual)들로 구성되며, 보다 정확히는 다음과 같은 요소들을 포함하고 있다.

- 클래스 간의 택소노미(taxonomy) 관계
- 데이터 프로퍼티, 즉 클래스의 요소인 속성값에 관한 기술
- 객체 프로퍼티, 즉 클래스 요소간의 관계 기술
- 클래스들의 인스턴스
- 속성들의 인스턴스

(1) 택소노미

OWL에서 owl:Thing은 모든 개체(individuals)를 담고 있는 집합(set)을 표현한다. 모든 클래스는 owl:Thing의 서브클래스이다. 즉 owl:Thing은 최상위 클래스에 해당된다. 따라서 건물공간 온톨로지에 있어서 다른 클래스의 하위개념이 아닌 건물(Building), 서브시스템(SubSystem),

공간(Space), 컴포넌트(Component) 등의 클래스조차도 모두 owl:Thing의 서브클래스이다.

OWL 온톨로지에서 클래스는 개체(individuals)들이 속하는 집합(sets that contain individuals)이다, OWL에서는 형식화된(수학적인) 서술을 통해서 클래스의 구성원이 되기 위한 조건을 정확하게 기술하여 표현한다.

여기서 클래스들 간에는 상위클래스(superclass)와 하위클래스(subclass)의 위계관계가 성립하는데 이를 택소노미(taxonomy)라고 한다. 서브클래스는 상위클래스를 특화(specialize)한 것이며 논리학 또는 인공지능 분야에서는 서브클래스가 상위클래스에 '포섭된다'(subsumed)라고 한다(유석인 1998).

예를 들어 건물컴포넌트(BuildingComponent)라는 클래스와 창호(Window)라는 클래스와의 관계에서 창호는 건물컴포넌트의 서브클래스이다. 이는 '모든 창호는 건물컴포넌트이다' 라는 뜻이 된다. 즉 창호의 구성원은 건물컴포넌트의 구성원도 되는 것이며, '창호에 속한다면 그것이 건물컴포넌트임을 묵시적으로 나타낸다' 또는 논리적으로 '창호는 건물컴포넌트에 포섭된다(subsumed)'라고 할 수 있다. 이러한 상위클래스-서브클래스 관계는 OWL-DL의 추론기(reasoner)에 의해 자동적으로 계산될 수 있다. 클래스는 개념(concept)이라고도 표현되는데 보다 정확히 정의하자면 클래스는 개념의 구체적인 표현이다.

OWL에서 서브클래스는 필요조건을 암시한다. 즉, 창호가 건물컴포넌트의 서브클래스라는 것은 모든 창호의 인스턴스는 예외 없이 건물컴포넌트의 인스턴스라는 것을 의미한다. 또는 그것이 창호라면 이는 또한 건물컴포넌트이기도 하다는 것을 암시한다.

(2) 프로퍼티

OWL 프로퍼티에는 객체형 프로퍼티(Object Property)와 데이터형 프로퍼티(Datatype Property)가 있다. 객체형 프로퍼티는 개체와 개체를 연결하며 데이터형 프로퍼티는 개체를 XML 스키마 데이터형이나 RDF 상수에 연결시킨다. 즉, 객체형 프로퍼티는 클래스 요소들 간의 관계를, 데이터형 프로퍼티는 클래스 요소가 취해야 하는 데이터의 형식과 값을 기술한다. 예를 들어 *hasObject* 관계는 객체형 프로퍼티로서 이는 공간(Space) 클래스 요소와 오브젝트(Object) 클래스 요소와의 관계를 기술한다. 반면 *volume* 은 데이터형 프로퍼티로서 이는 공간 클래스 요소의 체적에 대한 값과 형식(예를 들어 *real*값)을 나타낸다.

프로퍼티(properties)는 개체들 간의 관계로서 두 개체를 연결한다. 이러한 관계는 역관계(inverse relation)를 가질 수 있다. 예를 들면 공간(Space)과 사물(Object)의 개체들은 서로 *contains*와 *isContainedIn*이라는 관계들에 의해 연결되면 이들 관계는 서로 역관계이다.

프로퍼티는 프로퍼티 특성을 통해서 더욱 복잡한 의미를 표현할 수 있게 한다. 어떤 주어진 개체에 대하여 그

1) DAML+OIL, <http://www.w3.org/TR/daml+oil-reference>

2) RDF, <http://www.w3.org/RDF/>

3) OWL-Full은 자동추론이 불가능하다

4) 이는 온톨로지에 의한 지식표현의 핵심적 특성이다.

프로퍼티로 연결될 수 있는 개체가 하나 밖에 없을 경우 기능적 프로퍼티(functional property)라고 한다. 예를 들어 개구부(Opening)과 벽체(Wall)사이에는 *belongsTo*와 같은 프로퍼티를 생각할 수 있다. 개구부는 단 하나의 벽에만 속할 수 있으므로 *belongsTo*는 개구부 인스턴스가 하나의 개체에만 연결되는 기능적 프로퍼티이다. *belongsTo*의 역프로퍼티(inverse property)에 해당하는 *hasOpening*의 경우 하나의 벽체에 여러 개의 개구부가 있을 수 있으므로 기능적 프로퍼티가 아니다. 그러나 층(Story)과 그 층의 슬래브(slab)의 경우, 서로 *hasSlab*와 *isSlabOf* 관계의 역관계를 가지며, *isSlabOf*는 기능적 프로퍼티이므로 *hasSlab*는 역기능적 프로퍼티(inverse functional property)이다.

프로퍼티는 추이적인 프로퍼티(transitive)와 대칭적인(symmetric) 프로퍼티가 존재한다. 추이적인 프로퍼티의 대표적인 것으로는 *partOf* 프로퍼티가 있다. 예를 들어 A와 B가 *partOf* 관계로 연결되어있고 B와 C가 *PartOf* 관계로 연결되어 있다면 A와 C사이에도 *partOf* 관계가 성립하는 경우이다. 한편, a와 b가 p의 관계가 성립하고 b와 a도 p의 관계가 성립한다면 이는 대칭 프로퍼티이다. 예를 들어 개구부 oa와 개구부ob가 동일한 벽체 w와 *isOpeningOf* 관계로 연결되어 있다면 oa와 ob는 *hasSibling* 관계로 연결하는 것이 가능한데, 이는 전형적인 대칭 프로퍼티에 해당된다. 어떤 프로퍼티가 추이적이라면 역프로퍼티도 반드시 추이적이다. 또한 추이적인 프로퍼티는 기능적 프로퍼티가 될 수 없다.

프로퍼티 관계의 주체가 되는 개체가 속한 클래스는 도메인(Domain)이라고 하며 대상이 되는 개체가 속하는 클래스의 범위를 범위(Range)라고 한다. 범위는 여러 개의 클래스들로 구성될 수도 있는데 이 경우 프로퍼티의 범위는 클래스들의 합집합(union)으로 표현된다.⁵⁾

그림 2는 본 연구에서 사용된 Protégé의 Ontoviz 플러그인을 이용하여 표현된 온톨로지 그래프이다. 슬래브(Slab)나 벽체(Wall)는 공간과 공간을 수직적으로 또는 수평적으로 분리하는 Divider 클래스의 서브클래스로서 isa 관계에 의해 연결된다. 반면 Opening(개구부)이나 VertHole(슬래브에 속한 수직개구부)는 공간과 공간을 연결하는 Connector 클래스의 서브클래스로 isa 관계에 의해 연결되어 있다. 이러한 isa 관계는 객체형 프로퍼티가 아니라 OWL 텍스트노미에 의해서 구성되는 관계이다. 한편 Divider와 Connector는 서로 *hasOpening* 관계와 *isOpeningOf* 관계로 연결된다. Divider는 0 또는 하나 이상의 Opening을 가질 수 있으며(그림에서 *hasOpening**로 표현), 반면 Connector는 항상 하나의 Divider에만 속

할 수 있다(*isOpeningOf*로 표현). Connector는 2개의 공간을 연결하는 속성을 가지고 있다(*connects**).

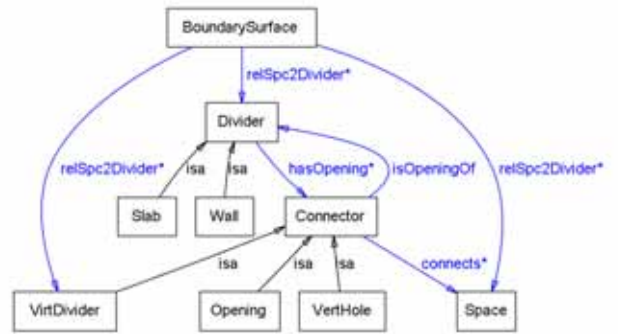


그림 2. 프로퍼티의 사례

(3) 인스턴스

인스턴스(Individuals 또는 Instances)란 논의대상 도메인의 사물 또는 개념(objects)을 표현하는 것이다. OWL에서는 인스턴스명의 유일성(Unique Name Assumption)을 가정하지 않으며 따라서 두개의 다른 이름이 실제로는 동일한 개체를 가리킬 수 있다. 예를 들면 ‘홍길동교수 연구실’과 ‘room_#21423’은 동일한 실을 가리키는 개체명일 경우이다. 따라서 OWL에서는 개체가 서로 다르거나 같음을 반드시 명시적으로 제시해야한다.

(4) 프로퍼티 제약에 의한 인스턴스 표현

존재한정사(existential quantifier)를 이용하여 $\exists hasSlab Slab \Rightarrow hasSlab$ 관계를 표현하면 Slab의 구성원과 적어도 하나와 연결되는 모든 개체들을 지칭한다. 예를 들어 본 연구에서 제시하는 Story(층)의 경우는 *hasSlab* 관계로 Slab를 취하는 유일한 클래스이므로 $\exists hasSlab Slab \equiv Story$ 이다. 그러나 한정사는 실제로 기술되고 있는 클래스의 익명적인 상위클래스를 기술하는 것으로서 보다 정확히 기술하자면 $\exists hasSlab Slab \ni Story$ 이다. 또한, 어떤 회의실(MeetingRoom)이 *hasObject* 라는 관계에 의해 MeetingTable을 취하고 있다면 (이는 어떤 방에 MeetingTable이 있다면 그 방은 회의실이 될 가능성이 있다는 것으로 해석할 수 있다) $\exists hasObject MeetingTable$ 은 실제로 회의실의 상위클래스인 Room이나 그 외 *hasObject* 관계에 의해 MeetingTable을 가지는 모든 것(예를 들면, MeetingTable은 연구실에 있을 수도 있다)에 해당되는 것이다. 즉 $\exists hasObject MeetingTable$ 은 MeetingTable을 (*hasObject* 관계에 의해) 적어도 하나 가지는 모든 인스턴스이다. 이러한 프로퍼티 제약(restriction)은 모든 *hasObject* 관계가 반드시 회의실에만 해당하지 않음을 의미한다. 또한 이러한 표현을 통해 어떤 에이전트(로봇)가 센서를 통해서 어떤 실에 놓인 물체를 MeetingTable로 파악했을 때 그 실이 회의실이거나 그 외 MeetingTable을 가질 수 있는 가능한 모든 실에 대한 확률적인 추측을 할 수 있다. 물론 이러한 확률치는

5) 본 연구에서 온톨로지 모델링 도구로 사용된 Protégé에서 프로퍼티는 슬롯(Slot)이라는 명칭으로 사용되며 서술논리에서는 역할(Role), 그리고 UML과 같은 객체지향적 개념에서는 관계(Relation)라고 불린다.

온톨로지를 통해서 표현될 수도 있지만, 다른 외적인 수단⁶⁾을 이용하여 표현될 수도 있다.

특정 클래스의 구성원에만 프로퍼티를 제한하기 위해서는 전체(universal) 제약을 사용해야 한다. 예를 들어 $\forall hasSlab Slab$ 는 *hasSlab* 관계에 의한 관계는 모두 Slab의 구성원에만 연결되는 개체를 기술한다. 예를 들어 Story는 Slab 외엔 다른 클래스의 멤버를 *hasSlab* 관계로 연결하지 않는다. 또한 제한된 도메인에서 우리는 “화장실의 경우 반드시 변기를 가지지만 욕조, 변기, 세면대를 동시에 가지면 욕실일 수밖에 없다” 라는 지식을 표현하기 위하여, $\exists hasObject ToiletBowl$ 은 결국 욕실이나 화장실을 포함하는 상위클래스를 나타내고, $\forall hasObject (Bahtub \cup ToiletBowl \cup Lavatory)$ 는 욕실을 가리킴을 표현할 수 있다.

4. 공간의 온톨로지화

4.1 기본적 문제

건물공간에 대한 기본적인 개념을 그려보면 우선 건물과 공간이 가지는 관계를 생각할 수 있다. 이는 하나의 건물을 무엇의 집합으로 보느냐의 문제이다. 앞서 논의한 것처럼 건물 자체를 공간의 특수한 형태, 즉 서브클래스로 보고 건물을 구성하는 층이나 실도 모두 공간의 서브클래스로 정의할 수 있다. 이러한 경우 공간(Space)는 계귀적으로 다른 공간을 포함하거나 다른 공간에 포함되는 관계로 개념화 할 수 있다. 이러한 개념화는 실제로 공간 중심의 개념설계를 지원하는 시스템에서는 유용하다. 프로토타입 중심의 시스템에서는 이보다는 컴포넌트 중심의 개념화가 유용하다.

두 번째는 공간과 공간을 구성하는 요소와의 관계이다. 이는 직관적으로는 건물공간과 외피, 혹은 실공간과 벽체와의 관계를 생각할 수 있다. 이러한 문제는 단순히 공간이 벽체의 존재에 종속적인가, 또는 벽체와 관계없이 존재하는 개념인가라는 두 가지 사유에서 연구할 수 있지만, 한편으로는 시스템 수행효율의 관점에서, 인지되지는 않지만 개념적인 요소를 도입할 필요를 고려할 수 있다. 이점에 대해서는 4.2절에서 다룬다.

공간과 관련된 세 번째의 내용은 공간과 공간을 사용하는 사용자와의 관계에 관한 것이다. 공간은 필연적으로 기능을 담는 용기로서 인간의 행위와 밀접한 관계를 가진다. 어떤 공간이 강연장이라는 명칭을 부여받는 것은 내재적인 속성에 의한 것일 수도, 임의적인 부여에 의한 것일 수도 있다. 미지의 공간의 용도와 의미에 대한 지식을 전혀 가지지 않고 있는 에이전트가 어떤 공간을 감지하고 “그 공간이 강연장이다”라고 파악하기 위해서는 우선 그 공간이 가지는 물리적인 크기와 구조를 통한 추론이 필요하다. 대부분의 오피스 공간의 경우 실기능이 처

음부터 부여되는 것이 아니고 사용자가 필요에 따라 기능을 부여한다. 따라서 공간을 특정기능공간의 서브클래스로 구현하기 보다는 특정 기능(function)에 관련되는 공간의 내재적 특성(intrinsic peroperties)과 공간사용주체의 행위(activity)와 기능과의 관계를 온톨로지화 하는 것이 온톨로지 모델링 관점에서 바람직할 것이다.

4.2 공간개념의 온톨로지화

본 연구에서 공간은 건물의 내부공간(IntSpace)과 외부공간(ExtSpace)로 구분된다. 외부공간의 구체적인 정의는 보류한 채, 내부공간은 건물의 구체적인 실공간(Room)으로서 정의된다. 높이가 일정한 내부공간을 가정하여 2차원만 고려했을 때, 그리고 문제를 단순화하기 위하여 곡률을 가진 벽체나 비정형의 벽체는 고려하지 않았을 때 기본적으로 생각할 수 있는 내부공간의 형상조건은 적어도 3개 이상의 벽체에 의해서 둘러싸여야 한다는 것이다. 즉, *isEnclosedBy*(IntSpace, Wall). 여기서 Wall은 공간의 종류에 따라 내벽일수도 있고, 외벽일수도 있다. 외부공간은 어떤 건물을 중심으로 보았을 때 역시 벽체에 의해 경계가 형성되는 개념(객체)이다. 즉 외부공간은 모두 외벽에 의해 둘러싸인다. *isEnclosedBy*(ExtSpace, eWall).

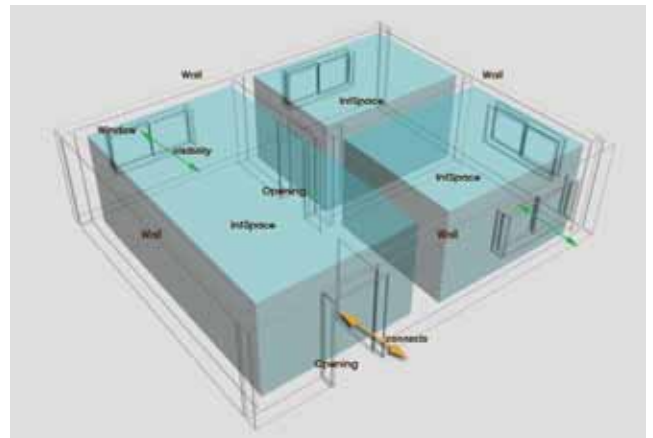


그림 3. 공간, 벽체, 개구부의 관계

그림 3에서 내부공간(IntSpace)는 인지적으로 벽체들에 의해 둘러싸이는 볼륨으로 표현된다. 개구부(Opening)은 벽체에 포함된 컴포넌트로 인지되며, 창호(Window)도 마찬가지이다. 개구부는 공간과 공간을 연결(connects)하는 속성을 가지고 있으며 창호는 공간과 공간을 가시성(visibility)관계로 연결한다. 외부공간과 내부공간이 connects 관계로 연결되는 개구부를 포함하고 있는 벽체는 외벽이며, 내부공간과 내부공간을 연결하는 개구부가 포함되는 벽체는 내벽이 된다.

여기에서 벽체는 본질적으로 두 개의 공간(Space)을 분리하는 분리자(Divider)로서 하나의 벽체가 두 개의 공간과 *isEnclosedBy* 관계에 의해 연결되는 것은 특정 공

6) 예를 들어 Bayesian Network는 개념들 간의 확률적인 인과관계를 그래프화하여 추론하는 환경을 제공한다.

간을 벽체와의 관계에 의해서 추출하려고 할 때 추가적인 추론을 요구한다는 점에서 문제가 있다. 따라서 벽체에 의해 직접 공간이 둘러싸이는 관계보다는 공간과 벽체를 매개하는 경계면(Boundary Surface)의 개념이 도입된다(그림 4). 그림2의 다이어그램에서 보듯이 공간은 relSpc2Divider라는 관계에 의해 벽체나 슬래브와 같은 분리자와 연결되고 있다.

연구의 초기 단계에서 모든 벽체는 실제 시공된 형상과는 관계없이 다른 벽체와 교차하는 경우 그 교차점을 기준으로 별개의 벽체로 분리하여 항상 단일 선분을 유지하도록 하였고 각 단일 벽체에 두 개씩의 경계면이 중속되는 것으로 모델링 하였다. 즉 벽체는 각각 두 개의 경계면을 가지고 있고 연결된 면(surface)이 형성하는 일종의 다각형이 공간을 형성하게 된다. 따라서 공간은 2차원적으로 보았을 때는 경계면들에 의해 relSpc2Divider 관계로 연결되는 개념으로서 내부공간의 경우 그러한 경계면이 적어도 3개 이상 되어야 하며 “외부공간의 경우 그러한 제약이 없다”라고 정의하는 것이 효율적이다. 외부공간은 일반적인 경우 특정 건물의 밖에 존재하는 공간으로서 특정한 경계면이 없거나 해당 건물의 관점에서 볼 때 건물의 외벽의 바깥쪽 경계면들에 의해 싸이는 공간이다.

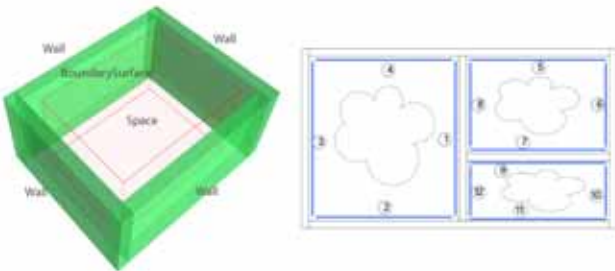


그림 4. 경계면의 개념

이러한 모델링의 문제점은 공간분할에서 큰 의미가 없는 가변적인 파티션이 일반 벽체와 교차할 경우에도 실제로는 하나의 벽체를 여러 개의 벽체와 그 벽체에 부속된 경계면을 파생시키는 것과, 이로 인한 모델의 거대화 및 활용의 어려움. 또한 실제 인지되는 벽체의 특성과의 괴리가 커지는 점 등을 들 수 있다. 또한 벽체에 중속된 개구부(Opening)나 창호(Window)에 대한 경계면의 개념 등이 모호해진다. 또한 IFC 모델과 같은 표준 AEC 교환 모델(IAI 1999)의 방식과도 호환되지 않는 문제점을 가진다.

따라서 본 연구에서 정의한 공간과 벽체와 다음과 같은 관계에 의해서 연결된다.

- 우선 공간은 벽체에 의해서 직접 둘러싸이지 않고 벽체와 독립적으로 존재하는 개념이다. 이는 공간에 실제적인 의미를 부여하고, 공간의 개념을 설계 초기단계에서부터 지원 가능하게 한다.

- 공간은 다각형 또는 곡선을 포함한 폐곡선의 형태를 extrude 한 볼륨, 또는 (그러한 방식으로 표현되지 않는 비정형적인 형상의 경우) boundary representation 에 의해 표현되는 볼륨이다. 그러한 볼륨의 각 면(facet)은 공간과 relSpc2Divider 관계에 의해 연결된다.

- 벽체는 일반적으로 시공된 형상을 그대로 반영한 것을 하나의 단위로 한다. 벽체 볼륨의 각 면(facet)은 공간과 연결됨과 동시에 relSpc2Divider 에 의해 벽체와 연결된다. 이렇게 구현된 공간과 벽체, 그리고 개구부 등의 요소들 간의 관계는 요소들 간의 관계를 위계적이 아닌 나열적 데이터 구조(flat data structure)로 만듦으로서 이를 사용하는 시스템에서 운용이 용이하고 표현에 융통성을 가질 수 있다는 장점이 있다. IFC 모델 등에서도 개구부를 벽체의 하위 컴포넌트로 모델링하지 않고 대등한 솔리드 컴포넌트로 정의하며, 개구부를 구성하는 문짝과 같은 요소도 Filler 라는 개념으로 별도의 클래스화 함으로서 동일한 효과를 도모하고 있다고 사료된다. 즉, 벽체가 CAD시스템의 그래픽환경에서 3차원 모델로 표현될 때 관련된 인스턴스 정보들을 추출하여 벽체로부터 개구부의 볼륨을 보이드(void)화하여 벽체형상을 표현하고 여기에 문짝의 형상들을 추가함으로써 벽체의 완성된 표현을 하는 것이다. 즉 표현(representation)은 데이터 모델을 사용하는 애플리케이션(application)에서 상황에 따라 다이나믹하게 표현한다는 구현철학이다.

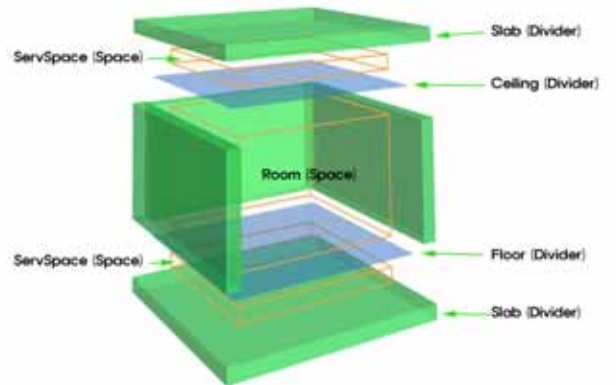


그림 5. 공간 클래스의 세분화

그림 5는 건물내부공간의 모델링에서 고려되는 다른 관점의 문제를 표현하고 있다. 하나의 실(Room)을 예로 들 때 공간은 벽체와 슬래브에 의해 둘러싸이는 (정확히는 그 벽체들과의 경계면에 의해 형성되는) 볼륨이다. 그러나 거주자가 아닌 엔지니어나 건물관리자의 관점에서 하나의 실공간은 내부공간 뿐만 아니라 천정(Ceiling)과 슬래브사이의 공간, 경우에 따라서는 바닥(Floor)과 슬래브 사이의 공간도 표현되어야 한다. 이들 공간 (그림 5에서 ServSpace로 표현)의 도입은 엔지니어링 관점에서의

건물공간 온톨로지가 결코 간단한 일이 아님을 보여주는 일례이다.

또한 설계자가 (기능적)공간으로 정의하였으나 벽체 없이 또는 기둥들에 의해 분리된 공간으로 정의된 경우 물리적으로 연결된 하나의 공간이지만 기능적으로 별개인 두 개의 인접공간을 연결해주는 연결자로서 가상벽(VirtDivider)이라는 개념을 도입한다(그림 6). 이는 연결자(Connector)의 서브클래스로 정의한다.

가상벽은 온톨로지 모델 측면에서 흥미로운 특성을 가진 요소이다. 그 이유는 가상벽은 그 이름이 암시하듯이 기능적으로는 연결자의 서브클래스이지만 형상표현 및 개념적으로는 분리자(Divider)의 속성을 가진다. 따라서 분리자와 경계면이 서로 연결되는 relSpc2Divider 관계를 이용하여 경계면에 연결되며, 형상적 특성은 분리자처럼 벽체의 형상을 가지되 *visible= false*라는 속성을 정의된다. 가상벽 개념을 도입함으로써 공간연결관계와 같은 추론에 있어서 connects 관계만을 추적함으로써 추론할 수 있다(그림 8).

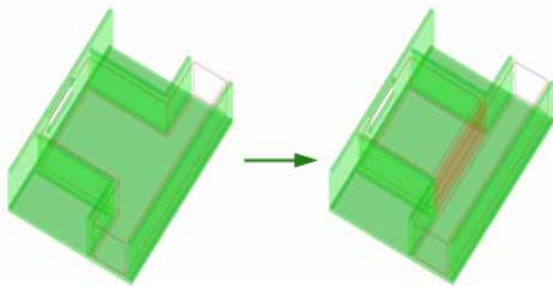


그림 6. 가상벽의 개념

연구 초기단계의 건물공간 온톨로지는 건물은 여러 개의 서브시스템으로 이루어지고 서브시스템 중에서 층(Story)이 공간들로 구성되며 공간은 구체적인 실들로 세분화되는 방식을 취하였다. 그러나 실공간의 기능은 많은 경우 내재적이라기보다는 일시적으로 부여된 임의의 속성일 가능성이 많으며(temporal relations), 특히 실공간의 부여된 기능에 대한 정보가 전혀 없는 상태에서 에이전트가 센서데이터로부터 파악된 오브젝트 등의 특성에 따라 공간의 기능을 추론해내야 하는 연구의 특성상 다음과 같은 공간-행위-오브젝트 관계를 도출하였다. 즉, 행위(Activity)는 관련된 오브젝트가 있다(Activity Association). 따라서 일단의 오브젝트 인스턴스들로부터 가능한 행위의 클래스가 파악되며, 공간은 *hasObjects* 관계에 의해 오브젝트를 가짐으로써 에이전트는 공간에 속한 오브젝트들로부터 공간의 기능(function)을 추론한다. 이로써 어떤 일단의 오브젝트들로부터 가능한 행위가 추론되며 그로부터 그 오브젝트들을 가지고 있는 공간의 기능에 대한 추론을 수행한다(그림 7).

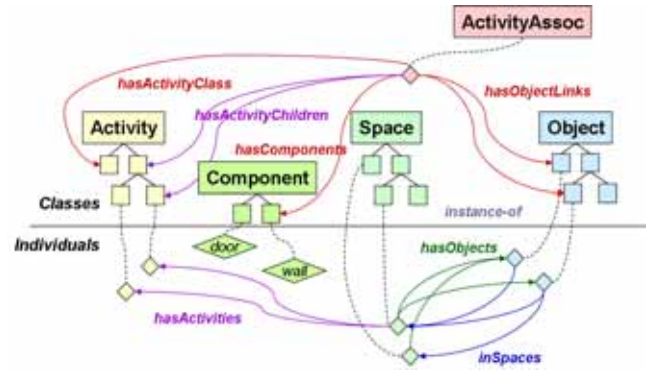


그림 7. 공간-행위-오브젝트의 관계

5. 온톨로지의 응용

그림 8과 그림 9는 본 연구의 환경 온톨로지를 이용한 공간추론사례를 보여준다. Java 기반의 규칙추론(rule inferencing) 도구인 JESS를 이용하여 OWL파일을 파싱하고, 여기에서 분리된 공간을 연결하는 (*connects* 관계)에 의해 연결되는 공간들로부터 공간연결관계 그래프를 생성하고 이로부터 특정 공간 사이의 네비게이션 경로를 추출하여 이를 그래픽 환경에 표현하는 방법을 취하고 있다. 공간연결관계의 추론은 건물공간 지식추론에 있어서 매우 기본적인 것이라고 할 수 있다. 또한 이러한 추론은 OWL과 같은 고수준의 온톨로지가 아니라 공간토폴로지를 표현하는 그래프 데이터구조에 의해서 쉽게 연산 가능할 것이다. 온톨로지 이용의 잠재적 가능성은 선언된 지식으로부터 추론된 지식을 추출해낼 수 있으며, 불완전한 모델을 통해서 접근 가능하다는 것이다.

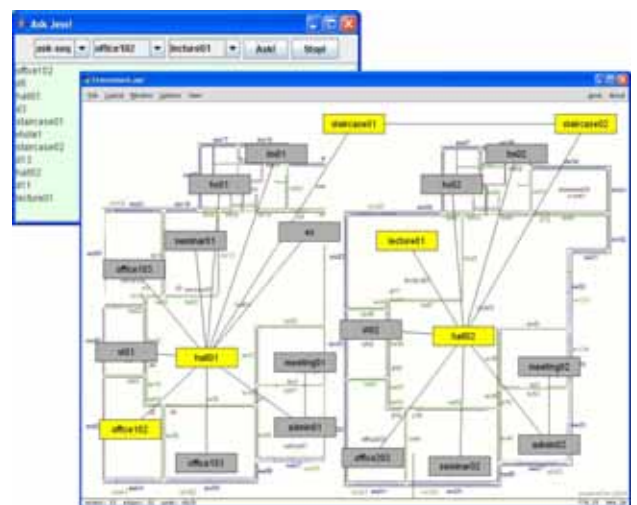


그림 8. JESS와 그래픽환경을 연결한 공간연결관계 추론 가시화의 사례 1

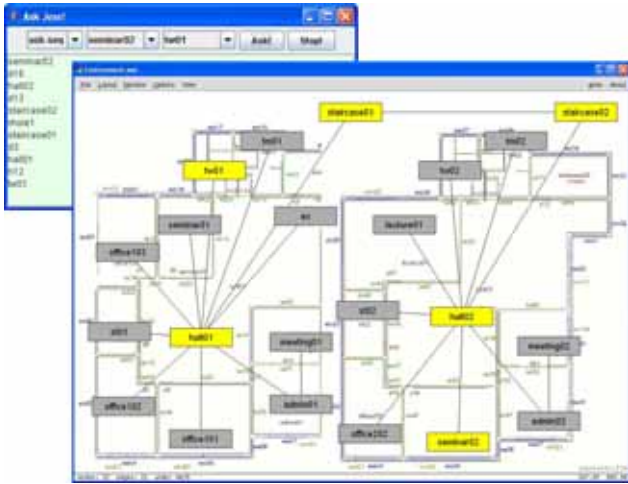


그림 9. JESS와 그래픽환경을 연결한 공간연결관계 추론
가시화의 사례 2

6. 결론

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

건축공간은 물리적인 실체를 가지지 않지만 건축설계에 있어서 설계초기단계에서부터 실체로서 취급되는 개념으로서, 단순히 건물 부재에 종속되는 개념으로만 모델링하기에는 어려움이 있다. 따라서 본 연구에서는 건물공간이 벽체 등의 건물구성요소와 가지는 상호관계를 유연성과 효율성을 고려하여 온톨로지화하였다. 특히, 온톨로지의 디자인 프로세스를 통해서 건물공간이 건물을 구성하는 다양한 컴포넌트와 상호관련되는 의미를 Protégé를 이용하여 온톨로지를 구현하였으며 rule inferencing 도구인 JESS 응용 프로그램을 통해서 이 온톨로지가 활용될 수 있는 기본적인 사례를 제시하였다.

본 연구는 본 연구는 추후 온톨로지의 행위 및 오브젝트 온톨로지와의 통합을 통한 추가적인 구현을 통해서 계속 보완될 것이다.

참고문헌

1. 김억, 최진원, 김성아, 객체지향 가변모델 개념을 적용한 주택설계 자동화에 관한 연구, 대한건축학회논문집, 15권 5호, 1999
2. 유석인, 인공지능원론, 교학사, 1998
3. Björk, B-C., "A Conceptual Model of Spaces, Space Boundaries and Enclosing Structures", Automation in Construction, Vol.1, No.3, pp. 193-214, 1992
4. Berners-Lee, T., J. Hendler, and O. Lassila. The Semantic Web. Scientific American, 2001.
5. Brase, J. & Nejd, W., "Ontology and Metadata for

- eLearning", Staab , S. & Studer, R. (eds.) Handbook on Ontologies, Springer, 200
6. Ding, Y, Fensel, D., Klein, M., Omelayenko, B., & Shulten, E., "The Role of Ontology in eCommerce", Staab , S. & Studer, R. (eds.) Handbook on Ontologies, Springer, 200
7. Eastman C. M. & Siabiris A., "A generic building product model incorporating building type information", Automation in Construction, Vol.3, No.4, pp. 283-304, 1995
8. Ekholm, A. & Fridqvist, S., "Concepts of space in computer based product modeling and design", in Proc. of the 15th ECAADE Conference, Vienna, Sep. 17-20, 1997.
9. Gruber, T.R., "A Translation Approach to Portable Ontology Specification", Knowledge Acquisition 5, pp. 199-220, 1993
10. Hahn, U. & Schultz, S., "Building a Very Large Ontology from a Media Thesauri", Staab , S. & Studer, R. (eds.) Handbook on Ontologies, Springer, 2004
11. IAI, IFC Release 2.0 Specifications. International Alliance for Interoperability, 1999
12. Musen, M.A., "Dimensions of knowledge sharing and reuse", Computers and Biomedical Research, 25, pp. 435-467, 1992
13. 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

(接受: 2005. 3. 31)