

3DF-GML 인스턴스 문서의 데이터베이스 저장을 위한 모듈 개발

Developing a Module to Store 3DF-GML Instance Documents in a Database

이 강 재* 장 건 업** 이 지 영***
Kang Jae Lee Gun Up Jang Jiyeong Lee

요약 최근 다양한 분야에서 GML 응용스키마로서 수행되는 여러 모델이 설계되어 왔다. GML 응용 스키마는 여러 응용 영역에 특화되어 활용되고 있고, GML 표준에 정의되어 있는 각종 기하 타입을 이용하여 객체 타입을 명시한다. GML 인스턴스 문서는 그러한 GML 응용 스키마에 근거하여 생성되어 진다. GML 인스턴스 문서는 일반적으로 엄청난 양의 지리적 객체를 표현하기 위해 많은 저장 공간을 필요로 한다. 따라서 GML 인스턴스 문서를 관계형 데이터베이스에 저장하는 것은 효율적인 관리와 이용을 위해 필수적이다. 관계형 데이터베이스는 상대적으로 사용하기 편리하며, 다양한 분야에서 이용되고 있어 호환성도 높다. 게다가 데이터베이스 구조는 기본적으로 파일 구조보다 많은 양의 데이터를 관리함에 있어 더욱 효율적이다. 현재까지 GML 문서를 저장하기 위해 수많은 연구가 진행되었지만, GML 응용스키마를 기반으로 작성된 GML 인스턴스 문서를 데이터베이스에 저장하기 위한 연구는 적은 편이다. 따라서 본 연구에서는 GML 인스턴스 문서를 관계형 데이터베이스에 저장하기 위한 저장 모듈을 개발하였다.

키워드 : GML 응용스키마, GML 인스턴스 문서, 저장 모듈, 관계형 데이터베이스

Abstract Recently, a variety of GML application schemas have been designed in many fields. GML application schemas are specific to the application domain of interest and specify object types using primitive object types defined in the GML standard. GML instance documents are created based on such GML application schemas. The GML instance documents generally require large volumes to represent huge amounts of geographic objects. Thus, it is essential to store such GML instance documents in relational database for efficient management and use. Relational database is relatively convenient to use and is widely applied in various fields. Furthermore, it is fundamentally more efficient than file structure to handle large datasets. Many researches on storing GML documents have been carried out so far. However, there are few studies on storage of GML instance documents. Therefore, in this study, we developed the storage module to store the GML instance documents in relational database.

Keywords : GML Application Schema, GML Instance Document, Storage Module, Relational Database

1. 서론

실세계의 공간을 모델링하기 위해 공간을 표현하고자 하는 범위는 2차원에서 3차원으로 확장되고 있으며, 이에 대한 연구가 꾸준히 이뤄지고 있다. 더불어

3차원 데이터 모델을 바탕으로 구축된 대용량의 3차원 공간 데이터를 효율적으로 저장하고 관리하려는 연구 또한 계속해서 수행되고 있으며, 이에 대해 현재까지의 3차원 데이터 모델을 비교 분석하는 연구도 진행되었다[26].

† 이 논문은 국토해양부 첨단도시기술개발사업 - 지능형 국토공간정보 구축 및 활용 기술개발 사업과제의 연구비지원(06국토정보B01)에 의해 수행되었음.

‡ 이 논문은 2011년도 정부재원(교육과학기술부 학술연구조성사업비)으로 한국연구재단의 지원을 받아 연구되었음.(NRF-201104302023)

* 서울시립대학교 공간정보공학과 석사과정, kjlee@uos.ac.kr

** 서울시립대학교 컴퓨터과학부 박사과정, atlas@venus.uos.ac.kr

*** 서울시립대학교 공간정보공학과 부교수, jlee@uos.ac.kr(교신저자)

OGC(Open Geospatial Consortium)에서 제안한 GML(Geography Markup Language)은 지리정보의 저장, 표현 및 웹상에서의 교환을 위한 표준으로 XML 기반으로 개발되었다[15]. 일반적으로 여러 분야에서 다루어지는 다양한 지리정보는 다른 환경 및 시스템에서 생성되었기 때문에 지리정보 공유에 어려움이 있다. 이러한 문제점을 해결하기 위해서 GML이 중요한 위치에 있으며, 개방형 GIS를 위해 다양한 분야에서 상호 정보공유가 가능하게 하고, 지리정보 데이터의 접근 및 처리를 가능하게 해준다. 이러한 면에서 GML은 지리정보를 여러 응용 분야의 필요에 따라 달리 구성하여 이용할 수 있고, 호환성이 좋다는 장점을 가지고 있지만, 각 응용분야의 특징에 따른 종합적인 모델 기준 및 제시가 없어 효율적인 활용에 한계가 존재한다.

국내의 공간정보를 표현하기 위한 데이터 모델은 CityGML[5] 및 국토해양부의 3차원 국토공간정보 구축을 통해 개발된 3DF-GML[24] 등이 있으며, GML 응용스키마의 형태로 표현 가능하다. GML 응용스키마는 응용 분야에 특화된 스키마로서 GML을 기반으로 생성되며, 특정 분야에서 데이터 상호 운용성을 보장한다. 응용 분야의 지리 데이터를 표현하기 위해서 특유의 응용 분야에 대해 XML 스키마를 생성하고, GML에서 정의된 기본 객체 타입을 참조하여 나타낸다. GML 응용스키마는 새로운 스키마 언어를 만드는 다른 마크업 언어와는 달리, 존재하는 XML 스키마 모델을 기반으로 한다. 그리고 이러한 GML 응용스키마를 참조 하여 GML 인스턴스 문서의 작성이 가능해진다. GML 인스턴스 문서는 GML 스키마에서 정의된 함수나, 어휘집을 참조하여 작성된 GML 문서들을 일컫는다. 어휘집이란 구조화된 데이터를 생성하기 위해 특정한 요소들의 집합을 만드는 집합체이다.

CityGML은 3차원 도시 모델의 저장 및 상호 정보 교환 접근을 위한 다목적, 멀티스케일 표현이 가능한 모델로서 개발되었다[22]. CityGML의 개발을 위한 기반 연구로 3차원 도시 객체를 표현하기 위해 요구되는 위상학적 개념과 모델에 대한 연구[4], 3차원 도시 모델에서 공간 객체의 표현을 위한 통합 모델, 특히 피쳐로 나타나는 실세계 객체의 기하학적 속성, 위상학적 속성 및 주제 속성에 대한 연구가 진행되었다[21]. 또 건물 모델을 구축하기 위해 의미적이고 공간적인 요소의 이용 및 세밀도의 각 단계별

건물의 세부 표현을 정의하는 연구[22]등이 진행되었다. 활용적 측면에서도 많은 연구들이 진행되어 왔다. CityGML을 활용하여 베를린의 가상 3차원 도시 모델 구축을 위한 활용에서 연구[9]가 진행되었고, 웹에서 CityGML의 서비스가 가능한 공간정보 서버인 CityServer3D에 대한 연구[10] 및 재해 관리 측면에서 3차원 도시 모델인 CityGML의 활용에 대한 연구[23]등 다양한 방면에서 활용에 대한 연구가 수행되었다.

CityGML 및 3DF-GML을 표현하는 GML 응용스키마는 건물, 지형, 식생 등의 피쳐 별 클래스, 클래스 간 상관관계 및 속성정보를 포함한 구조를 공간 및 지리적으로 표현함에 있어 기존의 GML에 비해 더욱 확장되어 설계되었다. 따라서 데이터베이스 스키마를 구축하고, 테이블에 데이터를 저장 시, 그러한 각 피쳐 모델 특징에 따른 많은 사항들을 고려해야 한다.

국외 3차원 도시 데이터 모델인 CityGML은 효율적인 활용을 위해 데이터베이스 저장에 관한 연구가 지속적으로 이루어지고 있는 반면, 3DF-GML은 그에 대한 연구가 미진한 실정이다. 특히 데이터베이스에 GML 인스턴스 문서를 체계적인 구조로 저장함으로써, 파일구조로 관리할 때보다 효율적이며 인코딩 및 디코딩 시 빠른 처리가 가능해진다. 또한 사용자의 활용 측면에서도 다양한 질의 및 공간적 분석이 가능해진다. 그리고 기존 3차원 데이터 모델인 CityGML 인스턴스 문서를 저장하기 위한 데이터베이스 스키마 구조를 활용하여 3DF-GML 인스턴스 문서를 저장할 수 있고, 3차원 객체 저장에 대해 테이블 구성에서 조정 및 이에 따른 저장 모듈의 개발이 필요하다. 그래서 CityGML 저장 방식의 활용으로 3차원 객체에 대해 시멘틱 정보를 활용하여 다양한 공간 질의 및 분석을 가능하게 할 수 있다.

따라서 데이터베이스를 이용한 GML 인스턴스 문서의 저장에 대한 심도 있는 연구가 진행되어야 한다. 데이터베이스 저장을 위해 GML 인스턴스 문서 구조 및 내용을 분석하는 일련의 처리과정을 담당하는 모듈의 개발이 요구되며, 자동으로 GML 인스턴스 문서를 처리할 수 있어야 한다.

본 연구를 통해 GML 인스턴스 문서를 데이터베이스에 저장하기 위한 모듈을 개발하였으며, 국내에서 개발된 3DF-GML을 바탕으로 작성된 GML 인스턴스 문서를 저장 모듈을 통해 데이터베이스에 실

제적으로 저장하는 연구를 수행하였다.

연구의 큰 흐름은 <그림 1>과 같으며, 첫째로 GML 응용스키마에서 각 피쳐 모델의 특징 및 UML 다이어그램을 참조하여 단순화하는 방식을 거쳐 데이터베이스 스키마를 설계한다. 둘째로 개발된 저장 모듈을 이용하여 GML 인스턴스 문서를 데이터베이스 테이블에 저장한다. 생성된 데이터베이스 스키마의 구조에 적합하게 GML 인스턴스 문서가 최종적으로 테이블에 저장된다.

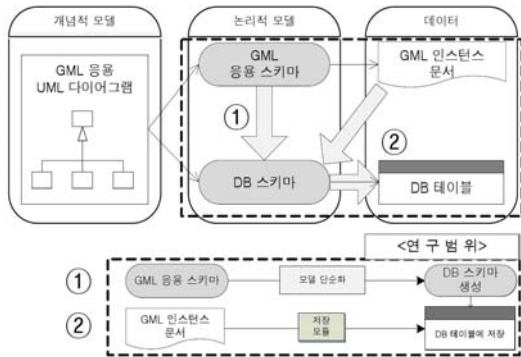


그림 1. 연구 흐름도

2. 관련 연구

2.1 GML 문서 저장 방법

GML 문서 저장은 기본적으로 XML 문서 저장 방법을 활용하고 있으므로[19], 이에 관한 기반연구가 선행되어야 한다. 현재까지 데이터베이스에 XML 문서를 효율적으로 저장하고, 검색하기 위한 연구가 활발히 이루어져왔으며, 특히 관계형 데이터베이스를 이용하여 XML 문서의 구조 정보를 저장하여 효율적인 관리 및 활용을 위한 연구를 중심으로 진행되어 왔다[20, 3].

XML 문서의 구조 정보를 저장하기 위한 방법에는 크게 두 가지가 있다. 첫째로 XML 문서 전체를 BLOB이나 CLOB 형태로 저장하는 비분할 저장 방법, 둘째로 XML 문서를 요소 기준으로 분류하여 테이블에 저장하고, 검색 결과를 반환할 때 저장된 데이터의 트리 구조 정보를 참조하여 데이터를 재구성해 나타내는 분할 저장 방법이 있다[3, 18]. 비분할 저장 방법은 XML 문서를 통째로 저장하고, XML 내에서 각 요소의 위치 값을 부여하기 때문에 검색 시 효율적이지만, XML 문서의 수정 및 갱신 시 해

당 요소뿐만 아니라 다른 요소의 위치 값도 대량으로 수정해야하기 때문에 비효율적이다. 그러나 반대로 분할 저장 방법은 검색 시 여러 테이블을 조인하여 처리해야 하기 때문에 검색 속도가 떨어지는 단점이 존재하지만, XML 문서의 전체 내용 중 일부만 수정되었을 시 관련 요소들의 수정을 통해 비분할 저장 방법보다 효율적인 관리가 가능하다.

분할 저장 방법은 DTD나 XML/GML 스키마의 구조적인 참조 여부에 따라 모델 사상 접근과 구조 사상 접근으로 분류가 가능하다. 첫 번째 방식으로 모델 사상 접근은 XML 데이터의 저장 시 구조적 특징을 참조하는 DTD 정보가 필요하지 않은 방법이다[20]. 모델 사상 접근은 Edge 기반 접근[3]과 Node 기반 접근[14]으로 나뉘며, Node 기반 접근 방식의 모델은 XRel과 XParent가 존재한다. 두 번째 방식으로 구조 사상 접근은 XML 문서를 특정 모델의 스키마를 기반으로 설계된 관계형 데이터베이스 스키마를 통해 저장시키는 형태이다[11, 13]. 대표적인 모델로는 Shared Inlining & Hybrid Inlining[19]과 LegoDB[16]가 있다. 본격적으로 국내에서 개발된 GML 저장 방식은 GParent 가 있으며, 이는 기존의 저장방법인 XParent를 확장한 것으로 공간데이터의 빠른 처리를 위해 공간 및 비공간 데이터가 저장되는 테이블을 개별적으로 구성한다[25].

2.1.1 XParent

XParent는 모델 매핑 기반 접근방법에 따른 XParent 매핑 스키마를 기준으로 테이블에 XML 및 GML 데이터를 저장한다[12]. 모델 매핑 방법에 의한 XML 및 GML 문서의 저장은 생성된 데이터베이스 스키마를 기준으로 수행된다.

XParent는 XML 및 GML 문서의 각 요소, 속성을 노드로 표현하고, 내용(content) 데이터 값으로 나타내는 트리 구조로 구성되며, 각 분류된 문서의 데이터 및 구조 정보는 LabelPath, Element, Data, DataPath 및 Ancestor의 5개의 테이블에 저장된다. LabelPath 테이블은 XML 및 GML 문서에서 각 대응되는 부분들의 경로 정보를 저장하고, 조상 노드로부터 각 노드간의 길이를 Length로 저장한다. Element 테이블은 각 노드마다 DataID를 부여하고, 이를 경로와 연결하는 역할을 수행함과 동시에 동일한 경로를 참조할 때 순서를 부여하기 위해 Ordinal 값을 부여한다. Data 테이블은 XML 및 GML 문서

에서 추출한 속성 값이나 내용(content)을 데이터로서 Value에 저장하고, 해당하는 노드에 대한 PathID 및 DataID를 저장한다. DataPath 테이블은 각 노드에 대해 상·하위 관계를 지정하고, 상위 노드는 PID, 하위 노드는 CID로 나타낸다. Ancestor 테이블은 각 노드의 한 단계 상위 노드부터 최상위인 조상 노드까지 모든 상위 노드들과의 관계를 Level 값으로 표현한다.

2.1.2 GParent

GParent는 XParent와는 달리 비공간 데이터는 Non-Spatial Index 테이블에, 공간데이터는 Spatial Index 테이블에 각각 분류하여 저장되기 때문에 효율적인 공간데이터의 관리가 가능하다. Non-Spatial Index 및 Spatial Index 두 테이블은 데이터 타입을 저장할 수 있다. GParent는 데이터를 탐색하기 위해서 테이블을 조인하여 연산하는 작업이 불가피함에 따라 특정 테이블을 조인하면서 조인에 수반되는 테이블을 설정한다. 그리고 조인에 수반되는 테이블의 개수를 변화시켜 저장 성능이나 데이터 검색, 저장 공간 사용량 등 특화된 목적 및 성능 향상을 위한 몇 가지 상이한 설계 방법이 존재한다[27].

2.2 3차원 공간 데이터의 저장

3차원 공간 데이터는 Geo-DBMS를 활용하여 저장 및 응용하는 연구가 이루어져 왔으며, 3차원 건물, 지표의 Geometry와 Topology를 저장하기 위한 방법 등이 Oosterom(2002), Arens(2005) 등에 의해 연구되어왔다[2, 17, 8]. Geo-DBMS는 대용량의 공간 데이터 셋을 관리하여 여러 사용자들이 동시에 접근 가능하게 해주며, 3차원 좌표로 3D 공간 객체를 표현하게 해준다.

J. Stoter & P. van Oosterom(2002)은 2차원 공간 데이터 및 3차원 공간 데이터를 하나의 DBMS에 저장하는 방안을 제시하였다[7]. 상용데이터베이스의 공간 데이터 타입을 이용하여, 2D와 3D 객체를 정의하고, 3D 볼륨 데이터의 정의가 가능할 수 있게 공간 데이터 타입의 확장을 제안하였다.

Arens(2005)는 3D 공간 객체를 3차원 기하요소를 이용하여 Geo-DBMS에 저장하는 방법을 연구하였다. 간단한 구조의 3D 기본 요소인 Polyhedron을 Geo-DBMS에 저장하였고, 더불어 3D geometry function을 구현하고, 2D 및 3D 공간 인덱싱 적용을

통한 효율성 향상을 증명하였다.

본 연구에서는 저장 모듈의 개발을 위해 GML 문서 저장 방법 중 구조 사상 접근 방식을 채택하며, 이는 GML 응용스키마를 기반으로 한 GML 인스턴스 문서의 데이터베이스 저장에 적합한 방법이다. 그리고 3차원 공간 데이터에서 공간 객체를 데이터베이스에 저장하기 위해 기존 연구된 방식을 활용 및 조정하여 각 피쳐 모델의 자세한 특징 및 시멘틱 정보를 고려한 데이터베이스 스키마 설계를 한다.

3. 저장 모듈 설계

본 연구를 통해 GML 인스턴스 문서를 최종적으로 데이터베이스 테이블에 저장하기 위한 저장 모듈을 제안한다. <그림 2>의 (a)와 같이 GML 인스턴스 문서 저장에 앞서 가장 먼저 GML 응용스키마를 이용하여 데이터베이스 스키마를 생성한다. City-GML 및 3DF-GML과 같은 3차원 데이터 모델은 지형지물의 구분을 위해 기본적으로 피쳐 모델이 구성되어 있다. 피쳐 모델에서 각 주제에 따른 피쳐들은 구조적인 측면에서 모두 다른 구성을 띄고 있기 때문에 효율적인 구조로 데이터베이스에 저장될 수 있어야 한다. 또한 건물과 같은 객체는 많은 면과 더불어 여러 구성 건물로 표현 가능하기 때문에 효율적인 계층 구조로 저장되어야 한다. 따라서 데이터 모델의 단순화 과정을 거쳐 3차원 데이터 모델 기반으로 작성된 GML 응용스키마가 효율적인 형태로 데이터베이스에 저장될 수 있게 해야 한다. 다시 말해 데이터 모델의 단순화 과정은 간결한 데이터베이스 스키마를 생성하기 위한 기반 작업으로써, 추후 저장 및 질의 처리를 효율적으로 수행하기 위해 필요한 작업이다. 모델 단순화 과정에서는 3차원 데이터 모델의 세부적인 특징을 모두 다루며, 도시와 같은 대용량의 데이터에 대해 데이터베이스 접근성의 수행 능력을 개선할 수 있다.

모델 단순화 과정에서는 특정 객체의 구성요소 간 계층 구조의 구현과 구성요소 관계에 대한 질의 수행능력을 향상시키기 위해 부모 요소와 루트 요소의 개념이 고려된다. 앞서 예를 들었듯이 건물과 해당 건물의 모든 구성 요소 간 관계에 대한 질의 시효 과적이며, 부모·자식 간 명시적 표현을 통해 정보 손실 없이 트리 구조 전체의 구현에 도움을 준다. 트리 구조의 적용에 대해서는 추후 저장 모듈 설계 과

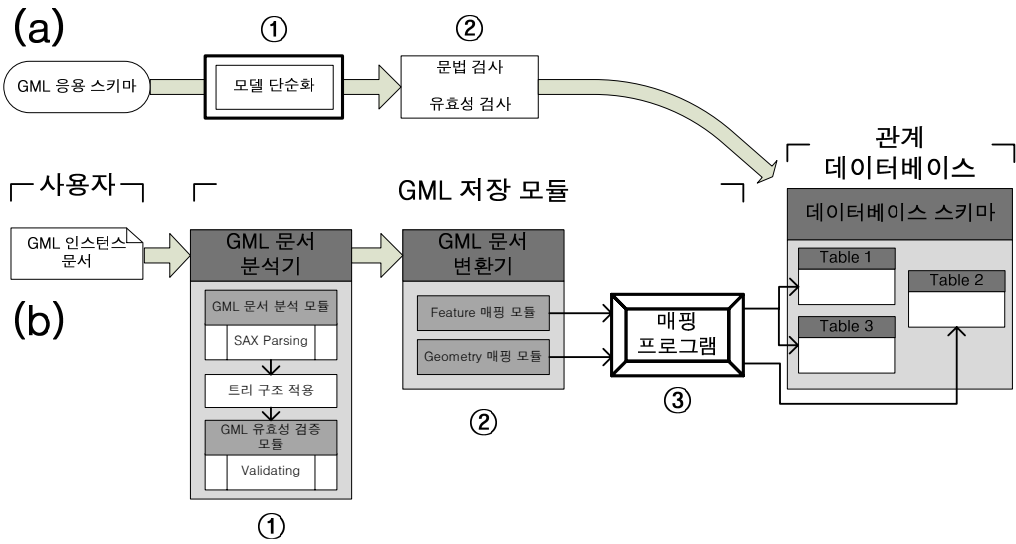


그림 2. 저장 모듈을 이용한 GML 인스턴스 문서의 저장

정에서 자세히 다뤄질 것이다. 또한 3차원 데이터 모델에 특화되어 정의된 데이터 타입을 데이터베이스에서 표현이 가능한 단순한 데이터 형태로 바꾸어 설계한다. 그리고 3차원 데이터 모델의 기하 모델에서 점, 선, 면 등을 표현하는 기본 기하 요소 및 여러 기하 요소로 표현되는 기하 집합에 대한 표현을 어떻게 효율적으로 데이터베이스 스키마에 표현할 것인지 설계한다. 전술한 기본 기하 타입을 저장하고, 공간 질의를 가능하게 하기 위해서 공간 데이터 타입이 고려된다. 단순화 과정을 거쳐 다음으로 문법 검사 및 유효성 검사를 통하여 GML 응용스키마가 XML 스키마의 기준에 명시된 요소 이름이나 데이터 타입을 이용하여 작성되었는지, 또한 마찬가지로 GML의 규칙을 잘 따르고 있는지 검사한다.

다음으로 데이터베이스 스키마를 생성한다. 3차원 데이터 모델의 객체를 구조적으로 저장하고, 질의 처리를 위한 효율적인 조인 연산 등을 고려하여 구축한다. 기존 CityGML 인스턴스 문서의 저장을 위한 데이터베이스 스키마 구조를 활용 및 수정하여 간결한 테이블을 구성한다. 테이블은 모든 공간 객

체에 관한 정보, 즉 메타데이터를 담고 있는 테이블, 공간 데이터를 저장하는 테이블 및 각 피쳐 모델에 따른 속성정보를 저장하는 여러 테이블로 구성된다. 특히 모든 공간 객체의 리스트를 관리하는 테이블에는 각 공간 객체의 고유성을 보장하기 위한 ID를 설계하는데, 여러 GML 인스턴스 문서를 데이터베이스에 저장할 시 ID의 중복이 일어날 수 있기 때문에 이런 점을 고려하여 추가적으로 ID외에 GML 인스턴스 문서가 저장되어 있는 경로를 속성으로 추가한다. 또한 데이터베이스 스키마에서 테이블 간 관계에서 상위 및 하위 클래스에 해당하는 테이블의 관계를 명확히 하고, 어떤 피쳐 모델에 속하는지 소속을 명확히 해주는 속성도 설계한다. GML 응용 스키마로부터 데이터베이스 스키마가 구성되는 (a) 과정은 간단하게 다음 <그림 3>과 같이 나타낼 수 있다.

<그림 3>에서 GML 응용스키마는 추상 부모 클래스인 객체 클래스가 건물, 교통 등과 같은 피쳐로 구성되어 있다. 이를 바탕으로 데이터베이스 스키마를 생성하며, 모든 3차원 객체에 대해서 정보를 관리하는 객체 테이블이 객체 클래스와 대응하여 구축

표 1. 기존 GML 저장과 GML 인스턴스 문서 저장 시 고려사항 비교

저장 방법	고려 사항	공간 및 비공간 데이터 구분	세밀도 모델	피쳐 모델	시맨틱 객체 관계
기존 GML 문서 저장 방법		O	X	X	X
GML 인스턴스 문서 저장 방법		O	O	O	O

된다. 그리고 건물, 교통 등과 같은 피처에 대해 테이블들이 구축되며, 기본적으로 객체 테이블과 연계된다. 각 피처에 대해 생성되는 테이블들은 피처의 특징을 반영하여 여러 테이블로 구성될 수 있다.

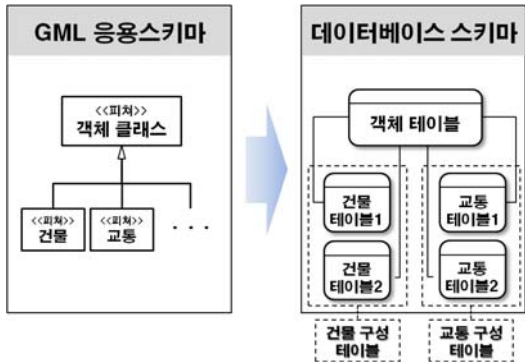


그림 3. GML 응용스키마와 데이터베이스 스키마의 대응관계

<그림 2>의 (b)에서는 실제적으로 GML 저장 모듈을 이용하여 GML 인스턴스 문서를 데이터베이스 스키마 기준으로 생성된 각 테이블에 저장한다. GML 저장 모듈은 GML 문서 분석기, GML 문서 변환기, 매핑 프로그램으로 구성되어 있다. 첫째로 GML 인스턴스 문서를 탐색하고 분석하기 위해 GML 문법 검사, 트리구조 적용 및 유효성을 검증하는 과정을 거치게 된다. 이를 위해 GML 문서 분석 모듈, GML 유효성 검증 모듈을 구성한다. 트리구조의 적용에서는 기존 GML 문서의 저장 방법과는 다른 방식을 적용한다. 기존 GML 문서 저장을 위한 트리구조는 요소, 속성, 내용 간 관계를 구조화하는데 그쳤지만, 본 연구에서는 이와 더불어 피처의 시멘틱한 정보 및 피처를 구성하는 요소 간 복잡한 구조 관계를 표현할 수 있게 설계한다. 두 번째로 공간 데이터를 보다 효율적으로 다루는데 있어 질의 및 수정이 용이하게 이뤄질 수 있게 GML 인스턴스 문서를 공간 데이터 및 비공간 데이터로 분류한다. 이를 위해 Feature 매핑 모듈과 Geometry 매핑 모듈을 구성하게 된다. 세 번째로, 분류된 공간 및 비공간 데이터를 직접적으로 데이터베이스 테이블에 저장하는 단계가 수행 된다. 이를 위해 매핑 프로그램을 제작하였으며, 이를 통해 GML 인스턴스 문서에서 데이터를 추출한 후, 설계된 데이터베이스 스키마 구조를 기준으로 생성된 테이블에 저장한다. 매

핑 프로그램은 기존 GML 문서의 저장방법에 대한 연구와 다르게 3차원 공간데이터 모델의 특징을 반영하여 제작된다. 세밀도에 따른 3차원 공간객체의 기하학적 표현의 변화에 대한 것뿐만 아니라 각 피처 모델별 특성에 따른 공간 및 비공간 데이터의 매핑과 건물과 같은 복잡한 공간객체의 구성요소가 기하학적이고, 시멘틱한 구조로 저장 가능하게 해준다. 연구 과정에 있어 방대하고 복잡한 구조를 갖는 3차원 공간데이터를 효과적으로 저장 및 관리하고 응용하기 위해서 비교적 활용이 편리하고 널리 사용되는 관계형 데이터베이스를 이용한다. 다음으로 저장 모듈에 대해 상세하게 살펴볼 것이며, 그 전에 본 연구에서 설계하는 저장 모듈이 기존 GML 문서의 저장 방법과는 어떤 면에서 다른지 살펴볼 것이다.

3.1 기존 GML 저장 방법과의 비교

기존 GML 저장 방법은 공간 데이터와 비공간 데이터의 분류 및 GML 문서에서 요소, 속성 및 내용 간 계층적 구조를 고려한 것으로 3차원 공간정보를 활용하기에는 한계가 있다. 하지만 본 연구에서 제안하는 저장 방법은 기존의 GML 저장 방법에 관한 연구에 더하여 3차원 공간 데이터 모델을 참고하기 때문에 크게 추가적으로 <표 1>과 같은 사항을 고려한다. 따라서 GML 인스턴스 문서의 저장을 위한 모듈 설계 시 저장과 직접적인 관련이 있는 트리 구조와 매핑 프로그램의 많은 부분이 확장되거나 변화된다. 첫 번째로 세밀도 모델을 고려하여 다양한 기하정보를 이용하면서 지형지물을 단순화하거나 혹은 반대로 자세하게 정의한 데이터를 저장할 수 있다. 따라서 동일한 지형지물을 표현하는 객체를 세밀도에 따라 달리 저장할 수 있다. 두 번째로 3차원 데이터 모델의 클래스에 따른 피처 모델을 고려하여 피처 간 공간 및 비공간적인 특징을 파악하여 데이터베이스에 저장한다. 피처 모델을 고려한 저장방법에 의해 여러 응용분야에서 상호운영이 가능한 단계에 이를 수 있다. 세 번째로 시멘틱 객체 관계의 지원을 통해 건물과 건물을 구성하는 벽, 지붕 등의 의미적인 관계에 대한 정보를 저장한다.

3.2 GML 문서 분석

GML 인스턴스 문서에서 원하는 데이터를 획득하여 활용하기 위해서는 그 문서를 분석하여 문법 규칙을 잘 따르고 있는지 판단해야 한다. 그리고 문서 안

에 정의된 각 요소, 속성, 내용(content) 간의 구조를 계층적으로 나타낸 이후 참조하는 GML 응용스키마를 정확히 준수하여 작성되었는지 판단해야 한다.

3.2.1 문법 검사

데이터베이스에 저장하기 위한 GML 인스턴스 문서를 탐색 및 분석하고, 보다 쉽게 GML 인스턴스 문서를 처리하기 위한 기법으로 DOM과 SAX가 있다. DOM(Document Object Model)방식은 XML과 같은 마크업 형식을 따르는 모든 문서를 처리하기 위한 표준 API이다. 특징으로서 문서를 탐색하기 위해 문서를 사용하기 전, 메모리 맵을 만든다. SAX(Simple API for XML)방식은 이벤트 구동 방식으로서 용량이 큰 GML과 같은 문서에서 사용자가 필요한 정보를 찾을 때마다 이벤트를 발생시켜 재구조화 및 데이터 획득이 가능하다. 결과적으로 본 연구를 위해 SAX 방식을 채택하였다. DOM 방식은 대용량의 GML 인스턴스 문서를 메모리에 로딩하여 많은 저장 공간을 요구하고, 처리 속도가 느린 방식이라 비효율적이기 때문이다. 따라서 SAX Parsing 과정을 통하여 GML 인스턴스 문서의 문법을 검사하는데, 이 과정에서 Well-formed 검증을 수행하여 GML 문서가 갖추어야 할 기본적인 조건을 검사한다.

<그림 4>에서 GML 인스턴스 문서는 예시와 같이 문법이 정확히 준수되고 있는지 SAX Parsing 과정을 통해 검증한다. 삼각 괄호로 표현된 태그 정보들 중 시작 태그가 하나의 끝 태그, 즉 슬래시가 들어간 태그와 짝을 이루는지, GML 문서가 하나의 루트 요소를 갖는지, 요소 이름이 XML의 명명 규칙을 따르는지 등의 문법 검사를 시행하여 올바르게 작성된

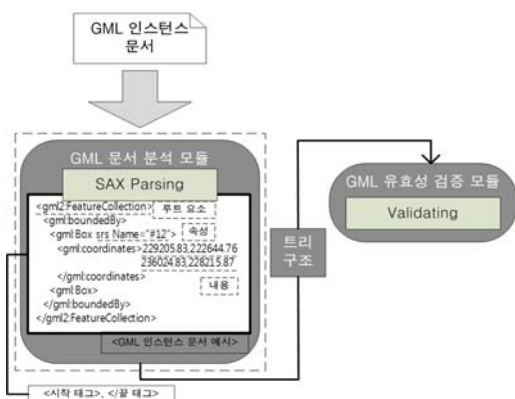


그림 4. GML 인스턴스 문서의 문법 검사

문서인지 증명한다. 따라서 기본적인 규칙에 따라 GML 인스턴스 문서가 작성되어 보편적으로 교환이 가능한지 판단하고, 데이터베이스에 이상 없이 저장하기 위한 기본적인 검사를 시행하는 과정이다. 앞서 설명된 과정은 GML 문서 분석기 중 GML 문서 분석 모듈을 통해 처리된다.

3.2.2 트리 구조 적용

문법 검사를 마친 후 데이터베이스 저장을 위한 색인 테이블 설계 및 효율적인 검색에 용이한 트리 구조를 적용한다. 트리 구조는 기존 GML 문서의 저장에 관한 연구에서부터 적용되어 왔으나, 본 연구에서는 지형지물의 복잡한 구조를 고려한 형태를 표현한다. 특히 현실에서 도심의 건물과 같이 복합적인 부분으로 구성된 지형지물을 데이터베이스에 저장하고, 구조화하기 위한 방법으로 기존 GML 문서 저장과는 다르게 시멘틱한 정보를 고려한다. 더불어 각 피쳐 클래스의 가장 상위개념인 객체 클래스와 피쳐 클래스 간 상속관계 등을 구조화하여 표현하는데 의미가 있다. 트리 구조는 각 클래스를 노드로 표현하며, 각 노드에 ID를 부여하여 참조하는 형식으로 부모-자식 관계를 나타낼 수 있다.

3.2.3 유효성 검증

GML 인스턴스 문서가 문서 타입을 정의한 GML 응용스키마를 참조할 때, 정의한 구조대로 올바르게 작성되었는지를 검사하는 Validating 과정을 거쳐 유효성을 검증해야 한다. <그림 5>에서 나타난 것과 같이 GML 인스턴스 문서의 Validating을 시행하여 요소의 시작 및 종료 태그의 이름 및 내용 타입 등이 GML 응용스키마에서 정의되어 있는 것과 일

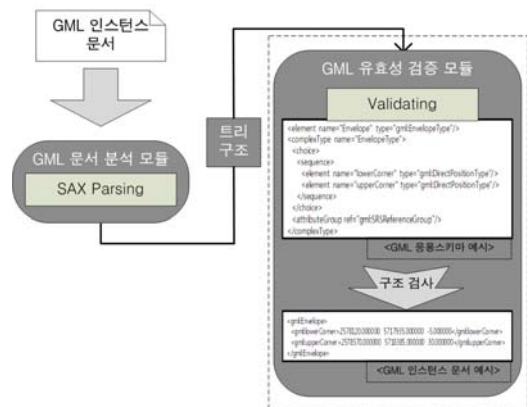


그림 5. GML 인스턴스 문서의 유효성 검증

치하는지 유효성을 검사하며, 이는 GML 문서 분석기 중 GML 유효성 검증 모듈을 통해서 처리된다. 따라서 GML 인스턴스 문서를 매핑하여 데이터베이스에 저장하기 전, 기준에 따른 명확한 문서 구조를 확립하여 데이터베이스에 일관성 있는 데이터를 저장할 수 있게 도와준다.

3.3 GML 문서 변환

데이터베이스에서 공간 데이터의 질의, 수정 등의 효율적인 처리를 위해 GML 인스턴스 문서에서 공간 및 비공간 데이터를 분류시킬 필요가 있다. 따라서 이를 위해 GML 문서 변환기를 구축하며, 비공간 데이터를 다루는 Feature 매핑 모듈과 공간 데이터를 다루는 Geometry 매핑 모듈을 설계한다. GML 인스턴스 문서에서 요소, 속성 및 내용(content)을 파악하여 공간 및 비공간 데이터로 구분하는 과정으로, 특히 좌표와 같은 공간데이터의 효율적인 관리를 위해 Geometry 매핑 모듈을 이용하여 공간 데이터를 개별적으로 처리한다. 공간 데이터는 상용 데이터베이스에서 제공하는 공간 데이터 타입을 통해 비공간 데이터와는 별도로 관리가 된다. 상용 데이터베이스에 저장된 공간 데이터는 공간 인덱스가 부여되어 효율적으로 데이터 처리가 가능해지며, 공간 연산 및 질의 처리 성능이 향상된다.

3.4 매핑 프로그램

GML 인스턴스 문서에서 공간 및 비공간 데이터가 추출되어 속성 및 데이터 타입 등의 구조가 정의된 여러 테이블에 매핑 프로그램을 거쳐 저장된다. 매핑 프로그램은 GML 저장 모듈에서 가장 핵심적인 부분으로 실제적인 데이터의 저장을 담당하고 있다. 기존 GML 저장 방식과는 가장 상이한 부분으로 3차원 공간 데이터 모델의 특징인 피쳐 모델 및 세밀도 등을 표현할 수 있게 프로그래밍 되며, 특히 건물과 같은 복잡한 형태의 지형지물 구조를 시멘틱 정보와 더불어 저장 가능하게 해준다. 매핑 프로그램은 기본적으로 데이터베이스 테이블에 접근하여 데이터 삽입을 위해 각종 SQL문을 수행한다. GML 인스턴스 문서에서 속성 값 또는 내용(content) 값을 매핑 프로그램을 통해 추출하고, 이를 해당하는 테이블에 저장한다. 매핑 프로그램은 특히 각 피쳐 모델 별 특징에 따라 앞서 설계된 데이터베이스 스키마를 기준으로 만들어진다. 예를 들어, 입력된 GML

인스턴스 문서 내용 중 건물과 관련한 부분이 명시되어 있는 요소 이름과 속성 값 등을 매핑 프로그램을 통해 판별하여 건물의 공간 및 비공간 데이터를 테이블에 저장한다. 건물 외에 다른 피쳐 모델도 이러한 과정을 거쳐 공간 및 비공간 데이터가 테이블에 저장된다. 이는 <그림 6>과 같이 나타낼 수 있으며, 매핑 프로그램에서 각 피쳐 모델에 해당하는 처리 부분을 나타내고 있다.

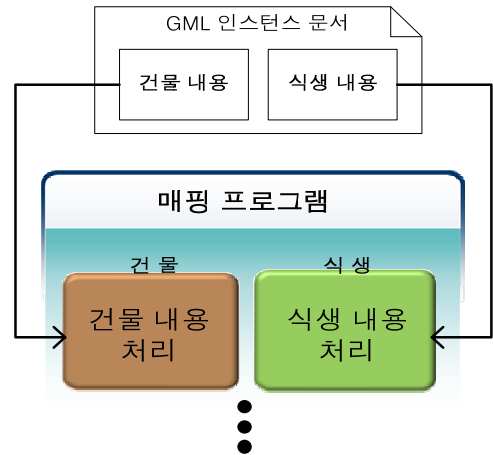


그림 6. 개별 피쳐 모델의 저장을 위한 매핑 프로그램의 구성

또한 앞서 트리구조 적용을 통하여 각 피쳐 모델을 트리구조로 형성하고 노드에 ID를 부여하였는데, 매핑 프로그램을 통해 각 ID를 참조하여 노드 간 연관성을 부여한다. 그리고 부모-자식 간 연관성을 표현하기 위해 Parent_ID 및 Root_ID를 적용한다.

<그림 7>과 같이 피쳐 모델 중 건물을 예로 들면 각 피쳐인 Building 1과 2 중 Building 1은 시멘틱 정보를 담고 있는 여러 Surface로 구성되어 있다.

각 Surface는 여러 단계의 세밀도 모델로 표현되며, 각 세밀도에 따른 Multi Surface는 수많은 Geometry Surface로 구성되어 나타난다. 각 건물은 매핑 프로그램을 통해 ROOT_ID 및 PARENT_ID 라는 속성을 부여받게 된다. PARENT_ID 및 ROOT_ID는 하나의 빌딩이 여러 부속 건물로 이루어졌을 시, 정의되는 하나의 빌딩의 ID를 참조하는 값으로 이용되며, 하나의 건물이 단일한 건물로만 구성되어 있을 시에는 각 건물의 고유의 ID값을 ROOT_ID에만 그대로 이용한다. 각 건물을 구성하는 여러 Surface는 건물의 ID 값을 참조하는 속성을

부여하여 구성 관계를 나타낸다. 또한 Surface는 상이한 단계의 세밀도에 따른 Multi Surface로 정의되며, 각 세밀도 Multi Surface의 ID 값을 참조하는 속성을 두어 관계성을 부여할 수 있다. Multi Surface는 건물과 같은 개념으로 ROOT_ID를 부여하며, 구성 부분이 하나의 Surface일 시에는 Multi Surface의 ID를 그대로 참조한다. 여러 Geometry Surface는 해당하는 상위 레벨 Multi Surface의 ID를 참조하는 ROOT_ID 및 PARENT_ID 속성을 이용하여 구성 요소라는 관계를 나타낼 수 있다. Geometry Surface를 더 세부적으로 나누어 표현하고자 할 시 PARENT_ID 속성 값을 이용하여 하위 단계의 Geometry Surface를 계속해서 생성해 나갈 수 있다.

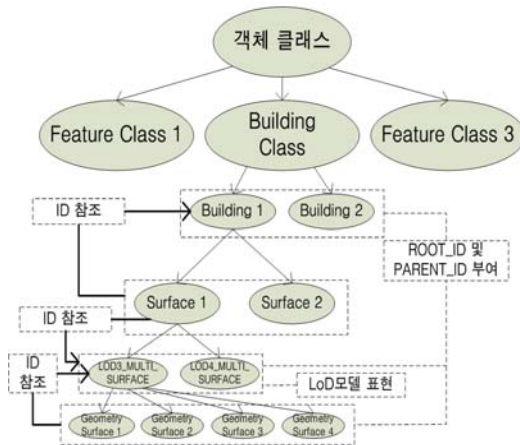


그림 7. 트리구조에서의 ROOT_ID 및 PARENT_ID 부여

4. 저장 모듈을 이용한 적용 사례

본 연구를 통해 개발된 저장 모듈의 실제 수행을 위하여 GML 응용스키마 중 국내에서 '3차원 공간정보 구축 시범사업'을 통해 설계된 3DF-GML의 샘플 데이터를 이용한다. 3DF-GML은 지형지물에 대한 표현이 GML에 비해 풍부해졌으며, 각 피쳐 모델마다 세밀도를 표현한다. 또한 국내 지형지물에 적합한 간결하고 충분한 표현력을 갖고 있으며, 피쳐 모델의 클래스 구조는 <그림 8>과 같다. _F3DObject는 3DF-GML의 Feature 모델 추상 부모 클래스로서 그 하위에는 시설물T, 교통T, 수자원T, 지형T 4개의 주제로 구성된 클래스가 존재하며, 3DF-GML

샘플 데이터가 표현하는 건물에 대한 클래스는 시설물T의 하위클래스로 존재한다. 그 중 본 연구 수행에 있어 중점적으로 다루지는 부분은 건물FC이다.

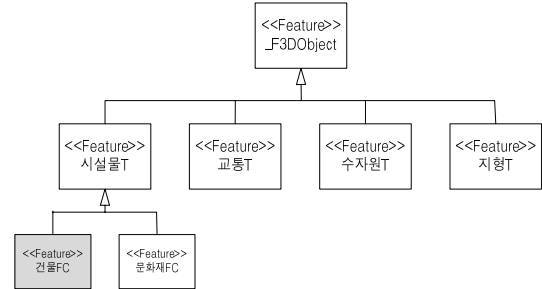


그림 8. 3DF-GML의 Feature 모델 클래스 구조

데이터베이스 스키마 생성과정을 거쳐 저장 모듈을 적용하여 샘플 데이터를 데이터베이스 테이블에 저장하고, 공간 데이터 및 비공간 데이터가 올바르게 저장되었는지 SQL 문을 이용하여 확인하였다. 샘플데이터는 <그림 9>와 같으며, (a)는 3DF-GML 응용스키마를 기반으로 작성된 GML 인스턴스 문서이다. <gml:name>과 같은 요소, gml:id 값과 같은 속성, 좌표로 표현된 3차원 공간데이터와 같은 내용(content)을 확인해볼 수 있다. (b)는 본 연구 수행에 있어 피쳐 모델 중 중점적으로 다루질 5개의 건물을 나타낸 모습이다. (c)에는 부가적인 문, 창문 및 굴뚝 등 설치물을 제외한 기본적인 건물 한 채의 구성요소를 바닥면, 4개의 벽면과 지붕으로 나타낸 GroundSurface, WallSurface 및 RoofSurface가 정의되어 있으며 이는 세밀도 3단계로서 표현된다.

샘플데이터의 저장을 위한 하드웨어 환경은 Intel Core 2 Duo E7500 2.93Ghz CPU, 3GB RAM, 1TB HDD 로 이루어진 시스템을 이용하며, 운영체제는 Microsoft Windows XP Professional이고, 상용 관계형 데이터베이스는 Oracle 11g Release 2를 사용한다. 개발 IDE로 Oracle JDeveloper를 이용하며, 프로그래밍 언어로 Java 6와 데이터베이스에 접근하여 각종 SQL문을 수행하도록 JDBC를 사용한다.

4.1 데이터베이스 스키마 생성

3DF-GML 샘플 데이터를 데이터베이스에 저장하기 위해서 가장 먼저 데이터베이스 스키마를 생성한

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CityModel xmlns="http://is1.cs.pusan.ac.kr/~kgml" xmlns:sch="http://gml_application.xsd">
  <cityObjectMember>
    <!--use:LandUse gml:id="UUID_8ca95cbc-d80b-437a-a448-7bd25dc06adf"-->
    <gml:name>lawn</gml:name>
    <gml:boundedBy>
      <gml:Envelope srsDimension="3" srsName="urn:ogc:def:crs:crs:E"
        <gml:lowerCorner>207.769884109 216.785572052 0.0</gml:lower
        <gml:upperCorner>284.642593384 288.987518311 0.0</gml:upper
      </gml:Envelope>
    </gml:boundedBy>
    <creationDate>2011-07-20</creationDate>
```

(a) 3DF-GML 인스턴스 문서



(b) 건물

	() bldg:GroundSurface	() bldg:RoofSurface	() bldg:WallSurface
1	▾ bldg:GroundSurface		
2		▾ bldg:RoofSurface	
3		▾ bldg:RoofSurface	
4		▾ bldg:RoofSurface	
5		▾ bldg:RoofSurface	
6		▾ bldg:RoofSurface	
7			▾ bldg:WallSurface
8			▾ bldg:WallSurface
9			▾ bldg:WallSurface
10			▾ bldg:WallSurface

(c) 건물 1개의 구성 Surface

그림 9. 3DF-GML 샘플 데이터

다. 데이터베이스 스키마는 3DF-GML 스키마 및 3DF-GML의 UML 다이어그램을 참조하여 생성한다. 첫 번째 과정으로 모델 단순화를 통해 각 피쳐 모델에 대해 테이블 구성 시 이용될 필수적인 정보를 바탕으로 클래스 및 속성의 재구성 작업을 수행한다. 다음으로 3DF-GML 스키마의 문법 검사 및 구조적 유효성 검증을 위하여 SAX 파싱 및 validating 과정을 수행한다. 이를 기반으로 구조적이고 일관성 있는 데이터베이스 스키마를 생성한다. <그림 10>은 3DF-GML 샘플 데이터 중 건물 부분을 저장하기 위한 데이터베이스 스키마이다. 가장 상위인 객체 클래스는 F3DOBJECT 테이블로 나타난다. 그리고 공간 및 비공간 데이터는 SURFACE_GEOMETRY 테이블에 저장되고, THEMATIC_SURFACE 테이블은 시멘틱 정보를 저장하며, SURFACE_GEOMETRY 테이블을 참조한다. SURFACE_GEOMETRY 테이블의 GEOMETRY

속성은 Oracle의 공간데이터 타입인 SDO_GEOMETRY를 이용하여 효율적인 공간데이터 처리를 이끌어낸다.

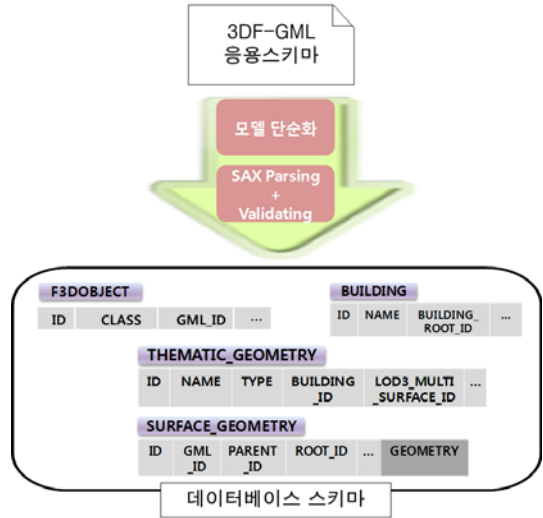


그림 10. 데이터베이스 스키마 생성

4.2 3DF-GML 샘플 데이터의 저장

본 절에서는 구축된 데이터베이스 스키마를 기준으로 3DF-GML의 샘플 데이터를 최종적으로 데이터베이스 테이블에 저장한다. 3DF-GML 샘플 데이터는 저장 모듈에서 문법적 오류 검사, 트리 구조화 및 구조적 유효성 검증 과정을 거쳐 유용한 데이터를 추출하여 공간 및 비공간 데이터로 분류된다. 다음으로 분류된 데이터는 데이터베이스에 의해 각 레코드에 고유의 ID가 부여되어 저장된다. 그리고 매핑 프로그램을 통해 트리 구조를 위한 ROOT_ID 및 PARENT_ID 값이 테이블에 저장된다. 또한 각 레코드에 3DF-GML 인스턴스 문서에서 추출된 gml:id 및 gml:name과 같은 속성에 해당하는 값이 데이터베이스 테이블에서 GMLID, NAME와 같은 속성에 저장된다.

저장 모듈을 이용한 3DF-GML 샘플 데이터의 저장은 <그림 11>과 같다. 저장된 데이터베이스 테이블을 살펴보면, F3DOBJECT, BUILDING, THEMATIC_GEOMETRY, SURFACE_GEOMETRY 4가지가 생성된다. 3DF-GML 인스턴스 문서에서 건물 부분에 관련된 구성 테이블은 기존 CityGML보다 간결해졌으며, 그 이유는 3DF-GML 건물 모델이

아직 실내 구조에 대한 정의가 마련되어 있지 않아 관련 테이블 및 속성 정보가 제거되었기 때문이다.

구성된 테이블 중 F3DOBJECT는 트리 구조에서 제일 상위의 객체 클래스로서 모든 피쳐 모델의 객체 정보를 담고 있다. <그림 11>에서는 각 객체의 ID 값, 객체의 종류에 따른 CLASS 값 및 GMLID 등의 속성 값이 저장된 것을 확인할 수 있다. 무엇보다도 각 건물뿐만 아니라 건물을 구성하는 Surface와 같은 객체 정보까지 저장되어 있다. BUILDING 테이블에는 각 건물에 대한 정보가 저장되며, 건물의 ID, 이름, 부모 ID 및 루트 ID 값이 저장된다. 루트 ID는 F3DOBJECT 테이블에서 각 건물의 ID 값을 참조하였으며, 부모 ID에 값이 저장되지 않은 이유는 각 5개의 건물이 모두 복합적으로 구성되지 않고, 하나의 건물로 정의되기 때문이다. THE-MATIC_GEOMETRY 테이블에는 각 건물을 구성하는 개별 Surface의 구별이 가능한 시멘틱 정보가 저장된다. 기존 GML 모델 저장방법에서 보다 확장된 부분으로 GML 응용스키마를 고려한 테이블 설계를 한다. <그림 11>에서 ID가 1인 'Teststreet1' 건물을 구성하는 바닥면, 지붕면, 벽면들에 대한 ID, NAME, TYPE, BUILDING_ID 및 LOD3_MULTISURFACE_ID 속성 정보 값이 정의되어 있는 것을 확인할 수 있다. BUILDING_ID 속성은 BUILDING 테이블에서 ID 값을 참조하여, 어떤 건물을 구성하는 Surface인지 정의한다. 그리고 LOD3_MULTISURFACE_ID 속성 정보는 세밀도 모델에 따른 건물 면을 나타내는 Multi Surface의 ID 값을

SURFACE_GEOMETRY 테이블에서 참조한다. SURFACE_GEOMETRY 테이블은 각 Multi Surface 및 Multi Surface를 구성하는 Geometry Surface에 대한 정보를 ID, GMLID, 부모 ID, 루트 ID 및 GEOMETRY 속성 값에 담고 있다. 부모 ID 값이 저장되지 않은 레코드들은 특정 세밀도의 바닥면, 지붕 및 벽면을 의미하며, 루트 ID는 해당 레코드 자신의 ID 값을 참조하였다. 부모 ID 값이 존재하는 레코드들은 하나의 Surface를 구성하는 Geometry Surface이다. 부모 ID 속성에는 SURFACE_GEOMETRY 테이블에 존재하는 해당 Surface의 ID 값이 저장된다. 또한 각 Geometry Surface의 좌표 값은 3DF-GML 샘플 데이터에서 매핑되어 공간 데이터타입인 SDO_GEOMETRY의 형태로 저장된다.

5. 결론

GML 인스턴스 문서는 기존의 GML 문서와는 달리 관계형 데이터베이스에 저장하기 위해 피쳐 모델을 고려한 별도의 처리 과정이 요구된다. 또한 일반적으로 DBMS를 이용하여 대용량의 3차원 공간 데이터 저장 시 파일 구조보다 효율적으로 관리가 가능하다. 따라서 본 논문에서는 관계형 데이터베이스에 GML 인스턴스 문서를 저장하기 위한 저장 모듈을 제안한다. 특히 각 기존 GML 문서의 저장과는 달리 피쳐 모델의 시멘틱 정보 및 3차원 객체의 세밀도를 고려한 GML 인스턴스 문서의 데이터베이스

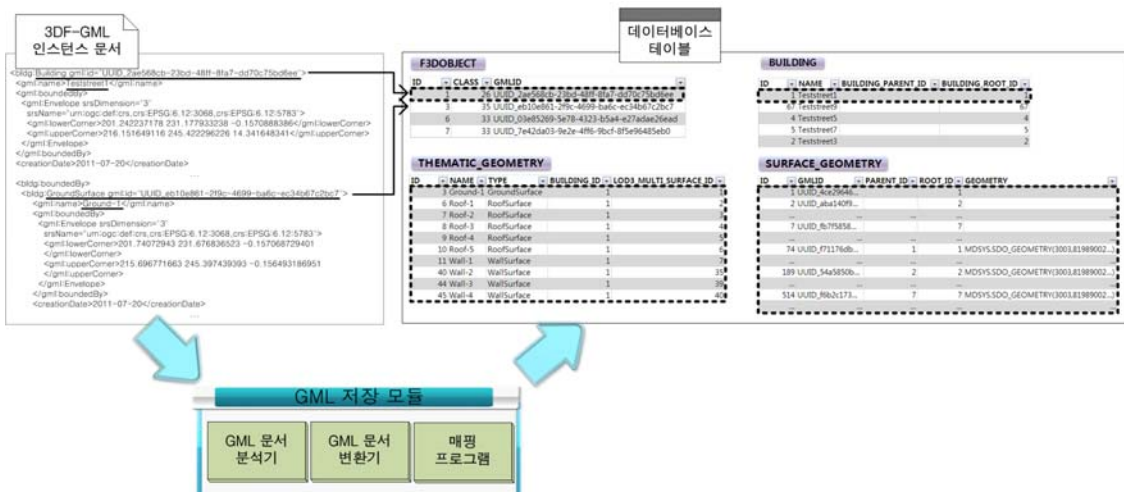


그림 11. 저장 모듈을 이용한 3DF-GML 샘플 데이터의 저장

저장을 연구하였다. 그리고 기존에 연구되었던 CityGML의 데이터베이스 스키마 구조를 활용하여 시멘틱 정보를 활용한 다양한 공간 분석 및 질의가 가능할 것으로 사료된다. 저장 모듈의 기능을 간략히 설명하면 다음과 같다.

1) GML 인스턴스 문서를 분석하여 문서의 교환 및 공유가 가능하도록 적법한 규칙을 준수하였는지 검사하고, 각 엘리먼트들의 계층적인 구조를 트리 구조로 구축한 다음, 표준화된 구조로 명세하기 위한 GML 응용스키마의 형태를 기반으로 작성되었는지 유효성을 검사한다.

2) GML 인스턴스 문서에서 데이터베이스 테이블에 저장될 값 부분에 해당하는 내용(content) 값이나 속성 값을 찾아 공간 데이터 및 비공간 데이터로 분류한다.

3) 매핑 프로그램을 이용하여 각 피쳐 별 특징을 기준으로 생성된 데이터베이스 테이블에 공간 데이터 및 비공간 데이터를 저장한다.

본 연구를 통하여 다양한 분야에서 이용되고 있는 GML 응용스키마의 보다 발전된 형태의 저장 방법을 통한 데이터 관리로 이어질 것으로 기대한다. 본 연구 이후 3DF-GML 인스턴스 문서 저장을 위한 데이터베이스 스키마에 대해 여타 다양한 저장 방법과의 비교, 정량적인 퍼포먼스 테스트 및 개선 방향에 대한 연구가 뒤따라야 할 것이다. 또한 추후 데이터베이스에 저장된 GML 인스턴스 문서의 데이터는 활용 측면에서 많은 연구가 필요할 것이다. 데이터베이스에 저장된 GML 인스턴스 문서의 데이터를 이용하여 원래의 온전한 GML 인스턴스 문서를 생성해내는 연구는 물론, 다른 형태의 데이터 파일로 전환할 수 있는 연구가 이루어질 수도 있을 것이다. 또한 향후 데이터베이스에 저장된 GML 인스턴스 문서의 데이터를 질의 과정을 통하여 사용자가 원하는 데이터의 공간 검색 및 획득에 대한 연구가 이루어져야 할 것이다. 마지막으로 데이터베이스에 저장된 데이터를 바탕으로 시각화 구현 기능 개발에 관한 연구 또한 수행될 수 있을 것이다.

참 고 문 헌

- [1] Arens, C.A., 2003, "Maintaining reality: modeling 3D spatial objects in a GeoDBMS using a 3D primitive", MSc. Thesis, Delft University of Technology, The Netherlands.
- [2] Arens, C.A., J. Stoter, Peter van Oosterom, 2005, "Modeling 3D spatial objects in a geo-DBMS using a 3D primitive", *Computer & Geosciences* 31, pp.165-177.
- [3] Florescu, D., Kossmann, D., 1999, "Storing and Querying XML Data using an RDBMS", *IEEE Data Engineering Bulletin* 22(3), pp.27-34.
- [4] Gröger, G., Plümer, L., 2005, "How to get 3-D for the price of 2-D-topology and consistency of 3-D urban GIS", *Geoinformatica* 9(2), pp. 139-158.
- [5] Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C., 2008. "OpenGIS City Geography Markup Language (CityGML) Encoding Standard", Version 1.0.0, International OGC Standard. Open Geospatial Consortium, Doc. No. 08-007r1
- [6] David Hunter et al., 2002, "Beginning XML 2nd Edition".
- [7] Stoter, J., van Oosterom P.J.M., 2002, "Incorporating 3D geo-objects into a 2D geo-DBMS", FIG, ACSM/ASPRS, April 19-26, Washington D.C. USA.
- [8] Stoter J., Zlatanova S., 2003, "Visualisation and editing of 3D objects organised in a DBMS", In: *Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering*, 22-24 January 2003, Enschede, The Netherlands, 16p.
- [9] Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., Teichmann, K., 2006, "The Virtual 3D City Model of Berlin - Managing, Integrating and Communicating Complex Urban Information", In: *Proc. of the 25th Intern. Symposium on Urban Data Management UDMS 2006 in Aalborg*, Denmark, pp.15-17.
- [10] Haist, J., Coors, V., 2005, "The W3DS-interface of cityserver3D", Kolbe, G.: *Next Generation 3D City Models. Workshop Papers: Participant's Edition*. Bonn, pp.63-67.
- [11] Córcoles, J.E., González, P., 2001, "A specification of a spatial query language over GML", *Proceedings of the ninth ACM Int'l symposium*

- on Advances in Geographic Information Systems, pp.112-117.
- [12] Haifeng Jiang et al., 2000, "Path Materialization Revisited: An Efficient Storage Model for XML Data", the 2nd Australian Institute of Computer Ethics Conference.
- [13] Yuzhen Li, et al., 2004, "GML Storage: A Spatial Database Approach", In: Wang S, et al., (eds) Conceptual Modeling for Advanced Application Domains, Springer Verlag, pp.55-66.
- [14] Yoshikawa, M., et al., 2001, "Xrel: A path-based approach to storage and retrieval of XML Documents using Relational Databases", ACM Transactions on Internet Technology, Vol. 1, No. 1.
- [15] OGC, 2007, "OpenGIS Geography Markup Language (GML), Encoding Standard Version 3.2.1, OGC Doc. No. 07-036".
- [16] Bohannon P., et al., 2002, "Lego DB - From XML scheme to relations : A cost-based approach to XML storage", In: Proceeding of International Conference on Data Engineering.
- [17] Oosterom, P.V., Stoter, J., Quak, W., Zlatanova, S., 2002, "The Balance Between Geometry and Topology", In: D. Richardson & P.Oosterom (eds), Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, pp.209-224.
- [18] Malaika, S., 1999, "Using XML in Relational Database Applications", 15th Int'l Conference on Data Engineering, Sydney, Australia, 167p.
- [19] Jayavel Shanmugasundaram, Gang, H., Kristin Tufte, Chun Zhang, David, J., DeWitt, Jeffrey, F., Naughon, 1999, "Relational databases for querying XML documents: Limitations and opportunities". In VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, pp.302-304.
- [20] Igor Tatarinov, et al., 2002, "Storing and Querying Ordered XML Using a Relational Database System", SIGMOD, pp.204-215.
- [21] Kolbe, T. H., Gröger, G., 2003, "Towards unified 3D city models", In: Schiewe, J., Hahn, M., Madden, M., Sester, M (eds): Challenges in Geospatial Analysis, Integration and Visualization II. Proc. of Joint ISPRS Workshop, Stuttgart.
- [22] Kolbe, T. H., Gröger, G., Plümer, L., 2005 "CityGML - Interoperable Access to 3D City Models", In: Proceedings of the first International Symposium on Geo-Information for Disaster Management, Springer Verlag.
- [23] Kolbe, T. H., Gröger, G., Plümer, L., 2008, "CityGML-3D city models and their potential for emergency response", In: S. Zlatanova, J. Li (Eds.), Geospatial Information Technology for Emergency Response, Taylor & Francis, London, pp. 257 - 274.
- [24] 건설교통부, 2005, "3차원 공간정보 구축 시범사업"
- [25] 김영국, 2006, "GML 문서 저장을 위한 저장 스키마 및 하부 저장 시스템의 설계 및 구현", 전북대학교 컴퓨터 공학과 석사 학위 논문.
- [26] 박세호, 이지영, 2009, "3차원 공간정보 데이터 모델 비교 분석", 한국GIS학회, 제17권 제3호, pp.277-285.
- [27] 장재우, 왕태웅, 이현조, 2007, "효율적인 GML 문서 저장을 위한 저장 스키마의 설계 및 성능평가", 한국공간정보시스템학회, 제9권 제2호, pp.35-53

논문접수 : 2011.07.28

수정일 : 1차 2011.09.27 / 2차 2011.11.30

심사완료 : 2011.12.12

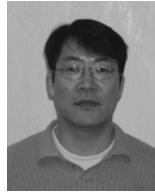


이 강 재

2011년 서울시립대학교 공간정보공학
공학사

2011년~현재 서울시립대학교 대학원
공간정보공학 석사과정

관심분야는 3차원 GIS, 데이터모델링



이 지 영

2001년 The Ohio State Univ. 지리학
이학박사

2001년~2004년 Minnesota State
University 조교수

2004년~2007년 University of North

Carolina at Charlotte 조교수

2007년~현재 기술표준원 지리정보전문위원

2008년~현재 ISPRS WG IV/8 부조직위원장

2007년~현재 서울시립대학교 공간정보공학과 부교수
관심분야는 3차원 GIS, 공간데이터분석, 데이터모델링



장 건 업

2005년 서울시립대학교 컴퓨터통계학
학사

2007년 서울시립대학교 대학원 컴퓨터
통계학 석사

2008년~현재 서울시립대학교 컴퓨터

과학부 박사과정

관심분야는 공간 데이터베이스, 질의처리 알고리즘