

# GML 기반 지리공간 정보 엔코딩 서비스 시스템의 설계 및 구현†

손훈수<sup>0\*</sup>, 김동오\*, 이용수\*\*, 한기준\*

건국대학교 컴퓨터공학과\*, 여주대학교 컴퓨터정보관리과\*\*

{hsson<sup>0</sup>, dokim, kjhan}@db.konkuk.ac.kr, diclee@mail.yeojoo.ac.kr

## Design and Implementation of a Geographic Information Encoding Service System based on GML

Hoon-Soo Son<sup>0</sup>, Dong-O Kim, Yong-Soo Lee, Ki-Joon Han

Dept. of Computer Engineering, Konkuk University, Dept. of Computer Information management, Yeojoo College

### 요 약

지리공간 정보가 사용되는 분야나 환경에 따라 사용되는 표준 데이터 포맷이 다르므로 다른 환경의 지리공간 정보를 활용하기 위해서는 자신의 환경에 맞는 데이터 포맷으로 변환하는 작업이 필요하게 되었다. 또한, 다양한 분야의 지리공간 정보를 상호 운용이 가능하도록 지리공간 정보를 엔코딩해 주는 지리공간 정보 엔코딩 서비스 시스템이 요구되게 되었다.

이에 본 논문에서는 지리공간 정보를 OGC의 GML 3.0으로 엔코딩해 주는 GML 기반 지리공간 정보 엔코딩 서비스 시스템을 설계 및 구현하였다. 특히, 지리공간 정보를 간편하고 효율적으로 엔코딩하기 위해 엔코딩 방법을 정의하는 매핑 규칙을 활용하였다. 본 논문에서 구현한 GML 기반 지리공간 정보 엔코딩 서비스 시스템을 활용해 공간 데이터베이스 시스템인 ZEUS에 구축된 지리공간 정보를 GML 3.0으로 엔코딩하거나, 역으로 GML 3.0을 ZEUS의 지리공간 정보로 엔코딩하여 상호 운용할 수 있다.

### 1. 서 론

최근 지리공간 정보에 대한 관심이 증가함에 따라 지리공간 정보의 활용 분야가 점차 다양해지고, 이를 효율적으로 관리하기 위해 다양한 지리 정보 시스템(GIS : Geographic Information System)이 구축되었다. 시간이 지남에 따라 개인 및 여러 조직에서 기존에 구축해 두었던 각종 지리공간 정보를 개인·조직 나아가 사회·국가 간에 자유롭게 유통 및 활용하려는 요구가 증가하였다. 따라서, 다양한 지리 정보 시스템에 독자적으로 구축된 대량의 지리공간 정보들을 다른 응용 분야의 기초 데이터로 활용하거나 다른 지리 정보 시스템과 지리공간 정보를 공유할 필요성이 발생하게 되었다[7,14,17].

그러나, 지리공간 정보가 위치, 위상(topology), 기하(geometry)등의 공간적 요소와 이들의 특징을 나타내는 비공간적 요소가 통합적으로 구성되기 때문에 일반적인 정보에 비해 월등히 복잡하다. 또한, 지리공간 정보를 처리하는데 있어서 다양한 형태의 데이터 포맷이나 처리 방법이 사용되고 있다. 사용자들은 이러한 특성을 갖는 다양한 환경의 지리공간 정보를 활

용하기 위해서는 자신의 환경에서 사용하는 데이터 포맷으로 변환해야 하는 문제점이 발생하였다[15].

그러므로, 사용자들은 지리공간 정보의 저장 데이터 포맷, 관리 프로그램, 저장 공간의 위치 등을 고려할 필요 없이 다양한 지리공간 정보를 효율적으로 상호 운용할 수 있는 방안이 요구되었다. 또한, 다양한 분야의 지리공간 정보를 상호 운용 가능하게 하기 위해 이질적인 환경의 지리공간 정보를 효과적으로 변환해 주는 지리공간 정보 엔코딩 서비스 시스템이 필요하게 되었다[1,6].

본 논문에서는 현재 지리공간 정보의 상호 운용성을 제공하기 위한 대표적인 방법인 OGC(Open GIS Consortium)의 GML(Geography Markup Language) 3.0[6] 명세를 이용해 ZEUS에 구축된 지리공간 정보를 GML 3.0 문서로 엔코딩하거나, 역으로 GML 3.0 문서를 ZEUS의 지리공간 정보로 엔코딩할 수 있는 GML 기반 지리공간 정보 엔코딩 서비스 시스템을 설계 및 구현하였다. 특히, 지리공간 정보를 효율적으로 엔코딩하는 방법을 정의하는 매핑 규칙을 활용하였다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장의 관련 연구에서는 GML 3.0 명세, DOM, MSXML 파서, ZEUS를 분석한다. 3장과 4장에서는 GML 기반 지리공간 정보 엔코딩 서비스 시스템의 설

† 본 논문은 중소기업청 산학연 공동기술개발 컨소시엄에 의해 지원 받았음.

계와 구현에 대해서 각각 설명한다. 마지막으로, 5장에서는 결론을 기술한다.

## 2. 관련연구

본 장에서는 본 논문에서 지리공간 정보 엔코딩에 사용되는 OGC의 GML 3.0 명세에 대하여 분석하고, GML 3.0 문서를 접근, 분석, 생성하기 위해 사용한 DOM과 MSXML 파서에 대하여 기술한다. 마지막으로, 본 시스템에 데이터 소스로 사용되는 공간 데이터베이스 시스템인 ZEUS를 소개한다.

### 2.1 GML 3.0

OGC에서는 다양한 분야의 지리공간 정보를 손쉽게 상호 운용하기 위해 이질적인 환경의 지리공간 정보를 XML[11]로 엔코딩하여 다양한 분야에 사용 가능하도록 하는 GML 명세를 제시하였다. GML 명세는 1999년에 GML 1.0 발표한 이후, 2001년에 XML 스키마[11,12,13]에 기반을 둔 GML 2.0[3]을 발표하였고, 2002년에 약간의 수정을 거친 GML 2.1.1[4]과 GML 2.1.2[5]를 발표하였다. 그리고, 2003년에 GML 3.0[6]을 발표하였다.

GML 명세는 GML의 사용 목적과 장점, GML에서 사용하는 객체(Object) 모델의 정의, GML로 엔코딩하는 방법, GML에서 제공하는 기본 스키마, 이 스키마를 기반으로 응용 스키마(Application Schema)를 제작하기 위해 지켜야 할 사항, 그리고 응용 스키마와 GML 문서 예제 등을 포함하고 있다.

#### 2.1.1 GML 3.0 스키마

GML 3.0에서는 지리공간 정보를 표현하기 위해 총 21개의 스키마를 정의하는데, 그 중 gmlBase 스키마, feature 스키마, geometry 스키마는 GML 3.0에서의 공간 데이터 모델을 표현하기 위한 스키마이다.

gmlBase 스키마는 GML에서 사용할 \_Object, AbstractGMLType, \_property, AbstractMetaData Type 등과 같은 추상 데이터 타입들을 정의하고 있는 스키마로서 거의 모든 GML 3.0 스키마가 gmlBase 스키마로부터 상속받아 생성된다.

feature 스키마는 Feature나 Feature Collection을 표현하기 위한 스키마이다. 이 스키마는 GML 2.1.2의 feature 스키마의 내용과 유사하나 기본 추상 객체에 대한 부분은 gmlBase 스키마로 분할되었으며 지리공간 속성에 대한 내용도 geometry 스키마로 옮겨졌다.

geometry 스키마는 공간 데이터를 표현하기 위한 스키마이다. 이 스키마는 GML 2.1.2의 geometry 스키마처럼 Geometry 객체를 표현해 주기 위한 스키마이다. 그러나, GML 3.0에서는 새로운 데이터 타입의

도입과 함께 스키마 내부 구조가 변경되었다. geometry 스키마에서 모든 Geometry 객체는 AbstractGeometryType으로부터 파생되어야 하며, 각 차원별로 AbstractGeometryType을 기반으로 몇 가지 추상 데이터 타입을 정의하고 있다. 사용자는 이러한 추상 데이터 타입을 상속받아 새로운 공간 객체를 생성하는 것이 가능하다.

geometry 스키마에서 제공하는 각 차원별 추상 데이터 타입은 다음과 같다. 먼저, 0차원 데이터 타입으로 PointType이 정의되어 있고, 1차원 데이터 타입으로는 AbstractCurveType과 AbstractCurveSegment Type이 정의되어 있다. 2차원 데이터 타입으로는 AbstractSurfaceType과 Abstract SurfacePatchType 이 정의되어 있고, 3차원 데이터 타입으로는 AbstractSolidType이 정의되어 있다. 그리고, 이를 기반으로 AbstractGeometricPrimitiveType, Abstract GeometricAggregateType, GeometricComplexType이 정의되어 있다. 마지막으로, 이러한 추상 데이터 타입 들을 기반으로 Circle, Arc, Circle, Surface, Triangle, Ring, Vector, Knot, BSpline, Bezier 등의 공간 데이터 타입이 정의되어 있다.

GML 3.0 명세는 이외에도 다양한 분야에서 사용하기 위해 여러 스키마를 정의하고 있는데, measures 스키마, valueObject 스키마, unit 스키마, basicType 스키마는 GML 3.0의 기본 스키마에서 사용하는 특정 값, 측정 기준 값, 데이터 타입 등을 정의하는 스키마로서 GML 3.0 문서 내에서 사용되는 값들의 의미나 기준치를 정의하고 있다.

temporal 스키마, direction 스키마, observation 스키마, dynamicFeature 스키마는 LBS 분야에서의 활용을 위해 동적 Feature에 관한 정보를 표현하기 위한 스키마로서 Feature의 위치, 방향, 속도 등의 다양한 정보를 저장하기 위한 프로퍼티를 정의하고 있다.

coverage 스키마, grid 스키마는 ISO 19123을 기반으로 OGC 추상 명세의 Coverage Type을 표현하기 위한 스키마이고, topology 스키마는 ISO 19107의 위상 관계를 나타내는 토폴로지를 정의하기 위한 스키마이며, defaultstyle 스키마는 GML 3.0 문서의 기본 스타일 등을 표현하기 위한 스키마이다.

#### 2.1.2 GML 3.0 응용 스키마

사용자가 표현하려는 다양한 지리공간 정보의 데이터 구조를 GML 명세에서 제공하는 기본 스키마로는 표현할 수 없으므로, 사용자는 GML 명세에서 제공되는 여러 스키마를 기반으로 실제 GML 3.0 문서에서 참조할 GML 3.0 응용 스키마를 제작하여 사용한다.

GML 3.0 명세에서 제시하는 GML 3.0 응용 스키마

생성 규칙으로는 Features 나 Feature Collections를 정의하는 방법, Coverages를 정의하는 방법, Observations를 정의하는 방법, Dictionaries 나 Definitions를 정의하는 방법, Coordinate Reference System을 정의하는 방법, Values를 정의하는 방법 등이 있다.

GML 3.0 응용 스키마 생성 규칙에서는 응용 스키마에서 사용자가 네임 스페이스 및 기타 설정 시 주의 사항, GML 엘리먼트 생성 규칙, 엘리먼트 정의 시 Import해야 할 스키마 등을 정의하고 있다.

## 2.2 DOM & MSXML 파서

본 논문에서는 MSXML 파서를 이용해 GML 3.0 문서를 파싱하여 트리 구조로 구성하고, 이러한 트리 구조에 접근하기 위한 인터페이스인 DOM(Document Object Model)[8,9]을 이용해 GML 3.0 문서에 접근한다.

DOM 인터페이스는 트리 조작에 필요한 각종 함수들을 제공함으로써 응용 프로그램이 쉽게 노드 탐색을 하도록 도와주며, XML 문서의 구조 변경과 같은 다양한 트리 조작이 가능하도록 해준다.

MSXML 파서는 XML 문서의 접근 방법을 지원하는 Microsoft XML 파서이다. MSXML 파서는 IE 6.0에 포함되어 있는 것으로, DOM Level 1 Core 사양과 확장 기능이 추가된 COM 컴포넌트로 되어 있다[2].

## 2.3 ZEUS

ZEUS는 객체 관계형 데이터베이스 엔진에 공간 데이터 처리 기술을 결합시킨 공간 데이터베이스 엔진이다[16]. ZEUS는 데이터베이스의 기본 타입으로 Point, Simpleline, Polyline, Polygon, Rectangle, Circle, Hpolygon, Shape의 공간 데이터 타입을 지원하고 있으며, 공간 클래스들은 모두 special\_object\_class를 상속 받아 생성하도록 되어 있다. 본 논문에서는 다양한 데이터베이스 시스템 중 공간 데이터 타입 및 공간 연산을 지원하는 ZEUS의 공간 데이터베이스를 이용하여 손쉽게 대량의 지리공간 정보를 GML 3.0 문서로 상호 변환하기 위한 데이터 소스로서 활용하고 있다.

## 3. 시스템 설계

본 장에서는 GML 기반 지리공간 정보 엔코딩 서비스 시스템의 모듈 구성을 살펴보고, 각 모듈에 대해 설명한다.

### 3.1 시스템 모듈 구성

그림 1은 GML 기반 지리공간 정보 엔코딩 서비스 시스템의 전체 시스템 모듈 구성을 보여준다.

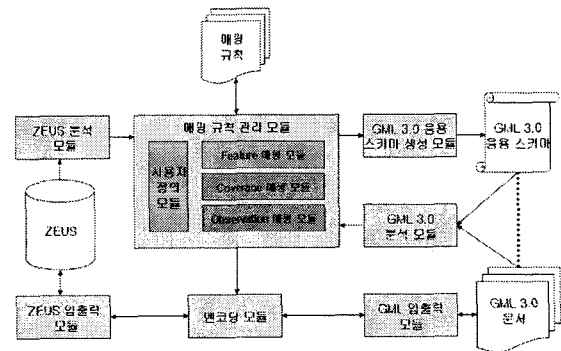


그림 1. 전체 시스템 모듈 구성

이 시스템은 ZEUS를 분석하는 ZEUS 분석 모듈, ZEUS의 지리공간 정보를 입출력하기 위한 ZEUS 입출력 모듈, 매핑 규칙을 생성하고 관리하는 매핑 규칙 관리 모듈, ZEUS의 지리공간 정보를 GML 3.0으로 엔코딩하거나 GML 3.0을 ZEUS의 지리공간 정보로 엔코딩하는 엔코딩 모듈, GML 3.0 문서가 참조하는 응용 스키마를 생성하는 GML 3.0 응용 스키마 생성 모듈, GML 3.0 응용 스키마나 문서를 분석하는 GML 3.0 분석 모듈, GML 3.0 문서의 데이터를 입출력하는 GML 3.0 입출력 모듈로 구성된다.

본 시스템에서는 매핑 규칙을 사용하는데, 매핑 규칙은 GML 3.0 문서를 엔코딩하는데 필요한 정보를 정의한 것으로서 엔코딩할 때의 변환할 데이터 소스의 구조와 변환될 GML 3.0 문서의 구조에 대한 정보 등을 가지고 있으며 각 구조 간의 매핑 관계가 표현되어 있다.

특히, 매핑 규칙은 분석 및 활용이 쉽도록 XML로 작성되었으며, ZEUS의 지리공간 정보와 GML 3.0간의 상호 엔코딩할 때 매핑될 엘리먼트와 데이터 타입에 대한 정보와 지리공간 정보 엔코딩에 필요한 시스템 정보를 가지고 있다.

본 시스템에서는 매핑 규칙을 생성하기 위해 UML 인터페이스를 사용하였다. 따라서, 사용자들은 쉽게 매핑 규칙을 생성할 수 있다. 이렇게 생성된 매핑 규칙에 따라 사용자가 원하는 GML 3.0 응용 스키마나 GML 3.0 문서가 생성되는데, 이 문서는 바로 활용되거나 다른 시스템에 전송될 수 있다.

#### 3.1.1 ZEUS 분석 모듈

ZEUS 분석 모듈은 ZEUS에 저장된 지리공간 정보의 구조 정보를 제공하는 모듈로서 ZEUS의 공간 클래스와 어트리뷰트 정보를 가져오는 기능을 가지고 있다.

ZEUS 분석 모듈은 GML 기반 지리공간 정보 엔코

딩 서비스 시스템에서 매핑 규칙을 생성하기 위해 지리공간 정보인 ZEUS의 공간 클래스만을 추출하는데, ZEUS에 속한 클래스 중에서 공간 데이터 타입인 Point, Simpline, Polyline, Polygon, Circle, Rectangle, Hpolygon, Shape를 어트리뷰트로 갖는 공간 클래스인 Point\_class, Sline\_class, Pline\_class, Polygon\_class, Circle\_class, Rect\_class, Hpolygon\_class, Shape\_class의 자식 클래스만을 가져온다.

### 3.1.2 매핑 규칙 관리 모듈

매핑 규칙 관리 모듈은 ZEUS 분석 모듈에서 얻은 ZEUS의 지리공간 정보를 사용자에게 보여주고, ZEUS에 구축된 지리공간 정보를 GML 3.0 문서로 인코딩하거나, 역으로 GML 3.0 문서의 지리공간 정보를 ZEUS로 인코딩하는데 필요한 매핑 규칙을 생성하고 관리하는 모듈이다. 매핑 규칙 관리 모듈은 사용자 정의 모듈, Feature 매핑 모듈, Coverage 매핑 모듈, Observation 매핑 모듈로 구성된다.

사용자 정의 모듈은 ZEUS의 공간 클래스와 어트리뷰트 정보를 사용자에게 보여주고 사용자의 입력을 받아들이는 모듈로서 ZEUS의 클래스 및 어트리뷰트와 매핑될 3.0 엘리먼트 및 데이터 타입을 입력받거나 갱신한다.

Feature 매핑 모듈은 지리공간 정보를 Feature Collection 모델로 인코딩하기 위한 정보를 생성하는 모듈이다. 이 모듈은 GML 3.0 지리공간 정보 인코딩 서비스 시스템에서 데이터 소스로 사용되는 ZEUS의 공간 클래스 구조를 사용자의 입력을 토대로 GML의 Feature로 매핑해 주는 기능과 ZEUS와 GML의 공간 데이터 타입을 매핑해 주는 기능을 가지고 있다.

Coverage 매핑 모듈과 Observation 매핑 모듈은 지리공간 정보를 Coverage나 Observation 객체로 인코딩하기 위한 정보를 생성하는 모듈로서 기능은 Feature 매핑 모듈과 같다.

### 3.1.3 GML 3.0 응용 스키마 생성 모듈

GML 3.0 응용 스키마 생성 모듈은 GML 3.0 문서의 구조를 나타내는 GML 3.0 응용 스키마를 생성하기 위한 모듈이다. 생성된 GML 3.0 응용 스키마는 해당 GML 3.0 문서가 참조하게 된다.

GML 3.0 응용 스키마 생성 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙과 GML 3.0 명세에서 제시한 GML 3.0 응용 스키마 생성 규칙에 따라 ZEUS의 공간 클래스 중 사용자가 원하는 공간 클래스와 어트리뷰트들을 GML 3.0 응용 스키마로 생성해 준다. 그러나, 이미 구조가 파악되어 사용 중인 GML 3.0 문서에 대해서는 반드시 GML 3.0 응용 스키마가 필요하지는 않다.

### 3.1.4 GML 3.0 분석 모듈

GML 3.0 분석 모듈은 GML 3.0 응용 스키마나 GML 3.0 문서를 분석하기 위한 모듈이다.

GML 3.0 분석 모듈에서는 GML 3.0 응용 스키마나 GML 3.0 문서에서 필요한 정보를 추출하여 매핑 규칙 관리 모듈에 전달해 주거나, GML 3.0 문서가 참조하는 GML 3.0 응용 스키마를 따르는지를 검사한다.

### 3.1.5 ZEUS 입출력 모듈

ZEUS 입출력 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 인코딩 모듈에서 필요한 ZEUS의 지리공간 정보를 입출력하기 위해 ZEUS의 공간 클래스와 해당 어트리뷰트들의 값을 입출력하는 모듈이다.

### 3.1.6 인코딩 모듈

인코딩 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 ZEUS의 지리공간 정보를 ZEUS 입출력 모듈을 통해 출력하여 인코딩한 후 GML 입출력 모듈에 전달해 주는 과정과 GML 3.0 문서의 데이터를 GML 3.0 입출력 모듈을 통해 출력하여 인코딩한 후 ZEUS 입출력 모듈에 전해 주는 과정을 담당하는 모듈이다.

인코딩 모듈에서는 ZEUS의 공간 클래스를 GML 객체로 인코딩하거나 GML 객체를 ZEUS의 공간 클래스로 변환한다. 또한, ZEUS의 공간 데이터 타입을 GML의 Geometry로 변환하거나 GML의 Geometry를 ZEUS의 공간 데이터 타입으로 변환한다.

### 3.1.7 GML 3.0 입출력 모듈

GML 3.0 입출력 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 인코딩 모듈에서 필요한 GML 3.0 문서의 지리공간 정보를 입출력하기 위해 GML 3.0 문서의 해당 엘리먼트를 입출력하는 모듈이다.

## 4. 시스템 구현

본 장에서는 GML 기반 지리공간 정보 인코딩 서비스 시스템에서 지리공간 정보를 인코딩하는 과정을 살펴보고, 시스템의 각 모듈 별 수행 모습과 처리 과정을 설명한다.

### 4.1 ZEUS -> GML 3.0

그림 2는 GML 기반 지리공간 정보 인코딩 서비스 시스템에서 ZEUS를 GML3.0으로 인코딩하는 과정을 보여주고 있다.

그림 2에서 보듯이 지리공간 정보를 추출하기 위해 ZEUS 스키마 분석 과정을 거쳐 ZEUS의 공간 클래스를 파악한 후, 필요한 매핑 규칙이 존재할 경우 편집하여 사용하고 존재하지 않을 경우 새로운 매핑 규칙을 생성하게 된다. 다음, GML 3.0 문서에서 참조할

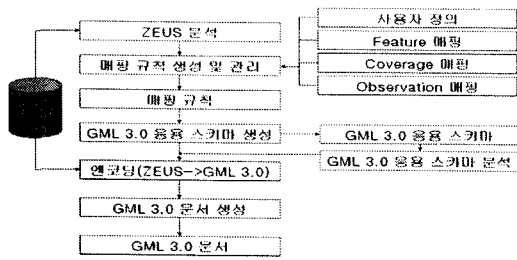


그림 2. ZEUS -> GML 3.0 과정

GML 3.0 응용 스키마를 생성한다. 그리고, 매핑 규칙에 따라 ZEUS 입출력 모듈을 통해 ZEUS 공간 클래스의 공간-비공간 데이터를 추출하여 GML 3.0으로 엔코딩하여 GML 3.0 문서를 생성하게 된다.

#### 4.1.1 ZEUS 스키마 분석

본 시스템에서는 ZEUS의 지리공간 정보를 GML 3.0으로 엔코딩하기 위해서 먼저 ZEUS의 데이터 소스 중 사용자가 원하는 공간 클래스가 들어있는 데이터 소스를 찾아내는 과정을 거치게 된다.

그림 3은 ZEUS 데이터 소스의 정보를 보여주고 선택하기 위한 ZEUS 스키마 분석창을 보여 주고 있다. 사용자가 ZEUS 스키마 분석 다이얼로그의 데이터 소스 리스트에서 원하는 하나를 선택하면, 선택된 ZEUS 데이터 소스에 속해 있는 공간 클래스들의 리스트를 가져와 보여준다. 그림 3에서 현재 ZEUS에 등록되어 있는 데이터 소스는 "testdb", "seoulC" 임을 알 수 있다.

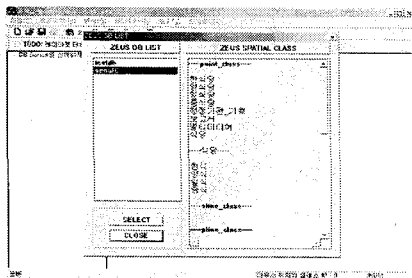


그림 3. ZEUS 스키마 분석 다이얼로그

#### 4.1.2 매핑 규칙 생성

그림 4는 매핑 규칙을 생성하는 GML 3.0 엔코딩 다이얼로그를 보여준다.

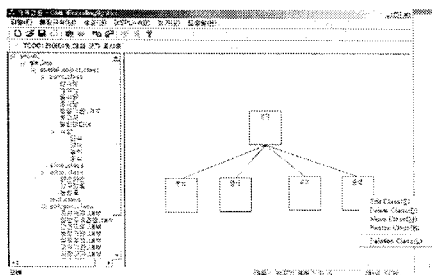


그림 4. GML 3.0 엔코딩 다이얼로그

그림 4에서 왼쪽 창은 ZEUS 스키마 다이얼로그에서 사용자가 선택한 데이터 소스의 공간 클래스 정보를 사용자가 보기 쉽게 트리 형태로 보여주는 창이다.

매핑 규칙을 생성하기 위해 GML 3.0 엔코딩 다이얼로그에서 사용자가 엔코딩하려는 공간 클래스를 왼쪽 창에서 오른쪽 창 화면으로 끌어다 놓으면 해당 공간 클래스의 속성을 갖는 클래스가 생성되게 된다. 그림 4에서는 "식당" 클래스와 "식당" 클래스의 하위 클래스인 "한식", "양식", "일식", "중식" 클래스가 생성된 것을 볼 수 있다.

각 클래스는 클래스 이름과 속성을 가지고 있으며, 사용자는 그림 5의 클래스 편집 다이얼로그를 이용해 해당 클래스에서 원하는 클래스 및 속성을 편집할 수 있다.

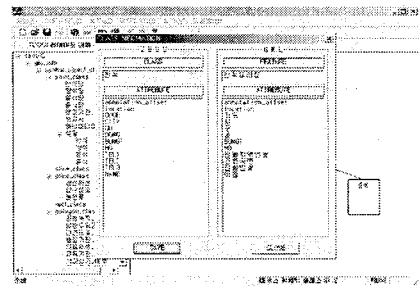


그림 5. 클래스 편집 다이얼로그

클래스 편집 다이얼로그는 해당 클래스에서 원하지 않는 속성을 제거할 수 있으며, 오른쪽 GML 정보 리스트에서 GML 매핑에 필요한 GML 엘리먼트 명을 갱신할 수 있다. 그리고, 이 클래스와 속성들이 GML 3.0에 Feature의 공간-비공간 속성들이 된다.

그림 6의 문서 생성 다이얼로그는 매핑 규칙, GML 3.0 응용 스키마, GML 3.0 문서를 생성하기 위한 다이얼로그이다. 그림 6에서 매핑 규칙의 "MAKE" 버튼을 누르면 매핑 규칙이 생성된다.

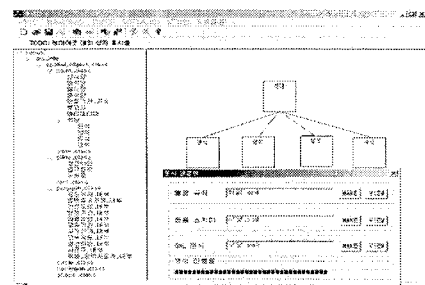


그림 6. 문서 생성 다이얼로그

그림 7은 ZEUS의 공간 클래스인 "식당"을 GML로 엔코딩하기 위한 "식당.mrd" 파일을 IE 6.0으로 본 모습이다. 이 매핑 규칙은 "Class" 엘리먼트의 속성 zeusName 값으로 ZEUS 공간 클래스인 "식당"을 가지고 있고, 속성 gmlName 값도 "식당"을 가지고 있

다. 그리고, 세 번째 속성 classType 값은 “식당” 클래스에 해당하는 공간 클래스인 “point\_class”를 가지고 있다.

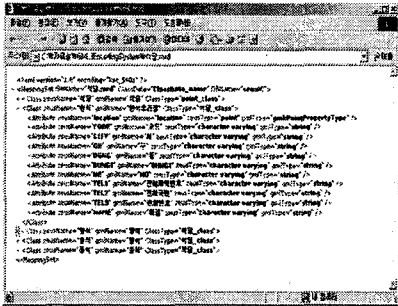


그림 7. 매핑 규칙

#### 4.1.3 GML 3.0 응용 스키마 생성

GML 3.0 응용 스키마는 생성된 매핑 규칙에 따라 GML 3.0 응용 스키마 생성 모듈을 통해 생성될 수 있다. GML 3.0 응용 스키마를 생성하기 위해 그림 6의 문서 생성 다이얼로그에서 GML 3.0 응용 스키마 이름을 입력하고 “MAKE” 버튼을 누르면 GML 3.0 응용 스키마가 생성되며, “VIEW” 버튼을 누르면 생성된 GML 3.0 응용 스키마를 메모장으로 볼 수 있다. 그림 8은 GML 3.0 응용 스키마를 IE 6.0으로 본 모습이다.

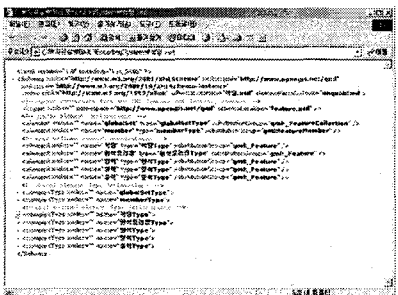


그림 8. GML 3.0 응용 스키마

GML 3.0 응용 스키마는 매핑 규칙에 따라 생성될 GML 3.0 문서의 구조를 정의하고 있으며, XML 스키마 정의 방법과 GML 3.0 응용 스키마 생성 규칙에 따라 매핑 규칙에 정의된 ZEUS의 속성과 타입에 기반하여 대응되는 XML 객체나 GML 객체가 정의된다.

#### 4.1.4 GML 3.0 문서 생성

GML 3.0 문서를 생성하기 위해 그림 6의 문서 생성 다이얼로그에서 GML 3.0 문서의 이름을 입력하고 “MAKE” 버튼을 누르면 GML 3.0 문서가 생성되고, “VIEW” 버튼을 누르면 생성된 GML 3.0 문서를 메모장으로 볼 수 있다. 이때, 매핑 규칙 화일명 또는 GML 3.0 응용 스키마 화일명을 반드시 입력해야 한다.

그림 9는 화일명이 “식당.xml”인 GML 3.0 문서를 IE 6.0으로 본 모습이다.

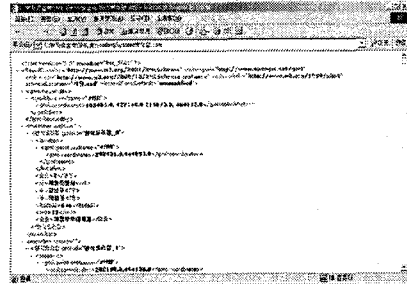


그림 9. GML 3.0 문서

### 4.2 GML 3.0 -> ZEUS

그림 10은 GML 기반 지리공간 정보 엔코딩 서비스 시스템에서 GML 3.0을 ZEUS로 엔코딩하는 과정을 보여준다.

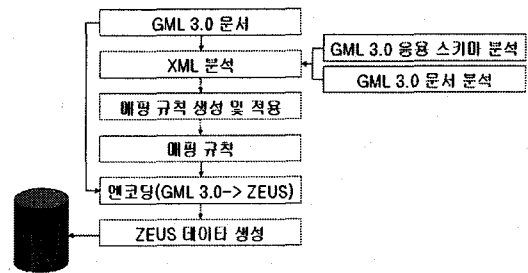


그림 10. GML 3.0 -> ZEUS 과정

본 시스템에서 GML 3.0 문서를 ZEUS에 저장하거나 갱신하려는 경우 GML 3.0을 ZEUS로 엔코딩하는 과정을 수행한다. 이 과정은 먼저 해당 GML 3.0 문서와 관련된 매핑 규칙이 존재한다면 편집하여 엔코딩 모듈로 전달한다. 만약 매핑 규칙이 존재하지 않으면 해당 GML 3.0 응용 스키마와 GML 3.0 문서를 분석하여 새로운 매핑 규칙을 생성하여 엔코딩 모듈에 전달해 준다. 그리고, 엔코딩 모듈에서는 GML 입출력 모듈을 통해 GML 문서에서 해당 엘리먼트에 대한 값을 출력하여 ZEUS 공간 클래스의 값으로 입력하게 된다.

#### 4.2.1 매핑 규칙 생성

그림 11은 GML 3.0 문서 “공원녹지\_내부.xml”을 ZEUS로 엔코딩해 주는 ZEUS 생성 다이얼로그를 보여주고 있다.

만약 해당 GML 3.0 문서에 맞는 매핑 규칙이 존재하지 않을 경우에는 GML 3.0 응용 스키마를 분석한 후에 새로운 매핑 규칙을 생성해야 한다.

그림 11의 ZEUS 생성 다이얼로그에서 응용 스키마 화일명을 기입하고 “매핑 규칙 생성” 버튼을 누르면 그림 12와 같이 사용자가 선택한 GML 3.0 응용 스키마를 분석한 결과를 보여주는 매핑 규칙 생성 다이얼로그가 활성화 되고, 사용자가 그림 12의 “MAKE” 버튼을 누르면 해당 매핑 규칙이 생성된다.

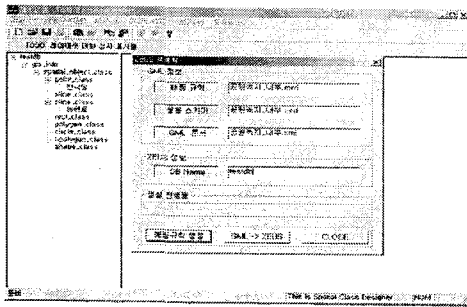


그림 11. ZEUS 생성 다이얼로그

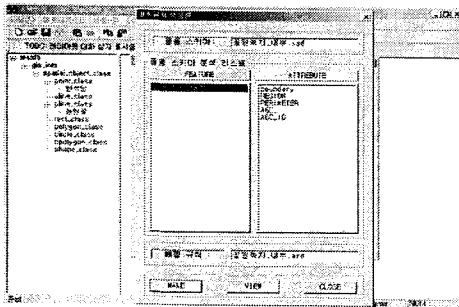


그림 12. 매핑 규칙 생성 다이얼로그

#### 4.2.2 ZEUS 생성

그림 11의 ZEUS 생성 다이얼로그에서 매핑 규칙 화일명과 GML 3.0 응용 스키마 화일명, GML 3.0 문서 화일명, ZEUS 데이터 소스 명을 기입하고 "GML->ZEUS" 버튼을 클릭하면 ZEUS가 생성된다.

그림 13은 "polygon\_class"의 자식 클래스 "공원녹지\_내부"가 생성된 것을 보여주고 있다.

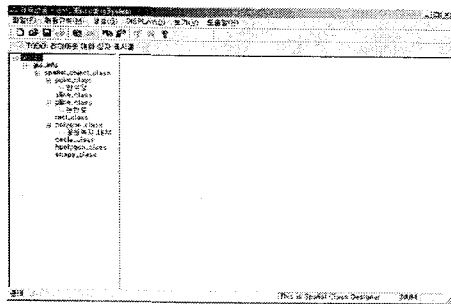


그림 13. ZEUS 생성 결과 화면

### 5. 결론

최근 여러 분야에서 지리공간 정보의 활용이 증가함에 따라 기존에 각자 구축해 두었던 각종 지리공간 정보를 엔코딩해 자유롭게 유통 및 활용하려는 요구가 증가하고 있으며, 국내외에서 이와 관련된 연구가 많이 진행되고 있다. 이를 위해 최근에 OGC에서는 GML 3.0을 제시하였다.

본 논문에서는 공간 데이터베이스 시스템인 ZEUS에 구축된 지리공간 정보를 GML 3.0으로 엔코딩하거나, 역으로 GML 3.0을 ZEUS의 지리공간 정보로 엔코딩할 수

있는 GML 기반 지리공간 정보 엔코딩 서비스 시스템을 설계 및 구현하였다.

본 논문에서 제시한 GML 기반 지리공간 정보 엔코딩 서비스 시스템을 활용하면 외부에서 제공된 GML 3.0 문서를 좀더 쉽고 효율적으로 공간 데이터베이스 시스템인 ZEUS에 엔코딩할 수 있다. 또한, OGC에서 제시된 GML을 활용한 서비스에 적용할 수 있으며, 지리공간 정보 재구축을 위한 시간 및 비용 절감 효과도 얻을 수 있다.

### 참고 문헌

- [1] ISO/TC 211, ISO 19136: Geographic Information - Geography Markup Language, N 1276, 2002.
- [2] Microsoft, The OLE DB Specification 2.0, 1998.
- [3] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.0, 2001.
- [4] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.1.1, 2002.
- [5] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.1.2, 2002.
- [6] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 3.0, 2003.
- [7] Tom, H., "GIS Standardization : The American Experience," 개방형 GIS 연구회 논문지, 1권 1호, 1999, pp.99-180.
- [8] W3Consortium, Document Object Model(DOM) level 1 Specification Version 1.0, 1998.
- [9] W3Consortium, Document Object Model(DOM) level 2 Specification Version 2.0, 2000.
- [10] W3Consortium, Extensible Markup Language (XML) 1.0, 1998.
- [11] W3Consortium, XML Schema Part 0: Primer, 2001.
- [12] W3Consortium, XML Schema Part 1: Structures, 2001.
- [13] W3Consortium, XML Schema Part 2: Data types, 2001.
- [14] 오병우, 한기준, "지리공간 정보 시스템을 위한 표준화", 한국정보과학회 정보과학회지, 13권 10호, 1995, pp.46-55.
- [15] 이혜진, 김동오, 윤재관, 한기준, "GML 3.0 기반 엔코딩 서비스의 설계 및 구현" 개방형 지리공간 정보 시스템 학술회의 논문집, 5권 2호, 2002, pp.153-156.
- [16] 임수미, 김장수, "객체관계형 공간 DBMS:GEUS," 한국개방형GIS연구회지, 1권 1호, 1999, pp.55-72.
- [17] 장영승, 윤재관, 한기준, "ZEUS 기반 OpenGIS 서버의 설계 및 구현," 개방형 지리공간 정보 시스템 학술회의 논문집, 2권 2호, 1999, pp.21-32.