

매핑 규칙을 활용한 GML 3.0 인코딩 시스템의 개발†

Development of a GML 3.0 Encoding System Using Mapping Rules

이용수*, 김동오**, 손훈수**, 한기준***

Yong-Soo Lee, Dong-O Kim, Hoon-Soo Son, Ki-Joon Han

요약 최근 여러 분야에서 지리공간 정보의 활용이 증가함에 따라 기존에 각자 구축해 두었던 각종 지리공간 정보를 개인·조직 나아가 사회·국가 간에 자유롭게 유통 및 활용하려는 요구가 증가되고 있다. 이를 위해 OGC에서는 다양한 분야의 지리공간 정보를 손쉽게 상호 운용하기 위해 이질적인 환경의 지리공간 정보를 XML로 인코딩하여 다양한 분야에 사용 가능하도록 하는 GML 명세를 제시하였으며, ISO/TC 211에서도 GML을 채택하여 국제 표준으로 추진 중에 있다.

따라서, 본 논문에서는 지리공간 정보 상호 운용 방법인 GML 3.0을 분석하고, 인코딩 과정을 좀더 간편하고 효율적으로 바꾸기 위해 인코딩 방법을 정의하는 매핑 규칙을 작성하여 활용하였다. 마지막으로 공간 데이터베이스에 구축된 지리공간 정보를 GML 3.0 문서로 인코딩할 수 있도록 매핑 규칙을 활용한 GML 3.0 인코딩 시스템을 설계 및 구현하였다. 특히, 본 논문에서는 인코딩 시스템을 테스트하기 위해 공간 데이터베이스 시스템으로 ZEUS를 사용하였다.

ABSTRACT Recently, because of the extensive use of geographic information in the various fields, the requirement for the easy circulation and utilization of the various geographic information among the individuals, organizations, societies and countries is raised. In order to support the interoperability of the heterogeneous geographic information in the various fields, OGC(Open GIS Consortium) proposed the GML(Geography Markup Language) specification that defines the XML encoding rule about the heterogeneous geographic information. In addition, ISO/TC 211 adopted the GML specification to make it as the international standard.

Therefore, in this paper, we first analyzed the GML 3.0 specification in detail that can support the interoperability of the heterogeneous geographic information. And then we suggested and applied the mapping rule that define the encoding method to improve the encoding process easily and efficiently. Finally, we designed and implemented the GML 3.0 encoding system using the mapping rule to encode the geographic information that was constructed in spatial databases into the GML 3.0 document. Especially, we used ZEUS as a spatial database system to test our encoding system in this paper.

주요어 : GML 3.0, ZEUS, 인코딩, 상호 운용, 응용 스키마, 데이터 변환

Key word : GML 3.0, ZEUS, Encoding, Interoperability, Application Schema, Data Conversion

1. 서론

지리공간 정보의 활용이 증가함에 따라 기존에 여러 분야에서 각자 구축해 두었던 각종 지리공간 정보를

개인·조직 나아가 사회·국가 간에 자유롭게 유통 및 활용하려는 요구가 증가하게 되었다[16,19]. 따라서, 다양해진 지리공간 정보를 효율적으로 유통하고 활용하기 위해 다양한 표준 데이터 포맷이 제시되어 활용

† 본 논문은 중소기업청 산학연 공동기술개발컨소시엄에 의해 지원받았음.

* 여주대학교 컴퓨터정보관리과 교수

diclee@mail.yeojoo.ac.kr

** 건국대학교 컴퓨터공학과 대학원생

{dokim,hsson}@db.konkuk.ac.kr

*** 건국대학교 컴퓨터공학과 교수

kjhan@db.konkuk.ac.kr

되고 있다. 그러나, 지리공간 정보가 사용되는 분야나 용도에 따라 사용하는 표준 데이터 포맷이 다르므로 사용자들이 다른 환경의 지리공간 정보를 활용하기 위해서는 또다시 다른 표준 데이터 포맷으로의 변환 작업을 거쳐야만 하는 문제점이 발생하였다[17].

그러므로, 사용자들이 지리공간 정보의 저장 데이터 포맷, 관리 프로그램, 저장 공간의 위치 등을 고려할 필요 없이 다양한 지리공간 정보를 효율적으로 상호 운용할 수 있는 방안이 필요하게 되었다. 따라서, 다양한 분야의 지리공간 정보를 손쉽게 상호 운용 가능하게 하기 위해 이질적인 환경의 지리공간 정보를 효과적으로 변환해서 다양한 분야에 사용 가능하도록 데이터 처리 과정을 수행하는 지리공간 정보 인코딩 시스템은 필수적이다[2,6-9].

OGC(Open GIS Consortium)에서는 다양한 분야의 지리공간 정보를 손쉽게 상호 운용하기 위해 이질적인 환경의 지리공간 정보를 XML[10]로 인코딩하여 다양한 분야에 사용 가능하도록 하는 GML(Geography Markup Language) 명세를 제시하였으며[9], ISO/TC 211에서도 GML을 채택하여 국제표준으로 추진 중에 있다[2]. 이에 따라, 국내 외에서도 기존에 구축된 지리공간 정보를 GML로 인코딩하는 연구가 많이 진행되고 있는 실정이지만, GML 2.0 명세에 기반하고 있으며 인코딩 과정이 복잡하므로 이를 개선할 방법이 절실히 필요한 실정이다 [16,17].

본 논문에서는 현재 지리공간 정보의 상호 운용성을 제공하기 위한 대표적인 방법인 GML 3.0을 분석하고, 지리공간 정보를 인코딩 하는 간편하고 효율적인 방법으로서 인코딩 방법을 정의하는 매핑 규칙을 작성하고 이를 활용하는 방안을 연구하였다. 그리고, 이를 바탕으로 공간 데이터베이스에 구축된 지리공간 정보를 GML 3.0 문서로 인코딩할 수 있도록 매핑 규칙을 활용한 GML 3.0 인코딩 시스템을 설계 및 구현하였다.

본 논문의 GML 3.0 인코딩 시스템은 공간 데이터베이스 시스템인 ZEUS[18]에 이미 구축된 지리공간 정보를 GML 3.0으로 인코딩 하는 시스템이다. GML 3.0 인코딩 시스템에서는 ZEUS와 GML 3.0 문서 간의 구체적인 매핑 규칙을 정의하고, 이를 실제 지리공간 정보를 GML 3.0으로 인코딩 하는 모듈에서 분석하고 이를 토대로 GML 3.0 응용 스키마와 GML 3.0 문서를 생성한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장의 관련 연구에서는 GML 3.0과 본 연구에서

활용한 공간 데이터베이스 시스템인 ZEUS를 분석한다. 3장은 매핑 규칙과 매핑 규칙 생성에 대해서 언급한다. 4장과 5장에서는 각각 매핑 규칙을 활용한 GML 3.0 인코딩 시스템의 설계와 구현에 대해서 설명한다. 마지막으로, 6장에서는 결론에 대하여 기술한다.

2. 관련 연구

본 장에서는 OGC의 GML 3.0 명세와 본 연구에서 사용한 공간 데이터베이스 시스템인 ZEUS에 대해서 분석한다.

2.1 GML 3.0

GML은 공간 Feature의 공간-비공간 속성을 포함하는 지리공간 정보를 모델링, 저장, 전송하기 위해 구조화된 문서인 XML로 인코딩한 것으로 OGC에서 표준으로 제정한 것이다. GML 명세는 GML 명세의 사용 목적과 장점, GML에서 사용하는 객체(Object) 모델의 정의, GML으로 인코딩하는 방법, GML에서 제공하는 기본 스키마, 이 스키마를 기본으로 응용 스키마를 제작하기 위해 지켜야 할 사항, 그리고 응용 스키마(Application Schema)와 GML 문서 예제 등을 포함하고 있다.

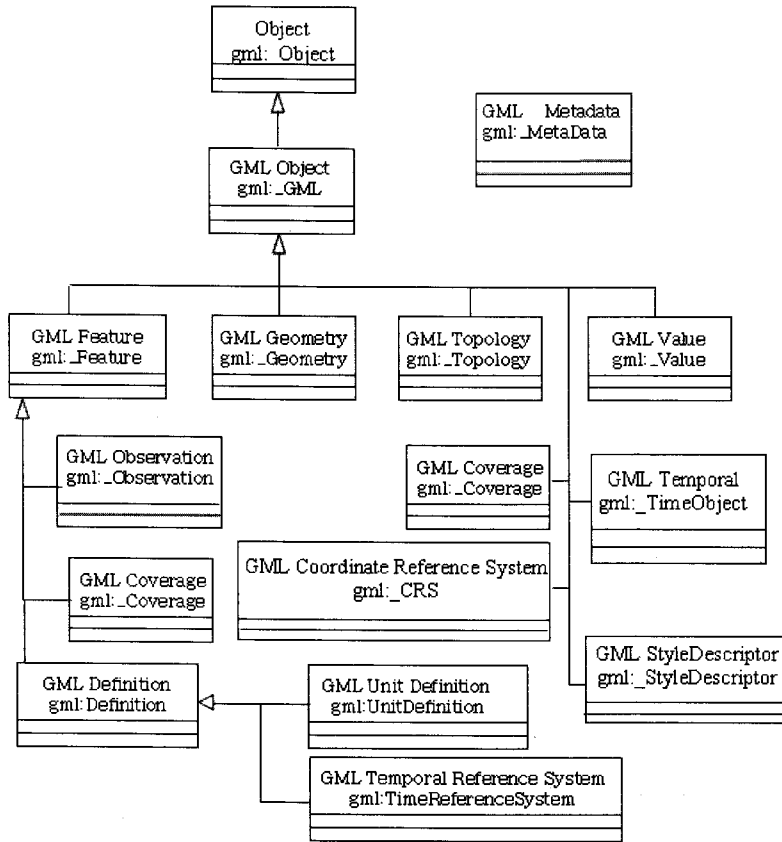
OGC에서는 1999년에 GML 1.0 발표한 이후, 2001년에 XML 스키마[11,12,13]에 기반을 둔 GML 2.0[6]을 발표하였고, 2002년에 약간의 수정을 거친 GML 2.1.1[7]과 GML 2.1.2[8]를 발표하였다. 그리고, 현재 GML 3.0[9]까지 제시하였다.

GML 3.0은 OGC와 ISO/TC 211의 다양한 표준을 지원하기 위해 약 20여 개의 GML 3.0 스키마를 사용하고 있으며, 다양한 응용 분야(예를 들면, LBS)에서 사용할 수 있도록 다양한 객체를 정의하고 있다. <그림 1>은 GML 3.0에서 제공하는 다양한 GML 객체의 구조를 보여준다.

<그림 1>에서 보듯이 GML 3.0은 feature, geometry, topology 등 많은 객체를 표현 가능하다. 이러한 클래스 구조는 GML 3.0에서 제공하는 여러 스키마로 표현된다. 각 스키마에 대한 설명은 다음 절에서 간략히 설명하겠다.

2.1.1 GML 3.0 스키마

GML 3.0에 제시된 스키마의 목록은 <표 1>과 같다. <표 1>의 GML 3.0 스키마 각각을 살펴보면 gmlBase.xsd는 GML에서 사용할 _Object,



〈그림 1〉 GML 클래스 구조

AbstractGMLType, AbstractMetaDataType, _property 등과 같은 추상 데이터 타입들을 정의하고 있는 스키마로서 모든 GML 3.0 스키마는 gmlBase.xsd로부터 상속받아 생성된다.

feature.xsd는 Feature나 Feature Collection을 표현하기 위한 스키마로서 GML 2.0의 Feature 스키마처럼 OGC의 Feature 명세(3,4)에 기반을 두고 있지만 기본 추상 객체에 대한 부분은 gmlBase.xsd로 이전되었으며 지리 속성에 대한 내용은 Geometry.xsd로 옮겨졌다.

geometryBasic0d1d.xsd, geometryBasic2d.xsd, geometryPrimitives.xsd, geometryAggregates.xsd, geometryComplexes.xsd는 공간 데이터를 표현하기 위한 스키마로서 GML 2.0의 Geometry 스키마처럼 Geometry 객체를 표현해 주기 위한 스키마로서 ISO 19107(1)을 따르는 Geometry 타입 엘리먼트 및 어트리뷰트를 정의하고 있다.

지금까지 살펴본 gmlBase.xsd, feature.xsd, geometryBasic0d1d.xsd, geometryBasic2d.xsd, geometryPrimitives.xsd, geometryAggregates.xsd, geometryComplexes.xsd는 GML 3.0에서 객체를 표현하는 기본적인 스키마이다.

GML 3.0은 이외에도 다양한 분야에서 사용하기 위해 여러 스키마를 정의하고 있는데, measures.xsd, valueObject.xsd, unit.xsd, basicType.xsd는 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.xsd, direction.xsd, observation.xsd, dynamicFeature.xsd는 LBS 분야에서의 활용을 위해 동적 feature에 관한 정보를 표현하기 위한 스키마로서 feature의 위치, 방향, 속도 등의 다양한 정보를 저장하기 위한 프로퍼티를 정의하고 있다.

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

〈표 1〉 GML 3.0 스키마

gml.xsd
gmlBase.xsd
basicTypes.xsd
units.xsd
measures.xsd
geometryBasic0d1d.xsd
geometryBasic2d.xsd
geometryPrimitives.xsd
geometryAggregates.xsd
geometryComplexes.xsd
dictionary.xsd
temporal.xsd
grids.xsd
topology.xsd
direction.xsd
feature.xsd
dynamicFeature.xsd
observation.xsd
valueObjects.xsd
coverage.xsd
defaultStyle.xsd

이러한 스키마 외에도 GML 3.0 문서의 지리 데이터의 공간 좌표 참조 체계 정보를 표현하기 위한 coordinateReferenceSystems.xsd, referenceSystems.xsd, coordinateSystems.xsd, datums.xsd, dataQuality.xsd가 존재한다.

2.1.2 GML 3.0 응용 스키마와 생성 규칙

사용자가 표현하고자 하는 다양한 지리 정보의 데이터 구조를 GML 명세에서 제공하는 기본 스키마로는 표현할 수 없으므로, GML 명세에서는 기본적으로 제공되는 여러 스키마를 기반으로 실제 GML 3.0 문서에서 참조할 GML 3.0 응용 스키마를 제작하여 사용하도록 한다.

GML 3.0 응용 스키마에서는 사용자가 사용할 Feature의 구조에 관한 정보를 가지고 있으며, 이를 정의하기 위해 GML 3.0 명세에서는 XML 스키마의 문법에 입각해 GML 3.0 응용 스키마를 생성하는 규칙을 제시하고 있는데, 이를 GML 응용 스키마 생성 규칙이라고 한다. 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. 〈그림 2〉는 GML 3.0 명세에서 제시하는 GML 3.0 응용 스키마 생성 규칙이다.

- ? defining Features and Feature Collections
- ? defining Coverages
- ? defining Observations
- ? defining Dictionaries and Definitions
- ? defining Coordinate Reference Systems
- ? defining Values

〈그림 2〉 GML 3.0 응용스키마 생성 규칙

2.2 ZEUS

ZEUS는 객체 관계형 데이터베이스 시스템에 공간 데이터 처리 기술을 결합시킨 공간 데이터베이스 시스템이다[18]. ZEUS는 데이터베이스의 기본 타입으로 Point, Simpleline, Polyline, Polygon, Rectangle, Circle과 같은 공간 데이터 타입을 지원하고 있으며, 모든 공간 클래스들은 special_object_class에서 상속받아 생성되도록 되어 있다. 본 논문에서는 공간 데이터 타입을 데이터베이스 시스템의 기본 데이터 타입으로 지원하는 ZEUS를 지리공간 정보를 추출하여 GML 3.0 문서를 생성하기 위한 데이터 소스로 사용하고 있다. ZEUS에서 제공하는 공간 데이터 타입과 공간 연산자에 대한 자세한 설명은 다음과 같다.

2.2.1 공간 데이터 타입

ZEUS에서 제공하는 공간 데이터 타입은 시스템 클래스나 사용자 정의 클래스로 정의된 것이 아니라, 데이터베이스 시스템의 기본 데이터 타입으로 제공되고 있다.

공간 데이터 타입은 크게 점, 선, 면으로 나눌 수 있으며, 각각은 0차원, 1차원, 2차원이다. 0차원의 공간 객체는 경계 값이 없고 점 자체가 내부 값이 되며, 1차원 공간 객체는 경계 값이 양 끝점이고 양 끝점을 제외한 선의 내부 값이다. 2차원 공간 객체는 경계 값이 면으로 둘러싸여 있는 폐곡선이고 내부 값은 폐곡선을 제외한 내부의 면이다.

2.2.2 공간 연산자

ZEUS에서는 두 가지 종류의 공간 연산자를 지원한다. 이는 위상 관계 연산자와 기하 연산함수이다. ZEUS에서 지원하는 공간 객체의 위상관계 연산자는 두 공간 객체 사이에 내부 값 간의 교집합이 없고 그 자체간의 교집합이 존재하는 경우를 나타내는 touch, 한 공간 객체가 다른 공간 객체를 포함하고 있으며 경계 값 간에 교집합이 없는 경우를 나타내는 contain/contained, 한 공간 객체가 다른 공간 객체를 포함하고 있으며 경계 값 간에 교집합이 있는 경우를 나타내는 cover/covered, 두 공간 객체의 교집합의 차원이 두 공간 객체 중 차원이 큰 공간 객체보다 1차원 작고 서로 포함관계가 없을 때를 나타내는 crossover, 두 공간 객체 사이에 교집합이 없는 경우를 나타내는 disjoint, 두 공간 객체가 동일한 경우를 나타내는 equal 등이 있다.

ZEUS에서 지원하는 공간 객체의 기하 연산 함수는 공간 객체에 대한 기하학적 정보를 얻기 위한 것으로서 면적을 구하는 area 함수, 무게중심을 구하는 center 함수, 두 공간 객체의 무게중심을 이은 직선과 x 축 사이의 각을 구하는 direction 함수, 공간 객체의 변을 구하는 edge/edges 함수, 시작점을 구하는 startpoint 함수, 끝점을 구하는 endpoint 함수, 둘레 길이를 구하는 length 함수 등이 있다.

3. 매핑 규칙

본 장에서는 GML 3.0 인코딩 시스템에서 활용하는 매핑 규칙과 매핑 규칙의 활용에 대해 기술한다.

3.1 매핑 규칙

매핑 규칙은 GML 3.0 문서를 인코딩 하는데 필요한 정보를 정의한 것으로서, 변환할 데이터 소스의 구조, 변환될 GML 3.0 문서의 구조에 대한 정보 등을 가지고 있다. 이러한 정보는 XML로 인코딩되어 저장된다.

매핑 규칙은 입력 데이터를 출력 데이터로 어떻게 변환하는가에 대해 기술하고 있다. 입력 데이터는 지리공간 정보를 저장하고 있는 시스템 및 파일을 나타내며, 출력 데이터는 GML 3.0 응용 스키마나 GML 3.0 문서가 된다. 즉, 지리공간 정보를 저장하는 시스템에서 GML 3.0으로 인코딩 하는 방법을 기술한다.

매핑 규칙은 공간 데이터베이스 시스템과 GML 3.0 문서 간의 매핑에 대해 정의하며, 이를 정의하고 활용하기 쉽도록 XML로 인코딩 된다. 인코딩 서비스를 통해 GML 3.0 응용 스키마나 GML 3.0 문서가 생성되는데, 이 문서

는 바로 활용되거나 다른 시스템에 전송될 수 있다.

매핑 규칙은 GML 3.0의 응용 스키마와 GML 3.0 문서 간의 정의를 하는 두 가지 변환 규칙을 정의할 필요가 없이 하나의 규칙만을 정의하여 사용한다. 그리고, 이를 응용 스키마 생성모듈에서 분석하여 GML 3.0 응용 스키마를 생성하는데 이용하며, 인코딩 모듈에서 분석하여 GML 3.0 문서를 생성하는데 이용한다. 단, 이를 위해 필요한 모든 정보가 매핑 규칙에 저장된다.

스키마 즉, GML 3.0 응용 스키마를 위해서는 응용 스키마 정의에 필요한 설정과 GML 3.0 응용 스키마 생성에 필요한 정보가 저장된다. 인스턴스 즉, GML 3.0 문서를 위해서는 변환할 데이터 구조의 인스턴스와 변환될 데이터 구조의 인스턴스와 관련된 GML 3.0 문서 간의 매핑 관계가 저장되며, 이는 일반적으로 응용 스키마 변환 정보로부터 유추된다.

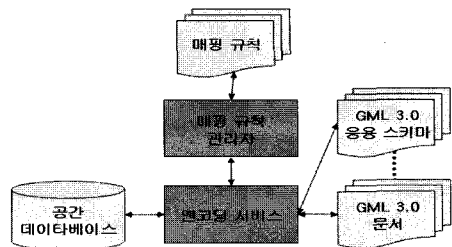
본 논문에서 사용된 매핑 규칙은 분석 및 이용이 쉽도록 XML로 작성되었으며, ZEUS의 스키마의 구조를 기반으로 GML 3.0으로 인코딩할 때 매핑될 엘리먼트와 데이터 타입에 대한 정보와 지리 정보 인코딩에 필요한 시스템 정보를 가지고 있다.

3.2 매핑 규칙 활용

매핑 규칙은 지리공간 정보를 변환할 두 데이터 소스 즉, 지리공간 정보 시스템 및 공간 데이터베이스와 GML 3.0 문서 간의 구체적인 인코딩 규칙을 정의하고 있으며, 공간 데이터 소스의 정보와 GML 3.0 문서의 정보를 저장한다.

이러한 매핑 규칙은 지리공간 정보를 GML 3.0으로 인코딩할 때 인코딩을 담당하는 인코딩 시스템에서 지리공간 정보의 GML 3.0 변환 방법을 파악하는데 이용되며, 기존의 인코딩 과정을 정의하고, 재활용하는데 이용될 수 있다.

〈그림 3〉은 매핑 규칙을 활용해 공간 데이터베이스와 GML 3.0 간의 문서를 변환하는 모습을 보여준다.



〈그림 3〉 매핑 규칙 활용

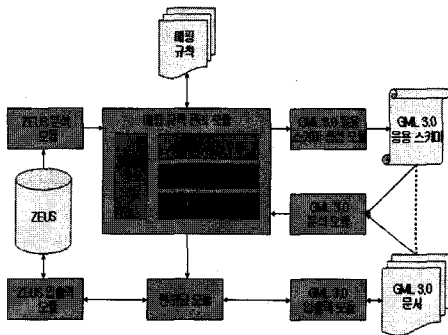
〈그림 3〉에서 보듯이 공간 데이터베이스에서 GML 3.0 응용 스키마 또는 GML 3.0 문서로 인코딩 하거나, 반대로 GML 3.0 문서에서 공간 데이터베이스로의 변환할 때 매핑 규칙 관리자를 통해 매핑 규칙을 활용하는 것을 볼 수 있다. 매핑 규칙은 사용자가 기존에 했던 인코딩 작업을 통해서 생성된 매핑 규칙일 수도 있고, 필요 시 직접 작성하거나 자동으로 매핑 규칙을 생성하는 것이 가능하다.

4. GML 3.0 인코딩 시스템의 설계

본 장에서는 매핑 규칙을 활용한 GML 3.0 인코딩 시스템의 구조를 언급한 후, GML 3.0 인코딩 시스템의 내부 모듈 구성에 대해 설명한다.

4.1 GML 3.0 인코딩 시스템 모듈 구성

그림 4는 매핑 규칙을 활용한 GML 3.0 인코딩 시스템의 모듈 구성을 보여준다. 〈그림 4〉에서 보듯이 GML 3.0 인코딩 시스템은 사용자가 원하는 지리공간 정보를 추출하기 위해 공간 데이터베이스 시스템인 ZEUS를 분석하는 ZEUS 분석 모듈, ZEUS의 지리공간 정보를 입출력하기 위한 ZEUS 입출력 모듈, ZEUS와 GML 3.0 문서 간의 매핑 정보를 정의하는 매핑 규칙을 생성하고 관리하는 매핑 규칙 관리 모듈, 매핑 규칙에 따라 지리정보를 GML 3.0으로 인코딩을 하는 인코딩 모듈, GML 3.0 문서의 응용 스키마를 생성하는 GML 3.0 응용 스키마 생성 모듈, GML 3.0 응용 스키마나 문서를 분석하는 GML 3.0 분석 모듈, 마지막으로 GML 3.0 문서에 데이터를 입출력하기 위한 GML 3.0 입출력 모듈로 구성된다.



〈그림 4〉 GML 3.0 인코딩 시스템 모듈 구성

4.1.1 ZEUS 분석 모듈

ZEUS 분석 모듈은 공간 데이터베이스 시스템인 ZEUS에서 제공하는 API(Application Programming Interface)를 이용하여 ZEUS에 저장된 지리공간 정보의 구조에 대한 정보를 제공하는 모듈로서 ZEUS의 공간 클래스와 어트리뷰트 정보를 가져오는 기능을 가지고 있다.

ZEUS 분석 모듈에서는 본 연구에서 필요한 지리공간 정보인 ZEUS의 공간 클래스만을 추출하기 위해 ZEUS에 속한 클래스 중에서 공간 데이터 타입인 Point, Spline, 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.

4.1.2 매핑 규칙 관리 모듈

매핑 규칙 관리 모듈은 사용자 정의 모듈을 통해 ZEUS 분석 모듈에서 얻은 ZEUS의 지리공간 정보를 사용자에게 보여주고, GML 3.0 응용 스키마와 GML 3.0 문서로 인코딩하는데 필요한 ZEUS의 공간 클래스와 그 어트리뷰트의 정보를 가지는 매핑 규칙을 생성하고 관리하는 모듈이다.

이 모듈은 매핑 규칙을 생성하는 기능과 매핑 규칙을 분석하고 활용하는 기능을 가지고 있는데, 매핑 규칙은 XML 파일로 저장된다. 여기서 매핑 규칙은 사용자의 입력을 받아 작성되거나, 자동으로 생성이 가능하며, GML 3.0 응용 스키마 생성 규칙을 따른다.

매핑 규칙 관리 모듈은 사용자에게 정보를 보여주고 사용자의 입력을 받는 사용자 정의 모듈, 지리공간 정보의 Feature로의 인코딩을 담당하는 Feature 매핑 모듈, 지리공간 정보의 Coverage로의 인코딩을 담당하는 Coverage 매핑 모듈, 지리공간 정보의 Observation로의 인코딩을 담당하는 Observation 매핑 모듈로 구성된다. 지리공간 정보의 공간 데이터 타입은 각각의 매핑 모듈에서 사용되는 공간 데이터 타입과 매핑되는 Geometry 타입으로 자동으로 변환된다.

사용자 정의 모듈은 ZEUS의 구조를 사용자에게 보여주고 사용자의 입력을 받아들이는 모듈로서 ZEUS의 내부 데이터 구조를 보여주는 기능과 사용자의 입력을 받아들이는 기능을 가지고 있다.

Feature 매핑 모듈은 지리공간 정보를 Feature Collection 모델로 인코딩하기 위한 정보를 생성하는

모듈이다. 이 모듈은 GML 3.0 엔코딩 시스템에서 데이터 소스로 사용되는 ZEUS의 공간 클래스 구조를 사용자의 입력을 토대로 GML의 Feature로 매핑해주는 기능과 ZEUS의 공간 데이터 타입에 대응되는 GML의 공간 데이터 타입으로 매핑해 주는 기능을 가지고 있다.

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

4.1.3 GML 3.0 응용 스키마 생성 모듈

GML 3.0 응용 스키마 생성 모듈은 GML 3.0 문서에서 참조되는 GML 3.0 응용 스키마를 생성하기 위한 모듈이다. 이 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙과 GML 3.0 명세에서 제시하고 있는 GML 3.0 응용 스키마 생성 규칙에 따라 ZEUS의 공간 클래스 중 사용자가 원하는 공간 클래스와 어트리뷰트들을 GML 3.0 응용 스키마로 생성해 주는 기능을 가지고 있다.

GML 3.0 응용 스키마는 GML 3.0 문서의 구조를 나타내기 위해서 반드시 필요하지만 이미 구조가 파악되어 사용 중인 GML 3.0 문서에 대해서는 반드시 GML 3.0 응용 스키마가 필요하지는 않다.

GML 3.0 응용 스키마를 생성하기 위해서는 응용 스키마로 제작하기 위한 지리공간 정보의 구조를 명확히 파악해야 하며, 파악한 지리공간 정보의 구조를 XML 스키마의 표현법을 사용하여 GML 3.0 응용 스키마로 표현해 주어야 한다.

GML 3.0 엔코딩 시스템에서는 사용자가 GML 3.0 응용 스키마를 생성하기 위해 별도의 공간 데이터의 스키마 분석 과정을 거치지 않고, 또한 XML 스키마 생성 방법을 이해할 필요 없이 다양한 응용 스키마를 GUI(Graphic User Interface) 환경에서 쉽게 생성할 수 있는 기능을 제공해 준다.

4.1.4 GML 3.0 분석 모듈

GML 3.0 분석 모듈은 선택된 GML 3.0 응용 스키마나 GML 3.0 문서를 분석하기 위한 모듈이다. 이 모듈은 GML 3.0 응용 스키마나 문서에서 필요한 정보를 추출하여 전달해 주는 기능, Feature나 Feature Collection 모델의 정보를 추출하여 전달하는 기능, GML 3.0 문서가 참조하는 GML 3.0 응용 스키마를 따르는지 검사하는 기능을 가지고 있다.

4.1.5 ZEUS 입출력 모듈

ZEUS 입출력 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 엔코딩 모듈에서 GML 3.0 엔코딩 시스템에서 필요한 ZEUS의 지리공간 정보를 입출력하기 위해 ZEUS의 공간 클래스와 해당 어트리뷰트들의 값을 입출력하는 모듈이다. 이 모듈은 ZEUS에 새로운 클래스를 생성하는 기능, ZEUS에 지리공간 정보를 입력하는 기능, ZEUS에서 지리공간 정보를 출력하는 기능을 가지고 있다.

4.1.6 엔코딩 모듈

엔코딩 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 ZEUS 입출력 모듈을 사용하여 지리공간 정보를 출력하여 엔코딩한 후 GML 입출력 모듈에 전달해 주거나 반대의 과정을 담당하는 모듈이다. 이 모듈은 매핑 규칙을 분석하는 기능, ZEUS공간 클래스를 GML 객체로 엔코딩 하는 기능, GML 객체를 ZEUS의 공간 클래스로 변환하는 기능, ZEUS의 공간 데이터 타입을 GML의 Geometry로 변환하는 기능, GML의 Geometry를 ZEUS의 공간 데이터 타입으로 변환하는 기능을 가지고 있다.

데이터 엔코딩 시 ZEUS의 공간 데이터 타입을 제외한 int, double, string 등과 같은 기본 데이터 타입은 XML 스키마에서 제공되는 기본 데이터 타입 중 대응되는 데이터 타입에 매핑되며, ZEUS의 공간 데이터 타입은 GML 3.0 명세의 Geometry 스키마에서 제공하는 공간 데이터 타입으로 매핑된다.

4.1.7 GML 3.0 입출력 모듈

GML 3.0 입출력 모듈은 매핑 관리 모듈에서 생성된 매핑 규칙에 따라 엔코딩 모듈에서 GML 3.0 엔코딩 시스템에서 필요한 GML 3.0 문서의 지리공간 정보를 입출력하기 위해 GML 3.0 문서의 해당 엘리먼트를 입출력하는 모듈이다. 이 모듈은 GML 문서를 생성하는 기능, GML 3.0 문서에 새로운 엘리먼트를 입력하는 기능, GML 3.0 문서에서 엘리먼트를 출력하는 기능을 가지고 있다.

5. GML 3.0 엔코딩 시스템의 구현

본 장에서는 GML 3.0 엔코딩 시나리오를 살펴보고, GML 3.0 엔코딩 시스템의 각 모듈 별 수행 모습과 처리 과정을 설명한다.

5.1 GML 3.0 엔코딩 시나리오

먼저 지리공간 정보를 추출하기 위해 공간 데이터베이스 시스템인 ZEUS의 공간 클래스를 ZEUS 스키마

분석 과정을 거쳐 파악한 후, 필요한 매핑 규칙이 존재할 경우 편집하여 사용하고 존재하지 않을 경우 새로운 매핑 규칙을 생성하게 된다.

다음, GML 문서에서 참조하기 위해 GML 응용스키마를 제작한다. 그리고, 매핑 규칙과 사용자의 입력에 따라 ZEUS 입출력 모듈을 통해 ZEUS 공간 클래스의 공간·비공간 데이터를 추출하여 GML 3.0으로 인코딩한 후 GML 입출력 모듈을 통해 입력하게 된다. 이때, 공간 데이터를 추출하는 과정에서 추출된 데이터가 속하는 SRS(Spatial Reference System)에 대한 정보를 입력하게 된다. 이렇게 생성된 GML 문서는 다양한 분야에서 활용 가능하다.

그리고, GML 3.0 문서를 ZEUS에 저장하거나 갱신하려는 경우 해당 GML 3.0 문서와 관련된 매핑 규칙을 편집하여 사용하거나 새로운 매핑 규칙을 생성하고, 이에 따라서 인코딩 모듈에서 GML 입출력 모듈을 통해 GML 문서에서 데이터를 출력하여 ZEUS 공간 클래스의 데이터로 입력하게 된다.

5.2 ZEUS 스키마 분석

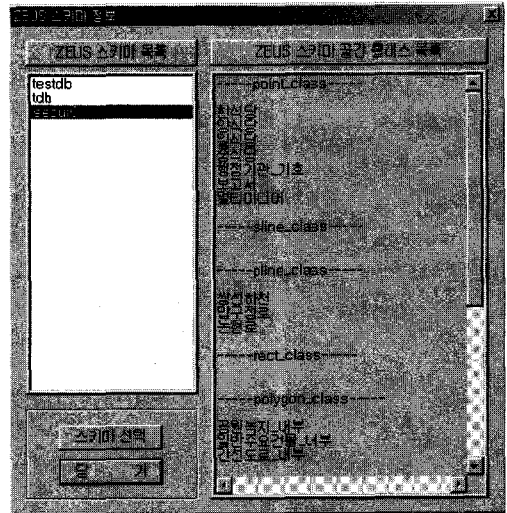
GML 3.0 인코딩 시스템에서는 지리공간 정보를 GML 3.0으로 인코딩하기 위해서 먼저 ZEUS의 데이터 소스 중 사용자가 원하는 공간 클래스가 들어있는 데이터 소스를 찾아내는 과정을 거치게 된다.

사용자가 원하는 ZEUS 데이터 소스를 찾기 위해 ZEUS에 등록되어 있는 데이터 소스의 정보를 ZEUS 분석 모듈을 통해 얻어내게 된다. <그림 5>는 ZEUS 데이터 소스의 정보를 보여주고 선택하기 위한 ZEUS 스키마 다이얼로그를 보여 준다.

ZEUS 스키마 다이얼로그에서는 ZEUS에 등록되어 있는 데이터 소스의 리스트를 보여주는데, 이는 <그림 5>의 왼쪽에 보이는 리스트로서 현재 ZEUS에 등록되어 있는 데이터 소스는 "testdb", "tdb", "seoul"임을 알 수 있다.

사용자가 ZEUS 분석 정보창의 데이터 소스 리스트에서 원하는 하나를 선택하면, 선택된 ZEUS 데이터 소스에 속해 있는 공간 클래스들의 리스트를 가져와 보여준다. 현재 "Seoul"에는 point_class로 "한식당", "양식당", "일식당", "중식당", "멀티미디어", "보고서", "행정기관_기호"가 있고, Pline_class로 "쌍선하천", "암구정로", "논현로"가 있고, Polygon_class로 "공원녹지_내부", "일반주요건물_내부", "간선도로_내부", "행정기관_내부", "의료후생_내부", "교육기관_내부", "시장상가_내부", "고속도로_내부", "쌍선하천_내부", "시군구_내부", "특별_광역시도계_내부"가 있음을

알 수 있다.

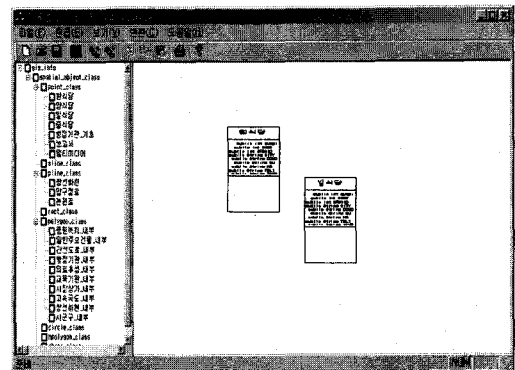


<그림 5> ZEUS 스키마 다이얼로그

마지막으로 사용자는 GML 3.0 인코딩 시스템에서 사용할 데이터 소스의 이름을 "스키마 선택" 버튼을 사용하여 등록하게 되며, ZEUS 스키마 다이얼로그를 통해 사용자가 선택한 데이터 소스에 등록되어 있는 ZEUS 공간 클래스들의 정보를 사용자에게 보여준다.

5.3 매핑 규칙 생성

매핑 규칙 생성에서는 3장에서 설명한 매핑 규칙을 생성하는 모습을 보여주는데, <그림 6>은 매핑 규칙을 생성하는 GML 인코딩 다이얼로그를 보여준다.



<그림 6> GML 인코딩 다이얼로그 1

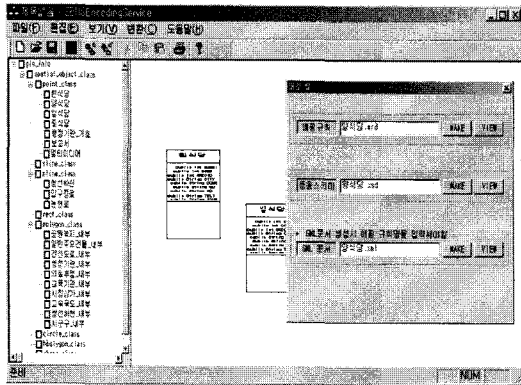
<그림 6>의 GML 인코딩 다이얼로그에서 화면의 오른쪽은 ZEUS 스키마 다이얼로그에서 사용자가 선택

택한 데이터 소스의 공간 클래스 정보를 트리 형태로 보여주는 창이다. <그림 6>에서 보듯이 ZEUS의 공간 클래스는 "spatial_object_class" 아래 공간 데이터 타입을 가지는 6개의 공간 클래스가 있으며, 사용자의 클래스는 사용하는 데이터 타입에 따라 6개의 공간 클래스 중 해당하는 공간 클래스 하부에 포함된다.

GML 인코딩 다이얼로그에서 사용자가 인코딩하려는 공간 클래스를 왼쪽 트리에서 오른쪽 창 화면으로 끌어다 놓으면 해당 공간 클래스의 속성을 갖는 클래스가 생성되게 된다. <그림 6>에서 "한식당"과 "일식당" 클래스가 생성된 것을 볼 수 있다.

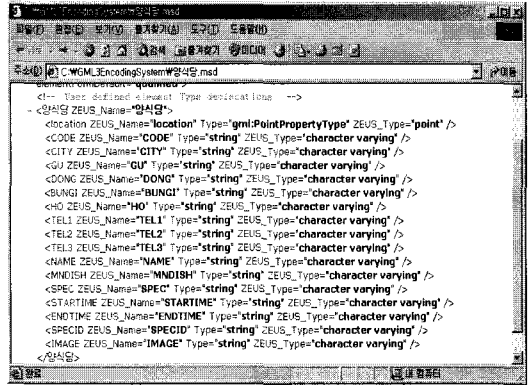
각 클래스는 클래스 이름과 속성을 가지고 있으며, 사용자는 해당 클래스에서 원하는 속성을 편집할 수 있다. 즉, 사용자가 원하는 클래스 만을 끌어 놓거나, 해당 클래스에서 원하는 속성을 제외한 모든 속성을 제거할 수 있다. 편집을 마치고 남은 공간 클래스와 속성들이 GML 3.0 응용 스키마와 GML 3.0 문서에 들어가게 될 Feature의 공간비공간 속성들이다.

<그림 6>의 클래스를 살펴보면 각 클래스는 해당 ZEUS 공간 클래스의 속성 정보와 매핑되는 GML 객체의 구조를 가지고 있으며, 이 구조는 XML로 인코딩되어 저장된다. 이러한 클래스 구조를 저장한 것이 매핑 규칙이다.



<그림 7> GML 인코딩 다이얼로그 2

<그림 7>은 GML 인코딩 다이얼로그에서 매핑 규칙을 생성하는 다이얼로그를 띄운 모습이다. 매핑 규칙을 생성하기 위해서는 매핑 규칙 화일명을 기입하고 "MAKE"라는 버튼을 누르면 약간의 시간 경과 후에 "매핑 규칙 생성 완료"라는 메시지 박스가 나타난다. 그러면 매핑 규칙이 생성된 것이다. 이때 "VIEW" 버튼을 눌러서 매핑 규칙을 메모장으로 볼 수가 있다.



<그림 8> 매핑 규칙

<그림 8>은 매핑 규칙을 IE 6.0으로 본 모습이다. <그림 8>은 ZEUS의 공간 클래스인 "양식당"을 GML로 인코딩하기 위한 매핑 규칙을 저장한 양식당.msd 화일로서 ZEUS_NAME 엘리먼트의 속성값으로 ZEUS 공간 클래스인 "양식당"을 가지고 있다. 이것은 ZEUS의 공간 클래스 "양식당"을 GML로 변환할 때 "양식당"으로 매핑한다는 것을 의미한다.

"양식당" 엘리먼트의 자식 엘리먼트들은 해당 공간 클래스의 속성을 나타내고, 속성값들은 GML로 인코딩하는 규칙을 정의한다. 즉, ZEUS_NAME은 ZEUS의 속성 이름을 나타내고, ZEUS_Type은 속성 타입을 나타낸다. 이러한 엘리먼트를 통해 ZEUS "양식당" 클래스의 "location" 속성은 타입이 "point"며, GML의 "gml:PointPropertyType"을 가지는 "location" 엘리먼트로 인코딩됨을 알 수 있다.

이 외에도 매핑 규칙에는 ZEUS 데이터 소스의 이름과 생성 날짜 등 필요한 메타 정보가 기록된다.

5.4 GML 3.0 응용 스키마 생성

생성된 매핑 규칙은 GML 3.0 응용 스키마 생성 모듈을 통해 GML 3.0 문서에서 참조할 GML 3.0 응용 스키마를 생성할 수 있다. GML 3.0 응용 스키마를 생성하기 위해 <그림 7>의 GML 인코딩 다이얼로그에서 GML 3.0 응용 스키마 이름을 입력하고 "MAKE" 버튼을 누르면 GML 3.0 응용 스키마가 생성되고, "VIEW" 버튼을 누르면 생성된 GML 3.0 응용 스키마를 메모장으로 볼 수 있다. <그림 9>는 GML 3.0 응용 스키마를 IE 6.0으로 본 모습이다.



〈그림 9〉 GML 3.0 응용 스키마

〈그림 9〉에서 GML 3.0 응용 스키마의 XML 버전은 "1.0"이고, 엔코딩은 "ksc_5601"로 구성되어 있으며, 네임스페이스로 XML Link인 "http://www.w3.org/2001/XMLSchema"와 gml과 xlink의 이름을 가지는 XML Link 어트리뷰트를 포함하고 있다.

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 타입을 기준으로 대응되는 XML 객체나 GML 객체가 정의된다.

본 논문에서 사용한 〈그림 9〉의 GML 3.0 응용 스키마에서는 ZEUS 데이터 소스의 이름인 "seoulC"를 FeatureCollection으로 선언하고, ZEUS의 각 공간 클래스를 갖는 Feature를 정의하여 "seoulC" 클래스에 포함시켰다.

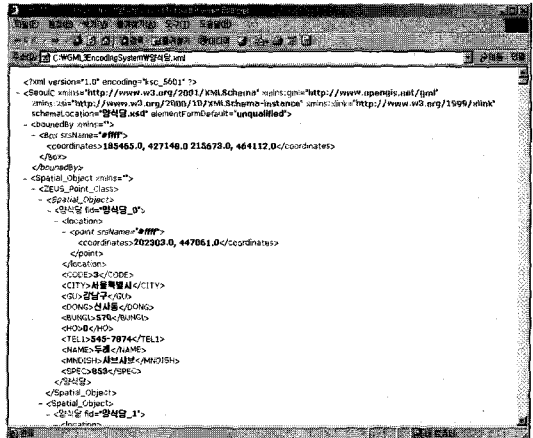
5.5 GML 3.0 문서 생성

GML 3.0 문서 생성은 엔코딩 모듈에서 담당하는데, 생성된 매핑 규칙을 분석하고, GML 3.0 분석 모듈에서 GML 3.0 문서의 필요한 정보를 분석하고, 이를 토대로 ZEUS 입출력 모듈과 GML 3.0 입출력 모듈을 이용하여 GML 3.0 문서를 생성한다.

GML 3.0 문서를 생성하기 위해 〈그림 7〉의 GML 엔코딩 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 파일명 또는 GML 3.0 응용 스키마 파일명을 반드시 입력해야 한다.

GML 3.0 문서도 GML 3.0 응용 스키마와 유사한 XML 문서 설정을 가지며, GML 3.0 응용 스키마를 Import 함으로서 해당 GML 3.0 응용 스키마와 연결되게 된다.

〈그림 10〉은 GML 3.0 문서를 IE 6.0으로 본 모습이다. 〈그림 10〉은 "양식당.xml" 파일로서 RootElement로 "seoulC" Feature Collection을 사용하였으며, MBB(Minimal Bounding Box)를 나타내기 위해 boundedBy 엘리먼트가 그 하부에 있고, 그 다음에 ZEUS의 공간 클래스인 "양식당"과 그 어트리뷰트들로 구성되어 있다.



〈그림 10〉 GML 3.0 문서

6. 결론

여러 분야에서 지리공간 정보의 상호 운용이 점차 증가되고 있는 시점에서 지리공간 정보를 손쉽게 상호 운용하기 위해 실질적인 환경의 지리공간 정보를 XML로 엔코딩하여 다양한 분야에 사용 가능하도록 하기 위해 국내외에서 기존에 구축된 지리공간 정보를 GML로 엔코딩하는 연구가 많이 진행되고 있는 실정이다.

본 논문에서는 기존의 지리공간 정보 엔코딩 방법을 개선하고, 좀더 편리하고 효율적으로 GML로 엔코딩하기 위한 방안으로서 엔코딩 방법을 정의하는 매핑 규칙을 제시하였다. 또한, 이를 토대로 공간 데이터베이스에 구축된 지리공간 정보를 GML 3.0 문서로 엔코딩할 수 있도록 매핑 규칙을 활용한 GML 3.0 엔코딩 시스템을 설계 및 구현하였다.

사용자들은 본 논문에서 제시한 GML 3.0 엔코딩

시스템을 이용함으로써 좀더 쉽고 효율적으로 ZEUS의 지리공간 정보를 GML 3.0으로 엔코딩하여 사용할 수 있게 되었으며, 이러한 엔코딩 기술을 기반으로 GML을 활용하는 다양한 OGC 기반의 서비스 즉, WMS나 WFS로의 확장에 대한 기반을 마련하게 되었다.

향후에는 본 논문에서 개발한 매핑 규칙을 활용한 GML 3.0 엔코딩 시스템을 확장하여 사용자의 입력이 없이 자동으로 다양한 GML 문서를 변환할 수 있는 시스템의 구현을 위한 기본 연구가 필요할 것이다.

참고문헌

[1] ISO/TC 211, ISO 19107: Geographic Information - Spatial Schema, DIS 19107, 2001.

[2] ISO/TC 211, ISO 19136: Geographic Information - Geography Markup Language, N 1276, 2002.

[3] OpenGIS Consortium, Inc., Topic 5 - The OpenGIS Feature, 1999.

[4] OpenGIS Consortium, Inc., Topic 10 - Feature Collections, 1999.

[5] OpenGIS Consortium, Inc., OpenGIS Simple Features Specification for OLE/COM Revision 1.1, 1999.

[6] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.0, 2001.

[7] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.1.1, 2002.

[8] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 2.1.2, 2002.

[9] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification 3.0, 2003.

[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] 김동오, 윤재관, 한기준, "GML 문서 관리 시스템의 설계 및 구현" 한국정보과학회 학술발표논문집, 제 28권 2호, 2001, pp.85-87.

[15] 김동오, 윤재관, 한기준, "ZEUS를 위한 GML 문서 생성기의 설계 및 구현" 개방형 지리공간정보 시스템학회 학술대회논문집, 2002, pp.53-58.

[16] 안영수, 박순영, 정원일, 배해영, "공간 데이터 베이스와 GML 문서간의 지리공간 정보 변환 방법" 정보처리학회 2002년 추계학술대회, 9권, 2002, pp.1-4.

[17] 이혜진, 김동오, 윤재관, 한기준, "GML 3.0기반 엔코딩 서비스의 설계 및 구현" 개방형 지리정보시스템 학회 2002년 추계학술대회 논문집, 2002, pp.153-156.

[18] 임수미, 김장수, "객체관계형 공간 DBMS:GEUS," 한국개방형GIS연구회지, 제1권1호, 1999, pp.55-72.

[19] 장영승, 윤재관, 한기준, ZEUS 기반 OpenGIS 서버의 설계 및 구현, 개방형 지리공간 정보 시스템 학술화의 논문집, 2권 2호, 1999, pp.21-32.



이영수

1986년 명지대학교 전자계산학과 졸업(공학사)
 1989년 건국대학교 대학원 전자계산학과 졸업(공학석사)
 1998년~현재 건국대학교 대학원 컴퓨터공학과 박사과정
 1995년~현재 여주대학 컴퓨터정보관리과 부교수

관심분야: GML, 데이터베이스, LBS, semantic web



김동오

2000년 건국대학교 컴퓨터공학과 졸업(공학사)
 2002년 건국대학교 대학원 컴퓨터공학과 졸업(공학석사)
 2002년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

관심분야: 지리정보 시스템, GML, LBS, MODB, XML



손훈수

2002년 배재대학교 컴퓨터공학과
졸업(공학사)
2002년~현재 건국대학교 대학원
컴퓨터공학과 석사과정
관심분야: GML, XML, 지리정보
시스템, 공간데이터베이스



한기준

1979년 서울대학교 수학교육학과
졸업(이학사)
1981년 한국과학기술원 전산학과
졸업(공학석사)
1985년 한국과학기술원 전산학과
졸업(공학박사)
1990년 Stanford 대학 전산학과
visiting scholar
1985년~현재 건국대학교 컴퓨터공학과 교수
관심분야: 지리정보시스템, 공간데이터베이스, LBS