

GML 기반 계층적 POI 정보 가시화를 위한 GVP 개발

송 은 하[†] · 박 용 진^{**} · 정 영 식^{***}

요 약

최근 GIS 서비스는 단순한 지도 가시화뿐만 아니라 각 개체의 위치 정보를 표시해주는 POI 서비스까지 요구한다. 하지만 POI 서비스는 개체의 위치 정보가 불충분하여 단순 지명 서비스만을 제공하고 있다. 게다가 GIS 서비스의 기반이 되는 대부분의 지리공간데이터 역시 표준화가 이루어지지 않았다. 본 논문에서는 계층적이고 사용자 중심적인 POI DB를 설계하고, 지도 가시화의 이질성과 비호환성을 극복하기 위해 GML 표준을 수용하는 GVP(GML Viewer POI)를 구축한다. GVP는 SHP 파일의 속성 정보를 관리하는 dBase 파일에 3-Layer 구조로 설계된 POI DB를 추가한다. 이에 따라서 POI 가시화는 디렉터리 형태의 그룹화된 POI 정보가 제공되므로 계층적인 검색이 가능하다. 지도 가시화와 POI 가시화로 구성된 GVP는 속성을 각각의 객체 형태로 재생성하여 사용자 이벤트에 대한 빠른 응답을 지원한다.

키워드 : SHP, GML, GIS, 모바일 지도 서비스, 지도 변환, POI

A Development of GVP for Hierarchical POI Information Visualization based on GML

Eun-Ha Song[†] · Yong-Jin Park^{**} · Young-Sik Jeong^{***}

ABSTRACT

Today, GIS service requires not only a simple map visualization but also geographical information of each object. However, POI service provides simply geographical naming service only because of lacking in geographical information of objects. In addition, geographical space data representation functioning that is the basis for most GIS has not been standardized yet. In this paper, it is designed that POI DB that is hierarchical and user-oriented and it is constructed that GVP(GML Viewer POI) that is the basis of GML specification to overcome difference and incompatibility of map visualization. GVP adds POI DB in a 3-Layer structure to dBase file managing the information of SHP file attributes. Therefore, POI visualization enables hierarchical search, by providing POI information in directory-type grouping. Consisting of map visualization and POI visualization, GVP regenerates attributes in the form of individual objects and responds to user events immediately.

Key Words : SHP, GML, GIS, Mobile Map Service, Map Convert, POI

1. 서 론

최근 GIS는 웹과 연동하여 지도 정보와 POI(Point Of Interest) 서비스를 제공하고 있다. POI는 건물, 도로, 상점, 관공서, 병원 등 지도 내에서 사용자가 관심 있어 하는 개체를 말한다. 이를 통해 사용자는 관심을 가지는 곳이나 필요한 정보를 쉽게 얻을 수 있다. POI 구성요소는 도로와 관련된 것, 배경과 관련된 것, 주기와 관련된 것이 있으며 추가적으로 전화번호부 주식회사에서 발간한 전화번호부 상호

데이터와 자체 고유 지번 데이터가 있다. 이러한 POI 정보를 지도 서비스에 추가함으로써 단순한 지도 가시화가 아닌 지도를 구성하는 각 개체들의 정보를 확인함으로써 효율성과 활용도를 향상한 GIS 서비스를 제공할 수 있다[1, 2, 3]. 그러나 현재 GIS 서비스는 기반이 되는 지리공간데이터의 표현과 POI 정보를 제공하는데 있어서 표준화가 이루어지지 않고 있는 실정이다. 따라서 서비스 개발업체들은 서비스 제공방법, 서비스 구축, 지도 표현 형식에 있어 특정 시스템이나 환경에 의존적이며 그에 따른 제약이 많은 형편이다. 특히 POI 서비스는 해당 개체의 정보가 불충분하여 단순한 지명 서비스에 그치는 정도이다. 따라서 POI 서비스 사용자는 요구하는 정보에 알맞은 체계적인 검색 기능을 제공하지 못하고 있다[4].

이에 본 논문에서는 OGC에서 제안하고 국제표준협회(ISO)에 의해서 표준화된 GML[5] 명세에 기반하여 지도를 가시

* 본 논문은 2006년 산학협동재단에서 지원하는 학술연구비 지원으로 수행되었음.

† 정 회 원 : 원광대학교 전기전자 및 정보공학부 전임강사

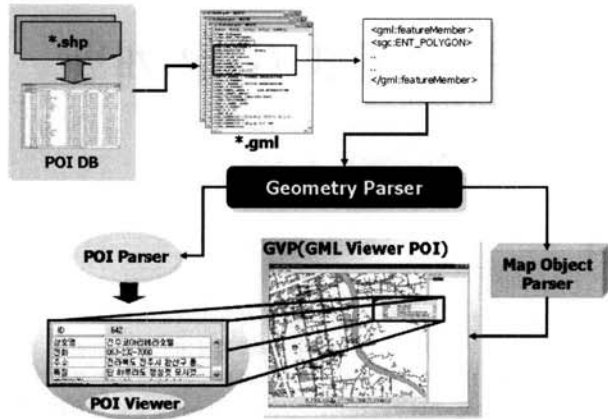
** 준 회 원 : 원광대학교 컴퓨터공학(공학석사)

*** 종신회원 : 원광대학교 전기전자 및 정보공학부 교수

논문접수 : 2007년 3월 8일

수정일 : 1차 2007년 8월 20일, 2차 2008년 1월 7일, 3차 2008년 5월 19일

심사완료 : 2008년 5월 21일



(그림 1) GVP 전체 개념도

화하고 POI 정보를 가시화하기 위한 모듈을 포함하는 GVP (GML Viewer POI)를 개발한다. 특히 GVP는 POI 정보를 표현하는데 있어서 DXF 표준 수치 지도를 참조하여 계층적 3-Layer 구조를 정의하여 사용자 중심의 POI 정보를 제공한다[6]. (그림 1)은 설계한 GVP의 전체 개념도이다.

2. 관련연구

2.1 GML 기반 지도 서비스

비영리 단체인 OGC에서 지리정보나 위치기반 서비스 정보를 플랫폼과 클라이언트 종류에 상관없이 이질적인 환경의 지리정보에 대해 모델링, 저장, 전송하기 위해서 구조화된 정보 표현이 가능하도록 XML에 기반한 GML 명세를 제시하였다. GML은 기존 GIS의 다양한 지리정보데이터 표준을 지원하기 위해 스키마를 사용하여 Feature, Geometry, Topology, TimeObject, Style 등 많은 객체의 표현이 가능하다. 현재 GML 3.1은 국제 표준화 기구에 의해 ISO 19136 으로서 표준화로 책정되었다.

기존의 GML 기반의 가시화 시스템들은 TatukGIS Inc.의 TatukGIS Viewer 1.4[7], SafeSoftware Inc.의 FME Universal Viewer[8], Snowflake Software Ltd.의 OS Master Map Viewer 2.0[9], eSpatial Inc.의 iSMART Explorer 4.4[10], ETRI의 GML Viewer 등이 있다.

TatukGIS Viewer 1.4는 지원하지 않는 포맷이 거의 없을 정도로 2Gbyte의 파일을 처리할 수 있는 능력을 가지고 있다. 또한 가시화된 지리정보를 PDF 파일로 변경할 수 있다. 이 시스템은 지점간의 거리와 점의 구성을 이루는 크기와 해당 지역을 측정하며 속성 처리는 SQL Builder로도 할 수 있도록 지원한다. 벡터 속성 정보를 기반으로 지도의 서로 똑같은 속성은 같은 색으로 표현하며, 마우스를 가지고 벡터그래픽을 클릭하면 해당 도형의 속성 창을 한눈에 볼 수 있는 속성 창 정보가 오른쪽에 나타난다. FME Universal Viewer 2003은 해당 도형의 속성 정보와 편집 기능을 제공한다. 또한 단순한 창안에 복수 점에 대한 측정 거리 및 기하학 정보를 보여준다. iSMART Explorer 4.4는 자동적으로

스키마를 발견하여 비공간 속성 데이터를 재검토할 수 있고 확대/축소/이동이 모두 가능하다. 또한 웹 브라우저를 통해 충분한 공간 편집이 가능하며 낮은 대역폭 연결을 위해 빠른 응답시간을 제공한다. 그 밖에 여러 GML 기반 지도 서비스가 있으나, 이들 가시화 시스템은 단순히 지도 가시화만 해줄 뿐 실질적인 POI 정보 서비스는 제공하지 않는다. 또한 이 시스템들은 GML 파일 내의 속성들만을 보여주기 때문에 전문 개발자들의 지도편집에 중점을 두어 실제 GIS를 필요로 하는 일반 사용자들이 사용하기에 편리한 GUI를 제공하지 않는다.

2.2 기존 POI Map Service

POI 구축과 POI 서비스 순으로 기존의 지도 서비스에 관한 내용을 소개한다. 먼저, POI 구축은 소정의 카테고리별로 분류하여 소정의 데이터베이스에 저장하고 분류된 카테고리별로 단순히 지도 위에 TIP 형태로 가시화하든지, DB 검색을 통한 테이블 형태로만 가시화하고 있다[11]. 또한 POI 파일 문서별로 모듈화하고 이를 바탕으로 자동차 네비게이션시 명칭검색, 지역검색 등을 통하여 경로 안내 목적지 표시에 활용하는 가시화 기술이 있다[12]. 이들은 POI 자체에 대한 가시화에 치중하고 있어 단순한 가시화 기술만 제공한다.

기존의 POI 서비스들을 주로 모바일과 PDA 환경을 위주로 개발되었다. 모바일과 PDA의 서비스들은 파일 경량화로 인해 상세한 정보를 나타내기에는 역부족이며 사용자의 편의성을 고려하는데 한계가 있다. 지도 가시화 측면에서는 도로와 건물의 색이 구분되지 않아 사용자가 알아보는데 매우 불편하며, 상하좌우 이동과 같은 대부분의 제어가 키패드와 메뉴를 선택한 이동만 가능하게 설계되었다. 특히 POI 정보는 상호명만을 보여주는 단순한 지명 서비스에 한계를 보이며, 실제 사용자가 필요로 하는 전화번호, 주소, 홈페이지, 특징 등의 정보는 제공하지 않는다.

Web상에서 POI 서비스를 해주는 대표적인 시스템으로는 콩나물달걀이 있다[13]. 이 시스템에서는 건물명이 주요 건물에만 표시가 되고 확대 축소를 하여도 상세한 건물명은 나오지 않아 사용자가 위치를 찾아 가거나 주변 정보를 살펴보는 데 있어 많은 불편하다. 또한 검색을 위해 건물명을 선택할 경우, 상세 POI 정보를 제공하지 않는다.

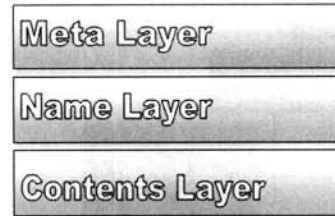
본 연구에서는 POI 분류 정보 및 POI 정보 가시화를 GML 기반 지도와 조합된 형태로 제공하여 보다 효율적이고 사용자 위주의 인터페이스 기능을 강조하고자 한다. 즉, 도로와 건물은 각기 다른 색을 적용하여 사용자가 한눈에 알아볼 수 있게 식별성을 높여 준다. POI 정보는 건물을 선택할 경우 상호명, 전화번호, 주소 그 건물의 특징, 홈페이지 등의 상세 정보를 제공한다. 또한 사용자의 빠른 정보 검색을 위해 계층적 POI 정보 구축과 함께 사용하기 편리한 검색창을 제공한다. 특히, 기존의 POI Map 서비스들은 지리데이터의 표준화가 이루어져 있지 않아 서로간의 상호호환이 불가능하여 데이터의 중복 구축의 단점을 GML 기반으로 보완한다.

3. 계층적 POI 설계

3.1 3-Layer POI 구조

본 논문에서는 POI 정보의 빠른 검색, 추가, 삭제, 갱신의 효율성을 위해 계층적인 3-Layer POI 구조를 설계한다. 3-Layer POI 구조의 기반 설계는 DXF 표준 수치지도를 참조하였다. DXF 표준 수치지도는 대, 중, 소, 세 분류의 계층 구조를 가지고 있으나 가장 하위 계층인 세 분류는 실제 사용자가 구체적인(specific) POI 검색을 지원하기엔 한계성이 있다. 이에 3-Layer POI는 DXF 표준 수치지도의 세 분류를 참조하여 이를 좀 더 구체적인 구조를 정의함으로써, 사용자의 구체적인 POI 검색을 지원한다. (그림 2)는 본 논문에서 정의한 3-Layer POI의 구조이다.

Meta Layer는 DXF 표준 코드의 레이어 코드가 '4'인 건물에 대해 사용자 중심 메타 코드를 정의한다. 동일한 그룹에 포함되는 항목에 특정 키 값을 임의로 주어 같은 그룹에 포함되도록 구분한다. 메타코드는 POI 정보를 분류하기 위한 기본적 구성 요소로서 <표 1>과 같이 U-Meta와 L-Meta로 이루어진다. <표 1>에서 진하게 표시된 셀은 메타 코드의



(그림 2) 3-Layer POI 구조

상위 그룹으로 정의된 U-Meta 코드이며, 총 11개의 분류 코드로 나눈다. 해당 U-Meta 코드는 그 특성별로 L-Meta 코드를 가진다.

Name Layer는 이름에 대한 분류로 해당 건물명에 키 값을 준다. Contents Layer는 실제 사용자에게 보여줄 POI 정보를 담고 있는 계층으로 이름, 주소, 전화번호, 특징, 홈페이지 등의 세부 정보가 있다.

(그림 3)은 정의된 3-Layer POI 구조의 사례이다. Meta Layer는 DXF 표준코드의 소분류와 유사한 교육학원을 가리키며, U-Meta 코드값으로 01을 주어진다. L-Meta 코드는

<표 1> U-Meta와 L-Meta 코드 분류표

Meta 코드	내 용	Meta 코드	내 용	Meta 코드	내 용
01	교육학원	0405	목욕탕	0801	사회복지
0101	초등	0406	사우나	0802	양로시설
0102	중등	0400	기타	0803	장애인시설
0103	고등	05	금융	0800	기타
0104	대학	0501	은행	09	아동복지
0105	학원	0502	보험	0901	탁아시설
0106	독서실	0503	조합	0902	영아시설
0107	체육관	0500	기타	0903	육아시설
0100	기타	06	문화종교	0904	놀이방
02	의료후생	0601	교회	0900	기타
0201	대학병원	0602	절	10	행정
0202	일반병원	0603	원불교	1001	시청
0203	보건소	0604	기도원	1002	동사무소
0204	약국	0605	문화재협	1003	면사무소
0205	한의원	0606	박물관	1004	경찰서
0206	한약방	0600	기타	1005	파출소
0207	의원	07	운수장고	1006	소방서
0200	기타	0701	역	1007	우체국
03	산업	0702	공항	1008	세무서
0301	산업	0703	터미널	1009	전화국
0302	공장	0704	버스정류장	1010	병무청
0303	변전소	0705	택시정류장	1000	기타
0300	기타	0706	카센터	00	기타
04	숙박	0707	주차장	0001	주택
0401	호텔	0708	주유소	0002	아파트
0402	모텔	0709	택배	0003	원룸
0403	여인숙	0700	기타	0004	기숙사
0404	민박	08	사회복지		



(그림 3) 3-Layer POI의 실례

DXF 표준코드의 세분류와 유사한 초등학교, 중학교, 고등학교 등으로 교육학원의 특성 분류로 나누어지며, 초등학교는 0101, 중학교는 0102, 고등학교는 0103의 코드 값을 부여한다. Name Layer는 중학교에 대한 학교명 분류로써 제일중학교, 중앙중학교, 봉남중학교 등에 대해 각각의 코드 값의 조합으로 식별키 값이 주어진다. DXF 표준 수치지도의 상세분류로 학교와 같이 포괄적으로 규정한 분류를 세분화하고 계층화된 코드 값을 부여한다.

3.2 계층적 POI DB

본 논문은 SHP(Shapefile)를 기반데이터로 설계한다. SHP는 국내 및 해외에서 Geospatial Data의 실질적인 대표 GIS 포맷이다. SHP는 도형정보의 위치를 얻을 수 있는 *.shp와 *.shp에 담겨있는 도형정보의 위치를 얻을 수 있는 인덱스 파일인 *.shx, 도형에 대한 속성정보를 담고 있는 *.dbf로 구성되어 있다. 본 논문에서는 SHP의 도형정보와 속성정보를 따로 관리하는 분리형 GIS 포맷의 특징과 함께 쉽게 접근 가능한 dBase IV 구조로 되어있는 *.dbf를 활용해 계층적 POI DB를 설계한다. 그러나 *.dbf 파일은 POI 서비스를 하기에는 좀 미비한 점이 있다[7].

첫째, 매칭되는 도형에 대한 최소한의 속성정보만을 가지고 있어 이를 이용해 POI 서비스를 했을 경우 단순한 이름들만을 보여주는 정도에 그친다. 둘째, 속성에 포함되는 분류체계는 기존 DXF 표준도엽 코드를 사용하여 분류되어 있다. 이는 대, 중, 소, 상세분류인 네 계층으로 나누어져 있으나 최하위 계층인 상세분류에서도 초등학교, 중학교, 대학교 등을 표현하지 못할 정도로 그 세분화 정도가 자세하지 못하여 사용자로 하여금 원하고자 하는 위치의 POI 정보에 대해 손쉬운 검색을 불가능케 한다. 이러한 문제점을 보완하고자 다양한 정보 추가와 세분화된 분류체계를 두어 계층적 POI DB를 설계한다.

계층적 POI DB는 총 15개의 필드로 정의되며, 주요 필드의 상세는 다음과 같다.

ObjectId는 각각의 도형에 관한 식별자를 가리키며 Gu_cd와 Dong_cd는 도형이 위치한 행정구역을 가리킨다. Blg_nm은 도형이 가지고 있는 이름이며, X_coord와 Y_coord는 도형의 위치, Shape_area는 도형건물의 공간을 나타낸다. 이외에도 Bldg_id, Blg_no, Road_no, Cls_cd, Fac_cd, Land_cd,

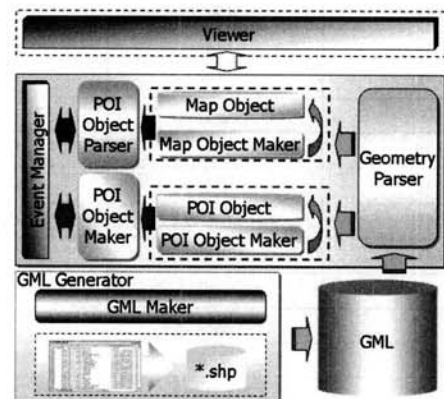
Bonbun, Bubun, Image, Recno, Bldg_cd, Fac_cd, Shape_len 필드가 있으나 이 필드들은 사용자들에게 제공될 POI 정보에 포함할 필요 없는 의미 없는 정보로 GML로의 변환 과정에서 삭제된다. 기반데이터인 SHP가 지원하는 POI와 관련된 기본 정보 이외에 사용자 중심의 실제 POI 속성 정보가 추가된다. 추가된 정보는 주소에 해당하는 add1, 상세주소인 add2, 전화번호인 Telephone, 상세 정보 및 특징을 제공하는 EX, 홈페이지 URL이 포함된다. 특히, 표준수치지도에서 참고한 도엽코드가 실제 사용자에게는 보이지 않지만 계층적 구분을 위한 키 값과 POI 정보의 표현 유무를 결정하기 위해 O_X 필드를 포함한다.

4. GVP 설계 및 구현

4.1 GVP 구조

GVP는 크게 데이터 분석 및 분류를 하는 Parser 영역과 분석된 데이터를 기반으로 내부 속성 데이터를 만드는 Maker 영역으로 분류되며, 이들 내부에는 5개의 세부모듈을 포함한다. 이 세부모듈은 기 개발된 GML Generator를 통해 생성된 GML 파일의 속성을 분석하여 Map Data와 POI 정보로 분류하는 Geometry Parser, 분류된 속성들을 바탕으로 지도 및 POI 가시화에 기반이 될 오브젝트 형태의 새로운 내부 속성 데이터를 만드는 Map Object Maker와 POI Object Maker, 생성된 오브젝트들을 가지고 가시화를 담당하는 Map Object Parser와 POI Object Parser가 있다. GVP의 전체 구조는 (그림 4)와 같다.

먼저 원시 데이터인 SHP에서 변환된 GML 파일을 Geometry Parser에 의해 지도 가시화에 필요한 속성과 POI 가시화에 필요한 속성을 분류하여 해당 오브젝트들을 생성할 Map Object Maker와 POI Object Maker로 보내준다. Map Object Maker는 지도 가시화를 위한 오브젝트인 Map Object를 생성한다. POI Object Maker에서는 POI 가시화를 위한 POI Object를 생성한다. 이렇게 해서 생성된 Map Object와 POI Object는 Map Object Parser와 POI Object Parser에서 가시화 및 이벤트 처리를 하고 최종적으로 Viewer에 가시화 된다.



(그림 4) GVP의 전체 시스템 구조

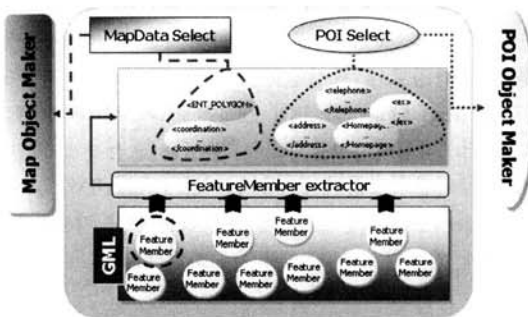
4.1.1 Geometry Parser

Geometry Parser는 File Importer에서 불러온 FeatureMember를 분석하여, 지도를 표현하는 요소와 POI를 표현하는 요소로 분류하여 해당하는 Maker에 전달하는 역할을 한다. GML은 FeatureMember 엘리먼트 단위들로 구성되며, 이 엘리먼트는 지도를 그리는데 있어 하나의 독립적인 요소들을 표현하는 독립객체이다. FeatureMember extractor는 FeatureMember를 하나씩 불러와 해당 FeatureMember의 하위 엘리먼트들을 분석한다. MapData Select는 지도 표현을 위한 엘리먼트인 Polygon, Polyline, Circle 등의 Geometry Type과 해당 객체를 표현할 정점들에 대해 TM 좌표들로 이루어진 coordinates들을 추출하여 Map Object Maker로 보내준다. POI Select는 POI 표현을 위한 주소, 상세주소, 전화번호, 홈페이지, 특징 엘리먼트들과 POI 검색을 위한 Key_cord, S_Cord, O_X 엘리먼트를 추출하여 POI Object Maker로 보낸다. (그림 5)는 Geometry Parser 모듈의 세부 구조이다.

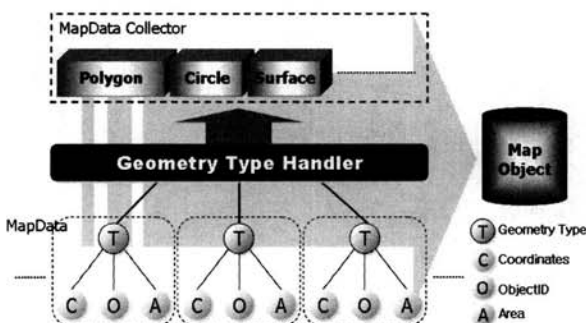
4.1.2 Map Object Maker

Map Object Maker는 Geometry Parser로부터 전달된 지도 표현 엘리먼트들을 지도 가시화와 이벤트 발생의 빠른 데이터 처리를 위한 Map Object를 생성하는 역할을 한다. (그림 6)은 Map Object Maker의 세부 구조이다.

Geometry Type Handler는 도형속성인 Polygon, Polyline, Surface 등으로 정의된 Geometry Type을 분류한다. MapData Collector는 분류된 속성을 따르는 좌표정보와 객체의 식별



(그림 5) Geometry Parser 모듈 구조



(그림 6) Map Object Maker 모듈 구조

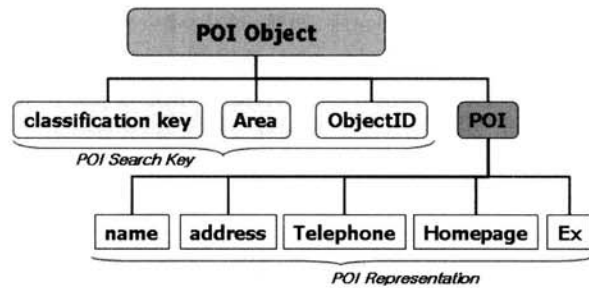
을 위한 ObjectID, Area를 포함하여 Type별로 묶어 블록별로 수집한다. 이렇게 수집된 Type 블록들은 Extractor를 통해 Map Object로 생성된다.

4.1.3 POI Object Maker

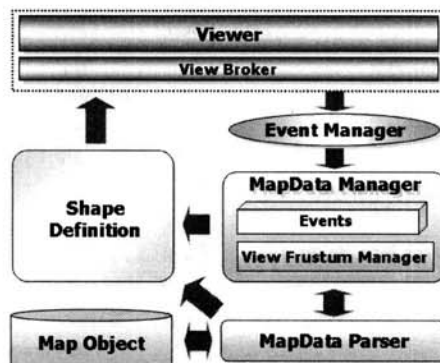
POI Object Maker는 Geometry Parser로부터 전달된 POI 엘리먼트들을 POI Analyzer에 의해 POI Search와 POI Representation로 분류된 POI Object를 생성한다. POI Search는 3-Layer 구조를 따르는 계층적 검색을 위한 식별키가 전달되며, POI Representation은 화면에 가시화될 실제 POI 정보 표현 부분이 전달된다. POI Search는 Area, ObjectID, Key 등을 전달받아 사용자가 POI 검색을 요청할 때, setArea, setObjectID, setKey 등의 메소드를 통해 설정된다. POI Representation는 실제 화면에 가시화될 정보들인 주소, 이름, 전화번호, 홈페이지, 특징 등을 전달받으며, getAddress, getBLG_NM, getTelephone, getHomepage, getEx 메소드에 의해 원시 코드인 엘리먼트 형식의 코드를 얻는다. POI Object Maker는 POI Search에 의해 키가 설정되고 POI Representation에 의해 가시화 정보를 트리구조 형태로 POI Object Extractor를 통해 POI Object를 생성한다(그림 7).

4.1.4 Map Object Parser

Map Object Parser는 Map Object Maker에 의해 생성된 Map Object를 Viewer에 가시화 하며 사용자에게 의해 발생되는 이벤트 처리를 담당한다. (그림 8)은 Map Object Parser의 세부 구조이다.



(그림 7) 생성된 POI Object 형태



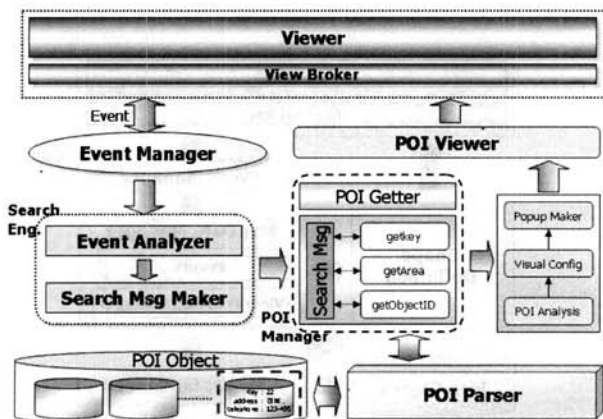
(그림 8) Map Object Parser 구조

Map Object Parser는 Map Object의 MapData를 Geometry Type에 따른 Draw 메소드들이 정의되어 있는 Shape Definition을 이용하여 Viewer에 가시화 한다. 초기에 로드된 파일은 특정 경로를 거치지 않고 Viewer에 가시화된다. Viewer에서 지도와 관련된 사용자 이벤트는 확대/축소/이동/전체보기를 제공한다. 사용자 이벤트가 발생되면, 이벤트를 처리하는 공통 모듈인 Event Manager는 지도 관련 이벤트인 확대/축소/이동 등을 판단하여 MapData Manager로 보내준다. MapData Manager는 사용자 이벤트를 처리하기 위해 지도의 재가시화가 필요하므로, View Frustum Manager를 통해 가시화 영역을 재설정한다. MapData Parser는 전달된 가시 영역에 해당하는 MapData를 추출하며, Shape Definition에 의해 Viewer에 재가시화 한다.

4.1.5 POI Object Parser

POI Object Parser는 POI Object Maker에서 생성된 POI Object를 이용하여 사용자가 POI 검색을 요청할 경우 Viewer에 팝업창 형태로 가시화해주는 역할을 한다. (그림 9)는 POI Parser의 세부 구조이다.

Viewer에서 POI 검색을 요청할 경우, 공통 모듈인 Event Manager는 POI 관련 이벤트를 분류해 POI Parser로 이벤트를 알린다. POI 관련 이벤트는 POI 정보 보기와 계층적 POI 검색을 제공한다. 사용자로부터 POI 보기 이벤트가 발생되면 공통 모듈인 Event Manager는 POI 이벤트가 발생된 건물들의 ObjectID, Area, X/Y 좌표를 Search Eng.에 전달한다. 전달된 이벤트는 Search Eng.의 Event Analyzer에서 검색 요청된 건물들의 ObjectID, Area 등을 메시지화 시켜주는 Search Msg Maker에 의해 메시지를 생성한다. POI Manager는 검색 메시지가 넘어 왔을 경우 해당 메시지 값을 POI Parser로 보낸다. POI Parser는 POI Object에 접근해 메시지 값을 검색하여 요청된 식별키를 리턴한다. POI Getter는 POI Manager로 전달된 키와 Search Eng.에서 넘어온 검색 메시지를 매칭한 후, POI Object에서 가시화할 POI 정보를 가져온다. 이렇게 얻어진 POI 정보는 POI Analyzer에서 최종 POI 분석을 하고 Visual Config에서 가



(그림 9) POI Parser 모듈 구조

시화할 부분만 추출된다. POI 정보는 PopUp Maker를 통해 팝업 형태로 가시화 한다.

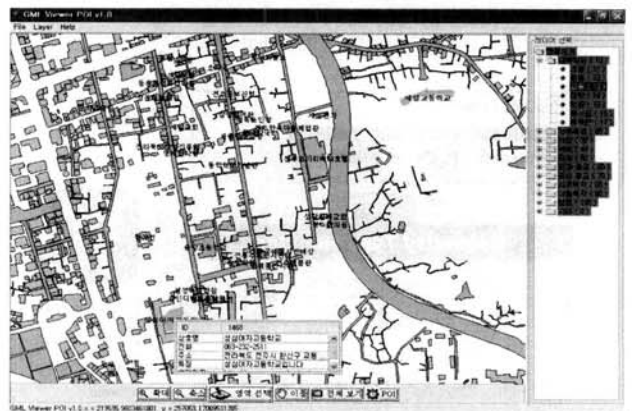
4.2 GVP 지도 및 POI 가시화

GVP 가시화 영역은 전주시 SHP의 일부인 “전라북도 전주시 완산구 풍남동 한옥마을”을 실례로 보인다.

GVP는 지도 가시화 관점에서 도로와 건물의 색상을 구분하며, 사용자 인터페이스는 각종 버튼 정의로 사용자의 편리성에 최대한 초점을 두었다. GVP가 제공하는 기능은 확대/축소/영역선택/상하좌우 이동/지도 전체보기/POI/계층적 검색이 있다.

POI 정보 가시화는 계층적 검색을 통한 POI 정보 보기와 어떤 상태에서든 POI 아이콘을 눌러 POI 정보를 보고 싶은 건물을 클릭해서 보는 방법으로 두 가지로 제공한다. 3-Layer POI 구조를 따르는 계층적 검색창에서는 U-Meta의 교육학원과 그 하위 L-Meta의 고등학교에 해당하는 Name Layer의 건물과 지명이 강조되어 표시된다. 즉 DXF 수치지도 지형 코드의 세 분류로 일반화된 정의를 계층적인 POI 구조 정의로 구체화한다. 또한 (그림 10)과 같이 Name Layer의 건물과 지명 정의에 해당하는 “성심여자고등학교”와 Contents Layer의 상세 정보를 팝업창으로 제공한다.

GVP는 (그림 10)에서와 같이 “성심여자고등학교”의 POI를 가시화하기 위해 먼저 원시파일인 SHP의 *.shp와 *.dbf 내용 전체를 추출해 도형정보와 POI를 매칭시킴과 동시에 각각 객체마다 GML 문법의 FeatureMember로 구성한다. SHP 파일의 모든 객체들이 FeatureMember로 구성되면 GML 기본 스키마와 함께 하나의 GML 파일을 생성한다. 생성된 GML 파일은 GVP에서 불러와 하나 하나의 FeatureMember들을 구성하는 도형정보와 속성정보를 따로 분류하여 저장하고 이 각각을 Map Object와 POI Object로 저장한다. 이후 사용자로부터 “성심여자고등학교”의 POI를 요청하게 되면 지도상에서 클릭된 “성심여자고등학교”의 ObjectID와 Area값이 POI Object로 전달되고 이에 해당하는 POI 정보를 팝업으로 구성해 최종적으로 화면에 가시화하게 된다.



(그림 10) GVP Map 및 계층적 POI 가시화

5. 결론 및 향후 연구

대부분의 지도 서비스는 서로 다른 지도 포맷을 가지고 있으므로 사용자가 표준화된 POI 정보 서비스를 제공받기가 어렵다. 본 논문에서는 지도 포맷과 POI 정보 서비스의 이질성과 비호환성을 극복하기 위해 GML 명세에 기반한 GVP를 구축하였다.

2차원 벡터 데이터의 사실상 표준으로 사용될 만큼 광범위한 사용자층을 확보하고 있는 SHP 파일은 도형정보와 속성정보를 따로 관리하는 장점이 있다. 본 논문에서는 SHP 파일의 속성정보를 관리하는 dBase 파일을 계층적 POI 구조로 재구성하여, 기존 DXF 표준 레이어코드의 한계성을 극복함으로써, 사용자에게 좀 더 구체적인 POI 검색을 지원한다. 계층적 POI는 Meta, Name, Contents Layer의 3-Layer 구조를 가진다. 특히 Meta Layer에서는 U-Meta와 L-Meta 코드 분류표를 정의하여 POI 정보를 구조적으로 세분화하였다. 또한 사용자들이 실질적으로 원하는 정보들을 Contents Layer에 포함하여 좀 더 상세한 사용자 중심의 POI 정보를 제공한다.

GVP는 GML 파일내의 지도 정보와 POI 정보를 따로 분류하였으며, 각각의 Object Maker로 오브젝트를 생성하여 관리한다. 이로 인해 GVP는 GML의 단점 중에 하나인 파일의 무거움에 따른 이벤트 발생시 파일에 접근이 느리다는 단점을 보완하여 지도 가시화 및 POI 가시화의 효율성 지원과 사용자의 이벤트에 대한 빠른 응답을 제공한다. 또한 GVP는 편리한 사용자 인터페이스를 지원한다. 지도 가시화 측면에서는 확대/축소, 영역 선택/이동/전체보기 등의 컨트롤러 아이콘을 제공하고 건물과 도로의 색상을 구분하였다. POI 가시화 측면에서 계층적 POI 구조를 설계하여 디렉터리 형태의 그룹화된 POI 정보 제공 및 팝업 형태의 상세 정보를 나타낸다.

향후 연구과제로는 GVP를 휴대성이 용이한 모바일 디바이스 환경으로 적용하기 위해 디바이스 환경의 제약사항을 고려한 POI 정보의 경량화 방법을 연구하고, 이를 통해 하여 좀 더 편리한 서비스를 제공한다. 또한 현재 위치에서 목적지까지 길을 안내해주는 길안내 서비스 제공과 주요 건물에 아이콘을 삽입하여 사용자들로부터 각 건물들에 대한 식별성 향상시키고 지도 서비스의 2D에서 3D로 정보 표현에 의한 현실감을 제공한다.

참고 문헌

- [1] OpenGIS Consortium, Inc., "OpenGIS Location Service Core Services," <http://www.opengeospatial.org/>
- [2] Mehdi Essid, Omar Boucelma, Franis, Marie Colonna, Yassine Lassoued, "Query Processing in a geographic mediation system," GIS: Geographic Information Systems, pp.101-108, 2004. 11.
- [3] Omar Boucelma, Jean, Yves Garinet, ZoLacroix, "The

virGIS WFS-based spatial mediation system," CIKM : Conference on Information and Knowledge Management, pp.370-374, 2003. 11.

- [4] 김대식, 김형진, 손봉수, 유완, "생활 지리정보 검색 및 안내를 위한 POI 구축 및 활용," 한국콘텐츠학회추계종합학술대회논문집 Vol.1, No.2, pp.423-430, 2003.
- [5] OpenGIS Consortium, Inc., Geography Markup Language (GML) Implementation Specification, <http://www.opengeospatial.org/docs/02-023r4.pdf>
- [6] 국립지리원, "수치지형도 레이어/심볼/코드", <http://www.ngic.go.kr/NGIC3/user/book/OpenPDSSearch.jsp>
- [7] TatukGIS Inc. TatukGIS Viewer 1.4, <http://www.tatukgis.com>
- [8] Saft Software Inc. FME Universal Viewer 2003, <http://www.saft.com/products/fme/index.php>
- [9] Snowflake Software Ltd. OS Master Map Viewer 2.0, <http://www.snowflakesoftware.co.uk/products/viewer/index.htm>
- [10] eSpatial Inc. iSMART Explorer4.4, <http://www.espatial.com/productsismartexplorerversion44.htm>
- [11] 안성우, "텔레메틱스 환경에서의 목적지 검색과 교통정보 전달 기법 연구", 석사학위논문, 아주대학교정보통신전문대학원, 2006. 2.
- [12] 지영민, "개인화를 위한 POI 검색 및 지도 제공 방법 및 시스템", 특허출원(특허번호: 10-0707714-0000, 2007. 4.)
- [13] 콩나물닷컴, <http://www.congnamul.com/>



송 은 하

e-mail : ehsong@wku.ac.kr

1997년 원광대학교 통계학과(이학사)

2000년 원광대학교 컴퓨터공학과(공학석사)

2006년 원광대학교 컴퓨터공학과(공학박사)

2007년~현 재 원광대학교 전기전자 및 정보공학부 전임강사

관심분야: LBS, 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템



박 용 진

e-mail : yjpark1@wku.ac.kr

2006년 원광대학교 전기전자 및 정보공학부 (공학사)

2008년 원광대학교 컴퓨터공학 (공학석사)

관심분야: LBS, GIS



정 영 식

e-mail : ysjeong@wku.ac.kr

1987년 고려대학교 수학과(이학사)

1989년 고려대학교 전산학과(이학석사)

1993년 고려대학교 전산학과(이학박사)

2004년 웨인 주립대학교 객원교수

1993년~현 재 원광대학교 전기전자 및
정보공학부 교수

관심분야: 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템, LBS