

## 계층적 POI 정보 가시화를 위한 GVP 개발

박용진<sup>○</sup> 김흠 한원희 서세광 송은하 한성국 정영식  
원광대학교 컴퓨터공학과

{yjpark1<sup>○</sup>, kimgECKO, samsung, skseo, ehsong, skhan, ysjeong}@wku.ac.kr

### A Development of GVP for Hierarchical POI Information Visualization

Yong-Jin Park<sup>○</sup> Jin Xin Won-Hee Han Se-Kwang Seo Eun-Ha Song  
Sung-Kook Han Young-Sik Jeong

Dept. of Computer Engineering, Wonkwang University

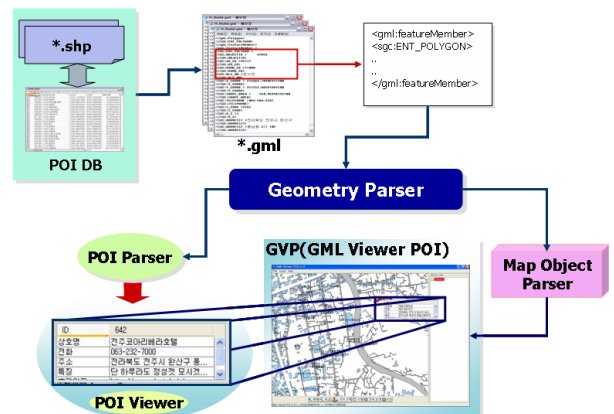
#### 요 약

현재 컴퓨터 성능의 발달과 함께 GIS 서비스는 기존의 단순 지도 서비스를 비롯한 각 개체들의 위치 정보인 POI 서비스까지 요구되고 있다. 하지만 대부분의 GIS 서비스는 그 핵심적 기반이라 할 수 있는 지도 데이터를 표현하는데 있어서 각 개발 업체 및 연구기관마다 각기 다른 지도데이터 포맷을 사용하고 있다. 이는 곧 데이터의 중복 구축뿐만 아니라 기존 GIS간의 상호 운용성이 결여되어 서로 간에 호환 불가라는 문제점을 야기 시킨다. 또한, POI 서비스에서도 POI를 이루는 개체들에 대한 위치 정보가 불충분하여 단순히 지명 서비스만을 제공하고 있어, 다양한 정보 서비스를 제공 받고자 하는 사용자의 요구를 충족시키지 못하고 있다. 이에 본 논문에서는 지도 표준인 GML을 기반으로 하여 지도 가시화의 중복 구축 및 비호환성을 극복하고, 단순한 POI 정보에 대해서는 계층적이고 사용자 중심적인 위치정보를 구축하고 이를 가시화하는 GVP를 개발한다.

#### 1. 서 론

최근 GIS는 기존의 단순 지도서비스에서 좀 더 나아가 사용자들에게 더 많은 정보를 제공해주기 위한 POI 서비스까지 제공하고 있다. POI는 건물, 도로, 상점, 관공서, 병원 등 지도 내에서 사용자가 관심 있어 하는 개체를 말한다. 이를 통해 사용자는 관심을 가지는 곳의 정보를 손쉽게 얻을 수 있다. POI는 전화번호나 지번데이터 같은 해당 개체가 이루고 있는 모든 특성 및 정보들을 표현한다. 이러한 POI 정보를 지도 서비스에 추가하여 지도 가시화와 함께 각 개체들의 정보들까지 표현함으로써 GIS의 효율성과 활용도를 한층 증대시킬 수 있다. 그러나 현재의 GIS 서비스는 핵심 기반이 되는 지리공간데이터의 표현과 함께 POI정보를 제공하는데 있어서 표준안이 제정되었음에도 불구하고 아직까지 그 표준화를 제대로 따르지 않는 실정이다. 이는 곧 각 개발업체들은 서비스 제공에 있어 특정 시스템이나 환경에 의존적이며 그에 따른 제약이 많은 형편이다. 특히 POI 서비스는 해당 개체의 정보가 불충분하여 단순한 지명 서비스에 그치는 정도이

다. 따라서 POI 서비스 사용자는 요구하는 정보에 알맞은 체계적인 검색 기능을 제공하지 못하고 있다[2]. 이에 본 논문에서는 OGC에서 제안하고 현재 우리나라의 지리정보 엔코딩 표준인 GML 3.0에 기반하여 지도를 가시화하고 POI 정보를 가시화하기 위한 모듈을 포함하는 GVP(GML Viewer POI)를 개발한다[1][3][7]. 특히 GVP는 POI 정보를 표현하는데 있어서 DXF 표준 수치지도[4]를 참조한 계층적 3-Layer 구조를 정의하여 사용자 중심의 POI 정보를 제공한다. (그림 1)은 본 논문에서 설계 및 개발한 GVP의 전체 개념도이다.



(그림 1) GVP 전체 개념도

\* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음. [2007-S-030-01, 유비쿼터스 환경기반 확장형 Mobile Modular Device 기술 개발]

2. 관련연구

2.1 기존 지도 서비스

OGC에서는 현재 서로 이질적인 환경에서 운용되고 있는 지리정보데이터를 좀 더 구조화된 정보 표현을 할 수 있고, 모든 종류의 데이터에 적용될 수 있는 XML을 GIS 환경에 도입하여 상호 운용을 하기 위해 GML 명세를 제시하였으며, 현재 표준으로 각광 받고 있는 GIS 포맷이다. 이러한 GML을 기반으로 한 가시화 시스템들은 TatukGIS Inc.의 TatukGIS Viewer 1.4, SafeSoftware Inc.의 FME Universal Viewer, Snowflake Software Ltd.의 OS Master Map Viewer 2.0, eSpatial Inc.의 iSMART Explorer 4.4, ETRI의 GML Viewer 등이 있으나, 이들은 단순히 지도가시화만 해줄 뿐 실질적인 POI 정보 서비스는 제공하지 않는 단점이 있다.

2.2 기존 POI 서비스

Web상에서 POI 서비스를 해주는 대표적인 시스템으로는 콩나물닷컴이 있다. 이 시스템에서는 건물명이 주요 건물에만 표시가 되고 확대 축소를 하여도 주위 건물에 대한 상세한 정보들이 나오지 않아 사용자가 위치를 찾아 가거나 주변 정보를 살펴보는데 있어 많은 불편함이 있다. 또한 검색을 위해 건물명을 선택할 경우, 상세 POI 정보를 제공하지 않는다.

본 논문은 사용자가 사용하기 쉬운 사용자 인터페이스에 대한 편리성에 중점을 준다. 지도 가시화에 있어서는 도로와 건물의 색을 다르게 하여 식별성을 증대시켰으며, POI 정보는 특정 건물을 선택했을 경우 그에 따르는 상세 정보들을 팝업창 형식으로 표현한다. 또한 기존 서비스들의 단점인 지리 데이터의 비 표준화에 대해서도 본 논문은 GML 표준을 기반으로 하여 단점을 보완한다.

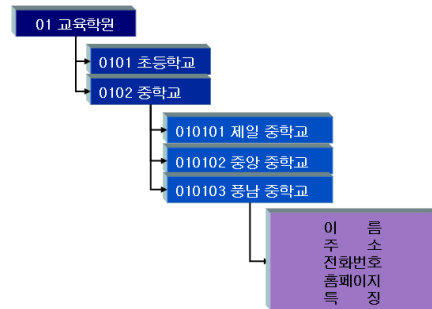
3. 계층적 POI 설계

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 참조하여 Meta Layer, Name Layer, Contents Layer를 구성하는 3계층으로 설계한다. 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 기

본적 구성 요소로서 U-Meta와 L-Meta로 이루어진다. L-Meta코드는 U-Meta에 비해 좀 더 하위계층의 분류이며 각 U-Meta는 특성별 하위 계층으로 L-Meta 코드를 가진다.

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



(그림 2) 3-Layer POI의 실례

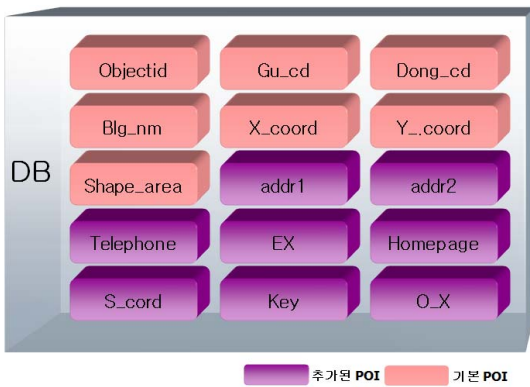
(그림 2)은 3-Layer POI 구조로 정의된 실례이다. Meta Layer는 DXF 표준코드의 소분류와 유사한 교육학원을 가리키며, U-Meta 코드값으로 01을 주어진다. L-Meta 코드는 DXF 표준코드의 세분류와 유사한 초등학교, 중학교, 고등학교 등으로 교육학원의 특성별 세부분류로 나누어지며, 초등학교는 0101, 중학교는 0102, 고등학교는 0103의 코드 값을 부여한다. Name Layer는 중학교에 대한 학교명 분류로써 제일중학교, 중앙중학교, 풍남중학교 등에 대해 각각의 코드 값의 조합으로 식별키 값이 주어진다. DXF 표준 수치지도의 상세분류로 학교와 같이 포괄적으로 규정한 분류를 세분화하고 계층화된 코드 값을 부여한다.

3.2 계층적 POI DB

본 논문은 SHP 파일을 기반데이터로 계층적 POI DB를 설계한다. SHP파일은 도형정보와 속성정보를 다른 파일로 관리하고 있는 특성이 있어 POI를 추가 및 삭제하기가 기존의 다른 포맷에 비해 이점이 있다. 그러나 이러한 SHP파일의 속성정보를 담고 있는 dBase 파일은 POI 서비스를 하기에는 좀 미비한 점이 있다. 가장 큰 문제로는 도형에 대한 최소한의 속성만을 가지고 있어 이를 이용하여 POI 서비스를 한다면 기존 단순 지명서비스의 틀을 크게 벗어나지 않을뿐더러 그 속성들에 포함되어 있는 분류체계는 기존의 DXF 표준 도엽 코드를 사용하고 있다. DXF 표준 도엽 코드는 대, 중, 소, 상세분류인 네 계층으로 이루어져 있으나 최하위 계층인 상세분류의 분류체계로는 교육학원의 좀 더 하위계층이라 할

수 있는 초, 중, 고등학교 등의 표현이 불가능하다는 단점이 있다. 이는 곧 사용자들의 맞춤형 서비스를 하기에는 그 분류단계가 너무 광범위하다는 단점이 있다. 이에 본 논문에서는 좀 더 다양한 정보의 추가와 세분화된 분류체계를 두어 계층적 POI DB를 설계한다.

계층적 POI DB는 총 15개의 필드로 정의되며, 기존 dBase에 정의되어 있는 필드에 사용자 중심의 실제 POI 속성 정보를 추가한다. 추가된 정보는 주소에 해당하는 add1, 상세주소인 add2, 전화번호인 Telephone, 상세 정보 및 특징을 제공하는 EX, 홈페이지 URL이 포함된다. 특히, 표준수치지도에서 참고한 도엽코드, 그리고 실제 사용자에게는 보이지 않지만 계층적 구분을 위한 키값과 POI 정보의 표현 유무를 결정하기 위한 O\_X 필드가 추가된다(그림 3).

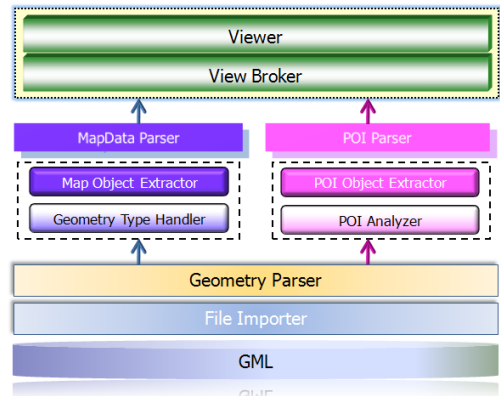


(그림 3) POI DB 필드 구성

#### 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 세부 모듈을 포함한다. 이 세부모듈은 기 개발된 GML Generator를 통해 생성된 GML 파일의 속성을 분석하여 지도 및 POI가시화를 한다. GVP의 전체 구조는 (그림 4)와 같다.

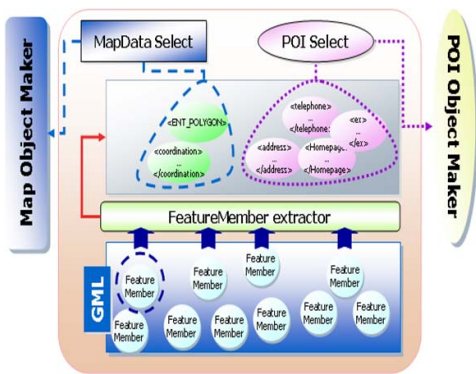


(그림 4) GVP 전체 구조

먼저 원시 데이터인 SHP에서 변환된 GML 파일을 Geometry Parser에 의해 지도 가시화에 필요한 속성과 POI 가시화에 필요한 속성을 분류하여 각각의 오브젝트들을 생성할 Map Object Maker와 POI Object Maker로 보내준다. Object Maker는 지도와 POI의 가시화를 위한 내부 데이터형인 Map Object와 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.

##### 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 그리는 데 있어 하나의 독립적인 요소들을 표현하는 독립객체이다. FeatureMember extractor는 FeatureMember를 하나씩 불러와 해당 FeatureMember의 하위 엘리먼트들을 분석한다. MapData Select는 Polygon, Polyline, Surface 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.



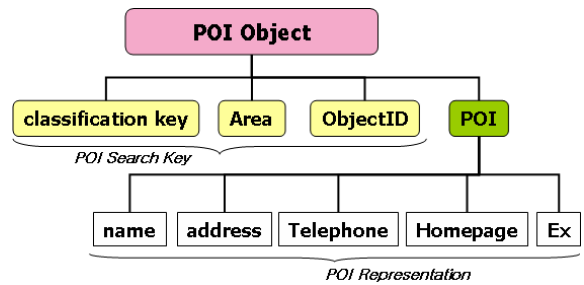
(그림 5) Geometry Parser

#### 4.1.2 Map Object Maker

Map Object Maker는 Geometry Parser로부터 전달된 지도 표현을 위한 엘리먼트들을 지도 가시화 및 사용자 이벤트 발생에 대해 빠른 데이터 처리를 위한 Map Object를 생성하는 역할을 한다. 도형속성인 Polygon, Polyline, Surface등의 분류를 위해 Geometry Type Handler를 두었으며, 분류된 도형속성들은 각각의 좌표 정보 및 객체의 식별을 위한 ObjectID, Area를 포함하여 Type별로 묶어서 블록별로 수집한다. 수집된 Type 블록들은 Extractor를 통해 Map Object를 생성한다.

#### 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 Representation은 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 6).



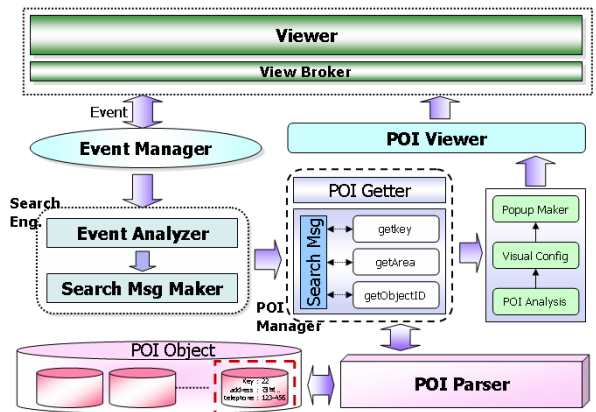
(그림 6) 생성된 POI Object 구조

#### 4.1.4 Map Object Parser

Map Object Parser는 Map Object를 Viewer에 가시화하고 사용자 이벤트에 대해 처리를 담당하게 된다. Map Object Parser는 Map Object의 MapData를 Geometry Type에 따른 가시화 컴포넌트들을 정의해 놓은 Shape Definition을 이용하여 Viewer에 가시화 한다. 초기에 로드된 지도는 특정 루트를 거치지 않은채 바로 Viewer에 가시화 된다. Viewer는 지도에 관련된 사용자 이벤트로 확대, 축소, 이동, 전체보기를 제공한다. Viewer로부터 사용자 이벤트가 발생되면, 이벤트를 처리하는 공통 모듈인 Event Manager는 지도 관련 이벤트를 판단하여 MapData Manager로 보내준다. MapData Manager는 사용자 이벤트를 처리하기 위해 지도의 재가시화가 필요하므로, View Frustum Manager를 통해 가시화 영역을 재설정한다. MapData Parser는 전달된 가시화영역에 해당하는 MapData를 추출하며, Shape Definition에 의해 Viewer에 재가시화 한다.

#### 4.1.5 POI Object Parser

POI Object Parser는 POI Object를 이용하여 사용자가 POI 검색을 요청할 경우, Viewer에 팝업창 형태로 가시화해주는 역할을 한다. (그림 7)는 POI Parser의 세부 구조이다.

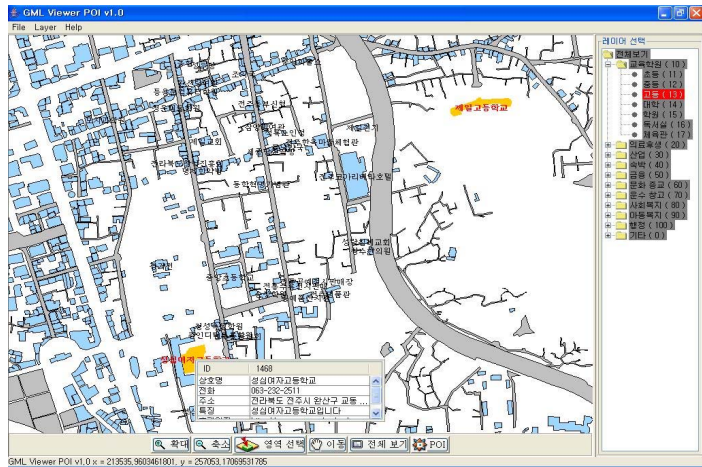


(그림 7) POI Parser 모듈 구조

Viewer에서 POI 검색을 요청할 경우, 공통 모듈인 Event Manager는 POI 관련 이벤트만을 분류해 POI Parser로 이벤트를 알린다. POI 이벤트로는 POI 계층적 검색과 POI 검색이 있다. POI 이벤트는 Search Eng.의 Event Analyzer에서 검색 요청된 건물의 ObjectID와 Area를 분석하여 Search Msg Maker를 통해 검색을 위한 메시지로 변환한다. 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 가시화할 부분만 추출된다. POI 정보는 팝업 형태로 가시화되는데, 이는 PopUp Maker를 통해 팝업창으로 생성하여 가시화한다.

4.2 GVP 지도 및 POI 가시화

GVP 가시화 영역은 전주시 SHP 파일의 일부인 “전라북도 전주시 완산구 풍남동 한옥마을”을 실례로 한다. GVP는 지도 가시화 관점에서 도로와 건물의 색상을 구분하며, 사용자 인터페이스는 각종 버튼 정의로 사용자의 편리성에 최대한 초점을 두었다. GVP가 제공하는 기



(그림 8) GVP Map 및 계층적 POI 가시화

능은 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 Meta Layer인 교육학원과 그 하위의 고등학교를 선택하

였으며, 해당 Name Layer의 건물과 지명이 강조되어 표현된다. (그림 8)은 성심여자고등학교 상세 정보인 Contents Layer를 팝업창으로 나타낸다.

5. 결론 및 향후 연구

현재의 지도 서비스들은 서비스 업체마다 각각의 다른 지도 데이터 포맷을 가지고 서비스를 제공하고 있어 사용자 하여금 일률적인 서비스를 제공받지 못하고 있다. 또한 POI 서비스에 있어서도 그 정보가 불충분하여 사용자들이 원하는 정보들을 충분히 제공해주지 못하고 있다. 본 논문에서는 기존 POI서비스를 보완하여 좀 더 확장된 POI 서비스를 제공한다. POI의 확장을 위해서는 SHP파일의 도형정보와 속성정보를 따로 관리하는 특징을 활용하여 기존 속성정보에 추가적으로 POI를 삽입하였으며, 추가된 POI는 계층적 구조로 되어 있어 검색 및 추가, 삭제가 용이하도록 하였다. 확장된 POI정보를 포함하는 SHP파일은 기 개발된 GML Generator를 통해 GML 표준 포맷으로 통합하고, 이를 바탕으로 사용자에게 지도 및 POI 서비스를 할 수 있는 GVP를 구축하였다. GVP는 파일 내의 지도 정보와 POI 정보를 따로 관리하여 GML의 단점인 파일의 무거움에 따른 데이터 액세스 시간을 단축한다. 또한 사용자들로 하여금 편리한 인터페이스를 제공한다.

향후 연구과제로는 GVP를 휴대성이 용이한 모바일 디바이스 환경으로 적용하여 좀 더 편리한 서비스를 제공한다. 또한 현재 위치에서 목적지까지 길을 안내해주는 길 안내 서비스 제공, 주요 건물에 아이콘을 삽입하여 사용자들로부터 각 건물들에 대한 식별성 향상, 지도 서비스의 2D에서 3D로 정보 표현에 의한 현실감을 제공한다.

5. 참고문헌

[1] OpenGIS Consortium, Inc., "OpenGIS Location Service Core Services", <http://www.opengeospatial.org/>

[2] 김대식, 김형진, 손봉수, 유완, "생활지리정보 검색 및 안내를 위한 POI 구축 및 활용", 한국콘텐츠학회 2003 추계종합학술대회 논문집 Vol. 1 No. 2, pp423-430.

[3] OpenGIS Consortium, Inc., Geography Markup Language(GML) Implementation Specification, <http://www.opengeospatial.org/docs/02-023r4.pdf>

[4] 국립지리원, "수치지형도 레이어/심볼/코드", <http://www.ngic.go.kr/NGIC3/user/book/O>

penPDSSearch.jsp

- [5] "A GML-based Mobile Device Trace Monitoring System," Eun-Ha Song, Sung-Kook Han, Laurence T. Yang, Minyi Guo, Young-Sik Jeong, Aug. EUC06, 2006.
- [6] ESRI, "ESRI Shapefile Technical Description", <http://www.esri.com>, 1998.
- [7] Shashi Shekhar, Ranga Raju Vatsavai, Namita Sahay, Thomas E. Burk, Stephen Lime, "GML, Interoperability, and Standards: WMS and GML based interoperable web mapping system", Proceedings of the 9th ACM international symposium on Advances in geographic information systems, Nov. 2001.