

# Folksonomy를 위한 POI 관리도구 개발 (Development of POI Management Tool for Folksonomy)

김 흠<sup>†</sup>                      박 용 진<sup>†</sup>                      송 은 하<sup>\*\*</sup>  
(Jin Xin)                      (Yong-Jin Park)                      (Eun-Ha Song)

안 시 훈<sup>\*\*\*</sup>                      남 승 훈<sup>\*\*\*</sup>                      정 영 식<sup>\*\*\*\*</sup>  
(Si-Hoon Ahn)                      (Seung-Hoon Nam)                      (Young-Sik Jeong)

**요 약** 오늘날 웹은 가장 폭넓게 사용되는 기술이고 과거 몇 년 동안 웹은 여러 단계를 통해 진화되고 성숙된 기술이다. 특히 폭소노미, 태깅, RSS 등의 웹 인터페이스 기술이 사용자 중심으로 발전, 변화하고 있다. 본 논문에서는 POI 정보 관리를 일반적인 구조적 관리 방법 뿐만 아니라, 태깅 기술을 적용하여 POI 정보관리를 위한 POIM(POI Management tool)를 개발한다. 사용자 중심의 POI 서비스를 위하여 사용자가 직접 POIM를 이용하여 특정 POI 객체들에 XML 기반 태깅 정의, 수정, 삭제 등을 수행하면 각 POI 객체들이 GML로 저장된다.

키워드 : 폭소노미, 태깅, GIS, GML, POI

**Abstract** The web might be today's most widely-used technology and over the past few years, the web has evolved through various phases and is a mature technology Specially, the web interface technologies, such as folksonomy, tagging, RSS have been developed and changed with respect to user-friendly. In this paper, we should apply to not only the general structure scheme but also the tagging technology for POI information management and develop POIM(POI Management tool)for it. For POI service with user-friendly, the user can available to define, modify, and delete for a specific POI object based on XML by using POIM and then these should be saved on GML.

**Key words** : Folksonomy, Tagging, GIS, GML, POI

## 1. 서론

웹이 진화됨에 따라 블로그나 UCC등을 활용한 개인 콘텐츠 및 여러 웹 자원들의 양이 대폭적으로 증가되고 있다. 이러한 상황 속에서 무수히 쏟아져 나오는 정보들에 대해 기존의 카테고리 형식의 분류체계는 그 한계를 드러내고 있다. 소수의 전문가들이 정한 기준에 의한 분류체계로 정보를 분류한다는 건 시시각각 새로 생겨난 정보에 어울리는 적당한 카테고리가 존재하지 않는다면 그 정보는 정확한 위치에 분류될 수 없다. 또한 대용량 정보에 대하여 분류해주는 계층적 구조에 의해 각 개인마다 원하는 정보의 검색 결과를 얻기 위한 검색 단계에 대해서도 점점 복잡해지고 까다로워지고 있다. 이에 좀 더 효율적이고 사용자 중심적인 정보 분류 및 정보 검색을 위한 새로운 대안들이 나타나고 그러한 대안들 중에 가장 대두되고 있는 기술이 바로 태깅(tagging)이다. 태깅은 웹 자원들에 대해 태그형식의 사용자 정의 메타데이터를 기록하는 것이다. 웹 자원은 사용자의 자

· 본 연구는 국토해양부 첨단도시기술개발사업 -지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C03)에 의해 수행되었습니다.

† 비 회 원 : 원광대학교 컴퓨터공학과  
kimgecko@wku.ac.kr  
yjpark1@wku.ac.kr  
\*\* 정 회 원 : 원광대학교 전기전자및정보공학부 전임강사  
ehsong@wku.ac.kr  
\*\*\* 비 회 원 : 비트컴퓨터 기술연구소  
ashinfo@bit.co.kr  
nsh91@bit.co.kr  
\*\*\*\* 종신회원 : 원광대학교 전기전자및정보공학부 교수  
ysjeong@wku.ac.kr  
논문접수 : 2008년 3월 7일  
심사완료 : 2008년 8월 25일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지 : 컴퓨팅의 실제 및 레터 제14권 제7호(2008.10)

유로운 연상 작용을 통해 떠오르는 다수의 키워드를 이용하여 데이터를 분류함으로써 기존의 틀을 따를 필요 없이 각 사용자들의 선호도를 반영하는 다채로운 분류 체계가 생성된다. 또한, 기존의 어떤 계층구조에도 속할 수 없는 특이한 분야나 새롭게 생성되는 분야에 대해서도 태깅을 통해 분류될 수 있다[1-4].

이에 본 논문에서는 태깅 기술을 POI 서비스의 사용자 중심 정보 분류 및 정보 검색 등에 활용한다[5]. 먼저, POI 분류 기준은 기존 계층적 정보 분류체계를 그대로 사용하며, 여기에 사용자들이 직접 개인의 특성에 맞게 POI 객체들에 XML 기반의 태깅을 적용한다. POI 객체에 XML 기반 태깅을 함으로써 여러 사용자들에게 POI 정보를 공유시킨다. 이와 같이 기존의 계층적 POI 정보 관리 방법과 함께 POI 객체들에 태그 생성, 추가, 삭제, 수정 등의 기능을 구현하여 폭소노미(folksonomy)를 위한 POI 관리 도구를 개발한다. 본 논문에서는 POI 객체 내의 세부 항목 및 데이터 모든 부분에 대해서 태깅 뿐만 아니라 모든 편집이 가능한 POIM (POI Management tool)을 개발한다. 그림 1은 본 논문에서 제시하고자 하는 POIM의 논리적인 기능 개념도이다.

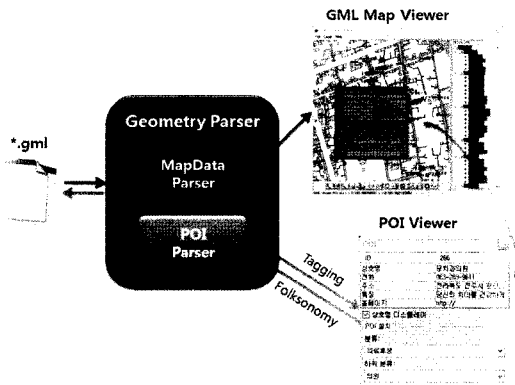


그림 1 시스템 전체 개념도

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 기술하고, 3장에서는 POI 태깅의 기반이 되는 개인화 태그 정보 생성을 기술한다. 4장에서는 POIM 설계를 기술하고, 5장에서는 POIM의 가시화와 6장에서 결론 및 향후 연구과제를 기술한다.

## 2. 관련연구

### 2.1 GML 기반 지도서비스

OGC에서는 이질적인 환경에서 운용되는 지리정보데이터를 구조화된 정보 표현이 가능하도록 GML 명세를 제시하였으며, 이는 현재 표준으로 각광받는 GIS 포맷

이다[6,7]. 이러한 GML을 기반으로 한 기존의 가시화 시스템들은 TatukGIS Inc.의 TatukGIS Viewer 1.4 [8], SafeSoftware Inc.의 FME Universal Viewer[9], Snowflake Software Ltd.의 OS Master Map Viewer 2.0, 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 있으며, 지점간의 거리와 점의 구성을 이루는 크기, 해당 지역을 측정 할 수 있으며 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 있으며, 확대, 축소, 이동 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.

### 2.2 기존의 태깅 서비스와 폭소노미

태깅은 현재 웹 2.0을 적용한 사이트들에 의해 지원되고 있다. 태깅 서비스의 대표적인 사이트로는 Flickr[11], del.icio.us[12], Technorati[13] 등이 있다.

Flickr는 현재 전 세계적으로 큰 인기를 얻고 있는 이미지 공유 사이트이다. 사용자는 자신의 이미지를 업로드하면서 이를 설명하는 태그를 부착 가능하다. 또한, 사용자 인터페이스는 AJAX를 사용하였다. 업로드할 사진의 지리적인 위치 정보를 담고 있는 위치 정보 태그를 추가할 수도 있다. 그리고, 자신이 찾고자 하는 이미지를 태그를 활용하여 검색할 수 있다. 태그 검색으로는 이진 표현 검색인 and, or, not등의 연산자를 지원하며 태그 이외의 텍스트 검색도 지원한다.

del.icio.us는 자신의 즐겨찾기를 저장하고 타인들과 공유하기 위한 소셜 북마킹(social bookmarking) 사이트이다. 사용자는 즐겨찾기를 추가하며 태그를 입력한다. 또한 사용자들은 그림 3과 같이 즐겨찾기에 사용할 태그를 추천 받을 수도 있다. 개개의 사용자들은 같은 URL을 북마킹하면서 서로 다른 태그를 자유롭게 사용할 수 있다. 태그 검색으로는 모든 and, or, not과 xor

의 이진 표현 검색을 지원한다.

Technorati는 태그 기반의 블로그 검색엔진이다. 사용자는 Technorati에서 자신의 블로그가 검색되도록 하기 위해서 자신의 블로그를 등록하고 인증을 받는다. 그리고 사용자가 블로그에 글을 쓸 때 클레임링크를 글에 삽입하면 Technorati는 크롤링을 통해서 태그와 블로그를 수집한다. 클레임링크는 Technorati에서 태그를 쉽게 수집하기 위해 사용하는 메타데이터이며 HTML의 형태로 되어 있다.

이러한 태그 서비스는 태그를 활용한 정보의 분류에 대해 사용자 기준의 자유로운 연상 작용을 통해 떠오르는 다수의 키워드를 이용하여 데이터를 분류함으로써 기존의 정해진 분류 체계의 틀을 따를 필요 없이 각 사용자들의 개성을 반영하는 다채로운 분류체계를 지원한다. 또한 정보의 검색에 있어서도 각 사용자의 기준에 맞는 분류체계를 따르기 때문에 동의어에 대한 검색에 오류를 범하지 않으며 기존의 트리 구조의 탐다운 검색 방식의 검색이 아닌 태그를 통한 수평적으로 검색하는 등의 신속하고 정확한 검색이 가능하다. 본 논문에서는 태그 기술을 POI 서비스에 적용하여 POI 정보의 분류 및 검색에 활용한다. 또한 POI 태그 정보를 XML파일로 생성해 POIM을 사용하는 사용자들간에 공유하도록 하여 POI 정보의 폭소노미를 제공한다.

### 3. 개인화 태그정보 생성

개인화 태그정보는 GML의 POI 정보들에 태그를 달

기 위한 기반으로 POI의 분류체계를 사용자가 직접 정의한 후 이를 XML포맷으로 분류정보를 패키징하여 PCode.xml이라는 파일로 저장한다. 저장된 분류정보와 GML의 <key> 태그는 서로 매칭이 되면서 POI 객체에 분류체계를 적용한다. 그림 2는 개인화 태그정보 생성 과정이다.

GML은 지도표현을 위한 마크업 언어로써 전체적인 구조는 GML 스키마에 의존적이다. 이런 특정 스키마에 국한된 GML파일은 각 객체의 POI 속성을 담당하는 태그들에 사용자가 임의적으로 수정하거나, 각각의 객체마다 POI 태그들을 다른 이름으로 작성한다는 건 문법상의 오류를 범하게 된다. 이에 본 논문에서는 GML파일의 기본 골격이라 할 수 있는 GML 스키마를 해하지 않으면서 각 객체의 POI마다 분류태그를 달기 위해 POI 정보에 <key> 태그를 추가하였다. <key> 태그의 값은 개인화 태그정보에 의해 결정된다.

개인화 태그정보는 사용자가 정의한 태그정보를 XML 파일로 따로 분류하고 GML 파일 내의 <key> 태그에 개인화 태그정보의 key값으로 설정하여 POI 객체에 분류체계를 적용시킨다. 사용자가 새로운 분류를 추가하면 NewChild라는 기본이름을 갖는 분류와 그에 따르는 기본 key값이 생성되고 이는 개인화 태그정보에 <NewChild key="">라는 엘리먼트로 추가 삽입된다. 기본 이름으로 정의된 NewChild는 Edit를 통해 사용자가 원하는 이름으로 변경이 가능하며 변경된 내용이 적용된다. 개인화 태그 정보는 최초 사용자에게 의해 분류가 되면

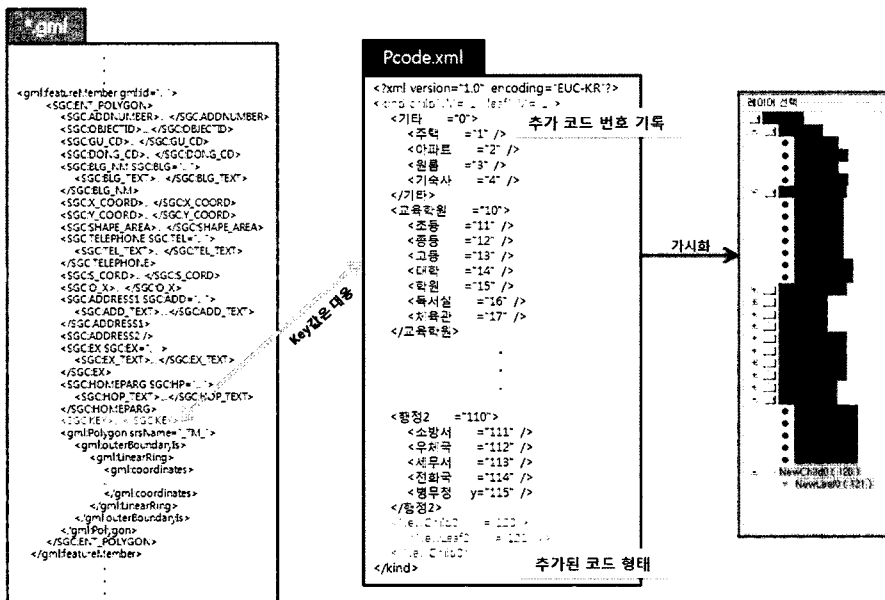


그림 2 개인화 태그정보 생성 과정

이는 단순히 태깅 정보에 불과하지만 POIM을 사용하는 사용자들에게 배포함으로써 해당 개인화 태그정보를 공유하는 사용자들간에 집단이 형성되며 이러한 집단을 통해 수시로 업데이트 되는 새로운 버전의 개인화 태그정보를 사용자들의 POIM에 적용 가능하다. 이는 웹 자원이 아닌 GIS의 POI 데이터에 대해서도 태깅 기술을 적용하고 태그정보들을 배포 및 공유를 통해 집단을 구성하여 폭소노미를 실현한다.

#### 4. POIM 설계

POIM은 GML 표준을 기반으로 하여 핵심 데이터들의 추출과 지도가시화 및 POI 가시화를 위해 중간계층의 내부 데이터로 분류하는 파서 영역과 파서를 통해 분류된 맵 데이터와 POI 데이터를 기반으로 가시화를 담당하는 가시화 영역으로 분류된다. 그림 3은 POIM의 전체 모듈 구조이다.

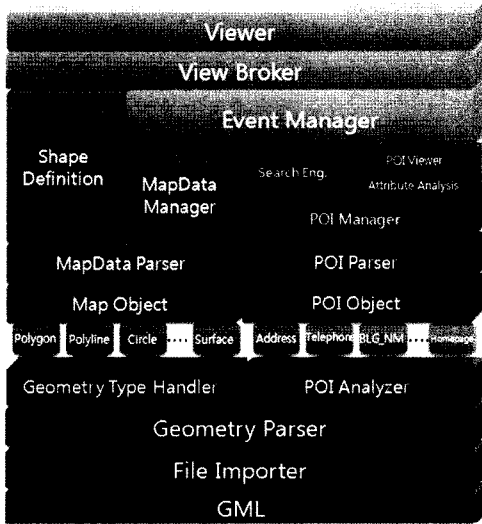


그림 3 POIM의 전체 시스템 구조

#### 4.1 Geodata 파서

Geodata 파서는 지도의 핵심 데이터들을 파싱하는 Geometry 파서, 파싱된 데이터를 Map과 POI라는 속성상의 분류를 담당할 MapData 핸들러와 POI 핸들러로 나뉘진다.

##### 4.1.1 Geometry 파서

Geometry 파서는 File Importer에서 불러온 FeatureMember를 분석하여 지도를 표현하는 요소와 POI를 표현하는 요소로 분류하여 해당하는 핸들러에 전달하는 역할을 한다. GML은 FeatureMember 엘리먼트 단위로 구성되며, 이 엘리먼트는 지도를 그리는데 있어 하나

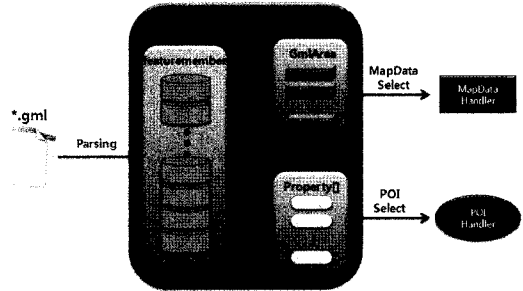


그림 4 Geometry 파서 구조

의 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 점점들에 대해 TM 좌표들로 이루어진 coordinates들을 추출하여 MapData 핸들러로 보내준다. POI Select는 POI 표현을 위한 주소, 상세주소, 전화번호, 홈페이지, 특징 엘리먼트들과 POI 검색을 위한 Key\_cord, S\_Cord, O\_X 엘리먼트를 추출하여 POI 핸들러로 보낸다. 그림 4는 Geometry 파서 모듈의 세부 구조이다.

##### 4.1.2 MapData 핸들러

MapData 핸들러는 Geometry 파서로부터 전달된 지도 표현 엘리먼트들을 지도 가시화와 이벤트 발생의 빠른 데이터 처리를 위한 Map Object를 생성하는 역할을 한다. 그림 5는 MapData 핸들러의 세부 구조이다.

Geometry Type 핸들러는 도형속성인 Polygon, Polyline, Surface 등으로 정의된 Geometry Type을 분류한다. MapData Collector는 분류된 속성을 따르는 좌표정보와 객체의 식별을 위한 ObjectID, Area를 포함하

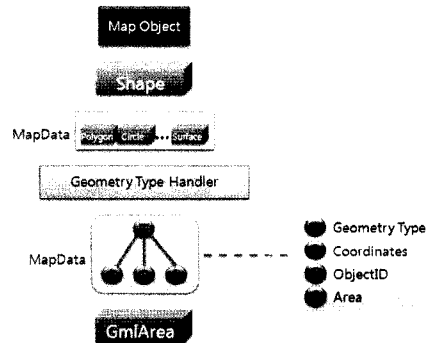


그림 5 MapData 핸들러 구조

여 Type별로 묶어 블록별로 수집한다. 이렇게 수집된 Type 블록들은 Extractor를 통해 Map Object로 생성된다.

4.1.3 POI 핸들러

POI 핸들러는 Geometry 파서로부터 전달된 POI 엘리먼트들을 POI Analyzer에 의해 POI Search와 POI Representation으로 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 트리 구조 형태로 POI Object Extractor를 통해 POI Object를 생성한다. 그림 6은 POI 핸들러의 세부 구조이다.

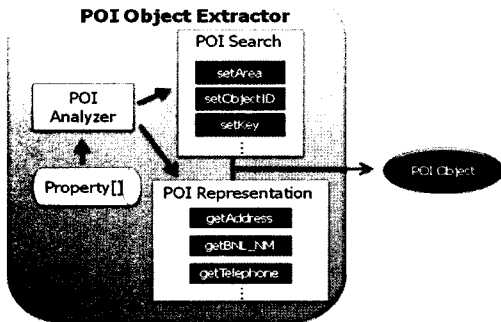


그림 6 POI 핸들러 구조

4.2 가시화 컴포넌트

가시화 컴포넌트는 Geodata 파서로부터 생성된 각각의 Map Object와 POI Object를 가지고 화면상에 가시화한다. 두 개의 컴포넌트로 구성되어 있고 지도와 POI를 가시화한다. 특히 POI 가시화는 단순한 POI 가시화와 POI의 편집을 통해 사용자 정의 POI를 지원한다.

4.2.1 지도 가시화 컴포넌트

지도 가시화 컴포넌트는 Mapdata 핸들러에 의해 생성된 Map Object를 Viewer에 가시화 하며 사용자에게 의해 발생하는 이벤트 처리를 담당한다. 그림 7은 Map 가시화 컴포넌트의 세부 구조이다.

지도 가시화 컴포넌트는 Map Object의 MapData를

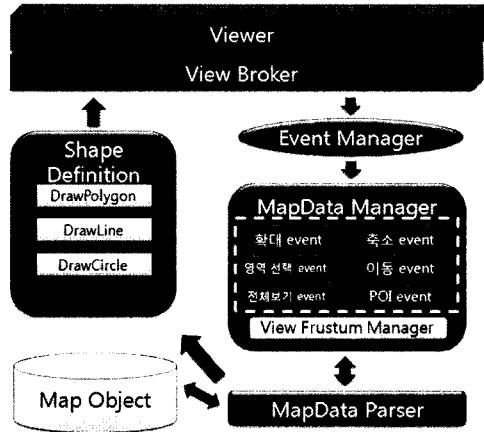


그림 7 Map 가시화 컴포넌트 구조

Geometry Type에 따른 Draw 메서드들이 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 이벤트를 판단하여 MapData Manager로 보내준다. MapData Manager는 사용자 이벤트를 처리하기 위해 지도의 재가시화가 필요하므로, View Frustum Manager를 통해 가시화 영역을 재설정한다. MapData 파서는 전달된 가시영역에 해당하는 MapData를 추출하며, Shape Definition에 의해 Viewer에 재가시화 한다.

4.2.2 POI 가시화 컴포넌트

POI 가시화 컴포넌트는 POI 핸들러에서 생성된 POI Object를 이용하여 사용자가 POI 검색을 요청할 경우, Viewer에 팝업창 형태로 가시화해주는 역할을 한다. 그림 8은 POI 가시화 컴포넌트의 세부 구조이다.

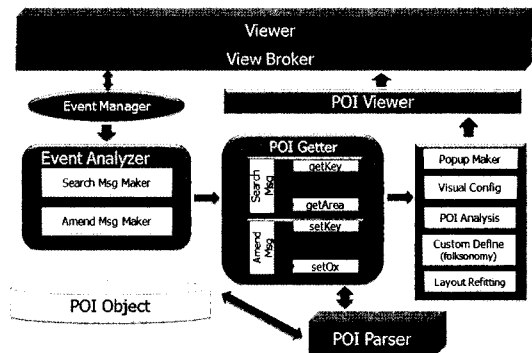


그림 8 POI 가시화 컴포넌트

Viewer에서 POI 검색을 요청할 경우, 공통 모듈인 Event Manager는 POI 관련 이벤트만을 분류해 POI 파서로 이벤트를 알린다. POI 관련 이벤트는 POI 정보 보기와 계층적 POI 검색을 제공한다. POI 관련 이벤트는 Search Eng.의 Event Analyzer에서 검색 요청된 건물의 ObjectID, Area 등을 메시지화시켜주는 Search Msg Maker에 의해 메시지를 생성한다. POI Manager는 검색 메시지가 넘어 왔을 경우 해당 메시지 값을 POI 파서로 보낸다. POI 파서는 POI Object에 접근해 메시지 값을 검색하여 요청된 식별키를 반환한다. 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 부분만을 추출된다. POI 정보는 팝업 형태로 가시화되는데, 이는 PopUp Maker를 통해 팝업창이 생성된다.

#### 4.2.3 POI Manager

POI Manager는 POI 가시화 컴포넌트 내부에 존재하며, 기존의 고정된 POI 항목에서 내용만을 편집했던 서비스에서 내용뿐만 아니라 항목까지도 수정하여 사용자로 하여금 서비스에 참여하여 서비스 제공에 직접적으로 관여하도록 지원한다. POI Manager는 사용자로부터 POI 검색이 요청되면 파서로부터 POI 정보들을 불러와 팝업형식의 User Interface로 가시화 한다. 또한 User Interface의 POI 정보의 편집에 따라 POI Object의 원본 POI 정보들을 수정하는 역할을 한다. 본 논문의 기반으로 사용되는 GML파일의 POI는 객체의 이름을 표현하는 BLG\_Name, 전화번호를 의미하는 TELEPHONE, 주소인 ADDRESS1+2, 객체의 홈페이지인 HOMPAGE, 기타 해당 개체의 특징을 나타내는 EX로 총 5개의 항목으로 이루어져 있다. 초기 POI 검색은 기존 5개의 POI 항목을 User Interface에 기본적으로 표현한다. POI 편집은 기존 5개 항목을 포함한 항목의 추가/삭제와 각 항목에 따른 내용 수정을 제공한다. User Define POI는 User Interface에서 넘어온 수정 값들에 대한 데이터를 저장하고, POI Editor에 의해서 각 항목 엘리먼트의 포인터에 접근하여 각 수정 항목들에 매칭하여 User Define POI의 정보로 수정한다. 일정 맵 안의 모든 객체들의 POI표현에 대해서는 사용자가 원하는 POI만 표현되도록 setVisualization에서 설정하도록 지원한다.

태깅에 대해서는 본 논문의 3절에서 기술한 개인화 태그정보의 PCode 태그정보들을 바탕으로 해당 POI의 분류를 설정한다. 태깅에 대해서는 GML 파일 내부의 POI를 구성하는 엘리먼트 중 <key> 엘리먼트에 설정

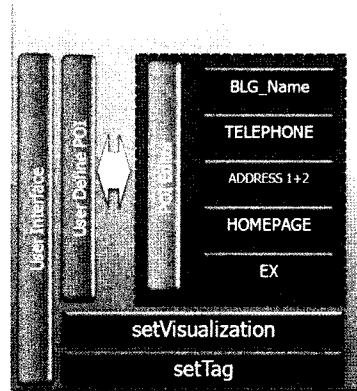


그림 9 POI Manager 세부 구조

되며 이를 통해서 태그 정보로 검색을 제공한다. 그림 9는 POI Manager의 세부 구조이다.

## 5. POIM 구현

### 5.1 POIM 가시화

그림 10은 전라북도 전주시 한옥마을의 전통문화센터를 본 논문에서 구현한 POIM을 통해 POI 가시화와 GML 파일 내의 POI 원시 데이터이다.

POIM은 지도 가시화창을 중심으로 오른쪽엔 POI 검색과 사용자정의 분류체계를 적용할 수 있는 레이어 선택창(①)과 하단에는 지도의 기본 이벤트 아이콘(③)들이 있으며 POI 검색이나 객체 선택을 통해 POI를 가시화하는 POI Viewer(②)로 구성되어 있다.

레이어 선택창은 POI의 계층구조를 사용자가 직접 정의하고, 정의된 계층구조를 바탕으로 POI 검색을 한다. 기본 이벤트 아이콘은 지도의 확대/축소/영역선택/이동/전체보기/POI보기로 이루어져 있다. POI Viewer는 POI를 가시화 할 뿐만 아니라 POI의 항목 등에 대한 편집이 가능하며, 상호명 디스플레이를 통해 상호명을 지도에 표현 할 것인지에 대한 유/무 체크가 가능하다. 또한 레이어 선택창의 사용자 정의 계층구조를 바탕으로 POI태깅을 통해 계층구조를 적용한다.

### 5.2 POIM의 개인화 태그정보 생성

기존 분류체계에서 팝업메뉴의 ADD를 통해 임시적으로 NewChild라는 이름의 상위계층과 NewLeaf라는 이름의 하위계층으로 구성된 새로운 분류가 추가된다. 이 분류는 사용자가 Edit 메뉴를 통해서 새로운 이름으로 정의할 수 있으며, POI와 동기화를 위한 Key값의 정의 또한 가능하다. 항상 상위계층과 하위계층으로 생성되는 것은 태깅의 단점이라 할 수 있는 데이터간에 계층이 없는 점에 대한 보완을 위함이다. 그림 11은 역사라는 상위계층과 유적지라는 하위계층으로 구성되는

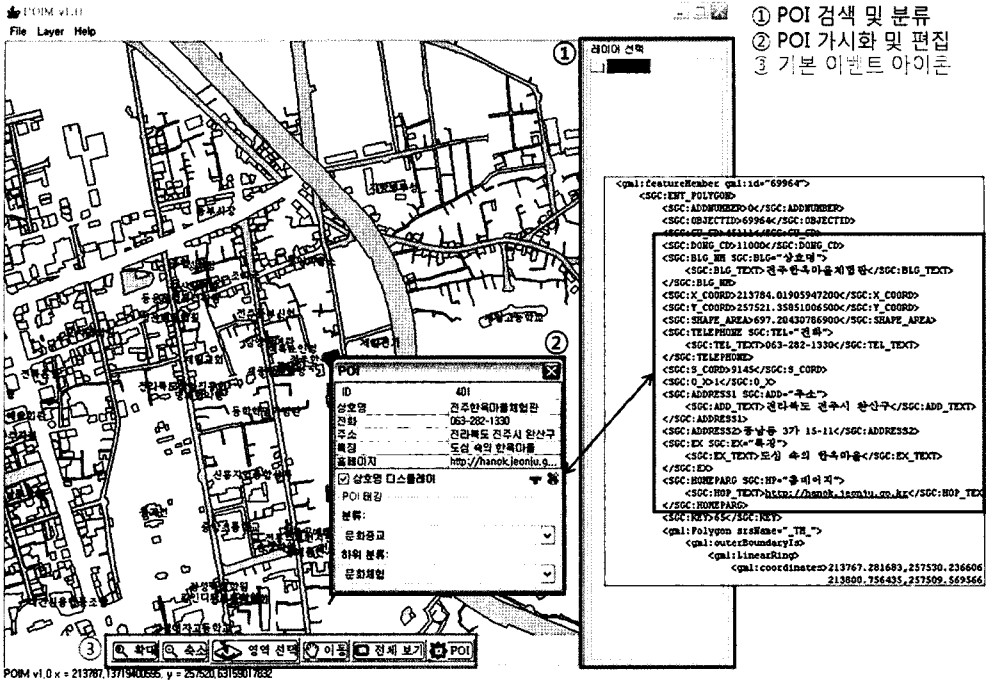


그림 10 POIM 가시화 사례

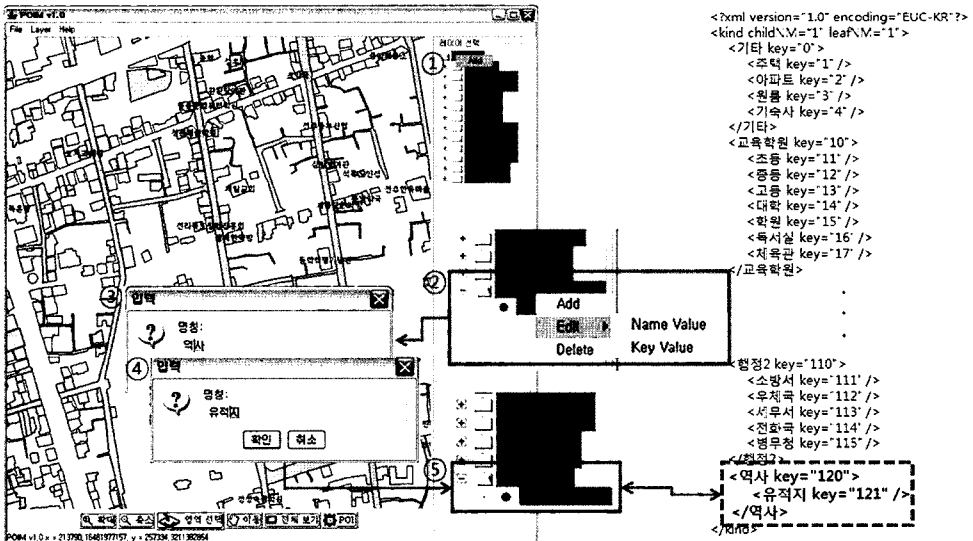


그림 11 개인화 태그정보 생성

새로운 분류를 추가하는 과정과 생성 후 본 논문의 분류체계 정보를 담고 있는 PCode.xml파일에 새로운 분류가 추가된 것을 보인다.

### 5.3 POIM의 POI 편집 및 태깅

아직 등록되지 않은 객체에 대해 POI를 가시화하면 빈 테이블만 보이며, 빈 테이블에 항목과 항목에 따르는

데이터를 삽입한다. 항목은 기본으로 5개 리스트가 제공되나 사용자가 원하는 만큼의 항목 리스트를 추가 가능하여 객체마다 POI 항목을 다르게 하여 좀 더 완전한 사용자 맞춤형 POI 서비스를 지원한다. 또 자주 확인하는 POI에 대해서는 지도 가시화와 함께 상호명을 같이 디스플레이 할 수 있도록 디스플레이 체크박스가 존재한다.

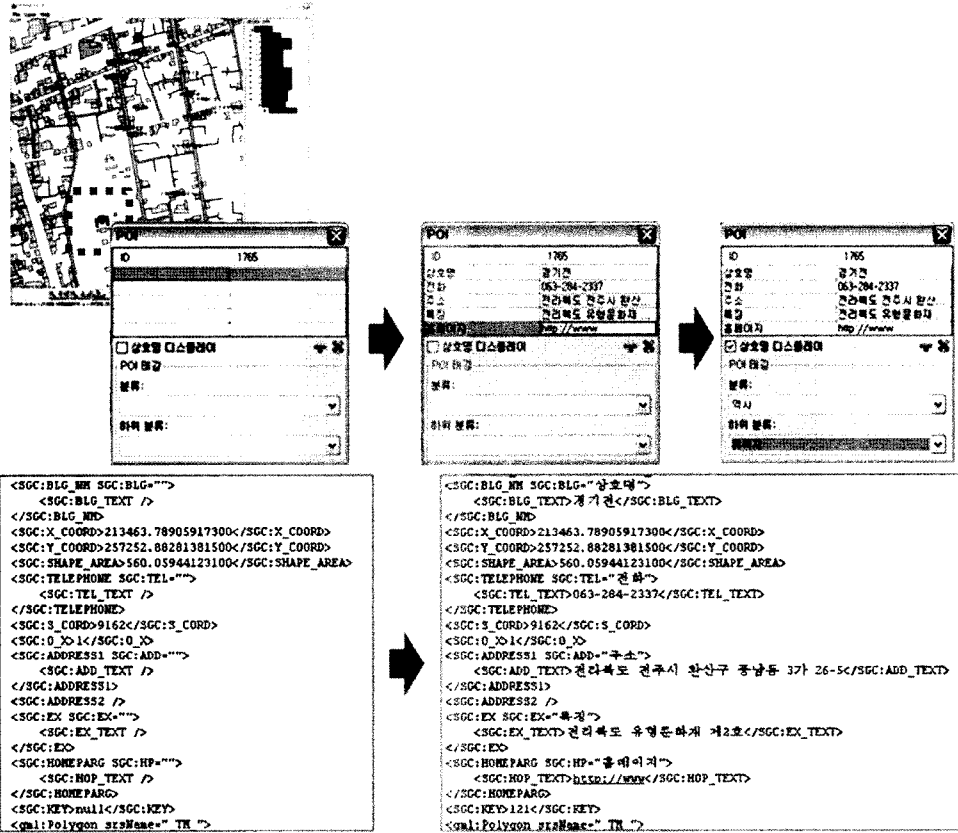


그림 12 POI 편집 및 태깅 과정

POI 태깅은 사용자 정의 계층구조에 의한 계층을 적용할 수 있으며 적용과 함께 레이어 선택창과 동기화되어 레이어 선택창에서의 검색에 반응하여 해당 객체에 주황색으로 가시화된다. 그림 12는 전라북도 전주시 한옥마을에 위치하는 “경기전”에 대한 POI 데이터 삽입과 5.2절에서 새롭게 정의한 역사 계층의 하위계층인 유적지를 태깅하는 과정이다.

### 6. 결론 및 향후 연구

본 논문에서는 정보의 홍수속에 수시로 생겨나는 다양한 종류의 웹 자원들을 분류 및 관리하기 위해서는 기존 카테고리식 분류체계의 한계점을 인식하고 이를 극복하는 새로운 분류체계의 대안으로 나타난 태깅 기술을 POI 서비스에 적용하여 POI의 폭소노미를 실현하는 POIM을 구현하였다. POI 서비스에 태깅 기술의 적용은 기존에 웹 자원들을 중심으로 태깅 기술이 활용되었던 점을 미루어 봤을 때, 데이터 변동이 심하며, 그 데이터들간의 분류체계가 필요로 하는 시스템들에 대해

서도 태깅의 장점을 활용할 수 있음을 보였다. 또한 태그 정보들을 XML포맷으로 패키징하여 이를 다른 POIM을 사용하는 사용자들로 하여금 서로 공유하도록 하여 집단지성의 특성을 갖추는 폭소노미를 실현하였다. POI 서비스에 있어서도 서비스의 기반이라 할 수 있는 지도 데이터에 대해서도 GML이라는 표준을 수용하였으며, 사용자가 직접 POI 정보를 수정, 삭제 등의 편집이 가능하도록 하여 사용자가 정보의 생산자, 제공자가 될 수 있는 사용자 중심의 POI 서비스를 지원한다.

향후 연구로는 폭소노미의 단점으로 대두되는 동의어나 유의어 관리에 대해 좀 더 효율적인 대안을 연구하여 좀 더 폭넓고 완전한 분류체계로 거듭나도록 한다. POIM은 현재의 단일 어플리케이션이란 단점을 극복하기 위해 서버-클라이언트 형식으로 재구성하여 지도 및 POI 서비스들에 대한 온라인화를 실현함으로써 단순히 POI 서비스 뿐만 아니라 위치 추적이나 길 찾기 등의 위치 기반 서비스 등의 폭넓은 서비스를 제공함으로써 통합 GIS&LBS 솔루션으로 재 구현되도록 한다.



## 참고 문헌

- [1] Emanuele Quintarelli, "Folksonomies : Power to the people," <http://www.dimat.unipv.it/biblio/isko/doc/folksonomies.htm>
- [2] Scoot A. Golder and Bernardo A. Huberman, "Using Patterns of Collaborative Tagging System," *Journal of Information Science*, 32(2), pp.198-208, 2006.
- [3] Laura Gordon-Murnane, "Social Bookmarking, Folksonomies, and Web 2.0 Tools," *Searcher*, Vol.14, Iss. 6, pp. 26-39, 2006.
- [4] Owen Kaser and Daniel Lemire, "Tag-Cloud Drawing: Algorithms for Cloud Visualization," *Tagging and Metadata for Social Information Organization, WWW 2007*, 2007.
- [5] OpenGIS Consortium, Inc., "Geography Markup Language(GML) Implementation Specification," <http://www.opengeospatial.org/docs/02-023r4.pdf>
- [6] Eun-Ha Song, Sung-Kook Han, Laurence T. Yang, Minyi Guo, Young-Sik Jeong, "A GML based Mobile Device Trace Monitoring System," *EUC06, LNCS 4097*, pp. 234-243, 2006.
- [7] 김대식, 김형진, 손봉수, 유완, "생활지리정보 검색 및 안내를 위한 POI 구축 및 활용", *한국콘텐츠학회 추계 종합학술대회 논문집 Vol.1 No.2*, pp. 423-430, 2003.
- [8] eSpatial Inc. iSMART Explorer4.4, <http://www.espatial.com/productsismartexplorerversion44.htm>
- [9] Saft Software Inc. FME Universal Viewer 2003, <http://www.saft.com/products/fme/index.php>
- [10] TatukGIS Inc. TatukGIS Viewer 1.4, <http://www.tatukgis.com/products/Viewer/viewer.aspx>
- [11] Flickr, <http://www.flickr.com/>
- [12] del.icio.us, <http://del.icio.us/>
- [13] Technorati, <http://technorati.com/>



김 흠

2004년 중국길림농업대학교 컴퓨터과학 및 기술 학사. 2008년 한국원광대학교 컴퓨터공학과 석사. 2008년~현재 다우(대련)기술 한국사업부. 관심분야는 LBS, GIS



박 용 진

2006년 원광대학교 전기전자 및 정보공학부 학사. 2008년 원광대학교 컴퓨터공학 석사. 2008년~현재 공간정보기술(주) 제품개발팀. 관심분야는 LBS, GIS



송 은 하

1997년 원광대학교 통계학과 졸업(학사) 2000년 원광대학교 대학원 컴퓨터공학과 졸업(석사). 2006년 원광대학교 대학원 컴퓨터공학과 졸업(박사). 2007년~현재 원광대학교 전기전자 및 정보공학부 전임강사. 관심분야는 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템



안 시 훈

1997년 인제대학교 의용공학과 학사. 1999년 부산대학교 의공학 협동과정 석사 2001년 부산대학교 의공학 협동과정 박사 수료. 2000년~2006년 (주)비트컴퓨터 기술연구소 원격진료팀장. 2005년~2006년 보건복지부 의료선진화위원회 원격진료워킹그룹 위원. 2007년~현재 (주)비트컴퓨터 u-Healthcare팀장. 2007년~현재 대한의사협회 사이버학술위원. 2008년~현재 산업자원부 차세대의료기기사업 연구기획위원. 관심분야는 u-Healthcare 서비스 및 시스템, Telemedicine



남 승 훈

1998년 인제대학교 의용공학과 학사. 2000년 인제대학교 의용공학과 석사. 2001년 한국전자통신연구원 연구원. 2007년~현재 (주)비트컴퓨터 선임연구원. 관심분야는 U-Healthcare, Wearable computing, Bio-medical Engineering



정 영 식

1987년 고려대학교 수학과 졸업(학사) 1989년 고려대학교 대학원 전산학과 졸업(석사). 1993년 고려대학교 대학원 전산학과 졸업(박사). 2004년 웨인 주립대학교 객원교수. 1993~현재 원광대학교 전기전자 및 정보공학부 교수. 관심분야는 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템