

Drupal 기반 맵 응용 생성기 (MapAppGen) : 맛집탐방 응용 생성 사례

음 두 현[†]

요 약

웹과 스마트폰의 사용이 일상화 되면서 맵 응용에 대한 수요가 웹 및 모바일 환경에서 급증하고 있다. 웹 기반 맵 응용들은 대부분 ArcGIS와 같은 값비싼 개발 환경이나 MapServer와 같은 공개 개발 환경에서 작성되고 있고 모바일 맵 응용들은 Google Maps나 Yahoo Maps와 같은 API 수준에서 작성되고 있다. 그러나 이러한 개발 환경들은 자동화 지원 정도가 미약하여 맵 응용 개발의 많은 부분이 아직 수작업에 의존하고 있다. 우리가 설계하고 구현한 MapAppGen은 신속한 웹 기반 맵 응용의 작성을 지원하기 위해 Drupal CMS(Contents Management System)에 적용 가능한 맵 관련 모듈들을 생성해 줌으로써 맵 응용에 대한 생산성을 향상시켜 준다. MapAppGen으로 생성한 응용은 Drupal이 지원하는 기존의 GMap이나 NodeMap이 생성한 응용에 비해 관심의 대상인 지형/지물과 함께 이 지형/지물과 연관된 모든 정보를 검색해 표시한다. MapAppGen은 Google Maps API를 사용하며 Drupal은 콘텐츠의 생성, 구성, 관리를 지원하는 모듈 기반 시스템이다. 향후, MapAppGen을 확장하여 모바일 맵 응용의 자동 생성도 지원할 예정이다.

키워드 : 맵 응용, 맵 응용의 생산성, 맵 응용 생성기, Drupal 응용

Drupal-based Map Application Generator(MapAppGen): an Application Generation Example for Famous Restaurants

Doohun Eum[†]

ABSTRACT

The demand for map applications in both Web and mobile environments has been rapidly increased with the population of Web and smart phone usage. Web-based map applications are mostly developed on such environments as ArcGIS and MapServer and mobile map applications are developed on such API levels as Google Maps and Yahoo Maps. But many parts of map applications are still constructed by coding because these environments don't support high level of automation. Our MapAppGen that we have designed and implemented enhances the Web-based map application productivity by generating the map related modules that can be applied to the Drupal that is one of popular content management systems(CMS's). Comparing the applications that are constructed by the Drupal-supported GMap or NodeMap, the applications that are constructed by MapAppGen provide information on not only the interested geographical feature but also its related geographical features. MapAppGen uses Google Maps API and Drupal is a module-based system that supports the creation, composition and management of contents. We are now working on automatic generation of mobile map applications with MapAppGen.

Keywords : Map Applications, Map Application Productivity, Map Application Generator, Drupal Applications

1. 서 론

맵 서비스를 이용하는 산업의 성장 잠재력으로 인하여 맵 시장은 전 세계적으로 연평균 20% 이상 급성장하고 있으며 2012년 국내 공간정보산업의 규모도 11조원까지 확대될 것

으로 예측된다[1].

맵 응용은 단순한 맵 검색을 뛰어 넘어 사용자의 특정 기호에 맞게 발전하고 있으며 Google을 시작으로 포털 사이트들이 Maps API를 공개하면서 맵 응용에 대한 관심과 수요가 급증하고 있다[2]. 그러나 웹 기반 맵 응용의 작성은 ArcGIS[3]와 같은 값비싼 개발 환경이나 MapServer[4]나 GeoServer[16]와 같은 공개 개발 환경을 사용한다고 해도 많은 부분에 아직 수작업이 필요하며 모바일 맵 응용들도 Google Maps, Yahoo Maps, UPMAP[15] 등과 같은 API 수준에서 작성되고 있기 때문에 맵 응용에 대한 생산성이 떨

※ 본 논문은 2011년 덕성여대 교내 연구비 지원으로 연구되었음.

† 정 회 원 : 덕성여자대학교 디지털미디어학과 교수

논문접수: 2012년 2월 11일

수정일: 1차 2012년 3월 14일, 2차 2012년 3월 30일, 3차 2012년 5월 25일

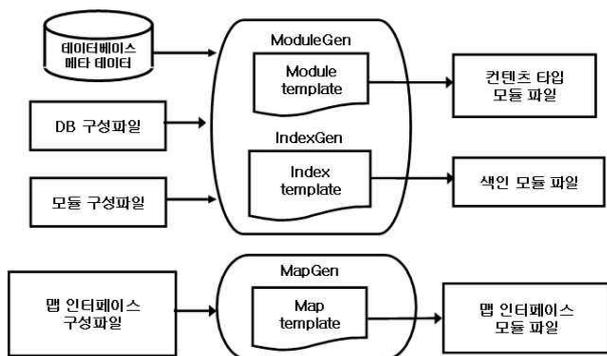
심사완료: 2012년 5월 30일

어질 뿐만 아니라 유지·보수에도 많은 시간과 노력이 요구된다[5, 6, 7, 8, 18].

CMS(Contents Management System) 중의 하나인 Drupal은 소셜 네트워킹 사이트의 신속한 개발을 위해 콘텐츠를 생성하고 구성하며 관리해주는 모듈 기반 시스템으로서 기본 모듈들 외에 추가적인 기능 및 콘텐츠를 제공하는 모듈을 생성해 Drupal에 적용시킴으로써 그 기능을 확장시킬 수 있다.

우리가 설계하고 구현한 MapAppGen은 웹 기반 맵 응용을 위해 맵 관련 Drupal 모듈들을 자동 생성해 줌으로써 맵 응용에 대한 생산성을 향상시킨다. 맵 응용을 Drupal과 같은 콘텐츠 관리 시스템을 기반으로 하면 로그인, 접근 제어, 기본 콘텐츠 관리 등과 같은 기본적인 기능들을 제공하는 코어 모듈들과 서드 파티(third party)들이 제공하는 다양한 기능의 서드 파티 모듈들을 Drupal에 적용해 활용함으로써 맵 응용의 생산성을 개선할 수 있다. 또한, 추가적인 패키지의 설치 없이 표준 웹 브라우저만으로 맵 인터페이스를 사용하는 응용을 작성할 수 있다.

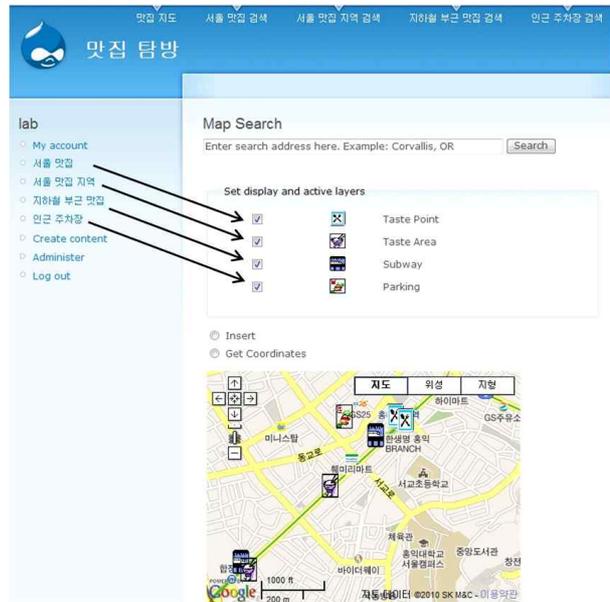
MapAppGen은 (그림 1)과 같이 ModuleGen, IndexGen, MapGen 도구들로 구성된다. MapAppGen은 개발자가 정의하는 선언적 구성과일에 따라 Drupal 모듈들을 생성한다. ModuleGen 도구는 사용자 정의의 지형/지물 콘텐츠 타입 모듈을 생성한다. IndexGen은 콘텐츠 타입 모듈로 생성된 지형/지물의 신속한 유형별 검색을 지원하는 색인 모듈을 생성한다. MapGen은 생성된 지형/지물들을 맵 상에서 마커를 통해 한눈에 보고 검색할 수 있게 하는 맵 인터페이스 모듈을 생성한다.



(그림 1) MapAppGen의 구조

본 논문에서는 MapAppGen의 기능을 설명하기 위해 MapAppGen을 이용해 작성한 (그림 2)의 맛집탐방 응용을 예로 든다. (그림 2)는 서울에 위치한 맛집들의 정보를 제공하는 맛집탐방 응용의 메인 화면을 보인 것이다. 사용자가 이 응용을 이용해 맛집을 맵상에서 유형별로 검색할 수 있고 등록할 수도 있다.

MapAppGen으로 생성된 모듈과 Gmap[9]이나 NodeMap[10]과 같이 Drupal의 서드 파티들이 제공하는 맵 모듈을 비교해 보면, Gmap과 NodeMap은 Drupal의 기본 콘텐츠



(그림 2) MapAppGen으로 생성된 맛집탐방 응용

타입 모듈(Story, Page 등)에 의존하기 때문에 사용자 정의의 지형/지물 타입들을 추가할 수 없다. Gmap이나 NodeMap은 호텔, 박물관 등과 같은 유형별 지형/지물 콘텐츠의 검색 기능도 지원하지 못한다. 또한, Gmap과 NodeMap 모듈은 콘텐츠 정보 페이지에 해당 콘텐츠와 연관된 콘텐츠들을 검색해 표시하지 못한다.

2절에서 MapAppGen으로 생성한 맛집탐방 응용의 기능을 살펴본다. 3절에서는 MapAppGen의 개요를 설명하고 4절에서는 MapAppGen의 도구들 중 ModuleGen 도구와 IndexGen 도구의 구현을 설명한다. 5절에서는 MapAppGen으로 생성된 맵 응용과 Gmap이나 NodeMap을 이용해 생성된 맵 응용을 비교한다. 마지막 절에서는 결론과 향후 연구 방향을 정리한다.

2. MapAppGen으로 생성한 MapAppGen 응용

MapAppGen으로 생성된 맛집탐방 응용(그림 2)의 메인 화면은 MapGen이 생성한 맵 인터페이스 모듈에 의해 제공되며 맛집 정보는 콘텐츠 타입에 따라 point 형태의 Taste Point 마커 (X), polygon 형태의 Taste Area 마커 (A), 지하철 부근의 맛집을 나타내는 Subway 마커 (S), 인근 주차장을 나타내는 Parking 마커 (P)들로 표시된다. 마커의 말풍선에는 각 콘텐츠 제목이 나타나며 이 제목을 클릭하면 해당 콘텐츠의 정보 페이지로 이동된다.

(그림 2)의 상단 메뉴들은 IndexGen으로 생성된 색인 모듈이 제공하는 (그림 3) 상단의 검색 폼들과 연결된다. 검색 폼에서 검색된 결과중 하나를 선택하면 (그림 3) 하단의 해당 콘텐츠 정보 페이지로 링크된다.

4개의 콘텐츠 타입 모듈 중에서 Taste Area 모듈의 콘텐츠는 자신과 연관된 Taste Area 타입의 콘텐츠들과



(그림 3) IndexGen으로 생성된 콘텐츠 검색 폼과 콘텐츠 정보 페이지

Subway 타입의 콘텐츠들을 함께 검색하여 표시하며 Subway 모듈의 콘텐츠는 자신과 연관된 Taste Point 모듈의 콘텐츠들을 검색하여 표시한다. 예를 들어, (그림 3)의 [서울 맛집 지역] 검색 폼에서 '홍대'로 검색한 결과 중 '마포구 서교동'을 선택하면 하단의 '마포구 서교동'에 대한 정보 페이지가 표시된다. 이 정보 페이지에서 Taste Area 항목은 '마포구 서교동' 콘텐츠와 연관된 '합정역 앞'을 함께 검색해 표시한다. Subway 항목은 '마포구 서교동' 콘텐츠와 연관된 [지하철 부근 맛집] 타입의 '합정역'과 '홍대입구역'을 검색해 표시한다. 이 Subway 목록에서 '홍대입구역'을 선택하면 '홍대입구역'에 대한 정보 페이지로 링크된다. 이 콘텐츠 페이지에는 Taste Point 항목이 표시되며 '홍대입구역'과 연결된 '아웃백 스테이크 하우스'와 '할리스 카페'를 검색해 표시한다. 여기서 '아웃백 스테이크 하우스'를 선택하면 우측의 '아웃백 스테이크 하우스'의 정보 페이지로 링크된다.

(그림 3)처럼 각각의 콘텐츠에 대한 정보 페이지를 생성하고 연결하기 위해 사용자는 (그림 2)의 좌측 메뉴 항목과 연결된 삽입/편집 폼(ModuleGen 도구가 생성)을 이용한다. (그림 4)는 [서울 맛집 지역]과 연결된 삽입/편집 폼의 일부를 보인 것이다.

Parent Taste Area Id 드롭다운 리스트에는 이미 생성된 [서울 맛집 지역] 타입의 콘텐츠들이 표시된다. 드롭다운 리스트에서 연결하고 싶은 항목을 선택하면 콘텐츠 생성 시 (그림 3)과 같이 자동으로 연결되어 표시된다. 하단의 맵 인

터페이스는 사용자가 point 또는 polygon으로 위치 지정할 수 있게 한다.

(그림 2)의 상단 메뉴 중 [맛집 지도]는 MapGen으로 생성된 맵 인터페이스 모듈이 제공하는 Map Search 폼(그림 2의 메인 화면)과 연결된다. Search 필드 아래 위치한 Set display and active layers 상자 안의 체크 박스를 선택하면 해당 지형/지물들이 유형별로 맵 상에 마커들로 표시된다. 표시된 마커를 마우스로 클릭하면 해당 콘텐츠의 정보를 볼 수 있는 정보 페이지로 이동된다. 또한, 사용자는 Insert 버튼을 통해 지형/지물의 정보를 맵 상에 입력할 수 있고 Get Coordinates 버튼으로는 사용자가 원하는 지점의 좌표를 얻을 수도 있다. 맵은 Google Maps API[11, 12]를 사용하기 때문에 기본 지도 타입, 위성 지도 타입, 지형 지도 타입의 3가지 형태를 지원한다.

3. MapAppGen의 개요

본 절에서는 MapAppGen으로 생성한 Drupal 기반의 맛집탐방 응용(그림 2)의 동작 원리와 MapAppGen의 모듈 생성 과정을 살펴본다.

생성된 맛집탐방 응용의 동작 원리는 (그림 5)와 같다. 사용자가 웹 브라우저를 통해 특정 지형/지물에 대한 맛집 정보 페이지를 요청하면 웹 서버 내의 Drupal은 해당 콘텐츠 타입 모듈 내의 함수들을 기동시킨다. 이 함수들은 SQL 질



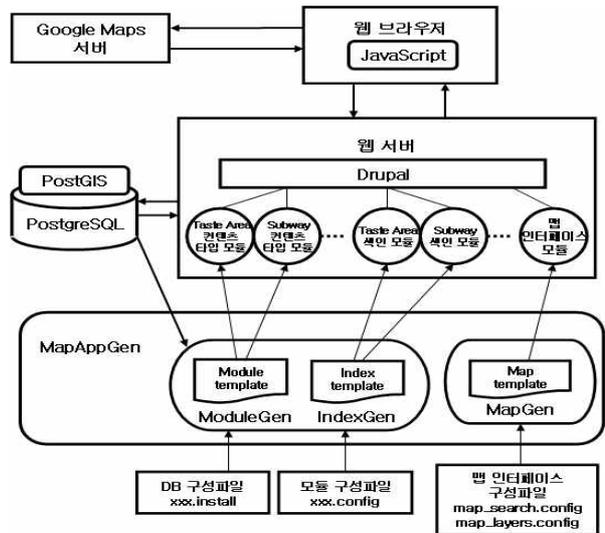
(그림 4) ModuleGen이 생성한 [서울 맛집 지역] 타입의 삽입/편집 폼

의를 통해 PostgreSQL[13] 데이터베이스의 테이블들로부터 필요한 데이터를 검색한다. 공간 연산이 필요한 SQL 질의는 PostGIS[14] 서브 시스템에 의해 처리된다. 검색된 데이터는 콘텐츠 타입 모듈에 의해 폼을 포함하는 HTML 페이지 형태의 맛집 정보 페이지로 구성된 후, 웹 서버에 의해 웹 브라우저에 전달된다.

맛집 정보 페이지는 맵 인터페이스를 포함할 수 있다. 맵 인터페이스가 필요한 연산은 Google Maps API를 사용하는 JavaScript 함수들에 의해 처리된다.

이 함수들이 Google Maps 서버에 맵 이미지를 요청해 표시하고 지형/지물 데이터는 위에 기술한 과정을 거쳐 PostgreSQL 데이터베이스로부터 검색해 맵 상에 마커로 표시된다. MapAppGen이 제공하는 JavaScript 함수들은 생성될 모든 맵 응용에 공통인 맵 인터페이스 기능을 지원한다.

Drupal에 적용된 콘텐츠 타입 모듈, 색인 모듈, 맵 인터페이스 모듈들은 MapAppGen에 의해 자동 생성된다. ModuleGen은 맛집 정보를 포함하는 지형/지물 콘텐츠들을 생성, 저장, 삭제하는 Taste Area와 Subway와 같은 사용자 정의의 콘텐츠 타입 모듈을 생성한다. 모듈 구성파일(xxx.config)은 Google Maps API를 사용하기 위해 필요한 Google Maps Key, 드롭다운 리스트에 포함될 속성, 맛집 위치 정보의 형태, 연결된 콘텐츠 타입 모듈의 테이블 정보 등, 생성하고자 하는 지형/지물 콘텐츠 타입 모듈에 대한 정보를 제공하는 php 파일이다. 이 파일은 php 문법을 이용한 선언문들을 포함하며 프로그램을 작성하기 위한 것은 아니다. DB 구성파일(xxx.install)은 생성될 지형/지물 콘텐츠 타입 모듈과 연동될 테이블들의 스키마 정보를 제공하는 php



(그림 5) 맛집탐방의 동작과 MapAppGen의 모듈 생성

파일이다. 이 테이블들은 사용자 정의의 지형/지물 콘텐츠 타입에 대한 속성들을 저장한다. 콘텐츠 타입 모듈 템플릿은 생성될 모든 콘텐츠 타입 모듈 파일들의 기본 틀을 제공하는 템플릿 파일이다.

ModuleGen은 모듈 구성파일, DB 구성파일 그리고 Drupal이 생성한 데이터베이스의 메타 정보를 입력받아 그 내용을 내장된 콘텐츠 타입 모듈 템플릿 파일에 적용해 지형/지물 콘텐츠 타입 모듈 파일을 생성한다. 이 모듈은 지정된 테이블들을 생성하고 맛집 정보 페이지와 그림 4와 같은 삽입/편집 폼을 제공한다.

(그림 6)은 Taste Area 콘텐츠 타입 모듈을 생성하기 위한 모듈 구성파일을 보인 것이다. \$map_interface는 생성될 콘텐츠 타입 모듈이 제공하는 맛집 지역 페이지와 삽입/편집 폼에 맵 인터페이스를 포함할지의 여부를 지정한다. \$geom_type은 맛집의 위치 정보 형태를 명시한다. Taste Area 모듈은 맛집들이 위치한 지역을 저장하는 모듈이므로 polygon으로 설정한다. \$google_map_key는 Google로부터 배정받은 Google Maps API의 키를 저장한다. \$options_parent_taste_area_id와 \$options_d_county_id에는 삽입/편집 폼의 드롭다운 리스트에 포함될 항목들을 지정한다.

\$linked_tables는 연결하려는 콘텐츠 타입 테이블에 대한 정보를 배열로 가진다. table_name은 연결하려는 테이블의 이름을 선언하는 변수이다. 자기 자신인 taste_area 테이블과 subway 테이블을 연결한다. type은 테이블간의 관계를 설정하는 변수인데 to_many는 1:N의 관계를 지정한다. parent_column은 연결하려는 테이블의 외래키로서 taste_area 테이블의 parent_taste_area_id 컬럼과 subway 테이블의 taste_area_id 컬럼을 통해 연결된다. 이렇게 명시하면 (그림 3)의 [서울 맛집 지역]의 정보 페이지에 표시된 Taste Area 항목과 Subway 항목과 같이 현재 콘텐츠와 연관된 항목들이 SQL 질의를 통해 검색되어 표시된다.

(그림 5)의 IndexGen은 콘텐츠 타입 모듈들을 통해 등록

```

<?php
$map_interface = TRUE;
$geom_type = "polygon";
$google_map_key =
"ABQIAAAAWmi9n0Pep_HNHokjC0ygChQbwIHzPvqgLr2VTI
O4ademzrStaxQNhhLd3GoaViYUdrfoXDOLCG-B5g";
( ... 중략 ... )
$options_parent_taste_area_id = $arResult;
$options_d_county_id = array(
1 =>'가로수길', '강남역', '논현동', '대치동', '도곡동', ...
);
$linked_tables = array(
array(
'table_name' => 'taste_area',
'type' => 'to_many',
'parent_column' => 'parent_taste_area_id'
),
array(
'table_name' => 'subway',
'type' => 'to_many',
'parent_column' => 'taste_area_id'
),
);
?>

```

(그림 6) Taste Area 모듈 구성파일

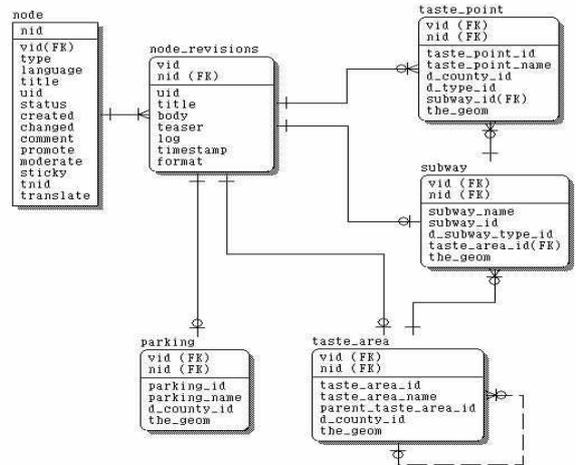
된 콘텐츠들에 대해 색인 정보를 추가하여 신속한 검색을 지원하는 색인 모듈을 생성한다. 이 모듈은 (그림 3) 상단의 검색 폼들을 제공한다. MapGen은 (그림 2)와 같은 맵 응용의 메인 화면을 제공하는 맵 인터페이스 모듈을 생성한다. Taste Point, Taste Area, Subway, Parking 모듈들을 통해 생성된 콘텐츠들을 모두 맵 상에 레이어별로 표시하고 검색할 수 있는 기능을 제공한다. 맵 인터페이스 구성파일은 map_search.config 파일과 map_layers.config 파일로 구성되는데 map_layers.config 파일에는 맵 인터페이스 모듈과 연계될 콘텐츠 타입 모듈들을 레이어별로 정의한다. map_search.config 파일에는 맵 인터페이스 모듈을 생성하기 위해 필요한 정보인 Google Maps Key와 드롭다운 리스트에 포함될 속성들을 정의한다.

4. MapAppGen의 구현

MapAppGen은 Drupal 6.16, PHP 5.2.6, PostgreSQL 8.3.9, PostGIS 1.3.3-3, Apache 2.2.11, Ajax 등의 공개 소프트웨어들을 사용해 구현되었다. 본 절에서는 ModuleGen이 사용자 정의 콘텐츠 타입을 지원하기 위해 Drupal의 기본 node 테이블을 확장하는 방법을 설명하고 이어서 ModuleGen의 콘텐츠 타입 생성 알고리즘을 살펴본다.

4.1 Drupal 기본 node 테이블의 확장

Drupal은 콘텐츠들을 노드들로 구성한다. 노드는 페이지,



(그림 7) node 테이블, node_revisions 테이블, 사용자 정의 콘텐츠 테이블 간의 관계

스토리, 블로그 항목 등과 같은 콘텐츠 타입의 개체를 나타낸다. Drupal을 설치하면 응용을 위한 기본 소프트웨어 패키지인 코어 모듈들이 포함된다. Drupal 코어는 사용자 관리, 기본 콘텐츠 관리, 테마 등과 같은 기능들을 제공한다.

노드는 기본적으로 타이틀과 바디의 속성을 가진다. ModuleGen이 생성하는 사용자 정의 콘텐츠 타입 모듈은 콘텐츠 타입의 속성을 지원하기 위해 기본적인 Drupal 노드 테이블의 구조를 (그림 7)과 같이 확장한다.

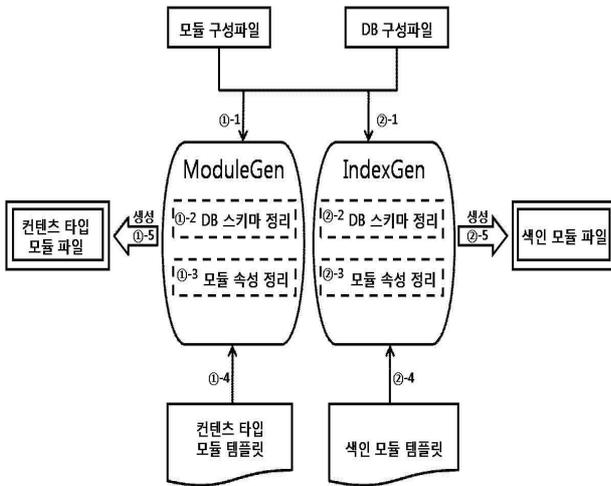
(그림 7)은 ModuleGen으로 생성된 콘텐츠 타입 모듈이 맛집탐방 응용의 Taste Point와 Taste Area, Subway, Parking 콘텐츠 타입의 속성들을 지원하기 위해 개발자가 제공하는 DB 구성파일에 따라 생성하는 taste_point, taste_area, subway, parking 테이블들과 Drupal 노드의 속성을 저장하는 node 테이블의 관계를 보인 것이다.

node 테이블은 노드 식별자(nid)와 함께 노드의 기본 속성들을 포함한다. taste_point와 taste_area, subway, parking 테이블은 node_revisions 테이블을 통해 Drupal의 node 테이블과 연결된다. 한 개의 노드는 여러 개의 확장(revision)을 갖는다. 각 확장에 대한 정보는 확장 식별자(vid)와 함께 node_revisions 테이블에 저장된다. 각 사용자 정의 테이블은 node_revisions 테이블과 vid를 통해 연결된다.

사용자가 Taste Point 또는 Taste Area 콘텐츠 타입 모듈이 제공하는 삽입/편집 폼을 통해 지형/지물을 생성하면 그 속성들이 콘텐츠 타입 모듈의 SQL 질의를 처리하는 함수들에 의해 taste_point와 taste_area 테이블에 반영된다. 사용자가 요구한 지형/지물에 대한 맛집 정보 페이지는 모듈 내 함수들의 SQL 질의를 통해 이 테이블들로부터 필요한 데이터가 검색되어 HTML 페이지 형태로 구성된 후, 웹 브라우저에 전달된다.

4.2 콘텐츠 타입 모듈 생성 알고리즘

(그림 8)은 ModuleGen의 콘텐츠 타입 모듈 생성 알고리즘과 IndexGen의 색인 모듈 생성 알고리즘을 도식화 한 것



(그림 8) 컨텐츠 타입 모듈과 색인 모듈의 생성 과정 (알고리즘의 도식화)

이다. 생성 과정은 크게 다섯 단계로 이루어진다. 단계 ①-1은 모듈 구성파일과 DB 구성파일의 디렉터리 상대경로를 파악해 저장하는 단계이다. ①-2의 DB 스키마 정리 단계는 DB 구성파일을 참조하여 테이블(그림 7)의 스키마를 속성 타입별로 정리하는 작업을 진행한다. 이 단계의 데이터는 Drupal에 생성된 모듈을 설치할 때 테이블 생성시 참조된다. 단계 ①-3의 모듈 속성 정리는 모듈 구성파일로부터 삽입/편집 폼에 제공될 드롭다운리스트의 항목들을 추출하는 단계이다. 단계 ①-4는 컨텐츠 타입 모듈 템플릿에 ①-1~①-3 단계에서 추출되고 정리된 데이터를 반영하는 단계이다. 또한, 폼을 정의하며 입력, 삭제, 수정, 보기에 대한 권한을 부여하고 입력, 삭제, 수정, 보기의 이벤트가 발생될 경우, 테이블에 저장된 값을 수정하기 위한 SQL 질의문을 생성하는 함수를 생성한다. 단계 ①-5는 Drupal 모듈 파일을 생성하는 단계로서 ①-4 단계에서 변경된 템플릿 파일을 출력한다. IndexGen도 이와 유사한 과정을 거친다(②-1,2,3,4,5).

(그림 9)는 Taste Area 컨텐츠 타입에 대해 (그림 6)의 모듈 구성파일을 (그림 8)의 다섯 단계에 적용해 생성된 Taste Area 컨텐츠 타입 모듈 파일(PHP 파일)의 일부를 보인 것이다. `taste_area_taste_area_links()` 함수와 `taste_area_subway_links()` 함수는 (그림 6)의 모듈 구성파일 중 `$linked_tables`의 속성들을 반영해 생성된 함수들이다. 이는 `$linked_tables`의 속성으로 정의된 테이블 간의 관계에 의해 연관 컨텐츠를 검색하기 위한 SQL문을 작성하는 함수이다. 사용자가 연결시키고 싶은 컨텐츠 타입의 해당 테이블에 대한 정보를 `$linked_tables` 속성으로 명시하면 ModuleGen은 `$linked_tables` 속성으로 정의된 컨텐츠 타입의 개수만큼 이 함수들을 생성한다.

`taste_area_insert()` 함수는 Taste Area 컨텐츠 생성시 DB 작업을 하는 함수이다. (그림 4)의 삽입 폼을 통해 Taste Area 컨텐츠를 생성할 때 Parent Taste Area Id 드롭다운 리스트를 통해 사용자가 연관시키려는 컨텐츠를 선택하면 `taste_area_insert()` 함수는 DB에 접속해 사용자가

```
<?php
    (... 중략 ...)
function taste_area_taste_area_links($nid, $edit = FALSE) {
    $sql = "SELECT n.nid, n.title
    (... 중략 ...)
    $result = db_query($sql, $nid);
    (... 중략 ...)
    $linked_node_links .= theme('table', $header, $rows); //
    produce table of links
    return $linked_node_links;
}
function taste_area_subway_links($nid, $edit = FALSE) {
    $sql = "SELECT n.nid, n.title
    (... 중략 ...)
    $result = db_query($sql, $nid);
    (... 중략 ...)
    $linked_node_links .= theme('table', $header, $rows); //
    produce table of links
    return $linked_node_links;
}
function taste_area_insert($node) {
    (... 중략 ...)
    $pgsql_handle = pg_connect("host=localhost port=5432
    user=postgres password=lab349 dbname=gisdb");
    $result = pg_exec($pgsql_handle, "SELECT taste_area_name
    FROM taste_area");
    $row = pg_fetch_array($result,
    ($node->parent_taste_area_id)-1);
    $result = pg_exec($pgsql_handle, "select taste_area_id from
    taste_area where taste_area_name='".$row[0]."'");
    $sarr = pg_fetch_array($result, 0);
    db_query(
        "INSERT INTO {taste_area} (vid, taste_area_id,
    taste_area_name, parent_taste_area_id,
    d_county_id$geom_column) VALUES (%d, %d, '%s', %d,
    %d" . $geom_val_clause . ")",
    $node->vid, $node->nid, $node->title, $sarr[0],
    $node->d_county_id);
    if(($node->parent_taste_area_id)==0)
        db_query("UPDATE {taste_area} SET
    parent_taste_area_id=0 WHERE vid = %d", $node->vid);
    }
?>
```

(그림 9) Taste Area 컨텐츠 타입 모듈 파일의 일부

선택한 항목이 `taste_area` 테이블의 `parent_taste_area_id` 컬럼의 속성으로 저장하는 SQL 질의를 실행한다. 이를 위해 먼저 PostgreSQL에 접속하는 함수인 `pg_connect()`을 통해 접속한다. 접속에 성공하면 `$pgsql_handle`에 접속 지시자를 돌려준다. `pg_exec()` 함수는 질의를 실행하는 함수로 질의가 성공적으로 실행되었을 경우 결과 지시자를 `$result`에 저장한다. 첫 번째 `pg_exec()` 함수를 통해 사용자가 생성한 컨텐츠의 `taste_area_name`을 `$result`에 저장한다. `pg_fetch_array()` 함수는 `pg_exec()`에서 주어진 `$result`가 가리키는

질의의 결과로부터 데이터베이스의 한 행을 배열 형태로 가져오는 함수이다. 이 함수에서 사용하는 \$node는 사용자가 콘텐츠를 생성하기 위해 작성 단계에 있는 삽입 페이지를 가리킨다. \$node->parent_taste_area_id를 통해 작성 단계에 있는 삽입 페이지의 Parent Taste Area Id 드롭다운 리스트에서 사용자가 현재 선택한 항목을 가져온다. 이 단계를 통해 사용자가 직접 선택한 항목의 taste_area_name은 \$row 배열에 저장된다. \$row[0]에 저장된 taste_area_name에 해당하는 taste_area_id를 두 번째 pg_exec() 함수를 통해 검색하여 결과를 \$arr 배열에 저장한다. 선택된 taste_area_id가 \$arr[0]에 저장되어 있으므로 db_query() 함수가 실행하는 Insert문에서 taste_area 테이블의 parent_taste_area_id 컬럼 속성에 \$arr[0]의 값을 삽입시키면 사용자가 선택한 항목의 taste_area_id가 속성으로 저장된다.

이러한 작업을 통해 생성된 콘텐츠 타입 모듈을 Drupal에 적용해 콘텐츠를 생성하면 (그림 7)과 같이 taste_area 테이블은 taste_area_id 컬럼과 parent_taste_area_id 컬럼을 통해 자기 자신과 1:N 관계로 연결되며 taste_area_id 컬럼과 subway 테이블의 taste_area_id 컬럼을 통해 subway 테이블과도 1:N 관계로 연결된다. 따라서 (그림 3)과 같이 Taste Area 모듈의 맛집 정보 페이지에 있는 Taste Area 항목과 Subway 항목에 현재 콘텐츠와 연결된 콘텐츠들이 검색되어 표시된다.

5. 기능 비교

MapAppGen처럼 맵 관련 Drupal 모듈을 생성하는 생성기는 현재 존재하지 않는다. 따라서 본 절에서는 MapAppGen으로 생성된 세 가지 종류의 모듈들을 사용해 작성한 맵 응용(MapAppGen 응용)과 Drupal 서드 파티들이 제공한 Gmap 모듈과 NodeMap 모듈을 사용해 작성한 응용(Gmap 응용, NodeMap 응용)의 기능을 비교 설명한다.

<표 1>은 Gmap 응용, NodeMap 응용, MapAppGen 응용의 기능을 비교한 것이다. MapAppGen 응용은 ModuleGen으로 생성된 콘텐츠 타입 모듈, IndexGen으로 생성된 색인 모듈, MapGen으로 생성된 맵 인터페이스 모듈이 통합되어 하나의 응용을 구성한다. Gmap을 사용하여 맵 응용을 만들기 위해서는 위도, 경도를 저장하고 지도를 표시해주는 Location 모듈이 필요하다. 또한, Drupal에서 기본적으로 제공하고 있는 콘텐츠 타입 모듈을 통해 콘텐츠를 생성하므로 Drupal의 기본 콘텐츠 타입 모듈이 필요하다. NodeMap 응용도 Drupal의 기본 콘텐츠 타입 모듈을 통해 콘텐츠를 생성하므로 Drupal의 기본 콘텐츠 타입과 NodeMap 모듈이 필요하다. Gmap 응용은 MySQL과 PostgreSQL 서버와 연동이 가능하고 NodeMap 응용은 MySQL과 연동된다. Gmap 응용과 MapAppGen 응용은 Google Maps API를 사용하며 NodeMap 응용은 Google Maps와 Yahoo Maps API를 사용한다.

위치 정보 형태만 맵 상에 위치 정보를 어떠한 형태로 표시하는지를 나타낸 것인데 Gmap 응용은 point, polygon,

<표 1> Gmap 응용, NodeMap 응용, MapAppGen 응용의 기능 비교 (음영 처리된 부분은 기술적 우위 기능)

	Gmap 응용	NodeMap 응용	MapAppGen 응용
구성 모듈	Location 모듈, Drupal 기본 콘텐츠 타입 모듈	Drupal 기본 콘텐츠 타입 모듈	사용자 콘텐츠 타입 모듈, 색인 모듈, 맵 인터페이스 모듈
연동 DB 서버	MySQL, PostgreSQL	MySQL	PostgreSQL (PostGIS)
Maps API	Google Maps	Google Maps, Yahoo Maps	Google Maps
위치 정보 형태	point, polygon, line, circle	point	point, polygon
사용자 정의 콘텐츠 타입	없음	없음	지원
유형별 검색	없음	없음	지원
연관된 콘텐츠 표시	없음	없음	지원 (1:N, M:N)
유지·보수	+	+	++

line, circle의 네 가지 형태로 저장이 가능하고, NodeMap 응용은 point 형태만 가능하며, MapAppGen 응용은 point와 polygon의 두 가지 형태를 지원한다.

Gmap 응용, NodeMap 응용, MapAppGen 응용의 콘텐츠 생성 과정은 대체적으로 사용자가 저장하고자 하는 위치를 맵 상에 클릭하고 해당 위치에 관한 정보를 입력하는 형태를 취한다. 그러나 세부적으로 콘텐츠를 생성해 내는 방법은 다르다. Gmap은 두 가지 방법을 제공하는데, 첫 번째 방법은 Drupal이 기본적으로 제공하는 콘텐츠 타입 모듈 (Story, Page 등)에 Location 모듈을 연결하여 생성하는 방법이다. 기본 콘텐츠 타입 모듈에 Location을 연결하면 해당 콘텐츠 타입에 맵이 포함되므로 사용자가 맵 상에서 위치를 지정할 수 있게 해준다. 두 번째 방법은 Gmap 매크로 생성기를 통해 생성된 매크로를 기본 콘텐츠 타입 모듈에 붙여 넣어 콘텐츠를 생성한다. NodeMap 응용은 기본 콘텐츠 타입 모듈과 연결을 설정하면 해당 콘텐츠 타입에 맵이 포함되어 맵 정보를 담고 있는 콘텐츠를 생성한다. 이렇게 두 모듈은 기본 콘텐츠 타입 모듈에 의존하기 때문에 사용자 정의의 콘텐츠를 추가 할 수 없다. 그러나 MapAppGen 응용은 ModuleGen을 통해 사용자가 원하는 형태의 콘텐츠 타입 모듈의 생성이 가능하다.

(그림 2)의 맛집탐방 응용에서 지하철역 근처에 위치한 맛집에 관한 콘텐츠들을 해당 역으로 분류하는 기능을 보였듯이, MapAppGen 응용은 콘텐츠 타입 모듈로 콘텐츠를 생성할 때 콘텐츠의 유형을 지정할 수 있으며 지정된 유형별로 검색이 가능하다. 또한, MapAppGen 응용은 한 콘텐츠와 연관된 모든 콘텐츠들을 검색해 표시할 수 있다. 콘텐츠 타입 모듈은 콘텐츠 타입의 속성들을 저장하기 위해 개발자가 제공하는 DB 구성파일에 따라 테이블을 생성하는데 연결시켜 표시하고 싶은 콘텐츠 타입의 테이블 정보들을 모듈 구

성파일에 명시하면 해당 콘텐츠 정보 페이지에 모두 검색되어 표시한다. 그러나 Gmap 응용과 NodeMap 응용은 이러한 기능을 지원하지 못한다.

MapAppGen 응용은 생성될 모듈들의 기본 틀인 템플릿을 기반으로 모듈들을 생성하기 때문에 응용 환경의 변화에 따라 템플릿만 수정해 재생성이 가능하기 때문에 생성된 응용의 유지·보수가 Gmap 응용이나 NodeMap 응용에 비해 용이하다.

6. 결 론

Drupal CMS(Contents Management System)를 기반으로 하는 MapAppGen은 웹 기반 맵 응용을 위해 Drupal 모듈들을 자동 생성해 주는 도구로서 맵 응용의 생산성을 향상시키고 생성된 응용의 유지·보수를 용이하게 한다. MapAppGen은 사용자의 맵 콘텐츠 생성을 위해 지형/지물 콘텐츠 타입 모듈을 생성하는 ModuleGen, 사용자의 지형/지물 검색을 위해 색인 모듈을 생성하는 IndexGen 그리고 모든 콘텐츠 타입 모듈들의 정보를 가지고 생성된 지형/지물들을 맵 상에서 마커를 통해 한눈에 보고 유형별로 검색할 수 있게 하는 맵 인터페이스 모듈을 생성하는 MapGen으로 구성된다.

MapAppGen 응용은 Drupal이 지원하는 기존 Gmap 응용과 NodeMap 응용에 비해 다음과 같은 점에서 비교 우위이다<표 1>.

첫째, Gmap 응용과 NodeMap 응용은 기본 콘텐츠 타입 모듈에 의존하기 때문에 개발자가 원하는 지형/지물 콘텐츠를 추가 할 수 없다. 그러나 MapAppGen 응용은 ModuleGen을 통해 사용자가 원하는 형태의 사용자 정의 지형/지물 콘텐츠 타입 모듈을 생성 할 수 있다.

둘째, MapAppGen 응용은 Gmap 응용과 Nodemap 응용에 비해 맵 상에서 키워드나 지형/지물의 유형별 정보 검색을 지원한다. 예를 들어, 유형에 병원, 호텔 등이 있다면 병원을 선택하여 병원 콘텐츠만 검색하여 표시할 수 있다. Gmap 응용과 Nodemap 응용은 사용자가 생성한 여러 콘텐츠들을 맵상에 마커들로 표시하여 주지만 유형별 검색은 지원하지 못한다.

셋째, MapAppGen 응용은 사용자가 원하는 지형/지물과 1:N 또는 M:N으로 연관된 콘텐츠들을 검색해 함께 표시해 준다.

넷째, MapAppGen 응용은 생성될 모듈들의 기본 틀인 템플릿을 기반으로 모듈들을 생성하기 때문에 응용 환경의 변화에 따라 템플릿만 수정해 재생성이 가능하기 때문에 생성된 응용의 유지·보수가 Gmap 응용이나 NodeMap 응용에 비해 용이하다.

마지막으로, MapAppGen이 갖는 가장 큰 장점은 MapAppGen은 Gmap이나 NodeMap과 같은 맵 모듈 자체가 아니라 원하는 맵 모듈을 자동으로 생성하는 모듈 생성기라는 것이다.

향후, 스마트폰과 같은 모바일 기기에서 동작하는 모바일 맵 응용도 자동 생성하는 방향으로 MapAppGen을 발전시킬 예정이다.

참 고 문 헌

- [1] MBN, <http://mbn.mk.co.kr/news/newsRead.php?vodCode=391462&category=mbn00000>.
- [2] 심화영, http://www.dt.co.kr/contents.html?article_no=2008040902010531727002, 디지털타임즈, 2008.
- [3] ESRI, <http://www.esri.com/software/arcgis/index.html>, 2010.
- [4] Open Source Geospatial Foundation, <http://mapserver.org/>, 2012.
- [5] 음두헌, 모바일 데이터베이스 응용을 위한 스크립트 자동 생성기의 설계 및 구현, 인터넷정보학회논문지, 제 10권 제 4호, pp.71-85, 2009.
- [6] Jia Zhang et al, Towards Increasing Web Application Productivity, Proceedings of the ACM Symposium on Applied Computing, March 14-17, Nicosia Cyprus, pp.1677-1681, 2004.
- [7] Shengjun Xue and Liang Zhang, Developing GIS Software with Component Technic, International Symposium on Intelligent Information Technology Application Workshop (IITAW '08), pp.726-728, Shanghai, 21-22 Dec., 2008.
- [8] Zhang Yi and Wu Li-li, A Study on Plug-in-Based GIS Application Framework, International Conference on Computational and Information Science (ICIS 2010), pp.85-88, Chengdu, 17-19 Dec., 2010.
- [9] Gmap, <http://drupal.org/project/gmap>, 2012.
- [10] NodeMap, <http://drupal.org/project/nodemap>, 2012.
- [11] Matt Butcher, Learning Drupal 6 Module Development, PACKT, 2008.
- [12] Michael Purvis, Jeffrey Sambells, and Cameron Turner, Beginning Google Maps Applications with PHP and Ajax, Apress, 2006.
- [13] PostgreSQL Global Development Group, <http://www.postgresql.org/>, 2012.
- [14] PostgreSQL Global Development Group, <http://postgis.refractor.net/>, 2012.
- [15] 김은형, GeoSaptial Web 플랫폼 기술 분석 및 기술개발전략, 한국GIS학회지 제 17권 제 2호 pp.171-181, 2009. 7.
- [16] Open Geospatial Consortium(OGC), <http://geoserver.org/display/GEOS/Welcome>, 2012.
- [17] ESRI Developer Network, http://edndoc.esri.com/arcobjects/9.2/Net_Server_Doc/developer/ADF/wma_intro.htm, 2012.
- [18] DNR GIS, DNR Interactive Web Mapping Applications, <http://dnr.wi.gov/maps/gis/applist.html>, 2012.

음 두 헌

e-mail : dheum@duksung.ac.kr

1984년 서강대학교 전자공학과(학사)

1987년 미국 Oregon State University 컴
퓨터공학과(석사)

1990년 미국 Oregon State University 컴
퓨터공학과(박사)



1991년 한국전자통신연구원 인공지능연구실(선임연구원)

1992년~현 제 덕성여자대학교 정보미디어대학 교수

관심분야: 객체지향 시스템, 모바일 응용 등