

# 위치기반 DBpedia 검색을 위한 반응형 웹 애플리케이션 개발

이수형, 이용주

경북대학교 컴퓨터학부

e-mail: poponaya@naver.com, yongju@knu.ac.kr

## Developing Responsive Web Application for Location Based DBpedia Retrieval

Su-hyoung Lee, Yong-ju Lee

School of Computer Science and Engineering, Kyungpook National University

### 요 약

본 논문은 링크드 오픈 데이터(Linked Open Data)의 일종인 DBpedia 개체를 위치기반으로 검색하여 지도상에 표시해주고, 표시된 개체중 하나를 선택하게 되면 개체에 관한 RDF 형태의 데이터를 파싱하여 개체에 관한 기본적인 정보와 사진과 외부로의 링크 등을 제공한다. 또한 개체가 가지는 특성을 지정하여 지도에 선택적으로 표시하는 필터링 기능을 제공한다. 이러한 웹 애플리케이션을 개발하기 위해 오픈소스 웹 프레임워크인 루비 온 레일즈(Ruby on Rails)를 사용하였고, HTML5와 Google Map API를 활용한 반응형 웹 애플리케이션으로 구현하였다.

### 1. 서론

유·무선 통신과 관련된 신기술들이 대중들에게 보편화됨에 따라 인터넷과 실생활은 이제 아주 밀접하게 연관되게 되었다. 이로 인해 많은 사람들이 인터넷을 사용하게 되었고 그 과정에서 발생한 많은 정보가 웹상에 존재하게 되었다. 하지만 그 정보들을 재사용하는 데에는 많은 어려움이 따른다. 이러한 점을 보완하기 위해 시맨틱 웹(semantic web)이 제시되었으며, 시맨틱 웹을 실현하기 위한 방안으로 링크드 오픈 데이터(LOD: Linked Open Data)가 사용되고 있다. 링크드 오픈 데이터는 공공 데이터에 HTTP, RDF, URI 등과 같은 국제 표준의 주소 정보와 의미 정보를 추가로 부착하여 데이터 간 결합성을 극대화한 데이터 형태이다[1]. 따라서 링크드 오픈 데이터를 통해 기존 문서 위주의 WWW 전달 방식을 페이지가 아닌 데이터간 연결 중심으로 전환하여 보다 풍부한 자원의 생산과 효율적인 활용이 가능한 방식으로 웹을 지능화시킬 수 있다[2].

본 논문은 위와 같은 링크드 오픈 데이터를 활용하는 반응형 웹 애플리케이션 개발 사례로, 링크드 오픈 데이터의 일종인 DBpedia를 간편하게 활용할 수 있도록 도와주는 위치기반 DBpedia 탐색 웹 애플리케이션 개발에 관한 내용이다.

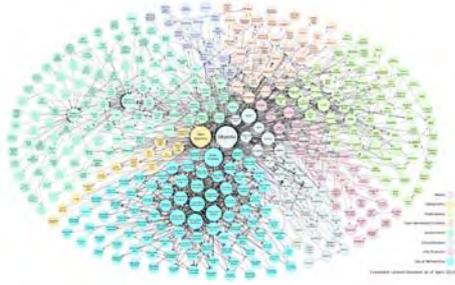
### 2. 이론적 배경

#### 2.1 링크드 오픈 데이터

링크드 오픈 데이터는 시맨틱 웹을 구현하는 한 방법으로 웹상에 산재되어 있는 데이터를 컴퓨터가 인식할 수 있도록 데이터 각각에 의미를 부여하여 연결시키는 의미론적인 방법이다[3]. 링크드 데이터는 웹상에 존재하는 데이터를 URI로 식별하고, URI를 통해 정보들이 서로 연결되는 형태의 데이터 구조를 말한다. 이러한 구조를 통해 얻는 이점은 기존 데이터의 재사용이 용이하고 데이터의 중복을 줄일 수 있다. 링크드 오픈 데이터는 링크드 데이터 중 오픈 라이선스에 따라 배포/재사용 가능한 것을 말하며, 이러한 링크드 오픈 데이터는 현재 빠르게 성장 중이며 국내에서도 공공/민간부문 모두 링크드 오픈 데이터를 이용한 데이터 구축 사례가 많이 존재한다.

링크드 오픈 데이터는 2017년 현재 1100여개의 데이터 세트가 구축되어 상호 연결되어 있으며 LOD의 중심에는 DBpedia가 존재한다. DBpedia는 온라인 백과사전인 Wikipedia 데이터를 바탕으로 하여 만들어진 데이터 세트로서, 온라인 백과사전을 기반으로 만들어진 만큼 분야를 가리지 않고 다양한 것들에 대한 데이터세트를 가지고 있다. 현재 영어버전 DBpedia를 기준으로 약 4백 58만 여개의 데이터를 가지고 있으며 RDF 쿼리 언어인 Sparql을 사용하여 웹 개체를 효과적으로 검색할 수 있다. (그림 1)은 LOD들이 상호 연결된 모습을 나타낸 도식화하여 나타낸 그림으로 수많은 LOD의 중심에 DBpedia가 위치해 다양한 데이터세트와 연결되어 있는 것을 확인할 수 있다.

이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2016R1D1B02008553).



(그림 1) LOD cloud

## 2.2 루비 온 레일즈(Ruby On Rails)

루비 온 레일즈는 루비 언어를 사용하는 오픈 소스 웹 프레임워크이며 Active Record, CoC(Convention over Configuration), DRY(Don't repeat yourself), MVC 패턴 등 잘 알려진 소프트웨어 엔지니어링 디자인 패턴과 Agile 개발 방식이 프레임 워크에 통합되어 있다[4]. 이로 인해 유지 보수성 및 확장성 등이 우수하고 전체 애플리케이션에 대한 코드 관리가 용이하다. 또한 외부 라이브러리를 추가하는 것도 매우 간단한 편이다. 위와 같은 특성으로 인해 루비 온 레일즈를 이용하면 상대적으로 빠른 속도로 웹 애플리케이션을 개발하는 것이 가능하기 때문에 프로토타입을 제작 할 때나 소규모로 프로젝트를 진행할 때 유용하게 사용할 수 있다.

## 3. 반응형 웹 애플리케이션 개발

### 3.1 애플리케이션의 구성

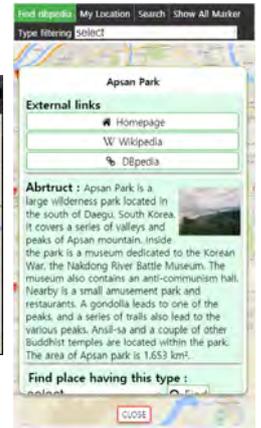
(그림 2)와 (그림 3)은 각각 데스크톱 환경과 모바일 환경에서의 화면으로 본 논문에서 개발한 반응형 웹 애플리케이션의 전체적인 모습을 보여준다. 반응형 웹은 말 그대로 웹브라우저의 가로 사이즈에 반응하여 최적화된 콘텐츠 UI를 제공해주는 서비스로서 별도의 신기술이라고 하기 보다는 CSS3의 속성을 활용한 웹 UI기법이라고 할 수 있다[5]. 반응형 웹은 1개의 HTML, 즉 원소로 구현하여, 새로운 디바이스가 등장해도 별도의 대응 없이 최적화된 화면을 구현할 수 있다. 두 그림은 반응형 웹으로 구현한 실행 화면으로 환경에 맞추어 레이아웃이 변경되어 표시되어 있다. 가장 위쪽에는 주변 DBpedia 개체 검색, 찾은 개체들의 특성을 바탕으로 필터링하는 기능 등을 가진 버튼들이 나열되어 있고, 나머지 부분은 지도가 차지하고 있다.

검색 기능을 이용하여 지도에 나타난 개체들 중에서 하나를 선택하면 이에 대한 자세한 정보를 확인 할 수 있다. (그림 2)와 (그림 3)에는 한 개의 개체를 선택하여 세부 정보를 보여주는 팝업 형식의 세부 정보창이 띄워져 있는 모습이다. 이 세부 정보 확인창의 가장 상단에는 개체 이름이 표시되어 있으며, 그 아래에 3부분으로 나뉘어서 외부로의 링크를 제공해주는 첫 번째 부분과, 간단한

요약정보와 사진을 보여주는 2번째 부분, 그리고 해당 개체와 같은 속성을 가지는 개체들을 필터링하는 3번째 부분으로 구성되어 있다.



(그림 2) 데스크톱 환경



(그림 3) 모바일 환경

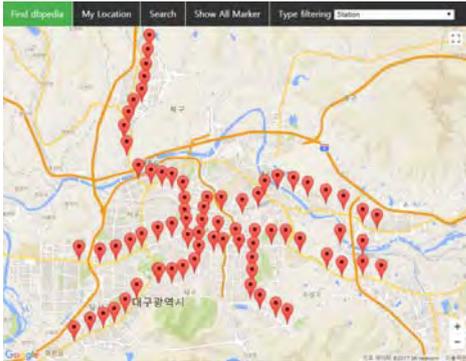
### 3.2 애플리케이션의 개발 기법

웹 애플리케이션의 개발 플랫폼은 루비 온 레일즈를 사용하였고, HTML/CSS3와 자바스크립트를 적절하게 섞어 사용하여 여러 기능들을 구현하였다. 또한 Google에서 제공하는 Google Map API를 이용하여 지도와 관련된 기능들을 구현하였으며, 반응형 웹을 구현하여 기기의 해상도 정보를 기준으로 데스크톱과 모바일 페이지에 적절한 형태로 표시하도록 CSS를 구성하였다.

첫 페이지에 접속 시 HTML5의 geolocation API를 이용하여 사용자의 위치를 파악한다. 이 geolocation API의 경우엔 IP, Wi-Fi, GPS 등 다양한 방법을 통하여 사용자의 위치를 파악한다. 파악한 위치를 위도/경도 값으로 받아 지도의 중심을 사용자 위치로 하여 표시한다. 이후 상단에 있는 메뉴 중 Search 버튼을 누르면 사용자의 위도/경도 값을 바탕으로 DBpedia에서 제공하는 Sparql 쿼리 서비스를 이용하여 주변의 DBpedia 개체를 검색한다. Sparql 쿼리 결과는 RDF/XML 형식의 결과 값을 다운받은 뒤 자바스크립트를 이용하여 개체들을 지도상에 마커 형태로 표시한다. 맵에 표시된 마커에 데스크톱 환경에서는 마우스를 올릴 경우(혹은 모바일 환경의 경우 터치했을 때) 개체의 이름이 표시되도록 했으며, 표시된 이름을 클릭했을 때 상세정보 팝업이 나타나도록 했다. 또한 쿼리문에 개체들의 타입(rdf:type)을 질의하는 부분을 추가하여 각 개체들의 타입을 알아내고, 이 값들을 바탕으로 상단 메뉴 중 타입 필터링 메뉴의 드롭다운 형식의 목록들을 검색된 개체들이 가지는 타입의 목록을 동적으로 구성한다.

이같이 구성된 목록의 값들 중 하나를 선택하면 검색된 개체들 중 해당 속성을 가지고 있는 값만 지도에 표시되는 기능을 구현하였다. (그림 4)는 목록 중 'Station' 속

성을 가지는 개체만 표시된 화면으로 지하철과 기차역이 여러 개의 선 형태로 나열되어 있음을 알 수 있다. 이와 같은 필터링 기능으로 원하는 특성을 가지는 개체를 손쉽게 찾을 수 있다. 혹은 하나의 개체를 선택하여 나타나는 세부 정보창에서도 해당 개체가 가지고 있는 타입을 선택하여 같은 특성을 가지는 개체를 찾는 기능을 갖추어서, 동일한 특성을 가지는 개체를 쉽게 찾을 수 있게 도와준다. 세부 정보 창에서 특성 선택 후 Find버튼을 눌렀을 때도 (그림 4)과 같이 동일하게 표현된다.



(그림 4) 속성 필터링

초기 계획은 클라이언트 측에서 모든 데이터를 처리하는 걸로 계획 했지만, 상세 정보 표시창의 표시내용은 서버로 키워드를 전송한 후 데이터를 다시 받아 표시하는 방식으로 구현하였다. 이와 같은 구조를 가진 이유는 HTML의 위치 정보를 사용하기 위해서는 보안 및 개인정보 보호를 위해 https프로토콜을 이용한 보안 연결이 요구되었으나, DBpedia 페이지가 이를 지원하지 않기 때문에 클라이언트 측에서 DBpedia에 연결을 시도하면 자동으로 연결이 차단되어 애플리케이션의 정상적인 작동이 불가능하기 때문이다.

그래서 개체를 선택하면 서버 측으로 개체에 관한 정보를 전송하고, 서버 측에서 DBpedia의 데이터를 읽어 들여서 필요한 내용만 다시 클라이언트 측으로 전송하는 방법으로 구현하였다. 서버에서 DBpedia에 해당 개체에 관한 데이터를 RDF/XML 형식으로 다운받아, 개체의 이름이나 홈페이지 링크, 요약정보, 사진 등 세부정보에 필요한 데이터들을 선택하여 클라이언트 측으로 전송해 표시하도록 구성하였다.

### 3. 결론

본 논문에서는 표준 웹을 이용하여 위치 기반의 DBpedia 탐색 반응형 웹 애플리케이션을 개발하였다. 사용자 주변의 DBpedia 개체의 간단한 정보와 추가적인 탐색 기능 등을 제공하며, 반응형 웹을 사용함으로써 표준 웹을 지원하는 브라우저로 접속만 가능하다면 사용자의 접속 기기의 종류나 운영체제, 기기의 크기 등 환경에 구

애받지 않고 다양한 환경에서 접속하여 사용이 가능하다. 이를 이용하면 사용자에게 환경에 구애받지 않고 간편하게 시맨틱 웹을 이용할 수 있는 기회를 제공한다.

### 참고문헌

- [1] 이현정, “우리나라 공공데이터의 이용활성화 방안에 관한 연구: 링크드 오픈 데이터화 전략을 중심으로”, 정보관리학회지, Vol. 31, No. 4, 2014, pp. 249-266
- [2] 이용주, “링크드 오픈 데이터를 활용한 시맨틱 모바일 메쉬업”, 한국정보기술학회논문지, Vol. 14, No. 11, 2016, pp. 93-100
- [3] J. H. Lee, “A Study on the Extension of Archival Information Service Based on LinkedOpen Data in the Presidential Archives”, 한국기록관리학회지. Vol. 15, No. 2, 2015, pp. 55-82
- [4] Muller, Henning. “A CoAP Server with a Rack Interface for Use of Web Frameworks such as Ruby on Rails in the Internet of Things.” Diploma Thesis, University of bremen, 2015.
- [5] 박준우, 유금 “스마트 미디어 환경에서의 N-스크린 기반 반응형 웹 콘텐츠 고찰” 디지털디자인학연구, Vol. 12, No. 3, 2012, pp.257-266.