JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 6 : 1174~1182 Jun. 2017

i-Manager : LOD 인스턴스 개발 시스템의 구현

문희경[1] · 한성국[2*]

i-Manager: An Implementation of LOD Instance Development System

Hee-kyung Moon[1] · Sung-kook Han[2*]

[1]Department of Computer Engineering, Wonkwang University, Iksan 54538, Korea
[2*]Department of Computer Engineering, Wonkwang University, Iksan 54538, Korea

요 약

웹상에서 이질적 형태의 다양한 데이터를 개방, 공유하여 차세대 데이터웹을 실현하고자 하는 연구개발이 활발하게 수행되고 있다. 이를 위해 표준 데이터 모델로 온톨로지 기반의 LOD가 개발되었다. LOD기반 시스템을 효과적으로 개발하기 위해서는 전문화된 인스턴스 생성 시스템이 필수적으로 요구되고 있다. 본 논문은 LOD 시스템의 요구사항과 다양한 응용분야의 개발환경을 고려하여, LOD 인스턴스 개발에 적합한 i-Manager를 설계 구현하였다. i-Manager는 LOD 인터페이스 시트를 이용해서 온톨로지와 인스턴스 계층을 분리하고, 인스턴스 편집/저장, 시각화, LOD 질의 처리 등 LOD 인스턴스 개발에 전문화된 기능들을 제공한다. 본 논문은 LOD 인스턴스 개발의 새로운 방향을 제시하고 있으며, 구현된 i-Manager는 다양한 분야에서 LOD 개발 범용환경으로 활용할 수 있다.

ABSTRACT

The research and development on Web of Data to realize the opening and sharing of diverse, heterogonous data on the Web has been actively accomplished. As a standard data model for this effort, LOD (Linked Open Data) based on ontology has been proposed. A specialized instance generation system is vital to the development of LOD-based system effectively. This paper implements i-Manager as an appropriate environment for the development of LOD instances, considering the requirements of LOD systems and the practical development environment of the diverse application domains. i-Manager separates the instance layer from the ontology layer by way of LOD Interface Sheet (LIS) and implements the specialized functions requested in LOD instance development, such as instance edit/store, visualization and SPARQL query processing. This paper proposes a new approach for LOD instance development, and i-Manager can be applied for the practical LOD development environment in the diverse application areas.

키워드 : 링크드오픈데이터, 데이터웹, 온톨로지, RDF, 트리플, SPARQL

Key word : LOD(Linked Open Data), Web of Data, Ontology, RDF, triple, SPARQL

Received 17 April 2017, **Revised** 24 April 2017, **Accepted** 09 June 2017

* **Corresponding Author** Sung-kook Han(E-mail:skhan@wku.ac.kr, Tel:+82-63-850-6749)
Department of Computer Engineering, Wonkwang University, Iksan 54538, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.6.1174 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

IBM의 왓슨(Watson), 구글의 알파고(AlphaGo)에서 보는 바와 같이 인공지능, 기계 학습의 딥러닝(Deep Learning) 등의 비약적인 발전으로 인간의 지적 수준을 능가하는 지능 정보시스템들이 다양한 분야로 확산되고 있다. 음성비서 또는 챗봇(chatbot)으로 알려진 아마존의 알렉사(Alexa)등의 지능 정보시스템이 일상생활에서 활용되고 있고, 고수준의 번역 앱이 스마트폰에 기본으로 탑재되는 등 지능 정보서비스 또한 일반화되고 있다. 이러한 지능 정보시스템과 서비스는 잘 정제된 고품질 데이터가 핵심 요소이다. 양질의 대용량 데이터가 있어야 정확한 기계 학습이 가능하고, 다양한 질의에 효과적으로 응답할 수 있어, 고품질 데이터의 확보가 정보기술 개발의 현안으로 대두되고 있다[1,2].

빅 데이터(Big Data), 데이터 사이언스(Data Science) 등 여러 분야에서 고품질 데이터 개발을 위한 연구 노력을 지속해 왔다. 특히, 컴퓨터가 의미를 이해하고 처리할 수 있는 온톨로지(ontology) 기반의 Linked Open Data(LOD)는 웹상에서 데이터의 개방과 공유를 위한 표준 데이터 모델이 되었다. LOD는 DBpedia, DBLP등 거대한 LOD 클라우드를 형성하고, 차세대 데이터의 웹(Web of Data)과 지능 정보서비스의 공통 플랫폼이 되고 있다[3-5].

문화, 교육, 경제, 교통 등 다양한 분야의 공공 데이터가 LOD로 공표(publication)되고 있고, 소셜미디어, IoT 등의 정보원으로부터 LOD 추출생성(population)에 노력하고 있지만, 유용한 LOD 데이터 셋(data set)의 개발은 난해한 과제이다. LOD 개발은 온톨로지와 밀접한 관계가 있어 온톨로지 개발 라이프 사이클(life cycle)을 주로 활용하여 왔다. 그러나 온톨로지 개발은 도메인 개념 모델링인 반면에, LOD 개발은 실제 인스턴스(instance) 구현을 목표로 하고 있어 차이가 있다[6,7].

본 논문은 LOD 개발 라이프 사이클을 새로운 관점에서 고찰하여 효과적인 LOD 개발 전략을 도출하고, 인스턴스 중심의 LOD 개발 시스템을 설계, 구현한다. 본 논문의 LOD 인스턴스 구축 시스템은 온톨로지 스키마로부터 사용자 친화적 인스턴스 편집기를 생성하고, 인스턴스의 편집, 검증, 시각화, 질의 처리 등 LOD 개발 핵심 기능을 제공하여, 체계적인 LOD 데이터 셋 개발을 실현한다.

본 논문은 제1장 서론을 포함하여 6장으로 구성되어 있다. 제2장에서는 기존의 LOD 개발 접근방식을 분석하여 새로운 LOD 개발 전략의 필요성을 서술한다. 제3장에서는 인스턴스 기반 LOD 개발의 접근 방식을 상술하고, 온톨로지 스키마로부터 인스턴스 편집기를 생성하는 방법을 기술한다. 제4장은 LOD 인스턴스 구축 시스템의 구조, 기능과 구현을 서술하고, 실제 활용사례를 제5장에서 설명한다. 제6장 결론은 본 논문의 성과와 향후 연구과제에 대해 서술한다.

Ⅱ. 관련 연구

공공 데이터를 중심으로 한 LOD개발이 확산되면서 다양한 개발 방법들이 활용되고 있다. 이들 LOD 개발 방법은 기존의 온톨로지 활용 방법, 관계 데이터베이스의 LOD 매핑(mapping) 방법, 소셜미디어, 뉴스, 웹 자원 등 비정형 데이터로부터 대량 생성 방법 등 3가지 방식으로 요약할 수 있다.

LOD는 인스턴스는 여러 온톨로지를 기반으로 생성되어 웹상에 개방, 공유하는 정보자원이다. 이러한 특성이 있기 때문에, LOD 개발에 온톨로지 개발 방법론과 도구(tool)를 활용해 왔다. 특히 인스턴스와 관련된 도메인 온톨로지 개발이 필요한 경우에, Protégé, TopBraid Composer 등 온톨로지 개발 환경을 활용하여 도메인 온톨로지와 인스턴스를 개발하였다. 그러나 이러한 접근방법은 온톨로지 개발에 중점을 두고 있어, LOD에서 요구되는 대용량 인스턴스 생성과 저장, JSON-LD 등의 직렬화(serialization), SPARQL 질의 처리 등을 지원하기에는 미흡하다[8].

방대한 데이터를 축적하고 있는 기존 관계 데이터베이스의 내용을 LOD로 변환할 수 있다면, 유용한 LOD 데이터 셋을 손쉽게 확보할 수 있다. 관계 데이터베이스를 LOD 기반 모델인 RDF로 변환하는 다양한 매핑 방법이 연구되었다[9]. W3C에서는 직접 매핑(direct mapping) 방식에 기초한 표준 매핑 언어 R2RML을 지원하여 이러한 방법을 적극 지원하고 있다. 그러나 R2RML 등의 매핑 방식은 지극히 복잡하여 완전한 R2RML 처리기를 구현하고 있지 못하며, 데이터 셋 생성의 중복성, 표준 온톨로지 활용 등이 어려운 상태이다.

온톨로지 추출생성 방식을 이용하여 LOD를 자동 생

성할 수 있다. 소셜미디어, 뉴스, 블로그 등 다양한 정보 자원에 자연언어 처리, 빅 데이터 분석, 기계학습 등의 기술을 적용하여 LOD 트리플을 추출할 수 있다[10]. 이 방식으로 대용량의 LOD 트리플을 간편하게 생성할 수 있지만, LOD 데이터 셋의 정확성과 품질 등 해결해야 할 과제가 많다[11]. 표준 데이터 모델을 가진 LOD가 사물 인터넷, 빅 데이터, 인공지능과 차세대 웹 등의 기반 구조가 될 것으로 기대되지만, LOD 데이터 셋 구축을 위한 전문화된 개발 전략과 도구 개발이 미진한 상태이다. LOD 고유의 특성을 고려한 사용자 친화적인 LOD 전문 개발환경이 요구되고 있다.

Ⅲ. 계층기반 LOD 인스턴스 개발

LOD 데이터 셋을 효과적으로 개발하기 위해서는 인스턴스에 전문화된 개발환경이 필요하다[12,13]. 본 장에서는 온톨로지와 인스턴스 계층을 분리한 새로운 LOD 개발 전략과 시스템 구현 방법을 제시한다.

3.1. 관점의 분리

분할정복(divide-and-conquer)과 함께 문제해결 방법으로 널리 사용되는 중요 전략으로 관점의 분리(Separation of Concerns: SoC)가 있다[14]. SoC는 소프트웨어 등 복잡한 시스템을 설계 및 구현하는 기본원리로 문제의 구성요소 또는 기능성(functionality)을 중복되지 않는 독립된 관심사로 분할하여 집중함으로써 문제를 조직적으로 해결하는 전략이다. SoC는 복잡한 문제를 단일 관점의 단순 문제들로 환원하여 문제의 복잡성을 극복하는 기본 원리인 것이다.

SoC의 복잡성 해소 특성은 소프트웨어 공학 등에서 시스템 개발의 기본 원리로 활용하고 있다. 소프트웨어 객체의 모듈화(modularization), 캡슐화(encapsulation) 등은 SoC에서 유래한 개념이며 CSS, JavaScript도 SoC의 사례이다. 그림 1처럼 HTML은 문서 또는 콘텐츠, CSS는 표현 스타일, JavaScript는 사용자와의 상호작용을 독립적으로 분리하여 정의함으로써 효과적인 웹 사이트 개발과 체계적인 유지관리를 실현할 수 있다. 응용시스템을 모델(Model), 뷰(View), 컨트롤러(Controller)의 3가지 구성요소로 정의한 MVC 디자인패턴도 SoC적 사고 방식이다.

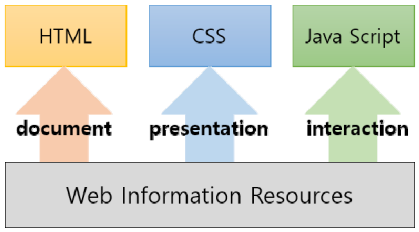

Fig. 1 Example of separation of concerns (SoC)

3.2. LOD 계층 구조

글로벌 분산 웹 환경에서 이질적 정보자원(heterogeneous resource)을 의미적으로 상호 연결하여 공유하는 정보 모델로 트리플(triple) 구조가 제시되었다. 트리플 모델은 웹상에서 고유한 URI를 갖는 정보자원을 표현하는 주어(subject), 정보자원의 속성, 관계, 특성 등을 표현하는 술어(predicate), 술어의 값을 지정하는 목적어(object)의 3요소로 구성된다. 트리플 모델은 단순한 구조이지만 웹의 링크처럼 지속적으로 연결되어 거대한 정보 공간을 구축할 수 있는 확장성이 있다. W3C에서는 트리플 모델을 표현하기 위한 표준 언어로 RDF를 제정하였다. RDF는 웹상의 정보자원을 구성하기 위한 트리플 기반의 프레임워크이다. 한편, RDF의 트리플 프레임워크에서 정보자원의 의미를 표현할 공유 어휘 정의를 위해서 RDF Schema가 개발되었다. RDF Schema는 메타데이터 수준의 도메인 온톨로지를 정의하는 언어이다[8,15].

RDF Schema는 그림 2처럼 웹상에서 도메인 온톨로지를 구현하는 기반을 제공하고, LOD 인스턴스는 이 기반 위에서 운영되는 실제 데이터로 분리 가능한 계층을 형성하고 있다.

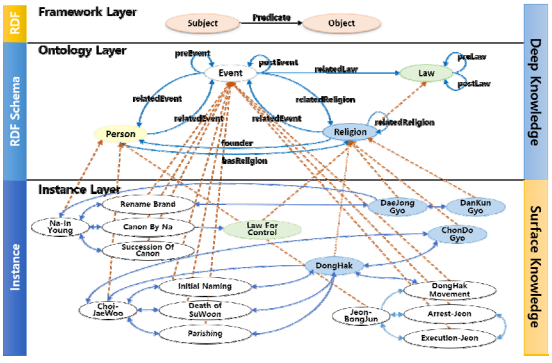

Fig. 2 Layered Structure of LOD data set

RDF와 RDF Schema는 도메인 개념구조에 해당하는 심층지식(deep knowledge), LOD 인스턴스는 개별 지식을 표현하는 표면지식(surface knowledge)이라 할 수 있다. OWL 온톨로지의 TBOX 및 ABOX와 동일한 맥락이다. 이처럼 LOD개발은 수평적 관점의 분리(horizontal SoC)가 이루어진 대표적 사례이다.

LOD 데이터 셋의 구조와 의미를 표현하는 RDF에도 이처럼 관점의 분리가 정의되어 있지만 기존의 LOD 개발은 이 관점을 분리를 활용하지 않았기 때문에 시행착오와 데이터 셋 검증 등에 문제가 있었다. LOD 개발에도 RDF의 관점 분리가 명확하게 활용되어야 고품질의 데이터 셋을 효과적으로 개발할 수 있다.

3.3. 인스턴스 계층 분리와 인터페이스

계층 분리를 구현하는 일반적인 방법은 두 계층 간에 인터페이스(interface)를 정의하는 것이다. LOD의 독자적 인스턴스 생성체제는 온톨로지와 인스턴스 계층 간의 인터페이스를 정의함으로써 실현할 수 있다. 이때, 두 계층 간의 인터페이스는 다음과 같은 요구사항을 만족하여야 한다.

- **(근원성)** 인스턴스는 온톨로지 기반 위에 정의되므로 인스턴스 생성체제는 RDF 온톨로지 구조로부터 유도되어야 하고 온톨로지 개념 특성을 효과적으로 표현하여야 한다.
- **(기술성)** LOD 개발과 관련된 저장, 공표, 질의, 응용 등에서 요구되는 다양한 기능을 제공하는 통합체계를 구축하여야 한다.
- **(적합성)** 트리플 구조로 표현되는 LOD 인스턴스 생성에 적합하도록 구조화, 시각화 되어야 한다.
- **(활용성)** RDF, LOD 등에 대한 전문지식이 미약한 다양한 분야의 사용자도 인스턴스를 구축할 수 있도록 실용적이어야 한다.
- **(편리성)** 인터페이스를 간편하게 정의할 수 있어야 하며, 사용자의 필요에 따라 기능을 추가, 확장할 수 있어야 한다.

본 논문에서는 이러한 요구사항에 기초하여 LOD 인스턴스 개발 시스템에서 사용될 LOD 인터페이스 시트(LOD Interface Sheet: LIS)를 정의하였다. LIS는 LOD 생성 화면 형식을 정의하는 XML 문서로 엔트리(Entry)를 중심으로 작성된다.

$$Entry := entryElement, entryType, entryAttr^{+} \quad (1)$$

여기서 entryElement는 엔트리 온톨로지의 프로퍼티(property)로 LOD의 술어이며, entryType은 LOD 목적어의 화면 표현 형식이며, entryAttr은 entryType의 속성을 의미한다. 본 논문은 RDF, OWL 온톨로지의 rdfs:literal, rdf:datatype과 XML Shema등의 데이터타입을 분석하여, LOD 개발에 필요한 entryType을 표 1과 같이 정의하였다.

Table. 1 Entry types defined in LOD Interface Sheet

entryType	contents	entryAttr
LiteralBox (LBox)	Input multipurpose string	width
TextBox (TBox)	Input text, explanation, etc	width, line
ClassBox (CBox)	Set class of instance	defaultClass Name, width
FileBox (FBox)	Specify URI for the relevant information resources, Documents, photos, images, videos, etc.	defaultFolder, width
NumberBox (NBox)	Input numeric data	width
DateBox (DBox)	Input date information	dateFormat
URIBox (UBox)	URI location of information resource	width
DataBox (VBox)	general data value information	datatype, width
FolderBox (GBox)	Folder location where information resources are stored	defaultFolder, width
SelectorBox (SBox)	Select possible data values	width, item

```
<Entry Eng="NameKor" Kor="한글이름" entryType="LBOX" width="30"
prefix="foaf" prefixValue="name" />
<Entry Eng="NameChi" Kor="한자이름" entryType="LBOX" width="30" />
<Entry Eng="PenName" Kor="호" entryType="LBOX" width="30" />
<Entry Eng="ChildName" Kor="아명" entryType="LBOX" width="30" />
<Entry Eng="NameEng" Kor="영문이름" entryType="LBOX" width="30" />
<Entry Eng="BirthDate" Kor="생년월일" entryType="DBox" />
<Entry Eng="DeathDate" Kor="사망월일" entryType="DBox" />
<Entry Eng="BirthPlace" Kor="출생지" entryType="CBox"
    defaultEngClassName="Region" defaultKorClassName="지역"
    width="30" duplicate="Y" />
<Entry Eng="Family" Kor="가족사항" entryType="LBOX" width="30"/>
<Entry Eng="ExternalLink" Kor="외부자료링크" entryType="FBox" width="30"
    duplicate="Y" />
<Entry Eng="References" Kor="참고자료" entryType="LBOX" width="30"
    duplicate="Y" />
<Entry Eng="hasReligion" Kor="종교" entryType="CBox"
    defaultEngClassName="Religion" defaultKorClassName="종교" width="30"
    duplicate="Y" />
<Entry Eng="RelatedEvent" Kor="관련사건" entryType="CBox"
    defaultEngClassName="Event" defaultKorClassName="사건" width="30"
    duplicate="Y" />
<Entry Eng="PhotoAndVideo" Kor="사진영상" entryType="FBox" width="30"
    duplicate="Y" />
<Entry Eng="Explanation" Kor="설명" entryType="TBox" width="30" line="10" />
```

Fig. 3 Part of LOD Interface Sheet (Person Class)

그림 3은 Person 클래스의 인스턴스 생성을 위한 LIS의 일부이다.

Ⅳ. i-Manager 설계와 구현

LOD 통합 개발 시스템인 i-Manager의 구현에 대하여 서술한다. i-Manager의 구조, 구현된 기능과 알고리즘을 설명한다.

4.1. i-Manager 요구사항 분석

LOD 개발은 일반 온톨로지 개발과는 달리, 웹상에 분산된 이질적 정보자원의 개방과 공유를 목적으로 하고 있기 때문에 고유한 독자적 특성이 있다. LOD 개발 시스템은 이러한 특성을 충족하는 전문화된 기능을 지원하여야 한다. 팀 버너스리가 제시한 LOD의 4대 원칙의 준수 등 LOD 개발 시스템은 다음 요구사항을 만족할 수 있어야 한다.

- 팀 버너스리가 제시한 LOD 4대 원칙을 실현하고 고품질 LOD 개발을 위한 제반 기능을 지원하여야 한다.
- 온톨로지와 연계하지만, 계층 분리된 인스턴스 중심의 개발환경이어야 한다.
- 인스턴스의 편집, 저장, 검증 등 핵심 기능이 인스턴스 품질과 수준을 제고할 수 있도록 되어야 한다.
- 인스턴스와 관련된 텍스트, 이미지, 사진, 동영상 등 다양한 정보자원을 용이하게 연결할 수 있어야 한다.
- 인스턴스와 온톨로지를 연계한 시각화(visualization) 기능이 제공되어야 한다.
- LOD 공유에 필수인 SPARQL Endpoint를 지원하여야 한다.

본 논문은 LOD 개발 시스템의 요구사항과 다양한 도메인의 개발자 실무 환경을 고려하여, LOD 인스턴스 개발에 적합한 i-Manager를 설계, 구현하였다.

4.2. i-Manager 아키텍처

i-Manager는 관점의 분리를 적용해 온톨로지 계층과 인스턴스 계층을 분리하고, 인스턴스 계층에 적합한 시스템 아키텍처를 갖고 있다. 온톨로지 계층은 기존의 우수한 온톨로지 개발 도구를 활용하고, i-Manager는 인스턴스 추출생성에 적합하도록 전문화 되었다. i-Manager의 시스템 아키텍처는 그림 4와 같다.

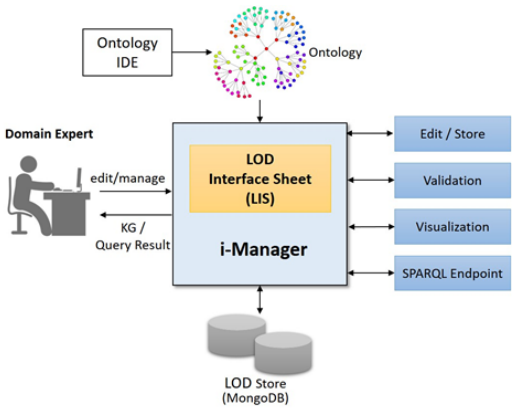

Fig. 4 i-Manager system architecture

- LIS: i-Manager의 구성 파일(configuration file)로, 관련 온톨로지로부터 반자동으로 생성된다. 사용자는 자신의 개발환경에 적합하도록 i-Manager를 설정(customization)할 수 있다.
- 편집/저장: 편집된 인스턴스는 JSON-LD 형식으로 변환되어 LOD전용 데이터베이스에 저장된다.
- 검증: 편집된 인스턴스는 add-on으로 추가된 검증기로 완전성을 확인할 수 있다.
- 시각화: LOD 트리플을 시각화한 지식 그래프와 두 개념 노드 간의 관계를 표현하는 지식 연관그래프를 지원한다.
- SPARQL Endpoint: 사용자의 SPARQL 질의어를 처리한다.

i-Manager는 LOD 인스턴스 개발에 전문화된 구조로 구성되어 있으며, 이외에 필요한 보조 기능은 add-on 형식으로 추가할 수 있다.

4.3. i-Manager 기능과 구현

i-Manager는 데이터센터에 설치된 서버에 구현하였다. 서버는 CPU Intel E5-2630(2.6GHz), RAM 64G이며, Visual Studio 2015, MongoDB, Angular JS, jQuery, Timeline JS 등을 활용하였다. 구현된 i-Manager 핵심 기능은 다음과 같다.

4.3.1. 인스턴스 편집과 저장

온톨로지 스키마는 RDF/XML 형태로 직렬화되어 XML 파서(parser)에 입력된다. XML 파서는 트리플의 주어, 목적어에 해당하는 클래스와 프로퍼티를 추출하

여 LIS의 기본구조를 생성하고, 사용자는 자신의 개발 환경에 적합하도록 entryType과 entryAttr을 설정한다.

생성된 LIS는 HTML로 변환되어 웹 브라우저상에 인스턴스 편집 화면을 만들고, 사용자가 입력한 인스턴스는 LOD 데이터베이스에 저장된다. 본 논문에서는 Jena와 같은 전통적인 RDF 저장 구조가 아니라 성능과 관리의 효율성을 실현하기 위해, LOD의 표준이 되고 있는 JSON-LD를 대표적 NoSQL인 MongoDB에 저장한다. 인스턴스의 편집과 저장 과정은 그림 5와 같다.

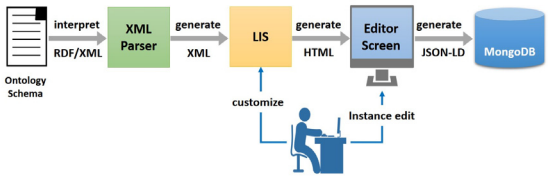

Fig. 5 instance edit and store

4.3.2. 지식 그래프의 생성

지식 그래프(knowledge graph)는 지정된 주어 노드를 중심으로 술어, 목적어를 지속적으로 확장하여 시각적으로 표현하는 것이다. 지식그래프는 LOD 트리플 전체 구조를 파악할 수 있는 도구로 LOD 개발과 활용에 중요한 기능이다. 지식 그래프 생성 알고리즘은 그림 6과 같다.

```
[Algorithm: Knowledge Graph (KG)]
input: a specified subject node, snode
output: knowledge graph, kgraph
data: LOD Store

input(snode); // initial subject node
repeat
while(snode==Null) do no-op; // wait until snode
kgraph(snode); // recursive node expansion
until false;

// recursive triple expansion of node
function kgraph(node)
while (node != null) do
{
triples = getTriples(node) from SPARQL Engine;
for all pred_obj pairs in triples
    connect(node, pred_obj);  // connect p-o pair
for all onode in triples
    kgraph(onode);          // expand all onode
}
return (kgraph);
```

Fig. 6 Algorithm Knowledge Graph(KG)

주어 노드(snode)에 대하여 다음의 SPARQL 질의 처리를 통해 관련 술어와 목적어 쌍을 연결하여 지식지도를 형성한 후, 모든 목적어 노드(onode)에 대하여 재귀적(recursive)으로 SPARQL 질의 처리를 수행하여 노드를 확장한다.

```
SELECT ?p ?o
WHERE {
        node ?p ?o  }
```

i-Manager가 SPARQL 질의 처리 기능을 가지고 있어 이러한 재귀적 질의 처리가 가능하며, 간결하고 효과적인 방법으로 지식 지도를 생성할 수 있다.

4.3.3. 지식 그래프의 구현

두 개념노드 S, T사이의 연관관계를 파악하는 것이 지식 연관그래프(knowledge association graph)이다. 지식 연관그래프는 직접적으로 연결되지 않은 개념간의 의미적 관계를 파악하는데 유용한 도구이다.

지식 연관그래프는 두 개념노드 사이의 연결통로(path)가 아니라, 방향성에 관계없이 모든 연관관계를 형성하는 것이다.

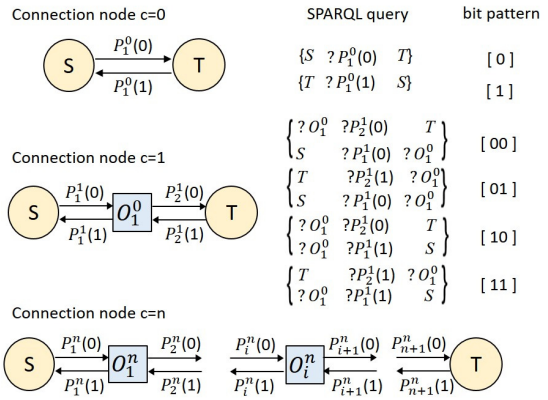

Fig. 7 Analysis of Knowledge Association Graph

그림 7처럼 두 개념노드 S, T가 주어지면 직접 연결관계도 있고, 다수의 중간 노드를 통해서 간접 연결되는 경우도 있다. 노드 간의 연관 술어를 $P_s^c(d)$, 중간의 연관노드를 O_s^c라 하자. 여기서 c는 연관노드의 수, s는 술어의 일련번호, d는 연결방향(순방향=0, 역방향=1)이다. 일반적으로 i개의 연관노드가 있을 경우, 연관관계

는 (i+1)개의 항(term)을 가진 $2^{(i+1)}$개의 SPARQL 질의 처리 결과에 의해 결정된다. 이때, SPARQL 질의어의 where절은 연관관계의 방향을 표시하는 비트 패턴(bit pattern)으로 형식화 할 수 있다. 지식 연관그래프를 생성하는 알고리즘은 그림 8과 같다.

[Algorithm: Knowledge Association Graph (KAG)]
input: two concept nodes, S and T
output: knowledge association graph, kagraph
data: LOD Store

input(S, T); // concept nodes
for connection c=0 to n {
  triples = $\bigcup_{i=1}^{2^{c+1}}$ getAssociation(c, i-1)
                    // all association at c level
  if (triples == NULL) return (kagraph)
  connectAssociation(kagraph, triples)
}

// evaluate each SPARQL query at c level one by one
function getAssociation(c, nterm)
bitPattern[] = makeBitPattern(c, nterm)
for all bit bi(i=0,1,2,…) in bitPatten[ ]
  if(bi=0) term = $O_i^c P_{(i+1)}^c(0) O_{(i+1)}^c$
  if(bi=1) term = $O_{(i+1)}^c P_{(i+1)}^c(1) O_i^c$
  concat(wclause, term)
$O_0^c = S, \quad O_{(c+1)}^c = T$ // rename initial nodes
triples = execSPARQL(wclause)
return(triples)

Fig. 8 Algorithm Knowledge Association Graph (KAG)

알고리즘 KAG는 모든 연관관계를 찾기 때문에 복잡도가 높다. 그러나 실무 사례에 적용해본 결과, 대부분 n=3 이하인 경우에 종료하며 MongoDB에서 SPARQL 질의 처리가 수행되어 성능에는 부담이 되지 않았다.

4.3.4. SPARQL Endpoint 구현

개발된 LOD 데이터 셋을 웹상에 개방, 공유하기 위해서는 SPARQL Endpoint가 필수적이다. 본 논문은 오픈소스 API인 dotNetRDF를 활용하여 SPARQL Endpoint를 구현하였다[16]. dotNetRDF는 SPARQL 1.1을 지원하고, 독립종단(SparqlRemoteEndpoint)과 연합종단(FederatedSparqlRemoteEndpoint)을 지원하므로 LOD의 SPARQL 질의 처리에 적합하다.

Ⅴ. i-Manager 적용 사례

한국 민족종교 지식지도 작성에 i-Manager를 활용하여 유용성을 검증하였다. 한국근대화 과정에서 민족의식을 고취하고 새로운 이상세계를 지향하는 다양한 민족종교가 태동하였다. 한국 민족종교는 상호간 관계가 복잡하고 인물, 사건, 종교, 법률 등 다양한 온톨로지 요소가 복잡하게 얽혀 있다. 또한, 이와 관련된 고문헌, 고전 간행물, 이미지, 사진, 각종 고문서 등과 같은 방대한 자료의 체계적 정리가 미흡한 상태이다. 이로 인하여 한국 민족종교에 관한 전반적 지식을 효과적으로 파악하는데 어려움이 있다. 한국 민족종교의 이러한 상황은 LOD 개발에 좋은 대상이며, LOD 지식지도는 한국 민족종교의 전반적 지식을 효과적으로 표출하는 도구가 될 것이다.

5.1. 온톨로지와 인스턴스의 편집 및 저장

한국 민족종교의 지식은 다양한 온톨로지 요소를 내포하고 있고 관련 정보자원이 방대하기 때문에, 도메인 전문가의 참여 개발이 필수적이다. 본 논문의 온톨로지는 종교 전문연구원이 Protégé로 개발하였다. 그림 9는 개발된 온톨로지로부터 생성된 LIS와 이에 대응하는 인스턴스 생성화면이다. 사용자가 친숙한 일반 데이터 입력 화면과 같으므로 손쉽게 인스턴스를 생성할 수 있다. 편집된 인스턴스는 검증을 거쳐 JSON-LD형으로 MongoDB에 저장된다.

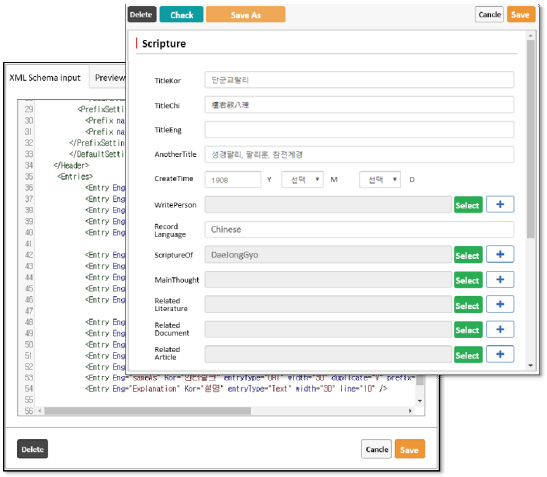

Fig. 9 LIS and its instance edit screen

5.2. 지식 그래프 생성

그림 10처럼 주어진 인스턴스에 대하여 모든 연관 인스턴스를 표현하는 것이 지식 그래프이다. 지식그래프의 각 노드를 지속적으로 확장하여 지식 공간을 탐색할 수 있다. 노드가 이미지, 사진, 문서 등의 정보자원과 관련이 있을 때는 해당 정보자원을 호출할 수 있다.

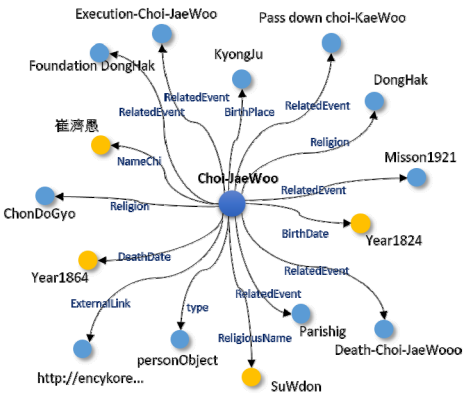

Fig. 10 Example of Knowledge Graph

5.3. 지식 연관 그래프 생성

지식 연관그래프는 그림 11처럼 지정된 두 인스턴스 간에 존재하는 모든 인스턴스를 보여준다. 두 인스턴스의 특성 비교, 공통점, 연계 인스턴스 등을 파악할 때 유용하다.

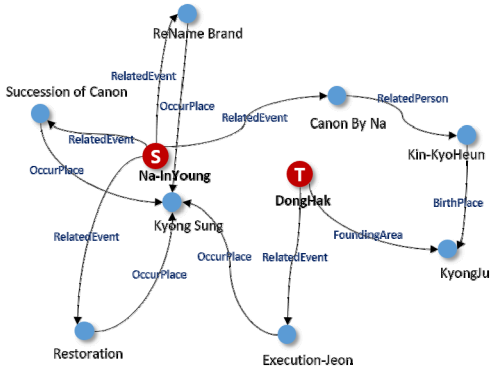

Fig. 11 Example of Knowledge Association Graph

5.4. SPARQL 질의 처리

많은 LOD시스템이 SPARQL Endpoint를 구현하고 있지 않지만, SPARQL Endpoint는 LOD 응용 서비스 개발, LOD 연합(federation)등에 중요한 요소이다.

i-Manager는 그림 12와 같이 사용자 또는 개발자의 SPARQL 질의 처리 기능을 제공한다. 질의 처리 결과는 RDF/XML, JSON, Turtle 등 다양한 형태로 제공되며, 추후 HTTP GET와 HTTP POST도 지원할 것이다.

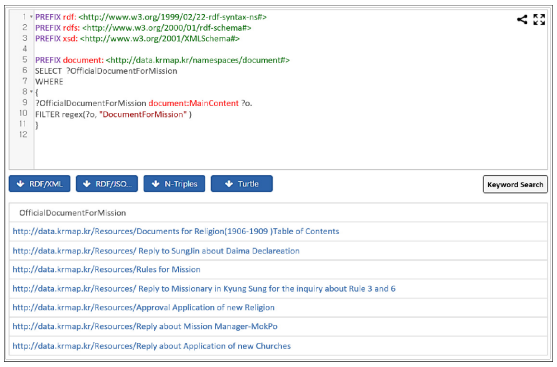

Fig. 12 SPARQL query processing in i-Manager

VI. 결 론

소셜 네트워크, 사물 인터넷, 빅 데이터 등 데이터 관련 정보기술의 획기적 발전으로 유용한 데이터를 용이하게 생성할 수 있게 되었다. 이러한 데이터는 정보시스템과 서비스의 인공지능화에 핵심적 역할을 하고 있다. 특히, 웹상에서 온톨로지 기반의 트리플 모델의 LOD는 시맨틱 웹(Semantic Web)을 실현하는 표준 플랫폼으로 정착되고 있다.

LOD의 핵심은 효과적인 데이터 셋 개발이다. 본 논문에서는 온톨로지와 인스턴스 계층을 분리하여, 인스턴스 개발에 전문화된 i-Manager를 구현하였다. i-Manager는 분리된 두 계층을 연결하는 인터페이스를 생성하여 NoSQL기반 MongoDB에 저장, 인스턴스의 관계를 시각화한 지식 그래프, 두 인스턴스간의 연관 상태를 표현하는 지식 연관그래프, SPARQL 질의 처리 등 인스턴스 개발에 필요한 다양한 기능을 제공한다. i-Manager는 LOD개발에 특화된 개발 환경으로 문화, 교육, 행정 등 다양한 분야의 LOD 데이터 셋 개발에 기여할 것이다. 온톨로지 계층과의 유연한 인터페이스, 네임스페이스 관리, SPARQL 질의 처리의 시각화 등의 기능 보완이 필요하다.

ACKNOWLEDGMENTS

This paper was supported by wonkwang university in 2017

REFERENCES

[1] F. M. Suchanek and G. Weikum, “Knowledge Bases in the Age of Big Data Analytics,” *Proceedings of the Very Large Data Base Endowment*, vol. 7, no. 13, pp. 1713-1714, Aug. 2014.

[2] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, “Quality Assessment for Linked Data: A Survey,” *Semantic Web*, vol. 7, no. 1, pp. 63-93, Jan. 2016.

[3] C. Bizer, “The emerging web of linked data,” I*EEE Intelligent Systems*, vol. 24, no. 5, pp. 8792, Sep, 2009.

[4] T. Davies and D. Edwards, “Emerging Implications of Open and Linked Data for Knowledge Sharing in Development”, *IDS Bulletin*, vol. 43, no. 5, pp. 117-127, Sep. 2012.

[5] W. Carrara, W. S. Chan, S. Fischer and E. Steenbergen, *Creating Value through Open Data: European Data Portal*, Belgium, EU:Digital Agenda for Europe, 2015.

[6] S. Auer, V. Bryl, and S. Tramp, *Linked Open Data - Creating Knowledge Out of Interlinked Data: Results of the LOD2 Project,* LNCS 8661, Heidelberg, Springer, 2014.

[7] M. L. Rösch and R. Heese, “Linked Data Authoring for Non-Experts,” in *WWW2009,* Madrid, Spain, pp 1-5, 2009.

[8] E. Klein, A. Gschwend and A. C. Neuroni, “Towards a Linked Data Publishing Methodology,” *International Conference for E-Democracy and Open Government (CeDEM)*, Lower Austria, pp. 188-196, 2016.

[9] F. Michel, J. Montagnat and C. Faron-Zucker, “A survey of RDB to RDF translation approaches and tools,” Ph. D. dissertation, Laboratoire d'Informatique, Signaux et Systemes de Sophia-Antipolis (I3S), France, 2014.

[10] P. Gocebe, O. Dikenelli and N. U. Kose "Bringing Agility into Linked Data Development," in *Proceeding of the 8th International Workshop About Linked Data On The Web(LDOW2015)*, Florence, Italy, pp. 1-10, 2015.

[11] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser and W. Hall, “Linked Open Government Data,” *IEEE Intelligent Systems*, vol. 27, no. 3, pp. 16-24, Jun. 2012.

[12] S. Auer, L. B¨uhmann, C. Dirschl, O. Erling, M. Hausenblas, R. Isele, J. Lehmann, M. Martin, P. N. Mendes, B. Nuffelen, C. Stadler, S. Tramp and H. Williams, “Managing the life-cycle of linked data with the LOD2 stack,” in *Proceeding of the 1th International Semantic Web Conference(ISWC 2012)*, Boston: MA, pp. 1-16, 2012.

[13] K. S. Yoshimura. (2016, August). Analysis of International Linked Data Survey for Implementers. *D ¯Lib Magazine* [Online]. 22(7/8). Available: http://www.dlib.org/dlib/july16/smith-yoshimura/07smith-yoshimura.html.

[14] D. Jackson and E. Kang, “Separation of concerns for dependable software design,” in *Proceedings of the FSE/SDP workshop on Future of software engineering research*. Santa Fe: NM, pp. 173-176, 2010.

[15] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space* , 1st ed. San Rafael, CA: Morgan & Claypool Pub., 2011.

[16] CodePlex Project Hosting for Open Source Software, dotNetRDF Project [Internet]. Available: https://dotnetrdf.codeplex.com/.

문희경(Hee-kyung Moon)

1993년 : 원광대학교 컴퓨터공학과 학사
1996년 : 원광대학교 컴퓨터공학과 석사
2016년 : 원광대학교 컴퓨터공학과 박사
2014년 ~ 현재 : 원광대학교 컴퓨터공학과 스마트기술연구소 실장
※관심분야 : 링크드오픈데이터, 시맨틱웹, 온톨로지, 그래프데이터베이스

한성국(Sung-kook Han)

1988년 : 인하대학교 전기공학과 박사
1984년 ~ 현재 : 원광대학교 컴퓨터공학과 교수
1990년 : University of Pennsyllvania 방문교수
2003년 : DERI 방문교수
2010년 : University of Innsbruck 방문교수
※관심분야 :인공지능, 자연어언처리, 링크드오픈데이터, 시맨틱웹, 온톨로지, 그래프데이터베이스