International Journal of Engineering & Technology, 7 (3.33) (2018) 225-228

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



LOD Development System for Medical Information Standard

Hee-kyung Moon[1], Sung-kook Han[2], Chang-ho An[3]*

[1, 2]Department of Computer Engimeering, College of Engineering, Wonkwang University, 460 Iksandae-ro, Iksan, Korea
[3]Department of Financial Information Engineering, Seokyeong University, 124 Seogyeong-ro, Seongbuk-gu, Seoul, Korea
**Corresponding author E-mail: choan@skuniv.ac.kr*

Abstract

This paper describes Linked Open Data(LOD) development system and its application of medical information standard as Observational Medical Outcomes Partnership(OMOP) Common Data Model(CDM). The OMOP CDM allows for the systematic analysis of disparate observational database in each hospital. This paper describes a LOD instance development system based on SII. It can generate the application-specified instance development system automatically. Therefore, we applied by medical information standard as OMOP CDM to LOD development system. As a result, it was confirmed that there is no problem in applying to the standardization of medical information using the LOD development system.

Keywords: *Medical Information Standard, Linked Open Data(LOD), RDF Schema, CDM, OHDSI.*

1. Introduction

Recently, research on standardization of medical information has been actively carried out. However, medical data have been constructed by adopting separate common data models with difficulties in standardization due to different data structures in hospitals [1, 2]. It is difficult to conduct multi-institutional research because the data structure and the data terms and code systems used by each institution are not standardized according to the purpose of utilization of medical information. In order to solve these problems, this paper proposes a method for applying medical information together with high quality data set by mapping with Linked Open Data (LOD) based common data model of Observational Health Data Sciences and Informatics (OHDSI). The CDM can accommodate both administrative claims and Electronic Health Records (EHR), allowing users to generate evidence from a wide variety of sources. It would also support collaborative research across data sources both within and outside the Korea, in addition to being manageable for data owners and useful for data users [3, 4]. In Korea, medical computerization has been carried out for the purpose of hospital management. With the spread of healthcare IT, medical information of hospitals was quickly collected and digitized on a digital platform. The application area of analysis including clinical decision support based on recently collected medical big data is expanding. In addition, personal medical information, which is individually managed for each medical institution, is being converted into a system capable of exchanging information between medical institutions. But, these information systems have been developed at widely differing points in time, by using different development paradigms as well as different software and hardware platforms. For this reason, it is difficult to exchange medical information or conduct multi-institutional research. A complementary study of CDM and LOD is required for solving these problems. LOD is a new Semantic Web set of best practice of standards to publish and link heterogeneous data. It can be applied to the representation and management of public level health data to alleviate such challenges [5, 6, 7].

In this paper, we present the hierarchical separation based LOD development system and focus on the data set design of medical information by standard as OHDSI.
The composition of this paper can be seen as follows. Section 2 shows the LOD Development system environments. Section 3 introduces applied medical information standard. The final Section summarizes this paper and discusses future research.

2. LOD Development System Environments

This section describes the architecture of LOD instance development system. A typical use-case is also presented to show the effectiveness of the layer separation approach [8].

2.1. Structure of LOD Instance Development

In general, LOD sets consist of two elements: ontologies that define the common vocabularies by means of Resource Description Framework (RDF) and the instances that represent the data of information resources [9, 10]. The domain ontology is usually developed by using the conventional ontology editors. Schema-Instance Interface (SII) is generated from the domain ontology represented by RDF and RDF Schema. The architecture of LOD instance development system *i-Manager* has three main functionalities as shown in Fig. 1. *i-Manager* creates the instance editing environment as the front end of the instance development. As the back end of the system, *i-Manager* provides many core functions such as instance editing and store, validation, visualization and navigation. Since the back end functions are implemented in add-on modules, additional subsystems such as SPARQL endpoint can be easily integrated.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

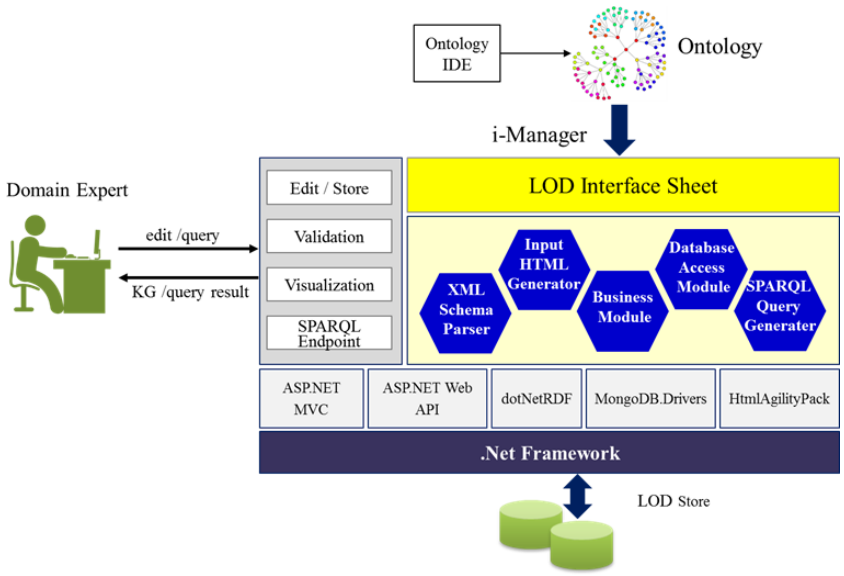

Fig. 1: Structure of LOD Instance Development System

2.2. Components of the LOD Instance Development system

- **LOD Interface Sheet (LIS):** Configuration file of *i-Manager*, semi-automatically generated from related ontology. The XML Schema parser extracts the classes and attributes corresponding to the subject and object of the triple in the input ontology schema, creates the basic structure of the LIS, and uses the *entry-Type* and *entry-Attr* to suit the development environment of the user. Can be customized.
- **Edit / Save:** The created LIS is converted into HTML and creates an instance editing screen in the web browser. Input HTML Generator dynamically creates an HTML edit instance screen based on the schema entered in XML form and converts the edited user into JSON-LD format, which is a standard of LOD by Database Access Module. In this paper, we use MongoDB, a typical NoSQL, as LOD repository.
- **Validation:** The edited instance can be verified by verifying that the instances are included in the ontology actually built and verifying their conformity with the verifier added as add-on.
- **Visualization:** Knowledge graph that visualizes the whole structure of LOD triple and knowledge association graph that expresses association between two concept nodes is supported.
- **SPARQL Endpoint:** To share the developed LOD dataset on the web, SPARQL Query Generator is used to dynamically generate the user's SPARQL query and query the LOD repository through dotNetRDF. It provides access to the individual information resources of LOD, which is the result of publishing through SPARQL Endpoint, and provides the results of the query in various formats such as RDF / XML, JSON, N-Triples and Turtle.

The main server modules that make up the *i-Manager* system architecture are the ASP.NET MVC and ASP.NET Web APIs of the .NET Framework, the software framework of Microsoft Windows [11]. ASP.NET MVC is a web application framework developed by Microsoft that implements the model-view-controller (MVC) pattern. With ASP.NET MVC, software developers can create Web applications consisting of three roles: model, view, and controller [12]. The ASP.NET Web API is a framework that makes it easy to build HTTP services that connect to a variety of clients, including browsers and mobile devices, and is an ideal platform for building RESTful applications on the .NET Framework. The dotNetRDF module parses, manages, and queries RDF documents in a .Net Framework environment and queries the LOD repository using SPARQL [13].

I-Manager handles all interactions with MongDB through the MongDB.Driver module. MongDB.Driver is a framework provided by MongoDB that accesses and queries MongoDB in a .Net Framework environment. The HtmlAgilityPack module is a .NET code library that can parse HTML files. It is an HTML parser that writes the read / write DOM and supports generic XPATH or XSLT [14]. *i-Manager* has a structure that is specialized for LOD instance development.

3. Applying medical information

In general, LOD datasets development usually focuses on the generation of the instances. Hence, the development methodology specialized for the instance generation is important. This section describes the principle and the approach to implement LOD instances development system as medical information.

The knowledge of medical information is complex and interrelated, and based on the patient, it can be used as a tool for dealing with observation period, specimen, death, visit occurrence information, surgical / treatment occurrence information, drug exposure information, device / material exposure information, Notes, and observational information; and ontology elements related to healthcare costs, extraction factors, and standard terminology [15].

Although RDF, which represents the structure and meaning of the LOD data set of medical information, defines the separation of viewpoints, the existing LOD development has not utilized separation of this viewpoint, so trial and error of LOD development process and data set verification have problems.

Fig. 2 shows the OMOP Common data model structure of medical information standards to represent the relevant domains, the CDM contains the following 39 tables [16]. The CDM is designed to support the conduct of research to identify and evaluate associations between interventions (drug exposure, procedures, healthcare policy changes etc.) and outcomes caused by these interventions (condition occurrences, procedures, drug exposure etc.). Outcomes can be efficacious (benefit) or adverse (safety risk). Often times, specific patient cohorts (e.g., those taking a certain drug or suffering from a certain disease) may be defined for treatments or outcomes, using clinical events (diagnoses, observations, procedures, etc.) that occur in predefined temporal relationships to each other. The CDM, combined with its standardized content (via the Standardized Vocabularies), will ensure that research methods can be systematically applied to produce meaningfully comparable and reproducible results.

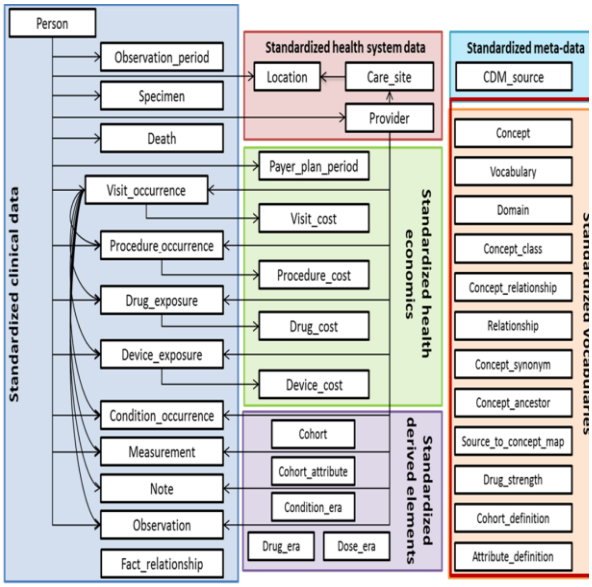

Fig. 2: OMOP Common Data Model Structure

The CDM defines table structures for each of the data in a Person-centric. The tables have foreign keys into the Person table and a date. This allows for a longitudinal view on all the healthcare-relevant events. The exceptions from this rule are the standardized health system data tables. Providers carrying out health care are linked to many of the events as well. Both are linked to healthcare organizations. Accordingly, data is extracted from the hospital information system (EMR, OCS), and the data is converted into a common data model.

A hierarchical instance development system designed and developed in this study is used to construct a medical information data set and create a knowledge map by selecting a medical information domain that has various complex ontologies. We demonstrate the usefulness of a specialized system for system modeling based on schema-instance hierarchy separation and hierarchical LOD instance development using *i-Manager*.

In Fig. 3, RDF and RDF Schema represent ontology information representing domain knowledge, and Instance Layer represents actual data as medical information. The event, disease, patient, and hospital are representative classes of medical information domain ontologies and are based on the Common Data Model (CDM) of OMOP. Patient instances were expressed as 'P1', 'P2', 'P3', hospital instances were expressed before and after, and the diseases were obesity, metabolic syndrome and heart disease, diabetes and rehabilitation. (BMI), waist circumference (WC), waist / height (WHtR), triglyceride (TG), blood pressure (Blood pressure), exercise, diet, heart rate (HR) and so on. Based on this, RDF and RDF Schema are provided as a basis for implementing the medical information domain ontology on the Web, and the LOD instance is mapped to the actual clinical data operated on this basis.

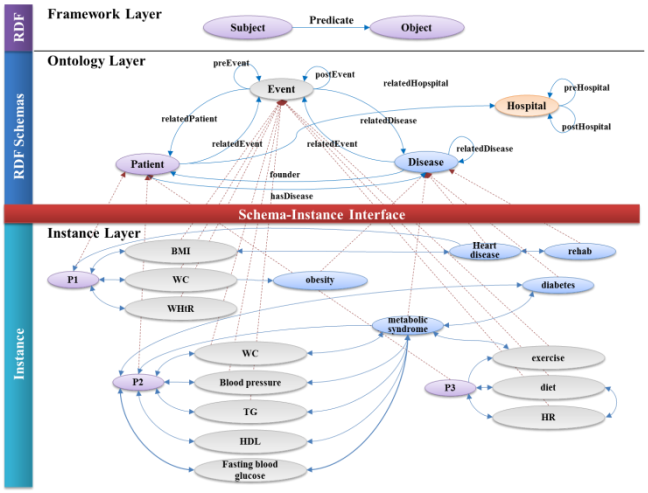

Fig. 3: Hierarchical structure of medical information LOD development

A hierarchical, proprietary instance creation system for LOD can be realized by defining the interface between the RDF / RDF Schema ontology layer and the instance layer as shown in Fig. 3.

4. Conclusion

In this paper, to use medical information data more actively and increase the value of medical information, we propose a method for knowledge of medical fields. Based on this, we propose a method for knowledge of medical information linked with CDM of OHDSI based on hierarchical LOD development environment. To achieve this, we describe the existing hierarchical LOD development system environment and show the LOD data set structure of medical information based on it.

The proposed system was applied to medical field and confirmed its effectiveness. However, it is not enough to show the result of using the constructed medical information. Since existing CDM data modeling is an RDB, it is difficult to convert to LOD. However, the proposed system provides a clue to solve the problems of existing medical information system. This study is an attempt to construct LOD - based data set of standardized medical information. Future research is as follows.

We will develop an effective medical information data set based on LOD, and conduct a performance evaluation study and a medical big data analysis study on the LOD dataset developed together with it. It will be helpful for the design of a multicentre validation study and providing the meaningful medical criteria.

References

[1] Y. Zhang, M. Qiu & C.W. Tsai (2017), "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data", *IEEE Systems Journal*, Vol. 11, No. 1, pp. 88-95.

[2] E. M. Rac-Albu, V. Ciobanu, M. Rac-Albu & N. Popescu (2016), "Interoperability of Medical Data Through e-Health Service in Romania", *Proceedings of the 7th International Conference on Exploring Services Science (IESS)*, pp. 683-692.

[3] OHDSI (Observational Health Data Sciences and Informatics), http://ohdsi.org.

[4] G. Hripcsak, J. D. Duke, N. H. Shah, C. G. Reich, C, V. Huser, M. J. Schuemie, ... & J. Van Der Lei, (2015), "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers", *Studies in health technology and informatics*, Vol. 216, pp. 574.

[5] B. Tilahun, T. Kauppinen, C. Keßler & F. Fritz (2014), "Design and Development of a Linked Open Data-Based Health Information Representation and Visualization System: Potentials and Preliminary Evaluation", *Journal of Medical Internet Research*, Vol. 16, No. 10, pp. 1-31.

[6] H.K. Moon, Z. Zhao, J.T. Choi & S.K. Han (2017), "A New LOD Development System by Schema-Instance Layer Separation", *In-*

ternational Journal of Applied Engineering Research, Vol. 12, No. 24, pp. 14931-14937.
[7] X. Zenuni, B. Raufi, F. Ismaili & J. Ajdari, (2015). "State of the art of semantic web for healthcare", *Procedia-Social and Behavioral Sciences*, Vol. 195, pp. 1990-1998.
[8] E. Klein, A. Gschwend, & A. C. Neuroni, (2016), "Towards a Linked Data Publishing Methodology", *Proceedings of the International Conference for E-Democracy and Open Government (CeDEM)*, pp. 188-196.
[9] N. Senthilselvan, V. Subramaniyaswamy & KR. Sekar (2017), "Information retrieval for biomedicine applications through linked open data based optimization model", *Biomedical Research*, Vol. 28, No, 18, pp. 7867-7870.
[10] C. Bizer (2009), "The emerging web of linked data", *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 87-92.
[11] Wikipedia, .NET Framework, available online: https://en.wikipedia.org/wiki/.NET_Framework.
[12] Wikipedia, ASP.NET MVC, available online: https://en.wikipedia.org/wiki/ASP.NET_MVC.
[13] CodePlex, Project Hosting for Open Source Software, dotNetRDF Project, available online: https://dotnetrdf.codeplex.com/.
[14] Html Agility Pack (HAP), What's Html Agility Pack?, available online: http://html-agility-pack.net/?z=codeplex.
[15] V. Zamborlini, R. Hoekstra, M. Silveira, C. Pruski, A. Teije & F. Harmelen, (2016), "Generalizing the detection of clinical guideline interactions enhanced with LOD", *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC)*, pp. 360-386.
[16] OHDSI, Data Standardization, available online: https://www.ohdsi.org/data-standardization/