Journal of KIIT. Vol. 18, No. 8, pp. 31-39, Aug. 31, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2020.18.8.31

RDF Data Management and SPARQL Query for Patent Information

Zaslyana Mozahker*[1], Jeong Rae Kim*[2], Ok Keun Shin**[1], and Hyu Chan Park**[2]

Abstract

Resource Description Framework (RDF) is a general framework to model and describe information within the Web. To cope with the growing size of RDF information, an efficient management and query system is required. SPARQL Protocol and RDF Query Language (SPARQL) is a well-known RDF query language to retrieve and manipulate data stored in RDF format. This paper applies these techniques to manage patent information systematically . To achieve this, the structure of patent information is first defined by analyzing several patent information, especially KIPRIS. And then, the RDF schema is designed to represent the structure of patent information. A prototype system was developed and tested to show RDF and SPARQL can be consistently applied to manage patent information.

요 약

RDF는 웹상에서 정보를 개념적으로 표현하고 모델링하기 위한 일반적인 방법을 제공한다. 이러한 RDF 정보의 양이 증가함에 따라 효율적으로 저장하고 질의하기 위한 시스템이 요구되고 있다. SPARQL은 이러한 RDF 형식으로 저장되어 있는 정보를 검색하고 관리하기 위한 질의언어이다. 본 논문에서는 이러한 기술을 특허정보의 체계적인 관리에 적용하고자 한다. 이를 위하여, 우선 KIPRIS와 같은 다양한 특허정보를 분석하여 특허정보의 공통적인 구조를 정의한다. 그리고, 이러한 특허정보 구조를 표현하기에 적합한 RDF 스키마를 설계한다. 프로토타입 시스템의 구현과 테스트를 통하여 RDF와 SPARQL이 특허정보의 일관성 있는 관리에 적용할 수 있음을 보인다.

Keywords
resource description framework (RDF), SPARQL protocol and RDF query language (SPARQL), patent information

Ⅰ. Introduction

Semantic Web is an extension of the World Wide Web to make Internet data machine-readable. With Semantic Web, Internet data can be searched and interpreted, and then shared and reused between applications and organizations. Internet users can also build vocabularies, create data stores, and write rules

* Dept. of Computer Engineering, Korea Maritime and Ocean University
- ORCID[1]: https://orcid.org/0000-0002-2417-5118
- ORCID[2]: https://orcid.org/0000-0001-9333-1981
** Professor, Dept. of Computer Engineering, Korea Maritime and Ocean University
- ORCID[1]: https://orcid.org/0000-0003-0171-5764
- ORCID[2]: https://orcid.org/0000-0003-3166-9287

· Received: Jul. 07, 2020, Revised: Aug. 04, 2020, Accepted: Aug. 07, 2020
· Corresponding Author: Hyu Chan Park
Dept. of Computer Engineering, Korea Maritime and Ocean University
Tel.: +82-51-410-4573, Email: hcpark@kmou.ac.kr

for handling data on the Web. Semantic Web is closely related to the Linked Open Data (LOD). It is interlinked with other data, so it becomes more useful through semantic queries. To empower the Semantic Web and LOD, technologies such as RDF and SPARQL may be applied[1].

RDF is a general framework to be applied for the description or modeling of information. It can be implemented with web resources by using plentiful notations. Various techniques have been developed to map RDF from relational data and other formats[2]. SPARQL is a query language for RDF that can store and retrieve information that can be expressed in the form of labelled graph. They include simple unstructured documents, semi-structured markup languages, structured databases, and so on.

This paper proposes a framework to apply RDF and SPARQL technologies to the management of patent information. To achieve this, patent information from various sources is first analyzed, and then defined the structure of patent information. With this structure, the RDF schema of patent information is designed.

This paper also proposes SPARQL queries to retrieve specific patent information from RDF storage. To show the applicability of the proposed framework for the management of patent information, a prototype system was also developed and tested.

II. Related Works

2.1 RDF and RDF Schema

RDF is a metadata model proposed by World Wide Web Consortium (W3C). It has been used for conceptual description and modeling of information on web resources. It supports a variety of notations and data serializations. With RDF, semantic information on the Web can be processed by machines as well as by humans. While there are many other standards, RDF may be the simplest and most efficient to handle data and relationships between data.

To define vocabularies for RDF data, RDF Schema (RDFS) has also been developed by W3C. It is composed of various classes with certain properties for the structure of RDF data[3][4].

RDF data is expressed in three terms of subject, predicate, and object about resources, called triples. The subject and object denote resources, and the predicate expresses a relation between the subject and the object. They can be stored in and retrieved from a triplestore with the query language SPARQL[5].

2.2 SPARQL

SPARQL is a language to be used to query RDF triples and merge results from multiple data sources. It also enables Linked Open Data for the Semantic Web and enriches information by linking it to other semantic resources. Thus, data can be merged, shared, and reused in a more meaningful way[6].

SPARQL is analogous to SQL used to create, store, and retrieve structured data. SQL is suitable to access tables in a relational database, but SPARQL can access RDF triples. Although SPARQL was developed to combine diverse sources of data, it can be used to access relational data as well. Moreover, the SPARQL query may be constructed across a range of datasets so long as they are presented as a directed labelled graph. The results of SPARQL queries can be in the form of RDF triples[7].

```
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix vs: <https://portmis.go.kr/>     ①

SELECT ?ship ?port     ②
WHERE {
     ?x vs:ShipName ?Ship .
     FILTER regex(str(?port), 'BUSAN')     ③
     ?x vs:PortName ?port .
}
```

Fig. 1. Structure of SPARQL query

Fig. 1 describes an example of a SPARQL query. The Block ① is PREFIX definition to specify URI of the related sites in the query. The Block ② is SELECT clause to retrieve variables. The Block ③ is WHERE clause to match with the triple patterns.

2.3 RDF Triplestore

RDF triplestore is a special-purpose storage and retrieval system for the management of RDF triples. Most of the triplestores support the standard SPARQL as a query language. Users can define their own query patterns by combining provided primitives. SPARQL queries can be parsed and transformed with the dataset.

This paper adopts Apache Jena for the triplestore and SPARQL. Apache Jena provides API to store and extract data from the RDF triplestore. It also supports access control at the level of server and endpoint within a dataset[8][9].

III. Patent Information Structure

A Patent is an intellectual property that gives its owner the right to exclude others from making and using the invention for years. Once a patent is licensed, only the individual or organization that is permitted can produce and use the patent. Patent owners can sell the patent and process registered technological issues. There are three classes of patent, that is design, utility, and plant patent. This paper focuses on the utility patent, which covers the machine, process, product, or combination of these three[10][11].

Although traditional keyword searches on the patent datasets can get back useful information, it might be insufficient, particularly in the engineering domain. To cope with the limitation, this paper adopts a semantic web with RDF and SPARQL framework. Besides keyword searching and linking one or more properties, the framework can determine semantic relations among patents.

3.1 Patent Information Structure in General

This paper analyzed patent information structure to build an efficient storage system and then to alleviate the burden of finding related patents for a certain technical problem.

Table 1 shows the patent information structure in general. It consists of ten main categories, which are certificate, registration, application, references, applicant, assignee, inventor, examiner, general information, and classification. In each category, detailed attributes have been assigned accordingly.

Table 1. Patent information structure in general

Group	Category
group1	Certificate
	Registration
group2	Application
	References
	Applicant
group3	Inventor
	Assignee
group4	Classification
	General
	Examiner

Table 2 shows two of the ten categories that are certificate and registration category. The certificate category holds the certificates of correction, re-examination, PTAB trial, and supplemental exam. Registration category includes the international details of the registration number, registration date, publication date, and filing date.

Table 2. Certificate and registration

Category	Attribute
Certificate	Cert. of correction
	Re-examination cert.
	PTAB trial cert.
	Supplemental exam cert.
Registration	International Reg. No
	International Reg. date
	International Reg. Pub. date
	Hague international filing date

Table 3 shows the next three categories, that are application, references, and applicant category. They include application type, application date, other references, reference by, foreign reference, applicant type, country, state, city, and name.

Table 3. Application, references and applicant

Category	Attribute
Application	Type
	Date
References	Other Ref.
	References By
	Foreign Ref.
Applicant	Type
	Name
	Country
	State
	City

Table 4 shows two other categories that are the inventor and assignee category. They have the same attributes, which are name, country, state, and city, respectively. The inventor can be either one or more persons who have developed the invention. Whereas, the assignee is one with the property right to the patent.

Table 4. Inventor and assignee

Category	Attribute
Inventor	Name
	Country
	State
	City
Assignee	Name
	Country
	State
	City

Table 5 shows the last three categories that are classification, general, and examiner category. Classification category lists current CPC, current CPC class, and international classification. General category includes patent number, issue date, reissue data, title, claim(s), abstract, description, prior filing date, attorney/ agent, and other attributes. The examiner category is divided into primary examiner and assistant examiner.

3.2 Patent Information Structure in KIPRIS

This paper also analyzed patent information structure from the Korea Intellectual Property Rights Information Service (KIPRIS). KIPRIS is a search site for intellectual property data in Korea.

Table 5. Classification, general and examiner

Category	Attribute
Classification	Current CPC
	Current CPC class
	International
General	Patent No.
	Issue date
	Reissue date
	Title
	Claim(s)
	Abstract
	Description
	Prior. filing date
	Attorney/Agent
	Gov. interest
	Patent family ID
	Related US application date
	PCT filing date
	Prior publication document date
	PCT information
	Foreign priority
	Related application filing date
	Re. patent application filing date
	Patent case information
Examiner	Primary
	Assistant

Table 6. Patent information structure in KIPRIS

Category	Attribute
General	Reg. No
	Invention Title
	Abstract
	Priority info
	IPC
	Status
Applicant	Applicant no.
	First name
	Last name
	Country
Inventor	Inventor no.
	First name
	Last name
	Country
Agent	Agent no.
	Organization name

Table 6 shows specific components of patent information within KIPRIS. It consists of an agent, applicant, inventor, and other attributes.

IV. RDF Schema Design

Fig. 2 depicts the RDF schema for the patent information structure of KIPRIS. *Patent* class has three subclasses of *Agent*, *Applicant* and *Inventor*, and two properties of *isUndertakenBy* and *hasInvention*. *Agent* then has two other resources, that are *Organization name* and *Agent no*. Domain and range property *isUndertakenBy* is connecting the classes *Patent* and *Agent*. Other resources are also connected in s similar manner.

Fig. 3 depicts a part of the RDF graph, where the applicant *STX Offshore & Shipbuilding Co. Ltd.* has detailed information. Table 7 explains the RDF graph in terms of triples, where the subject *Applicant* has a predicate *hasApplicantionNo* and an object *1020100060169*. Fig. 4 lists the RDF syntax to express corresponding applicant information. Generally, the formats of the RDF graph, triples, and syntax have similar characteristics.

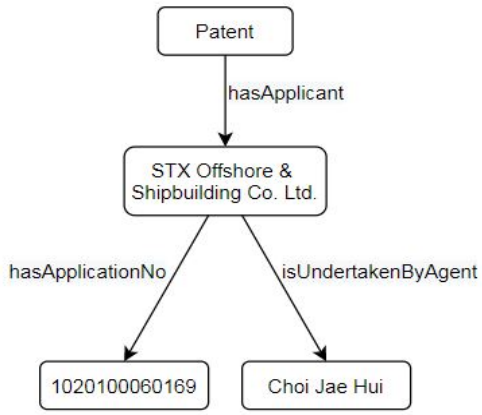


Fig. 3. RDF graph of applicant information

Table 7. RDF triples of applicant information

Subject	Predicate	Object
Patent	hasApplicant	STX Offshore & Shipbuilding Co. Ltd.
Applicant	isUndertakenByAgent	Choi Jae Hui
Applicant	hasApplicationNo	1020100060169

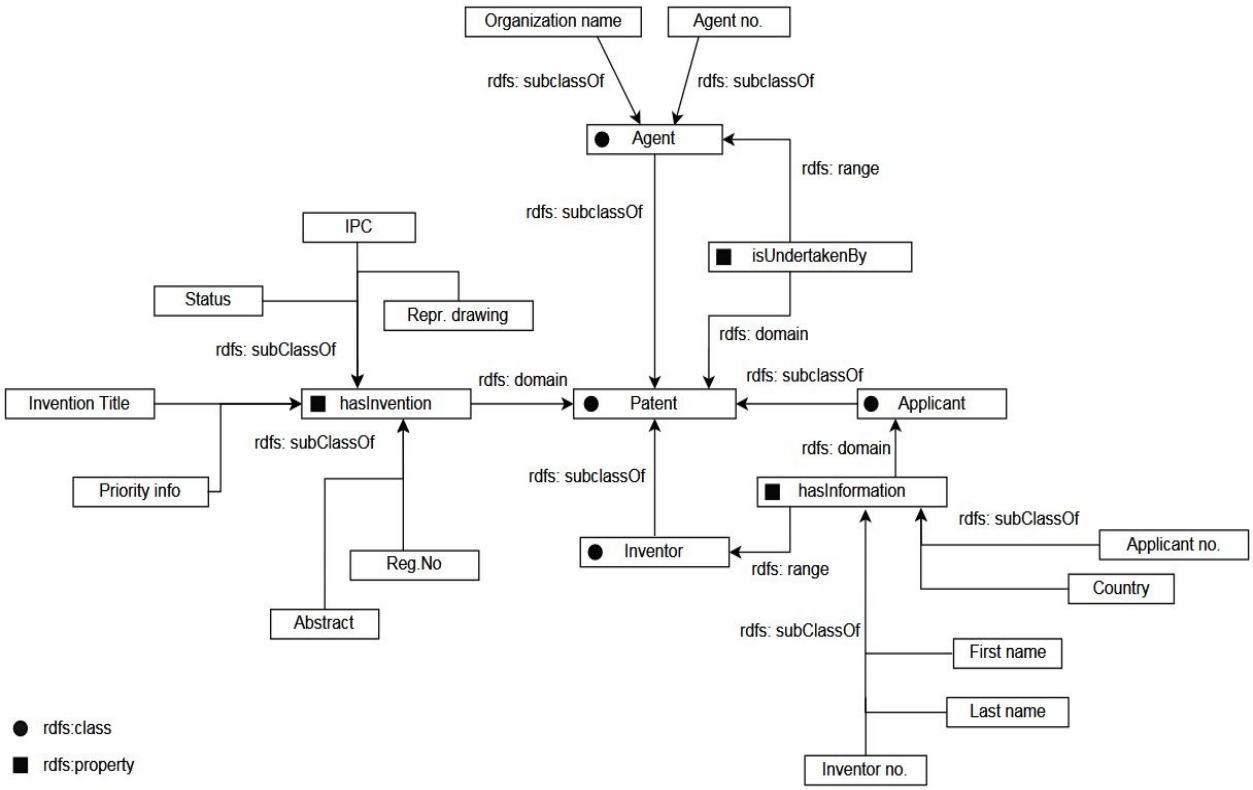


Fig. 2. RDF Schema for patent information in KIPRIS

A sample RDF syntax of applicant's information

```
<?xml version="1.0"?>
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:pt="http://eng.kipris.or.kr/patent#">

<rdf:Description ID="Patent">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf
  rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
</rdf:Description>

<rdf:Description ID="Applicant">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Patent"/>
</rdf:Description>

<rdf:Description ID="STX Offshore & Shipbuilding Co. Ltd.">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Applicant"/>
</rdf:Description>

<rdf:Description ID="Choi Jae hui">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Applicant"/>
</rdf:Description>

<rdf:Description ID="Application No">
 <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Applicant"/>
 <rdfs:subClassOf rdf:resource="#Agent"/>
</rdf:Description>

</rdf:RDF>
```

Fig. 4. RDF syntax of applicant information

V. Implementation and Testing

5.1 System Architecture

An architecture of RDF storage and query system is shown in Fig. 5. User can first construct its query and then accesses SPARQL server to process it. The system then processes the query conditions to match the datasets stored in the RDF storage. Next, the matched datasets are retrieved and analyzed thoroughly. Lastly, the result is returned to the user.

The illustration in Fig. 6 describes how uncategorized raw data are analyzed, converted into RDF format, and finally uploaded in the RDF storage.

Fig. 7 shows the process of the SPARQL query to retrieve the matched result. First of all, the SPARQL server has to be run in the background. The construction of queries needs to be done in order to retrieve data from RDF storage.

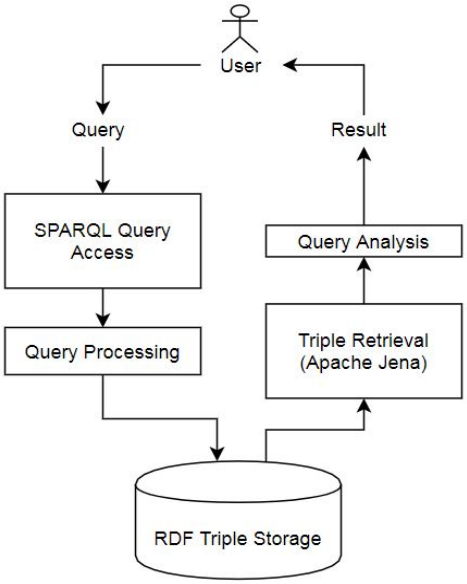


Fig. 5. Architecture of RDF storage and query system

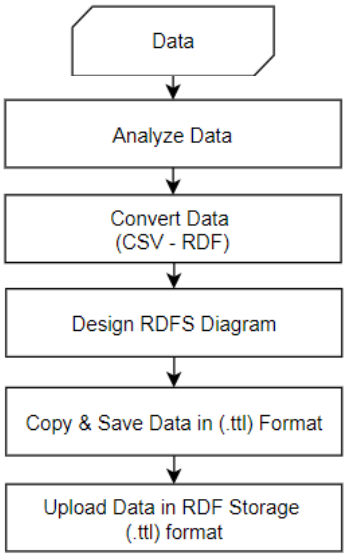


Fig. 6. Process of RDF storage

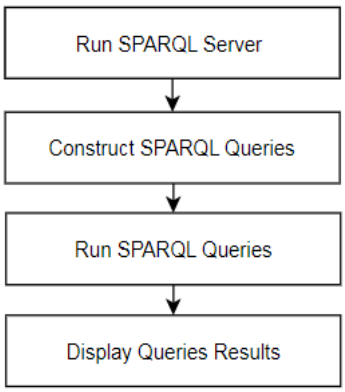


Fig. 7. Process of RDF query

5.2 SPARQL Query

Fig. 8 and Fig. 9 show an example of a SPARQL query and the retrieved result, respectively. Retrieving information from RDF storage is done with the SPARQL query within the SPARQL server.

Query. Patent Data based on KIPRIS

```
# Define the link of the data as prefix
PREFIX pt: <http://eng.kipris.or.kr/enghome/main.jsp#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

# Construct query CONSTRUCT define the tables, FROM define the file, WHERE define to
bind each variable
CONSTRUCT {
  ?URI a ex:Patent;
    pt:reprDrawing ?ReprDrawing;
    pt:applicationNo ?ApplicationNo;
    pt:titleOfInvention ?TitleOfInvention;
    pt:status ?Status;
    pt:ipc ?IPC;
    pt:applicant ?Applicant;
    pt:agent ?Agent;
    pt:inventor ?Inventor;
    pt:applicationDate ?ApplicationDate;
    pt:registrationNo ?RegistrationNo;
    pt:registrationDate ?RegistrationDate;
    pt:unexPubNo ?UnexPubNo;
    pt:unexPubDate ?UnexPubDate;
    pt:abstract ?Abstract;
}
FROM <file:patents.csv>
WHERE {
  BIND
URI(CONCAT('http://eng.kipris.or.kr/enghome/main.jsp/', ?Appl
icationNo)) AS ?URI)
  BIND (xsd:integer(?ApplicationNo) AS ?appNo)
  BIND (xsd:integer(?RegistrationNo) AS ?regNo)
  BIND (xsd:integer(?UnexPubNo) AS ?upNo)
}
```

Fig. 8. SPARQL query for patent information

Fig. 9. Query result for applicant information

5.3 User Interfaces

The followings are user interfaces that demonstrate the capabilities of the prototype system. Fig. 10 shows the datasets page that lists patent files in Turtle's format (.ttl) on the RDF triplestore.

Fig. 11 shows the query page where the SPARQL query can be constructed, and later the input query results will be displayed.

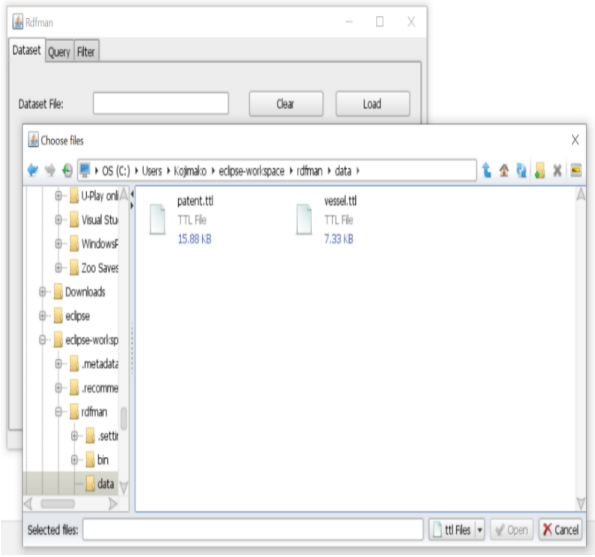


Fig. 10. Datasets page

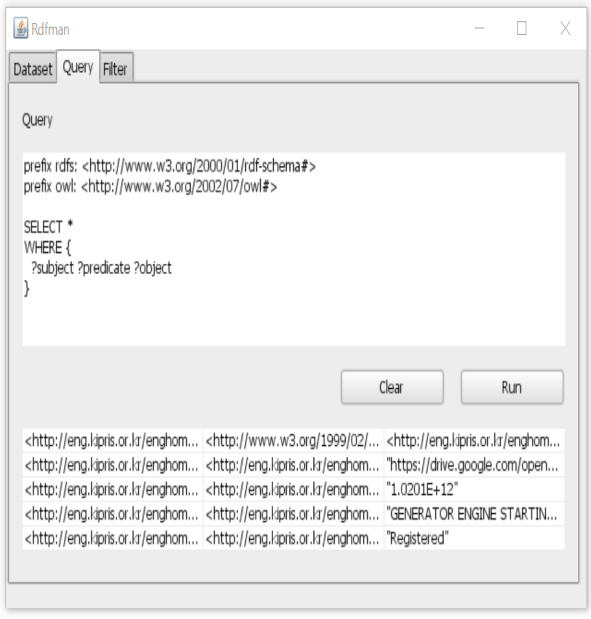


Fig. 11. Query page

Ⅵ. Conclusion

This paper proposed a framework to manage patent information by using RDF and SPARQL. In the framework, the structure of patent information was defined, and then the RDF schema was designed to represent the patent information consistently. With the framework, patent information can be represented in the form of RDF triples, and then stored in the RDF system. The stored patent information can be efficiently queried and retrieved by using SPARQL.

To show the possibility of the proposed framework, a prototype system was implemented. The implementation showed that patent information can be adequately managed. Although the proposed framework was tested through a prototype system, it needs to be verified through more application domains in the future.

References

[1] L. Lapeyra, "Introduction to the Semantic Web and Linked Data", 2016. [Online]. Available: https:// dlis.hypotheses.org/788

[2] RDF and SPARQL: Using Semantic Web Technology to Integrate the World's Data, 2007. [Online]. Available: https://www.w3.org/2007/03/VLDB

[3] D. Brickley, R. V. Guha, and A. Layman, "Resource Description Framework (RDF) Schemas", 1998. [Online]. Available: https://www.w3.org/TR/1998/WD-rdf-schema19980409

[4] D. Brickley and R. V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation", 2004. [Online]. Available: https://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[5] L. Curé and G. Blin (Eds.), "RDF Database Systems Triples Storage and SPARQL Query Processing", RDF Data Management, pp. 24-25, 2015.

[6] RDF and SPARQL: Using Semantic Web Technology to Integrate the World's Data, 2007. [Online]. Available: https://www.w3.org/2007/03/VLDB/

[7] SPARQL vs SQL. [Online]. Available: https://www.cambridgesemantics.com/blog/semantic-university/learnsparql/sparql-vs-sql/

[8] E. Jimenez and E. L. Goodman, "Triangle Finding: How Graph Theory Can Help the Semantic Web", Joint Workshop on Scalable and High Performance Semantic Web Systems, Boston, USA, pp. 45-58, Nov. 2012.

[9] E. Gayo, E. Prud'hommeaux, I. Boneva, and D. Kontokostas, "Validating RDF Data", 2018. [Online]. Available: http://book.validatingrdf.com/

[10] Patents Definition – What is Patents. [Online]. Available: https://www.shopify.com/encyclopedia/patents.

[11] PARR, Intellectual Property, Valuation, Exploitation, and Infringement Damages: 2019 cumulative, JOHN WILEY & Sons.

Authors

Zaslyana Mozahker

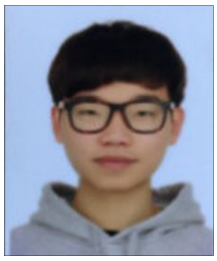
2018 : B.S. degree, Management and Science University, Malaysia

2020 : M.S. degree, Department of Computer Engineering, Korea Maritime and Ocean University

2020 ~ present : Online researcher

Research interests : Semantic Web, Database, Marine Information

Jeong Rae Kim



2018 : B.S. degree, Korea Maritime and Ocean University
2020 : M.S. degree, Department of Computer Engineering, Korea Maritime and Ocean University
2020 ~ present : Online researcher
Research interests : Big Data, Database, Marine Information

Ok Keun Shin

1983 : M.S. degree, Busan University
2005 : Ph.D. degree, Universite de Franche-Comte
1995 ~ present : Professor, Korea Maritime and Ocean University
Research interests : Signal processing, Embedded system

Hyu Chan Park

1987 : M.S. degree, Computer Engineering, KAIST
1995 : Ph.D. degree, Computer Engineering, KAIST
1997 ~ present : Professor, Korea Maritime and Ocean University
Research interests : Database, Data Mining, Big Data, Marine Information