

RDF와 SPARQL을 이용한 선박 정보의 관리

Zaslyana Mozahker¹ · 신 옥 근² · 박 휴 찬^{2*}

¹한국해양대학교 대학원 컴퓨터공학과 석사

²한국해양대학교 해사IT공학부 교수

Management of Vessel Information by using RDF and SPARQL

Zaslyana Mozahker¹ · Ok-Keun Shin² · Hyu-Chan Park^{2*}

¹Master, Department of Computer Engineering, Korea Maritime and Ocean University, Busan 49112, Korea

²Professor, Division of Marine IT Engineering, Korea Maritime and Ocean University, Busan 49112, Korea

[요 약]

RDF 및 RDF 스키마는 웹상에서 정보를 개념적으로 표현하고 모델링하기 위한 일반적인 방법을 제공한다. 이러한 RDF 정보의 양이 증가함에 따라 효율적으로 저장하고 질의하기 위한 시스템이 요구되고 있다. RDF Triplestore는 RDF 정보를 저장하는 특수한 데이터베이스이다. SPARQL은 이러한 RDF Triplestore에 저장되어 있는 정보를 검색하고 관리하기 위한 질의언어이다. 본 논문에서는 이러한 기술을 선박 정보의 체계적인 관리에 적용하였다. 이를 위하여, 우선 PORT-MIS와 같은 다양한 선박 정보를 분석하여 선박 정보의 공통적인 구조를 정의하였다. 그리고, 이러한 선박정보의 구조를 표현하기에 적합한 RDF 스키마를 설계하였다. 프로토타입 시스템을 RDF Triplestore상에서 구현하였으며 테스트를 통하여 선박 정보의 일관성 있는 관리에 적용할 수 있음을 보였다.

[Abstract]

Resource Description Framework (RDF) and RDF Schema are general methods for the conceptual description and modeling of information within the Web. The increasing size of RDF information requires an efficient storage and query system. RDF Triplestore is a special database for the storage of RDF data. SPARQL Protocol and RDF Query Language (SPARQL) is a well-known RDF query language to retrieve and manipulate data stored in RDF Triplestore. This paper applies these techniques to systematically manage vessel information. To achieve this, the structure of vessel information is first defined by analyzing several vessel information, especially PORT-MIS. And then RDF schema is designed to represent the structure of vessel information. A prototype system was developed on the RDF Triplestore and then tested to show its applicability for the management of vessel information.

색인어 : RDF, SPARQL, RDF 스키마, 선박 정보, Port-MIS

Key word : RDF, SPARQL, RDF Schema, Vessel Information, Port-MIS

<http://dx.doi.org/10.9728/dcs.2020.21.11.1939>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 22 October 2020; **Revised** 23 November 2020

Accepted 23 November 2020

***Corresponding Author; Hyu-Chan Park**

Tel: + [redacted]

E-mail: hcpark@kmou.ac.kr

I . Introduction

Semantic Web is a web extension in which data has meaning for humans and machines in such way that it can be searched, interpreted, exchanged and reused between applications, organizations, and communities. Its technology enables people to construct web-based data stores, develop vocabulary, and write data handling rules. This is known as the Web of data. In the Semantic Web framework, Linked Data is an essential concept within the Semantic Web. Technologies such as RDF and SPARQL are empowering Linked Data. The aim of Linked Data is to connect data to allow machines to browse the Web [1].

RDF and SPARQL are introduced, and various techniques were discussed to improve mapping between RDF and relational data [2]. The implementation with SPARQL, an RDF query language, can combine data from various databases, as well as documents, inference engines, or anything that could express its information as a directed labeled graph.

This paper proposes a framework for RDF and SPARQL technologies in the management of vessel information. To achieve this, vessel information from various sources is first evaluated, and then the structure of vessel information is established. The RDF vessel information schema is constructed with this structure. This implies the existing data does not need to be modified in order to benefit from the advantages of the design pattern, particularly the generation of labeled graph.

This paper also proposes SPARQL queries to retrieve specific data from RDF storage. Data retrieval has been made easy by pulling a specific keyword in searching the data. In order to demonstrate the applicability of the proposed structure for the management of vessel details, a prototype system has also been developed and tested.

II . Related Works

2-1 Semantic Web

Recently, the Web's Semantic data have increased, and the systems are designed to represent the real world in the data set as accurately as possible. Data symbols are organized linearly and hierarchically to give particular meanings as described later on in this study. Semantic data enables machines to communicate with real information without human interpretation by representing the real world in datasets [3].

RDF provides the framework for publishing, by linking these data since it is widely tested, and scalable technology has been

well known for modeling data. The framework was originally a part of the Semantic Web Stack. As shown in Fig. 1, it is used in Querying a Rules stack, Schema & Ontologies stack, and also Data Model stack representing the focus of this paper. The integrated representation of data in RDF helps the identification, disambiguation, and interconnection of information by software agents and various systems to read, analyze, and act on [4].

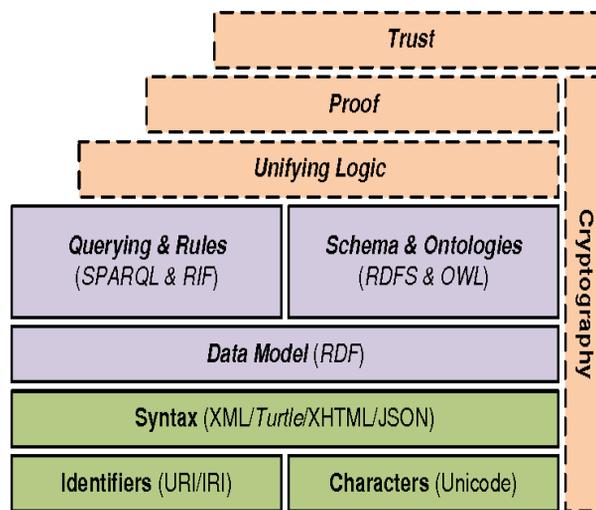


그림 1. 시멘틱 웹 스택 [5]

Fig. 1. Semantic Web Stack [5]

2-2 Linked Open Data

Linked Open Data (LOD) is a robust combination of Linked Data and Open Data where both are linked and open source. DBpedia, for example, is a community-driven initiative for extracting structured data from Wikipedia and making it available on the internet is a remarkable example of a LOD collection providing greater queries, enhancing untapped data exploration and effective results analytics.

The following, Fig 2 shows an illustration of Linked Data by DBpedia when converting Wikipedia content to RDF and linking the contents to other databases such as GeoNames. The data offered by DBpedia is complete and more reliable by incorporating other data. Since 2008, the LOD Cloud has an average amount of 50 published datasets. As of March 2019, the datasets increased rapidly with 1,239 datasets with 16,147 links [6]-[8].

Linked Data is one of the major elements of the Semantic Web. It is a set of design concepts to share interlinked machine-readable data on the Web [9][10].

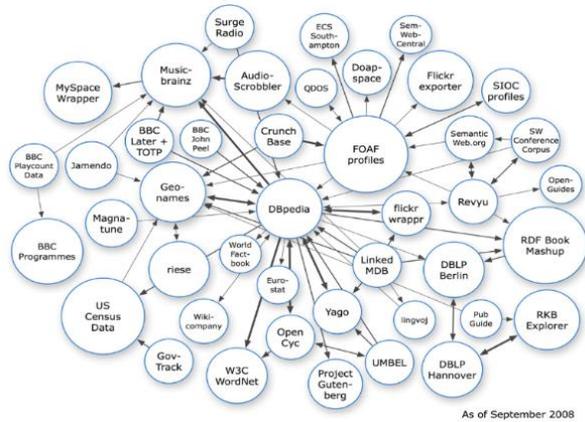


그림 2. 링크드 오픈 데이터 클라우드 [6]
 Fig. 2. The Linked Open Data Cloud [6]

2-3 RDF, RDF Schema, and RDF Triplestore

RDF is a data integration format developed and agreed upon by W3C. While there are many standard tools to handle data and relationships between data. RDF is the simplest, most efficient standard that has been designed as of now.

The two W3C standards, RDF and RDF Schema (RDFS), are designed to improve the Web with machine-processable semantic data. As a data modeling vocabulary for RDF data, schema data has been introduced. The first public draft of the scheme was published in 1998 [11] and adopted it in 2004 as a recommendation [12].

RDF is based on making statements in the form of subject, predicate and object expressions (i.e., triples) about resources, which is described as the Web. The subject denotes the resource and the predicate expresses the relation between the subject and the object. RDF is a conceptual data-modelling tool in general [13].

RDF Triplestore is a special database system for data management in RDF, storing interconnected data and gathering new facts out of the existing ones. Triplestore provides a standard SPARQL query language. By combining the provided primitives, users can define unique patterns and restrictions. A dataset is used to parse and transform SPARQL queries, while Apache Jena provides HTTP access to SPARQL.

Apache Jena is a component of a SPARQL server for other RDF query and storage systems with the integration of Jena TDB [16][17]. It provides access control at the level on the server, on datasets, on endpoints and also on specific graphs within a dataset.

2-4 SPARQL

SPARQL is originally used to query graph data represented as

RDF triples. Together, SPARQL and RDF make it easier to merge results from multiple data sources. It is designed to enable Linked Data to the Semantic Web and enrich data by linking it to other global semantic resources. Thus, sharing, merging and reusing data are performed in a more meaningful way [14].

According to Cambridge Semantics, the differences between SPARQL and SQL is that both languages give the user access to create, combine, and consume structured data. SQL access tables in relational databases and SPARQL access a web of Linked Data. SPARQL can be used to access relational data as well but was designed to merge disparate sources of data. Moreover, queries can be expressed across a range of datasets such as documents or any relevant information that could be presented as a directed labeled graph. The results of SPARQL queries can be in the form of sets or RDF graphs where the data is stored locally or viewed as RDF by middleware [15].

Relational data made up of rows of data collected into tables, which are also called “relations” in formal relational literature. The rows in a table conform to a set data types and constraints called a schema [15].

III. Vessel Information Structure

According to FleetMon [18], the world’s first terrestrial Automatic Identification System (AIS) data collection was in 2007. Satellite AIS data from 2013 onwards, the data were made available to access historical positions, schedules and port arrivals, trading patterns, static AIS data, port calls and database extracts.

Vessel information may not only consist of the technical specifications and management information, but it may also provide a real-time AIS vessel tracking service. In the study of this paper, two separate datasets were analyzed after they were obtained from various websites around the world to study the management of RDF data.

3-1 Vessel Information Structure in General

In order to improve a new concept of searching through the Web of Data, the structure design of datasets focuses on the data retrieved based on the type of information the user wants to search for. However, the vessel structure is divided into few categories.

As shown in Fig. 3, the overall structure consists of six main categories, including historical data, machinery, performances, capacities, general, and specifications. Each category includes the details of the vessel information.

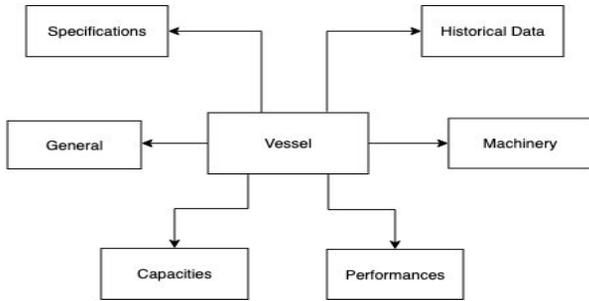


그림 3. 선박 정보의 전체적인 구조
Fig. 3. Overall structure of vessel information

In the first two categories mentioned in Fig. 4, the historical data are the name of the vessel, the name of the port, the call sign, the date of arrival, the date of departure, the MMSI and the flag containing the data from the previous location. The machinery shows the basic details of the vessel's engine, such as the mains, generators and the vessel's control.

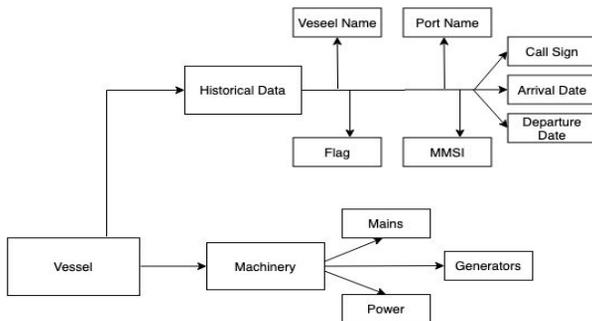


그림 4. 입출항 데이터 및 기계장치
Fig. 4. Historical data and machinery

Next, Fig. 5 shows the vessel performances and capabilities where specifics such as fuel consumption, speed, weight and overall volume are defined along with size of the vessel.

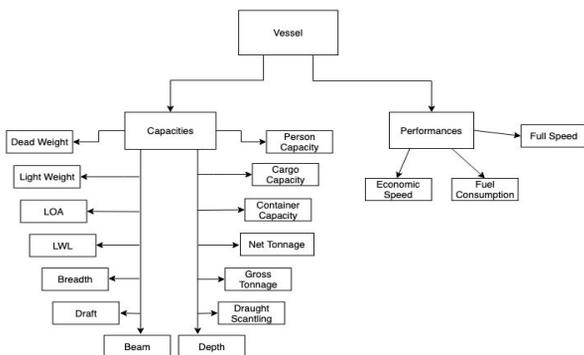


그림 5. 성능 및 용량
Fig. 5. Performances and capacities

The last two categories as shown in Fig. 6 include general information such as the MMSI number, type of vessel, year it was built, name, owner and etc. The specifications category defines the IMO number, port of registry, class, callsign and others.

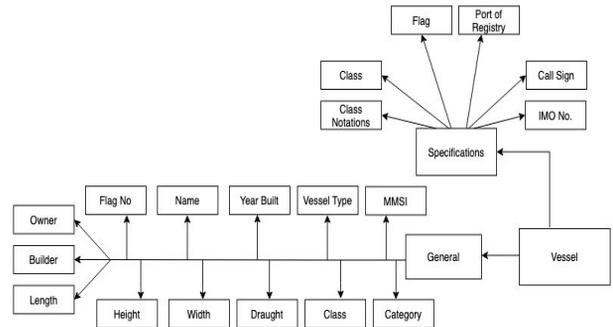


그림 6. 일반 사항 및 스펙
Fig. 6. General and specifications

3-2 Vessel Information Structure based on Port-MIS

Port Management Information System (Port-MIS) is an information system that has all administrative tasks such as the movement, the entry and departure of vessels, the entry cargo and transport of cargo, as well as the collection of taxes in thirty-one commercial ports across the country. The Port-MIS has been transformed into a Web-based system. In order to provide real-time information, several different types of civil complaint systems and advanced civil complaint services are provided through the wired and wireless internet [19].

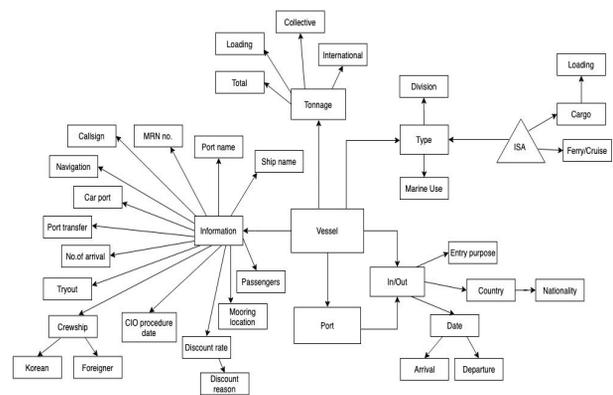


그림 7. Port-MIS에 기반한 선박 정보 구조
Fig. 7. Vessel information structure based on Port-MIS

As shown in Fig. 7, the diagram shows how the main subject vessel is connected to other related information, for example, its type, tonnage, related information, port, traffic flows and other information about the vessel.

IV. RDF Schema Design

Fig. 8 depicts the RDF Schema diagram showing one of the main RDF resources: Vessel. The Type resources are subclasses of the Vessel resource. Each instance of these resources has a URI associated with it. Resource related properties are also described in this schema.

The ranges of some properties are resources from other domains themselves. For example, resources Port has a property isConnectedBy. This property has its domain in the Port but its range is in the In/Out domain. The schema helps establish a data model that makes data storage and retrieval more efficient. It also helps to keep track of the relationship links specified within it.

Fig. 9 depicts a part of the RDF schema, where a sample is taken to further describe how it is organised and implemented in the RDF storage framework.

As shown in Table 1, the subject Vessel has a predicate hasTypes and an object ISA, and then ISA becomes the subject with a predicate providesShipType for the object Cargo and so forth. It was translated in the form of triples to make the information readable and understandable through the relationship links.

A sample RDF syntax in Fig. 10 is constructed to relate the type of the ship and its loading information to others.

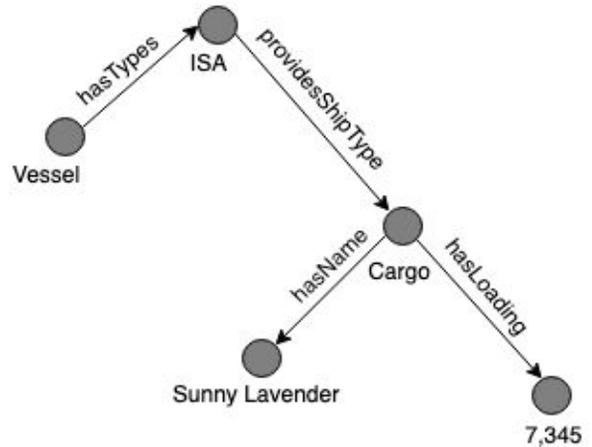


그림 9. 선박 타입 및 로딩 정보에 대한 RDF 그래프
Fig. 9. RDF graph of ship type and loading information

표 1. 선박 타입 및 로딩 정보에 대한 RDF 트리플
Table 1. RDF triples of ship type and loading information

Subject	Predicste	Object
Vessl	hasTypes	ISA
ISA	previdesShipType	Cargo
Cargo	hasName	Sunny Lavender
Cargo	hasLoading	7,345

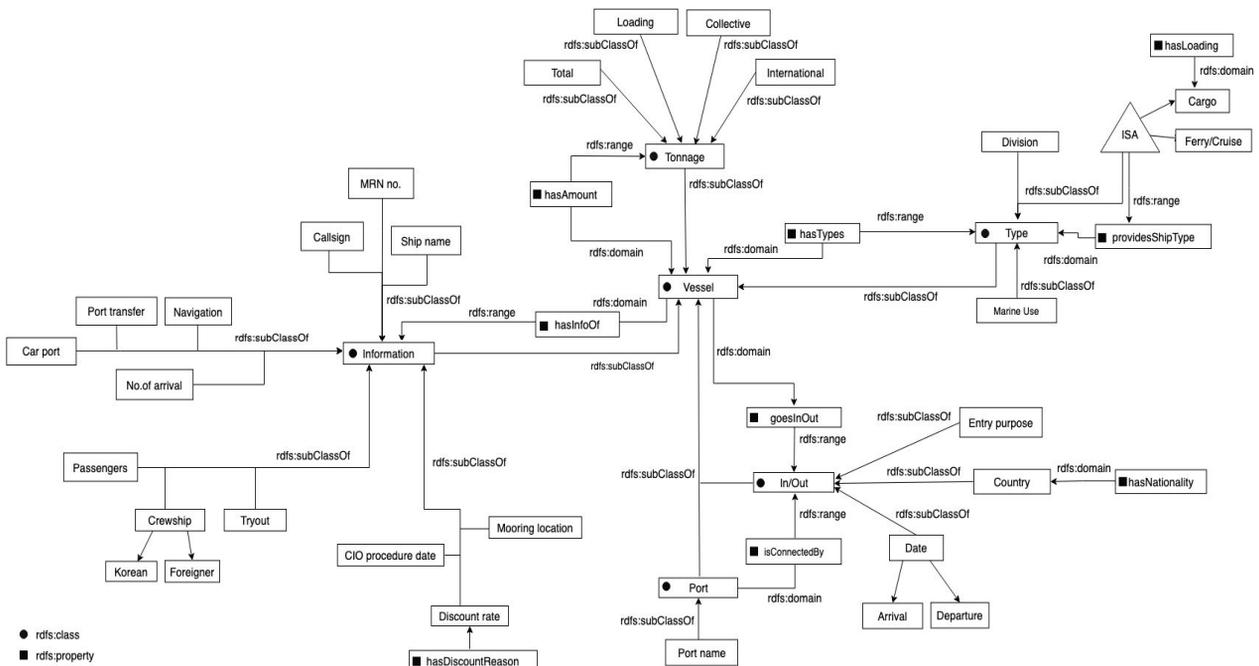


그림 8. Port-MIS에 기반한 선박 정보에 대한 RDF 스키마
Fig. 8. RDF schema for vessel information based on Port-MIS

```

Query. A sample RDF syntax of ship type and loading 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:vs="https://new.portmis.go.kr/port#">

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

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

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

<rdf:Description ID="7,345">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#Cargo"/>
  <rdfs:subClassOf rdf:resource="#Vessel"/>
</rdf:Description>

</rdf:RDF>
    
```

그림 10. 선박 타입 및 로딩 정보에 대한 RDF 문법
 Fig. 10. RDF syntax of ship type and loading information

V. Implementation and Testing

5-1 System Architecture

An architecture of RDF storage and query system is shown in Fig. 11. As illustrated, the user constructs a query and access the 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.

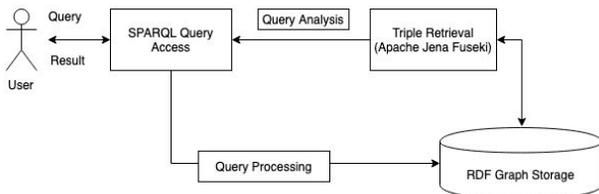


그림 11. RDF 저장 및 질의 시스템의 구조
 Fig. 11. Architecture of RDF storage and query system

The illustration in Fig. 12 describes the process of how raw uncategorized data are analyzed and converted into RDF format that is readable for both humans and machines.

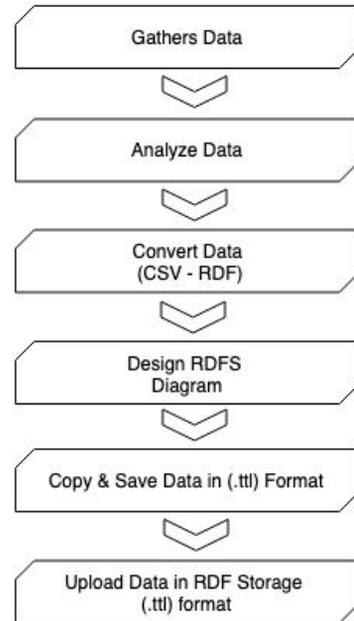


그림 12. RDF 저장 절차
 Fig. 12. Process of RDF storage

Fig. 13 shows the process of SPARQL query to retrieve the matched result. In order to start all the construction of queries, the SPARQL server must be running in the background. The construction of queries needs to be completed to retrieve data from RDF storage.

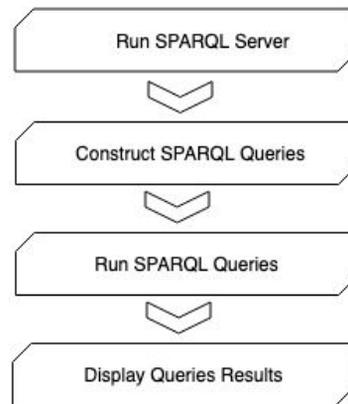


그림 13. RDF 질의 절차
 Fig. 13. Process of RDF query

5-2 SPARQL Query

Fig. 14 and Fig. 15 show an example of SPARQL query and the result of retrieved data. Retrieving information from RDF storage is done within the SPARQL server and the triplestore.

```

Query. Vessel Information based on Port-MIS
PREFIX pt: <http://eng.kipris.or.kr/enghome/main.jsp>
PREFIX vs: <https://new.portmis.go.kr/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

CONSTRUCT {
  ?URI a vs:Vessels;
  vs:PortName ?PortName;
  vs:CallSign ?CallSign;
  vs:ShipName ?ShipName;
  vs:NoOfArrivals ?NoOfArrivals;
  vs:Division ?Division;
  vs:Outgoing ?Outgoing;
  vs:Incoming ?Incoming;
  vs:TotalTonnage ?TotalTonnage;
  vs:InternationalTonnage ?InternationalTonnage;
  vs:CollectionTonnage ?CollectionTonnage;
  vs:DateOfArrival ?DateOfArrival;
  vs:DateOfDeparture ?DateOfDeparture;
  vs:CioProcedureDate ?CIOProcedureDate;
  vs:RepairDate ?RepairDate;
  vs:EntryPurpose ?EntryPurpose;
  vs:Navigation ?Navigation;
  vs:MrnNumber ?MRNNumber;
  vs:LoadingCargo ?LoadingCargo;
  vs:LoadingTonnage ?LoadingTonnage;
  vs:Country ?Country;
  vs:MooringLocation ?MooringLocation;
  vs:ShipPlaceA ?ShipPlaceA;
  vs:ShipPlaceB ?ShipPlaceB;
  vs:DiscountRate ?DiscountRate;
  vs:DiscountReason ?DiscountReason;
  vs:CarPort ?CarPort;
  vs:PortTransfer ?PortTransfer;
  vs:ShippingAgent ?ShippingAgent;
  vs:MarineUse ?MarineUse;
  vs:TrainmanKorean ?TrainmanKorean;
  vs:TrainmanForeigner ?TrainmanForeigner;
  vs:Passengers ?Passengers;
  vs:Tryout ?Tryout;
  vs:Ferry ?Ferry;
}
FROM <file:vessels.csv>
WHERE {
  BIND (URI(CONCAT('https://new.portmis.go.kr/', ?CallSign))
  AS ?URI)
}
    
```

그림 14. 선박 정보에 대한 SPARQL 질의
 Fig. 14. SPARQL query for vessel information

```

C:\tarql-1.2\data>tarql --write-base vessel.sparql vessels.csv
@base <file:///C:/tarql-1.2/data/vessel.sparql> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ex: <http://ex.org/a#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://ex.org/vessels/3FH08>
  rdf:type ex:Vessels ;
  ex:PortName "BUSAN" ;
  ex:CallSign "3FH08" ;
  ex:ShipName "SUNNY LAVENDER" ;
  ex:NoOfArrivals "2019-001" ;
  ex:Division "FINAL" ;
  ex:Outgoing "OUT" ;
  ex:Incoming "ARRIVAL" ;
  ex:TotalTonnage "9,865" ;
  ex:InternationalTonnage "9,865" ;
  ex:CollectionTonnage "9,865" ;
  ex:DateOfArrival "2019-01-01 00:35" ;
  ex:DateOfDeparture "2019-01-01 15:56" ;
  ex:CioProcedureDate "2019-01-01 00:35" ;
  ex:RepairDate "2019-01-02 09:27" ;
  ex:EntryPurpose "QUANTITATIVE" ;
  ex:Navigation "OTHER" ;
  ex:MrnNumber "19KMT0095I" ;
  ex:LoadingCargo "\nMiscellaneous goods" ;
  ex:LoadingTonnage "6,658" ;
  ex:Country "PANARA" ;
  ex:MooringLocation "SINSEONDAE-BUDU 4 SEONSEOG" ;
  ex:CarPort "ULSAN" ;
  ex:PortTransfer "SHIMIZU" ;
  ex:ShippingAgent "????(?)KORYO SHIPPING CO., LTD." ;
  ex:MarineUse "FULL CONTAINER SHIP" ;
  ex:TrainmanKorean "2" ;
  ex:TrainmanForeigner "16" ;
  ex:Passengers "1" ;
  ex:Tryout "Y" ;
  ex:Ferry "Y" .

<http://ex.org/vessels/A8ZP6>
  rdf:type ex:Vessels ;
    
```

그림 15. 선박 정보에 대한 질의 결과
 Fig. 15. Query result for vessel information

VI. Conclusion

This paper adopted RDF related technologies such as RDF Schema, RDF Triplestore, and SPARQL for the management of vessel information. The structure of vessel information was defined through the analysis of several vessel information, especially Port-MIS. To consistently represent the structure of vessel information, RDF Schema was also defined. Vessel information according to the RDF Schema was stored in the RDF Triplestore. The stored vessel information can be efficiently queried and retrieved by using SPARQL. A prototype system was implemented and tested to show that vessel information can be properly managed.

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] What is Semantic Data, 2009, [Online] Available: <http://www.semagix.com/what-is-semantic-data.htm>
- [4] Ontotext: What is RDF? Making Data Triple Their Power, 2017, [Online] Available: <https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf/>
- [5] A. Hogan, Linked Data and the Semantic Web Standards, 2013, [Online] Available: http://aidanhogan.com/docs/ldmgmt_semantic_web_linked_data.pdf
- [6] F. Yin, A Parallel Application, 2013, [Online] Available: <https://people.eecs.berkeley.edu/~driscoll/cs267/hw0/html/XiaotingYin/>.
- [7] Frank, Matthias, Zander, and Stefan, The Linked Data Wiki: Leveraging Organizational Knowledge Bases with Linked Open Data, 2019.
- [8] The Linked Open Data Cloud, 2007, [Online] Available: <https://www.lod-cloud.net/>
- [9] Star Open Data, [Online] Available: <https://5stardata.info/en/>
- [10] Ontotext: What Are Linked Data and Linked Open Data?, 2016, [Online] Available: <https://www.ontotext.com/knowledgehub/fundamentals/linked-datalinked-open-data/>
- [11] 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/>

[12] 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/>

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

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

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

[16] 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, pp. 45-58, 2012.

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

[18] C. Ducruet, S. W. Lee, and S. Roussin, Local strength and global weakness: A maritime network, 2009, [Online] Available: <https://www.researchgate.net/publication/41528614>

[19] Port-MIS: Yes! U-Port Your Future E-Business Safe Voyage, 2018, [Online] Available: <https://www.klnet.co.kr/resources/download/02.pdf>



Zasyana Mozahker

2018년 : Management and Science University, Malaysia (공학사)
2020년 : 한국해양대학교 대학원 (공학석사)

2020년~현재 : Online researcher

※ 관심분야 : 시멘틱 웹, 데이터베이스, 선박 및 해양정보



신옥근(Ok Keun Shin)

1981년 : 서강대학교 (공학사)
1983년 : 부산대학교 대학원 (공학석사)
1995년 : Universite de Franche-Comte (공학박사)

1995년~현재 : 한국해양대학교 해사IT공학부 교수

※ 관심분야 : 신호 처리, 임베디드 시스템



박휴찬(Hyu Chan Park)

1985년 : 서울대학교 (공학사)
1987년 : 한국과학기술원 (공학석사)
1995년 : 한국과학기술원 (공학박사)

1997년~현재 : 한국해양대학교 해사IT공학부 교수

※ 관심분야 : 데이터베이스, 데이터마이닝, 빅데이터, 선박 및 해양정보