An Approach to Map Geography Mark-up Language Data to Resource Description Framework Schema

Ammara Faqir[1], Aqsa Mahmood[2], Kiran Qazi[3], and Saleem Malik[4]

[1] Department of Computer Science & IT, National College of Business Administration & Economics Lahore, Punjab, Pakistan,
[2] Department of Computer Science & IT, Govt. Sadiq College Women University Bahawalpur
[3] Department of Computer Science &IT, The Islamia University of Bahawalpur
[4] Department of Computer Science, Federation University Australia, Victoria, Australia
ammarafaqir99@gmail.com, aqsa.mahmood@gscwu.edu.pk, kiranaliqazi@gmail.com

Abstract. GML serves as premier modeling language used to represent data of geographic information related to geography locations. However, a problem of GML is its ability to integrate with a variety of geographical and GPS applications. Since, GML saves data in coordinates and in topology for the purpose to integrate data with variety of applications on semantic web, data be mapped to Resource Description Framework (RDF) and Resource Description Framework Schema (RDFS). An approach of mapping GML metadata to RDFS is presented in this paper. This study focuses on the methodology to convert GML data in semantics to represent in extended and enriched form such as RDFS as representation in RDF is not sufficient over semantic web. Firstly, we have GML script from case study and parse it using GML parser and get XML file. XML file parse using Java and get text file to extract GML features and then get a graph form of these features. After that we designed methodology of prototype tool to map GML features to RDFS. Tool performed features by features mapping and extracted results are represented in the tabular form of mapping GML metadata to RDFS.

Keywords: Geography Markup Language, resource description framework, resource description framework schema, mapping, ontologies.

1 Introduction

Geospatial data is the extremely significant information regarding coordinate values of physical entities of our planet which can be represented in the form of numerical value in geospatial system. Geography markup language (GML) is a standard language for storing and transporting geography information to stimulate common policies, principles, framework guidelines and standards for interoperability and inter changeability for geospatial data and services. GML works for geography systems and be responsible for interchanged format for geographic transaction of data on internet. The ontology is the study of classifications of things that occur or may occur in some domain. It mainly deals with generating common terminology for different platforms. It may also be constructed on set of Features like things, events and relations that are specified in so-me way to create a communal terminology for exchanging information among different platforms. Ontology empower knowledge distribution in a way to facilitate different mindsets for same thing or vice versa.

Semantic web denotes information in well-define meaning, enable computer systems in a better way, and also empower people to work in corporation. Two vital technologies are in exercise to develop semantic web that is Extensible Markup Language (XML) and Resource Description Framework (RDF). In XML every user creates their own tag and it permit users to add random structure to their documents without giving any clue about the structure. Structural meaning is articulated by RDF approach which encodes it in set of triples then writes it in XML tags .RDF document make declaration about particular thing and their properties with certain values.

Resource Description Framework Schema (RDFS) is a semantic extension of RDF which is responsible for providing mechanism to describe the group of related resources, their relationships. RDFS is a basic schema language. RDFS and semantic web technologies permit data created by different teams for different uses at dissimilar times to be connected. RDFS describes classes of objects and inheritance.

Mapping of geography markup language data is to convert it into resource description framework. Only mapping in RDF do not provide comprehensive logic of resources. As geospatial data is accessed by numerous application systems which requires comprehensive description of these resources. So, it has become important to map GML data to RDFS that will describe related resources and their relationships.

We have GML script from case study and parse it using GML parser and get XML file. XML file parse using Java and get text file to extract GML features and then get a graph form of these features. After that we designed methodology of prototype tool to map GML features to RDFS. Tool performed features by features mapping and extracted results are represented in the tabular form of mapping GML metadata to RDFS.

2 Related Work

The related work presented in the era of mapping Geography Markup Language (GML) data to Resource Description Framework Schema (RDFS) to highlight its importance.

(Alam, M., & Napoli, A.2015, October) presented a study to represent interactive data searching paradigm using pattern structures. It took RDF triples and RDF schema and provided one navigation space resulting from several RDF resources. There is pattern mining algorithms with visualization tool to explore data and identified required patterns. These algorithms are advantageous on small datasets. Iterative process of data exploration was performed on Linked Open Data (LOD). Another approach was presented about deriving of linked data commencing GML data and highlighted its importance to represent GML data in RDF by transforming it from Unified

Modeling Language (UML) to Web Ontology Language(OWL) (Van den Brink, L., Janssen, P., Quak, W., &Stoter, J. E.,2013).

Conventional reasoning systems are not appropriate for massive quantity of semantic data due to resource limitations and existing reasoning system are inappropriate for enormous data. A solution was presented by (Gu, R., Wang, S., Wang, F., Yuan, C., & Huang, Y. ,2015, May) named as Cichild that considered as proficient reasoning engine, for extensively used RDFS and OWL Horst Rule sets. This engine was based on Spark. It implemented parallel reasoning algorithm with the Spark RDD (Resilient distributed dataset) Programming model. They suggested an optimized parallel RDFS reasoning algorithm. Swift understanding, easy searching and selection of schema and the web document was made possible in a study presented by (Troullinou, G., Kondylakis, H., Daskalaki, E., &Plexousakis, D.,2015, May) is a RDF digest, a novel base framework which provide automated summaries of the RDF/S Knowledge Base(KBs). To construct the Graph, algorithm utilizes the semantic, structure of schema and distribution of matching data or instances. Summarized Graphs created by proposed system are used to explore the Semantics of Schema, structure of the RDFS Graph, and distribution of the subsequent data to recognize important elements of ontology.

Linking data object to the dataset URI is presented by (Hietanen, E., Lehto, L., & Latvala, P. 2016) using the Vocabulary of Interlinked Datasets (VoID). The dataset is partitioned to the subsets which is given its persistent and specific URI. This permitted the entire dataset to be explored by using web browser and index each object using web crawlers. Another study represented by (Zhao, T., Zhang, C., & Li, W. ,2017) proposed an algorithm to convert RDF query to Web Feature Service(WFS) requests keeping in view that clients can request distributed WFS features as they were RDF instances. The algorithm avoided the cost of changing features to RDF objects while holding the advantages of RDF queries.

The possibility of handling billions of RDF triples on one service machine presented by (Corcoglioniti, F., Rospocher, M., Mostarda, M., & Amadori, M. 2015, April).This made possible by using streaming , sorting techniques , and focus on RDF processing tasks.These tasks are data filtering and transformation RDFS inference OWL that is same as statistics extraction . This paper introduce RDFpro (RDF processor) to handle this. RDFpro is an open source tool which provides streaming and sorting base processor for tasks. Semantic RDF services are become more confronted with many big data issues. In the presence of inference, query processing is one of them. To reduce memory footprint and make simple exchange of huge datasets (Cure, O., Naacke, H., Randriamalala, T., & Amann, B. 2015, October) presents structured resource identification scheme with help of clever TBox encoding of Features and property hierarchies for the efficient assessment of main and common RDFS entailment rules , while minimizing query rewriting and triple materialization. This paper presents how encoding computed through scalable parallel algorithm.

It becomes difficult for traditional systems to handle enormous data. For better utilization a semantically enriched data representation is required with keep it simple. (Malik, K. R., Ahmad, T., Farhan, M., Aslam, M., Jabbar, S., Khalid, S., & Kim, M.,2016) provide solution as representation in Resource Description Framework

Schema (RDF), introduced by World Wide Web Consortium (W3C). Data that is taken from different sources is formats into the RDF. Improvement is required to handle transition of information between all applications. This paper illustrate transformation of relational database textual data into RDF by using a case study.

To extract geospatial features from various sources (Patroumpas, K., Alexakis, M., Giannopoulos, G., &Athanasiou, S. 2014) present TripleGeo ETL utility. It transform geospatial data into triples for subsequent loading into RDF Stores. This tool have been tested and validate against OpenStreetMap layers with huge amount of geometries. Spatial information in data formats, schemas, platforms, systems, and web services. This Spatial data remains in databases. Geographic Information Systems (GIS) managed by vendors or by Government agencies. But Open Geo Spatial Consortium (OGC) and the initiatives for constructing Spatial data Infrastructures (SDI) under the EN INSPIRE directive drove path leading towards the Geospatial data Interoperability and distribution.

3 Proposed Methodology

Transformation by means of mapping of GML metadata into RDFS can make it more accessible for other domains of knowledge and can make it more advantageous. Following framework is used to map GML data to RDFS.

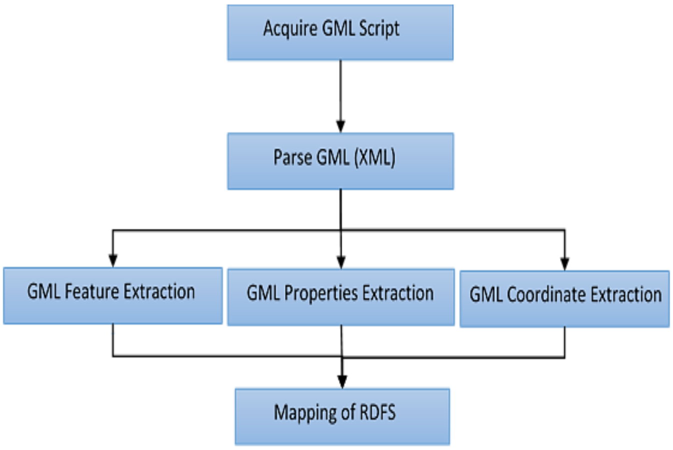

Fig. 1. Proposed Architecture for Mapping of GML Script to RDF Schema

3.1 Acquire GML Script

We input GML metadata in this step and will receive XML file of the given GML metadata as a result of this phase. This XML file have the three major types of tags, GML elements is the parent tag have child tag Feature tag, Description tag and Property sets tag.

3.2 Parse GML (XML)

In this phase parsing is performed on XML file that is generated as the output of the first phase. Following figure shows the steps to be performed during parsing of XML file. XML parsing is done in java. Java Programming facilitates to multiple ways to parse XML documents. But we used here DOM parse way and data received in text form.

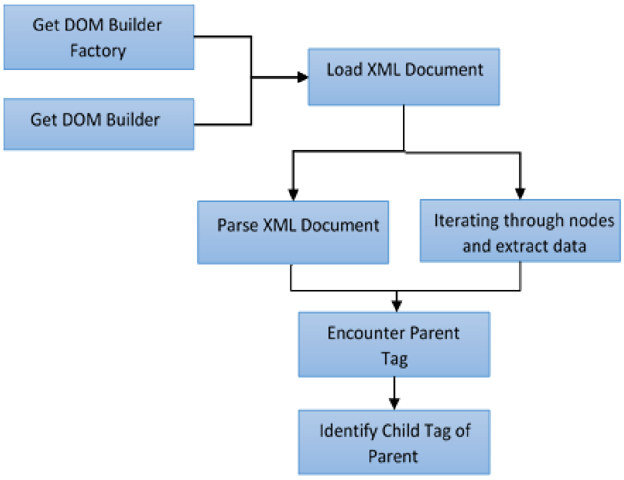

Fig. 2. Proposed Methodology to Parse GML (XML)

3.3 Extracting GML Features

In this phase, we input the textual data that was output of the previous phase to extract the GML elements. The general elements of GML are Features. Afterwards we apply parsing to distinguish between the noun Feature and Verb Features from the output of the first phase of extracting the Feature.

3.4 Extraction of Feature Property

In this phase parsing is performed on text file in order to extract GML elements. Their noun and verb feature are identified and extracted for further parsing process. Output of this phase is obtained in the form of graph.

3.5 GML Coordinates Extraction

In GML, the Coordinates are among the vital elements besides Features. These Coordinates denote coordinates of geometry instances. In this phase following three elements are used to specify details of a GML coordinate:

- *<gml:coordinates>*
- *<gml:pos>*

- *<gml:posList>*

XML DOM parser provides the complete string instead of individual coordinates. Here, we apply Java string functions to segregate each coordinate value from the resultant string of coordinates.

3.6 Mapping to RDFS

It is the last phase of our proposed methodology, it consists of text graph of RDF and RDFS. In the previous step we received GML terminology of elements. Elements are in two categories: one is Known Features and other is Verb/Facts Feature. In the last phase, noun Features became the node and fact/relation became the edges of the graph that symbolized the relation between nodes. Then our prototype tool map “Features” to “Resources” , GML Feature “Property” to the RDFS “Property”, “Property Name” to “RDFS:Label”, “Description” to “RDFS:Comment” RDFS elements. “Feature Type” GML element has no mapping in RDFS elements.

4 RESULTS AND DISCUSSION

We discuss the different features of our tool that included the phases of input, output, and processing of our tool. After it we make experiment by taking a case study to evaluate the performance of our tool. Our prototype tool of RDFS Generated automated transformation of GML statements to RDF and RDFS. Now, we discuss the following Module of our tool in details.

4.1 GML Input to RDF Generator

The Input for our tool is specified in XML file that is generated from GML editor. These GML statements are collected from several case studies .We input the Script rules about features, coordinates and properties of GML file to the editors Shown in Figure 3. Editor perform processing on GML statements and generate the XML file. Then GML editors again input GML statements and extract vocabulary as GML Noun, Verbs, Fact Types, Object Types, and Quantification etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ows="http://www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml" xmlns:gs="http://www.geoserver.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" numberOfFeatures="1"
  timeStamp="2014-08-07T12:22:06.508Z"
  xsi:schemaLocation="http://www.geoserver.org http://localhost:8080/geoserver/gs/wf
  <gml:featureMembers>
    <gs:CURVES gml:id="CURVES.0">
      <gs:NAME>Arc segment</gs:NAME>
      <gs:GEOMETRY>
        <gml:Curve srsDimension="2" srsName="urn:x-ogc:def:crs:EPSG:32632">
          <gml:segments>
            <gml:ArcString interpolation="circularArc3Points">
              <gml:posList>10.0 15.0 15.0 20.0 20.0 15.0</gml:posList>
            </gml:ArcString>
          </gml:segments>
        </gml:Curve>
      </gs:GEOMETRY>
    </gs:CURVES>
  </gml:featureMembers>
</wfs:FeatureCollection>
```

Fig. 3. Script of Rules

4.2 GML Output of RDFS Generator

GML editors generate output in Xml format and separate the noun Feature in tags such as name and terms and verb Feature in tags such as facts where verb show the relation between two nouns. Tagged XML file generated as the output of this phase is shown below.

```
<xsl:stylesheet version="1.0" ..... >
<xsl:template match="/">
<MapAtlasLab="Cartography"Date="2002-07-25"
xsi:noNamespaceSchemaLocation="E:\Spanaki\GML\XML2GML\MapAtlas_4POINTS.xsd">
    <xsl:for-each select="layer/ROW">
        <MapPoints>
            <xsl:attribute name="STSINM"><xsl:value-of select="STSINM"/></xsl:attribute>
            <xsl:attribute name="OBJECTID"><xsl:value-of select="OBJECTID"/></xsl:attribute>
            <PointsGeometry>
                <gml:coord>
                    <gml:X>
                        <xsl:apply-templates select="SHAPE/SDO_POINT/X"/>
                    </gml:X>
                    <gml:Y>
                        <xsl:apply-templates select="SHAPE/SDO_POINT/Y"/>
                    </gml:Y>
                </gml:coord>
            </PointsGeometry>
        </MapPoints>
        </xsl:for-each>
    </MapAtlas>
</xsl:template>
</xsl:stylesheet>
```

Fig. 4. Generated XML File

4.3 GML/ XML Processing

This step of our tool consists of input XML file that is generated from XML editor in previous phase to RDFS generator. Prototype tool parsed it in simple textual format that is the collection of the GML vocabulary. Then Processing is done and vocabulary of GML (Terms, fact type) is extracted. In vocabulary, terms are considered as nouns Features and facts types are considered as verb/ Relation. After extraction phase the vocabulary is written in text file and then text file is input into the graph generator which represent GML vocabulary in graph format as shown in the following figures.

```
<rdf:Description rdf:about="#FullSlide">
    <axsvg:GraphicsType>Chart</axsvg:GraphicsType>
    <axsvg:LabelledBy rdf:resource="#BottomLegend"/>
    <axsvg:ChartType>Line</axsvg:ChartType>
</rdf:Description>
<rdf:Description rdf:about="#BottomLegend">
    <axsvg:IsAnchor>true</axsvg:IsAnchor>
</rdf:Description>
```

Fig. 5. GML Vocabulary Extraction Text File

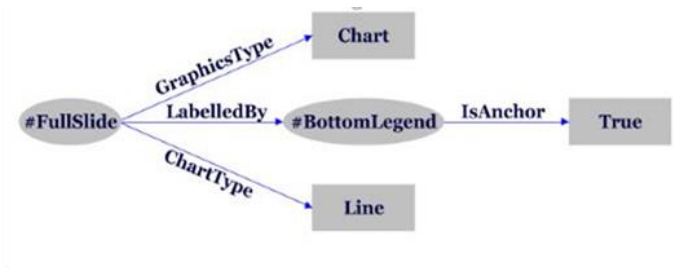

Fig. 6. GML Vocabulary Extraction Graph

4.4 Output of GML Script to RDFS Script

In this phase we input text file of previous phase that consists of Noun Features and facts type to generate the RDF script in text graph format. In this graph noun Features become the node of RDFS script and facts become the relation between the nodes of RDFS script that is called edges of graph.

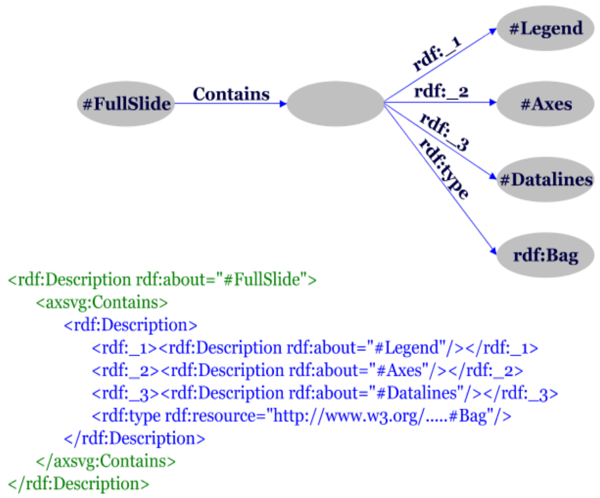

Fig. 7. Output of GML to RDFS Script

4.5 Mapping to RDFS

In the output of previous phase “Legend” is the parent node of the graph, and it has three child node “Axis”, “Data lines”, “Bag”. “Contains” in this word “has” is referent/label of edge of graph similarly “contains” in this word “is in” is the referent of graph. We represent this text graph in mapping of GML elements to RDFS elements where “Feature” element mapped to “Resource” in RDFS, GML element “Feature Type” not mapped in RDFS, GML” Property” element mapped to “Property” in RDFS, “Property Name” GML element mapped to “RDFS:Label” and “Description” element is mapped to “RDFS:Comment” in the following way.

Table 1. Mapping of GML data to RDFS

GML Elements	RDF/RDFS Elements
Feature	Resource
Feature Type	-
Property	Property
Property Name	RDFS: Label
Description	RDFS:Comment

4.6 Discussion of Results

There were Eight GML rules used in case study problem. The most important reason to select this case study was to test the performance of our tool with the complex GML rules. The correct, incorrect, and missing GML elements/ Rule in Graph using proposed methodology are shown in Table 2.

Table 2. Results of Graph generated from case study

Sr #	**Type/Metrics**	**N_{sample}**	**$N_{correct}$**	Nmissing
1	Features	4	3	0
2	Coordinates	8	7	0
3	Properties	14	13	1
	Total	23	21	1

According to our evaluation methodology, table shows sample elements are 23 in which 20 are correct 1 is incorrect and 2 are missing Graph elements.

Table 3. Results of Recall and Precision

Type/Metrics	GML Features
Nsample	23
Ncorrect	21
Nincorrect	1
Nmissing	1
Reccall%	91.30
Precision%	94.45

The results of this primary performance evaluation are extremely promising and supporting both the methodology presented and the capability of this prototype tool in general. There are some other case study that are used to test our prototype tool to achieve the performance results are shown below.

Table 4. Evaluation Results of Various Case Studies

Input	**N_{sample}**	**$N_{correct}$**	**$N_{incorrect}$**	**$N_{missing}$**	**Recall**	**Precision**	**F-value**
C1	24	22	1	1	91.16	95.65	93.35
C2	45	40	2	3	88.88	95.23	91.94
C3	33	31	2	0	93.93	93.93	93.93
C4	46	43	1	2	93.47	97.72	95.54
C5	56	45	5	6	80.35	90.00	84.90
Average	40.5	36.2	2.4	Average	89.38	93.88	91.57

5 CONCLUSION AND LIMITATIONS

To address the primary objective of this research study we have designed prototype tool to generate RDFS by overcoming GML rule (Generated from English) and Show the relationship between Nouns Feature and Fact type of GML vocabulary in RDFS. GML Generator is capable to parse XML file that is generated from Transaction editors of GML that input the GML rules received from English text. Extract GML vocabulary by using parsed text file which have nouns Features and fact /verb Features of GML, in the last step by using this GML vocabulary our tool generate the RDF and RDFS in text graph form. Then we perform mapping. We apply experiment on different case studies to check the performance of our proposed tool. Hence, the evaluation method proved that our proposed tool performance is satisfactory and work in accurate manner.

The output of our tool can be used to present RDF and RDFS view of GML rules for the better understanding and data interchange. As shown in the results section, the recall (89.38%) and precision (93.88%) results applying on the used case study for software requirements by using our tool are very satisfactory. Similarly, calculated F-value (91.57) is quite encouraging. Generated results also shows that the presented approach is easy and time saving to generate a semantically formal and controlled representation using our automated approach and our prototype tool".

6 FUTURE WORK

Keeping an eye towards the future GML and on the possible impact are to be explored. These aspects are discusses below:

- More research can be perform on the approaches based on mapping GML data to RDF and RDFS .Our tool can be expanded towards the skewed area charts (i.e. positive and negative) interpretation.
- Our tool can be expanded towards presenting RDF and RDFS in visual form.
- Implementation of GML in visual graph can be a good pictorial view of GML. Visual graph representation of GML rule by using the object-oriented Programming.

References

1. Alam, M., & Napoli, A. (2015, October). Interactive exploration over RDF data using formal Feature analysis. In Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on (pp. 1-10). IEEE.
2. Van den Brink, L., Janssen, P., &Quak, W. (2013).From geo-data to linked data: automated transformation from GML to RDF. Linked Open Data-Pilot Linked Open Data Nederland. Deel 2-De Verdieping, Geonovum, 2013, pp. 249-261.
3. Gu, R., Wang, S., Wang, F., Yuan, C., & Huang, Y. (2015, May). Cichlid: efficient large scale RDFS/OWL reasoning with spark. In Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International (pp. 700-709).IEEE.
4. Troullinou, G., Kondylakis, H., Daskalaki, E., &Plexousakis, D. (2015, May). RDF digest: Efficient summarization of RDF/S kbs. In European Semantic Web Conference (pp. 119-134).Springer , Cham.

5. Hietanen, E., Lehto, L., & Latvala, P. (2016). PROVIDING GEOGRAPHIC DATASETS AS LINKED DATA IN SDI. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 583-586.
6. Zhao, T., Zhang, C., & Li, W. (2017). Adaptive and Optimized RDF Query Interface for Distributed WFS Data. ISPRS International Journal of Geo-Information, 6(4), 108.
7. Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C. J., & Buytaert, W. (2015). Web technologies for environmental Big Data. Environmental Modelling & Software, 63, 185-198.
8. Saripalle, R. K., De la Rosa Algarin, A., & Ziminski, T. B. (2015, February). Towards knowledge level privacy and security using RDF/RDFS and RBAC. In Semantic Computing (ICSC), 2015 IEEE International Conference on (pp. 264-267). IEEE.
9. Qiu, R., Duan, H., Yu, X., Zheng, L., & Cheng, Q. (2015, June). An ontology matching method for GML application ontology to GeoSPARQL ontology. In Geoinformatics, 2015 23rd International Conference on (pp. 1-5). IEEE.
10. Lefrançois, M., Zimmermann, A., & Bakerally, N. (2016, November). Flexible RDF generation from RDF and heterogeneous data sources with SPARQL-Generate. In European Knowledge Acquisition Workshop (pp. 131-135). Springer, Cham.
11. Hor, A. H., Jadidi, A., & Sohn, G. (2016). BIM-GIS INTEGRATED GEOSPATIAL INFORMATION MODEL USING SEMANTIC WEB AND RDF GRAPHS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 3(4).
12. Corcoglioniti, F., Rospocher, M., Mostarda, M., & Amadori, M. (2015, April). Processing billions of RDF triples on a single machine using streaming and sorting. *In Proceedings of the 30th Annual ACM Symposium on Applied Computing* (pp. 368-375). ACM.
13. Cure, O., Naacke, H., Randriamalala, T., & Amann, B. (2015, October). LiteMat: a scalable, cost-efficient inference encoding scheme for large RDF graphs. In Big Data (Big Data), 2015 IEEE International Conference on (pp. 1823-1830). IEEE.
14. Malik, K. R., Ahmad, T., Farhan, M., Aslam, M., Jabbar, S., Khalid, S., & Kim, M. (2016). Big-data: transformation from heterogeneous data to semantically-enriched simplified data. Multimedia Tools and Applications, 75(20), 12727-12747.
15. Patroumpas, K., Alexakis, M., Giannopoulos, G., &Athanasiou, S. (2014, March). TripleGeo: an ETL Tool for Transforming Geospatial Data into RDF Triples. In EDBT/ICDT Workshops (pp. 275-278).