Mapping Processor of Resource Description Framework Based On RML

Dufang Fan [a], Xiushan Zhang [b] and Jian Huang [c]
Naval University of Engineering, Wuhan 430033, China.
[a]596344273@qq. com, [b]xiu3zh@139. com, [c]275970675@qq. com

Abstract. With the development of computer and communication technology, the Internet of things (IoT) is developing vigorously among people to realize interconnection, to link objects or people with any other objects. This paper is based on the related research results of a series of concepts such as the semantic Internet of things, sensor networks, and the middleware of the Internet of things. A method of automatic recognition of sensors can be effectively realized in the semantic Internet of things. In this system, the original sensor data are parsed and encapsulated first, and all kinds of data after encapsulation are uniformly transformed. Mainly in view of the shortcomings of the heterogeneous sensor data which is not conducive to the recognition and analysis of the computer, the concept of the ontology is used to describe the sensor and master the description rules of the resource description framework of RDF; An RML (RDF mapping language) processor is developed, which can map or convert different formats of sensor data into information expressed by RDF in application platform.

Keywords: Semantic IoT, RML, RDF.

1. Introduction

Semantic Web of Things (SWOT) is an upgrade of the Internet of things. It can effectively solve the heterogeneous problem of distributed systems. WWW can only display information, but it lacks specific description of information, so the computer still cannot recognize the meaning of information. The semantic Internet uses the ontology in semantic network to describe the information obtained by the sensor in the Internet of things in a certain rule, so that the main body of the information can obtain the understandable data description of the subject itself. That makes the machine understand the information on the Internet, and it can do any intelligent processing that meets the requirements. The semantic Internet of things, based on the four basic elements: human, machine, thing and matter in the information ecosystem, and the overall system architecture of their interaction, constructs a unified complex semantic space, which integrates machine space, physical space and social space on the basis of the semantic network, the Internet of things and the social network. Through semantic synergetic technology, intelligent information interaction [1] is realized.

The huge data information in the Internet of things is an important source of data from the Internet of things. However, the dilemma is, there is no uniform rule for the storage format of all kinds of sensor network data, so the development of the Internet of things is obstructing. In order to solve this problem, Linked Sensor Data is coming into full play. The link sensor data, as a new type of sensor data, is formed by generating the sensor network data annotated by the sensor ontology into the associated data and linking it with the related data. The Linked Open Data (LOD) uses the resource description framework (Resource Description Framework, RDF), which integrates the different data in various fields together.

The research significance of this topic is as follows:

RML mapping language is designed to solve the problem that the data format of the sensor network is not unified. The computer automatically identifies different types of sensors, and it can be converted into a unified source description framework, whether the sensor format is JSON or XML.

in view of the characteristics of heterogeneous data, a data generation system is developed. It can map or convert different formats of sensor data into the information expressed in RDF by the RML processor.

(3) in view of the disadvantages of heterogeneous sensor data which is not conducive to computer recognition and analysis, this paper studies how to use the concept of ontology to describe the sensor and master the description rules of the RDF resource description framework.

The rest of the paper is organized as follows: Section 2 discusses related solutions existing today. Section 3 describes how the RDF (Resource Description Framework) provide a method for describing data for information of Web resources and describes the RML language handles the mapping of hierarchical sources to RDF. In section 4, the paper describes how RML processor runs. Finally, sections 5 present the solution's evaluation, conclusions and future work.

2. Related Work

At present, the main research results are the OGC2 model proposed by the Open Geospatial Information Alliance (OGC) on the level of sensing layer and it is the description of sensor data. W3C has also developed SSN ontology. SSN ontology provides a large number of sensor resources and description framework of sensor observation data. Furthermore, taking into account the heterogeneity of sensor data, De and others put forward a semantic description model of Internet of things where resources are for open service. So far, most of the use of semantic technology is based on construction of domain ontology or general ontology design to handle the data processing of the Internet of things, and uses RDF, OWL (Web Ontology Language) and other languages to describe the data.

In the semantic annotation of sense data in IoT, different resources of the Internet of things need to be interrelated. The mainstream method is to generate linked data from the perceptual data stream: Bamaghi and other scholars have built a platform—Sense2web, which can publish linking sensor data. In the Sense2web, semantic annotation of acquired sensor data using the existing resources and attributes or general ontology concepts in the network, and give the sensor data meaning, and then the data meaning of the sensor data is given. After that, link resources with similar sensor description information, and finally, the link sensor data is formed.

Harshal Patni and some others implements a framework of Linked Sensor Data UW for publishing linked sensor data. It uses encoding method of O&M to process the original sensor data and convert it into a resource description framework—RDF format. It is finally stored in a common semantic knowledge base, waiting for the call of the application, but the method has not linked with the related Web data. Cory A. Henson and others implements a real-time data middleware—Linked Stream Middleware OLSM. In this generation method of middleware linking sensor data, the sensor data is annotated by semantics. In Web it can carry on the query of the related data set, and accurately set up link for the related data.

The link sensor data publishing system, which has been realized in China, uses the sensor ontology to add semantic information to the sensor data. It puts forward the related data query processing method of inheriting the concept group and the comparison method of graph similarity based on the heuristic attribute to realize the RDF link of semantic sensor network and data set of related Web [2].

In addition, it is not only a solution based on linked data, but also can be transformed into RDF by mapping the structure and sequence. For relational databases, the mapping language beyond R2RML has been defined and implemented long ago. Similarly, mapping languages are used to support conversion between data in files, spreadsheets and RDF data models. They include the conversion of data from various spreadsheets into the XL Wrap mapping language of RDF, declarative M2 mapping language [3], centering the OWL which converts the data of the spreadsheet to the Web ontology language (OWL), the Tarql4 and Vertere5 that follow the query rules.

3. Mapping Heterogeneous Resources to RDF Using RML

3.1 RDF.

Semantic web is based on RDF and uses semantic technology to deal with and describe the perceived data of the Internet of things. It mainly uses RDF, OWL and other ontology representing language.

RDF is recommended by W3C and popularized as a means to express and exchange semantic metadata. RDF is designed as a common language and method, which provides a way to describe information for information data of Web resources. It can be used to make different data readable and understandable through computer applications. RDF is a relational data model between network resource objects and other objects, and has simple semantics, followed by XML's syntax, structure and XML language compilation rules.

3.2 Representing Method

RDF provides a variety of representation methods, such as the triple method, chart representation, XML representation, etc.

(1) the triple law

The design of RDF is based on ideology: the described resource has a certain attribute, each attribute has a specific attribute value, and a resource refers to all objects that can be identified by the URI on the Web; the value of a resource can be a resource or a literal value, and if the attribute value is a resource, the corresponding attribute can be understood as the relationship between two resources; if the attribute value is a literal type, the attribute is the description of the resource property. The description of resources is a representation of resource attributes and attribute values, which we call declarations. The basic structure of the declaration is the triples of "resource-attribute-attribute value", also known as the triples of "subject-predicate-object". The declaration is used to describe the property owned by a resource. For example:

(Pride and Prejudice) (http://vip. book. sina. com. cn /book/index_37590. html), its author is Austen, Jane. The following triples can be used to describe:

(http://vip. book. sina. com. cn/book/index_37590. html foaf: author “Austen, Jane”.)

the subject of the statement: http://vip. book. sina. com. cn/book/index_37590. html; the predicate of the statement: foaf: author; the object of the statement: Austen, Jane

Among them, foaf is a prefix and represents a namespace that is http://xmlns. com/foaf/0. 1/.

(2) diagram method

RDF is represented by digraph, and digraph includes nodes and arcs. Each statement is composed of nodes and arcs, which are mainly used to represent the subject and predicate, that is, the value of resources or attributes. The arc is the directed edge, and it is used to express the predicate, that is, the attribute itself.

The subject and predicate are described by URI. The object can be a URI, a text, or a blank node, which indicates the relationship between the resource and the resource. The resource has the attribute value of the corresponding predicate, or the function of the connection only in the graph.

As shown in Figure 1, the above triples are described in the form of RDF diagrams.

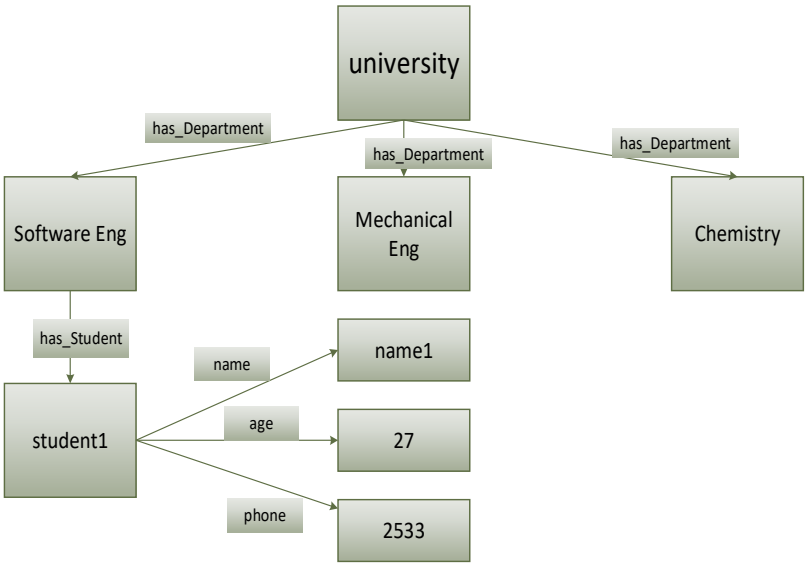

Fig 1. The graph model of RDF data

RDF diagram is a set of triples, whose token representation are recorded as $G = (V, E, IV, IE)$, V represents a set of nodes in all triples, E a set of directed edges in all triples, IV and IE a label function on V and E respectively. $subject(T)$ represents the set of subjects in all triples, $object(T)$ represents the set of all objects in the triples. Functions of $\text{from}()$ and $\text{to}()$ give the starting point and end point of the directed edge [4].

$$V = \{Vx \mid x \in subject(T) \cup object(T)$$

$$E = \{e_{s,p,o} \mid (s, p, o) \in T\}$$

$$l_v(V_x) = \begin{cases} (x, d_x) & \text{if x is a literal（} d_x \text{ is the form of data）} \\ x & else \text{ if} \end{cases} \quad (1)$$

$$from(e_{s,p,o}) = v_s$$

$$to(e_{s,p,o}) = v_o$$

$$l_E(e_{s,p,o}) = p$$

3.3 R2RML.

The R2RML mapping involves a logical table (logical table) that retrieves data from a relational database. The logical table can be a base table, a view, or a legitimate SQL query (called a R2RML view) of a relational database. Each logic table is converted to RDF by triple mapping (triples map), that is, every row instance data in the logical table is mapped to several RDF triples, which includes:
(1) subject map: lines of the logical table corresponding to the public subject of all RDF triples is usually an IRI generated by the primary key (PK) column of the table;
(2) multiple predicate-object map: each mapping is made up with the predicate map and object map or referencing object map. The structure of the R2RML map is shown in figure 1[5]. By default, all RDF triples will form a default graph of the RDF dataset; also, some RDF triples can be put into a named graph through the mechanism of graph map.

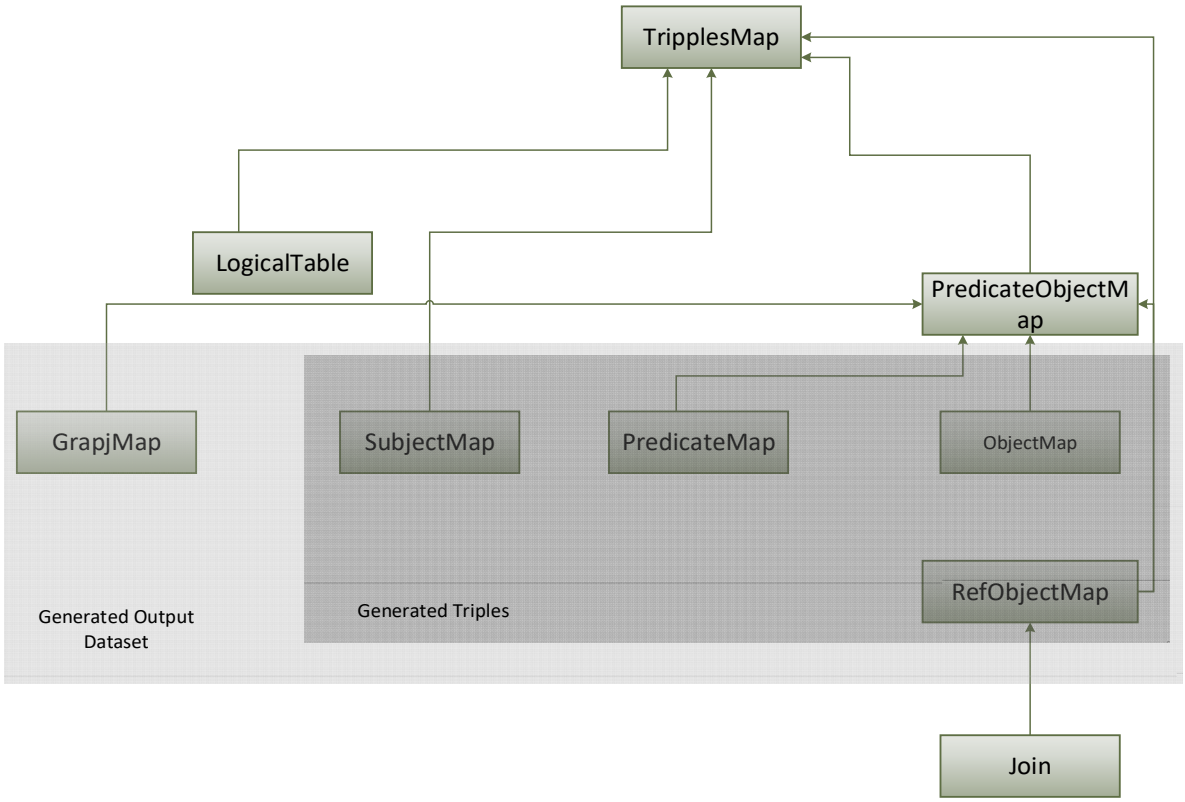

Fig 2. R2RML mapping structure

The entire R2RML mapping of a relational database consists of a mapping document, usually in the UTF-8 encoding format, written in the syntax format of Turtle [6], and the document extensional name is. TTL. The basic function of a R2RML processor is to map the instance data of relational database to an RDF dataset based on the definition in the R2RML mapping document, in which the R2RML mapping algorithm is the core [7].

3.4 RML.

RDF mapping language (RML) is a general mapping language, which defines the general mapping rules from heterogeneous data structure and serialization to RDF data model. RML is defined as a superset of W3C standardized mapping language—R2RML, which can map data from relational databases to RDF data models. RML defines data mapping in other formats, thus extending the applicability and the scope of R2RML. RML can also define rules to map any heterogeneous data, such as DB, CSV, XML and JSON, to RDF data models. A prototype of RML processor is a mapping driven model based on Java implementation.

In addition to reference to the special database in the core model, RML has the same mapping rules as R2RML. The main difference between the two is the input data. In R2RML, the input source restricts a specific database, and the use of RML can have a wide input source.

Fig 3. Mapping sources without and with RML

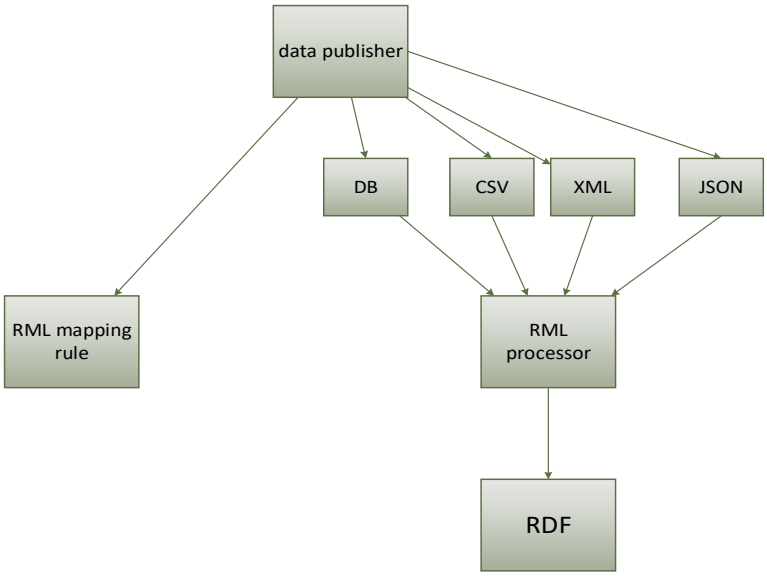

Fig 4. Mapping schematic diagram of RML

The definition of RML mapping follows the same syntax as R2RML. The RML vocabulary namespace is http://semweb.mmlab.be/ns/rml#, and the prefix is rml. More details about the RML mapping language can be found in http://semweb.mmlab.be/rml. Defining and using RML mapping requires users to provide an effective and well formatted input dataset and mapping rules (mapping documents). According to the mapping, the RDF data model (data output) is used to generate data representation. Data cleaning is beyond the scope of language definition and should be implemented ahead of time if necessary.

Triple mapping: triple mappings define rules that map to RDF triples. Triple mapping includes logical resources, subject mapping and predicate-object mapping.

Logical resources include the introduction of input sources, the concretization of how to introduce data rules and iterators about how to iterate data. The following rules are predefined but not limited to: ql: CSV, ql:CSS3, ql: JSONPath, rr:SQL2008 and ql: XPath.

```
RML Logical Source

<#PersonMapping>
  rml:logicalSource [
    rml:source "People.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.[*].Person" ].
```

Fig 5. RML Logical Source

Subject mapping: subject mapping includes how to define the subject of each triple and its optional type of URI.

Predicate-object mapping: including predicate mapping and object mapping. they describe how predicates are introduced and how the object is introduced respectively.

```
Predicate Object Map

<#PersonMapping>
  rr:predicateObjectMap [
    rr:predicate ex:name;
    rr:objectMap [ rml:reference "name" ] ].
```

Fig 6. Predicate-Object Map

The following is a case of JSON document as input and output in RDF form:

```
Input in JSON format (People.json)

[ ...   { "Title": "Multimedia Lab",
          "People": [ { "name": "Anastasia",  "surname": "Dimou" },
                      { "name": "Ruben", "surname": "Verborgh" } ],
        { "Title": "Media and ICT",
          "People": [ ... ] } ,  ... ]
```

Fig 7. Inputting in JSON format

```
Output RDF representation

ex:Anastasia_Dimou a ex:Person ;
  ex:name "Anastasia".

ex:Ruben_Verborgh a ex:Person ;
  ex:name "Ruben".
```

Fig 8. Outputting in RDF representation

4. RML Processor

Unlike R2RML, implementing a processor for RML is more complex, since it deals both with tabular (relational databases and CSV) and hierarchical sources (XML and JSON). Therefore, it demands a scalable and abstract approach to support the extracts of data in different structures (tabular and hierarchical) and different formats (e.g., XML and JSON serializations of hierarchical resources) in a uniform way [8].

While RML syntax is used for the mapping document definition, in order to deal with the a fore mentioned extraction's caveats, RML relies on expressions in a target language (rml: reference Formulation). Such an expression is used wherever values need to be extracted from the source, namely whenever an RDF Term Map or an iterator (rml: iterator) appears. To ensure consistency, the expression should be a valid expression according to the language specified in the Triples Map (rml: reference Formulation). In order to deal with these embedded expressions, an RML processor is required to have a modular architecture where the extraction and mapping modules are executed independently of each other. When the RML mappings are processed, the mapping module deals with the mappings'execution as defined at the mapping document in RML syntax, while the extraction module deals with the target language expressions. An extractor corresponding to the specified target language executes the expression and returns the specified value. Therefore, the role of the extractor is limited in parsing the defined source and providing to the mapping module the corresponding extract of data as specified.

An RML processor can be implemented using two alternative models: mapping-driven or data-driven. In the former case, the processing is driven by the mapping module. It requests an extract of data from the extraction module, considering the iteration pattern specified at the Logical Source. In the latter case, the processing is driven by the extraction module. It passes an extract of data to the mapping module, which applies the mapping rules valid for the particular extract. The expressivity of such languages is usually very high. However, to limit complexity and increase efficiency of implemented processors, a well-defined set of constraints is included in the RML definition. This advocate streaming solutions, since datasets do not always fit in the processor's memory. A side effect of a streaming approach, is the inability to support some features of expression languages. For instance, XPath has lookahead functionality that requires access to data which is not yet known. Nevertheless, in practice, most of the expressions only require functionality within this subset. As a result, the W3C XSLT3.0 Working Draft [9] already mentions a streaming specification. For functionality not supported by streaming, a fallback mechanism to in-memory processing can be provided. As a proof of concept, we created a Java implementation of an RML processor12. In the

remainder of this section, we describe the processing of Triples Maps in RML, disambiguating further the processing of Referencing-Object Maps.

An RML processor can be implemented using two alternative models: mapping-driven, data-driven or in a hybridic fashion following any combination of the two solutions that turns the processor to better perform.

Mapping-driven. In this model, the processing is driven by the mapping module. The processor processes each Triples Maps in a consecutive order. Based on the defined expression language, each Triples Map is delegated to a language specific sub-extractor. For each Triples Map, its delegated sub-extractor iterates over the source data as the Triples Map's Iterator specifies. For each iteration the mapping module requests an extract of data from the extraction module. The defined Subject Map and Predicate-Object Maps are applied and the corresponding triples are generated. The execution of dependent Triples Maps, because of joins, is triggered by the Parent Triples Map and a nested mapping process occurs.

Data-driven. In this model, the processing is driven by the extractor module, namely the data sources. The processor extracts beforehand the iteration patterns, if any, from the Triples Maps. Each defined dataset is integrated by its language-specific sub-extractor. Based on the defined expression language and the iterator, each Triples Map is delegated to a specific sub-mapper. For each iteration, a data extract is passed to the processor, which in turn, delegates the extract of data to the corresponding sub-mapper. The defined Subject Map and Predicate-Object Maps are applied and the corresponding triples are generated. The execution of dependent Triples Maps, because of joins, is triggered by the Parent Triples Map and a nested mapping-driven process occurs.

5. Summary

In this paper, except describing the rule and basic knowledge of RML mapping language, we presented the RML processor for mapping heterogenous hierarchical sources into RDF using RML.

The RML processor can propose mapping recommendations of enhanced quality as it builds a more comprehensive understanding of the domain. can propose mapping recommendations of enhanced quality as it builds a more comprehensive understanding of the domain. An RML based implementation defines triples of different subjects for a certain data extract, achieving semantically richer output. The users can also get instant feedback on their model by testing which triples are generated. For our evaluation, we reused the same RML mappings for the ten input files. An RML processor is source independent and thus easily extended to cover other input sources, as the core of the mappings execution is uniformly implemented and any new extractor can be easily configured.

Overall, whether in semantic quality of the RDF output, reusability of the mapping definitions, or scalability of mappings' execution, RML processor can satisfy users' demand. Our solution relying on RML mapping formalization is optimal as semantically richer and better interlinked output is achieved, while mappings are reused for sources describing the same domain and are interoperable across different tools.

For the future work, RML processer can be more efficient than before, and try to apply the prototype to visual dashboards. Next, we can develop a protocol about that different sensors can be recognized by computers once they are accessed. That is about rock-bottom protocol of data format. Finally, after combining the protocol and RML processor, it can truly realize automatic identification and management of heterogeneous and massive data in IoT.

References

[1]. Yafei Ding, Guanyu Li, Hui Zhang. The research of semantic collaboration method based on semantic space in semantic Internet of things. Computer application and software.Vol.33(2016) No.02,p.17-22.

[2]. Linna Cuan: Link sensor data generation system in semantic Internet of things (Master's degree, Maritime Affairs University of Dalian, China 2016).

[3]. MULLER H, CABRAL L, MORSHED A, et al. From RESTful to SPARQL: a case study on generating semantic sensor data. Proceedings of the 6th international Workshop in Semantic Sensor'Networks in Conjunction with 1SWC. Amsterdam: lOS Press. 2013, p.51-66.

[4]. Jicheng Song: Research and implementation of massive RDF data storage and query technology (Master's degree, Beijing University of Technology, China 2013).

[5]. DAS S, SUNDARA S, CYGANIAK R (Eds.). R2RML: RDB to RDF Mapping Language (W3C 335 Recommendation) (2012): http://www.w3.org/TR/r2rml.

[6]. BECKETT D, BERNERS-LEE T, PRUD'HOMMEAUX E, CAROTHERS G (Eds.). Turtle - Terse RDF Triple Language (W3C Working Draft) (2012).: http://www.w3.org/TR/turtle/.

[7]. Shufeng Zhou, Ching Xu. Mapping algorithm to resource description framework based on R2RML based relational database. Beijing: Chinese scientific and Technological Paper online (2013): http://www.paper.edu.cn/releasepaper/content/.

[8]. Anastasia Dimo, Miel Vander Sand, Jason Slepicka, et al. Mapping Hierarchical Sources into RDF using the RML Mapping Language. IEEE International Conference on Semantic Computing. Newport Beach, 2014, p.151-158.

[9]. Dimou, Anastasia, Vander Sande, Miel & Colpaert, et al. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. CEUR Workshop Proceedings (2014).