

Linked Sensor Data Generation using Queryable RML Mappings^{*}

Pieter Heyvaert, Ruben Taelman, Ruben Verborgh, and Erik Mannens

Ghent University - iMinds
pheyvaer.heyvaert@ugent.be

Abstract. As the amount of generated sensor data is increasing, semantic interoperability becomes an important aspect in order to support efficient data distribution and communication. Therefore, the integration of (sensor) data is important, as this data is coming from different data sources and might be in different formats. Furthermore, reusable and extensible methods for this integration are required in order to be able to scale with the growing number of applications that generate semantic sensor data. Current research efforts allow to map sensor data to Linked Data in order to provide semantic interoperability. However, they lack support for multiple data sources, hampering the integration. Furthermore, the used methods are not available for reuse or are not extensible, which hampers the development of applications. In this paper, we describe how the RDF Mapping Language (RML) and a Triple Pattern Fragments (TPF) server are used to address these shortcomings. The demonstration consists of a micro controller that generates sensor data. The data is captured and mapped to RDF triples using module-specific RML mappings, which are queried from a TPF server.

1 Introduction

In the Internet of Things paradigm, many real-world objects are connected through the Internet [1]. In most cases this is facilitated by using sensors, that collect data. Over the years these sensors have reduced in cost, and, hence, are easier to obtain for consumers. This resulted in hardware projects, such as Tessel¹ and Espruino², that offer modules that generate sensor data. Corcho and García-Castro [2] state that the addition of semantics to this data improves its understandability, management and usability. Furthermore, they identify a number of accompanying challenges, such as *the integration of (sensor) data* and *the rapid development of applications that produce semantic sensor data*. They conclude that the use of Semantic Web technologies and Linked Data can address these challenges.

^{*} The described research activities were funded by Ghent University, iMinds, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union.

¹ <https://tessel.io/>

² <http://www.espruino.com/>

Research is being conducted towards making the produced data available as Linked Data, specifically as Resource Description Framework (RDF) triples. Publishing sensor data as Linked Data enables finding other related data and relevant information, and facilitates interconnection and integration of data from different communities and sources [3]. Barnaghi and Presser [3] present Sense2Web, a platform to publish sensor data as Linked Data. They map the original data in XML to RDF triples using XSLT. However, when integration is required with data in other data formats, XSLT is not usable, as it is only limited to XML [4]. This hampers the integration with (sensor) data in other formats, such as the JSON format used by Schor et al. [6]. Patni et al. [5] developed an API that maps the original XML data to RDF. Similar to XSLT, the API is limited to data in the XML format. Furthermore, the mappings are not accessible, as users can only use the API. Therefore, when the mappings need to be updated or extended for a specific use-case, a new mapping needs to be created, which hampers the rapid development of applications.

In the demo, we will show how these challenges can be addressed. We will generate RDF based on the sensor data stream of modules connected to a Tessel micro controller. The sensor data, available as JSON objects, is mapped to RDF using the RDF mapping language (RML) [4]. Furthermore, we publish the mappings via a Triple Pattern Fragments (TPF) server [7]. This allows applications to query the correct module-specific mappings on demand, which improves the mappings' reusability and allows users to build upon them.

2 Challenges

In this section, we discuss how the two challenges are addressed by using RML mappings and making them available through a TPF server.

Integration of (Sensor) Data We use RML to define how sensor data is mapped to RDF triples, because RML supports multiple heterogeneous data sources. Data might come from different sensors that each use a different data format. The use of different solutions for each data format to generate triples requires users to install, learn, use and maintain these solutions. This hampers the generation of triples. Furthermore, the data can come from multiple sensors and other sources. Therefore, a solution is required that allows to integrate multiple data sources at the same time. RML solves these two problems by offering a declarative way to define how data in multiple heterogeneous data sources is mapped to RDF triples.

Rapid Development of Applications We use a TPF server to publish the RML mappings, because this allows applications and users to reuse the existing mappings. The rapid development of applications that need to map (a stream of) sensor data to RDF triples is only possible when existing mapping methods are reusable and extensible. RML mappings can be published using any Linked Data Fragments (LDF) server, because the mappings are expressed as RDF triples. This

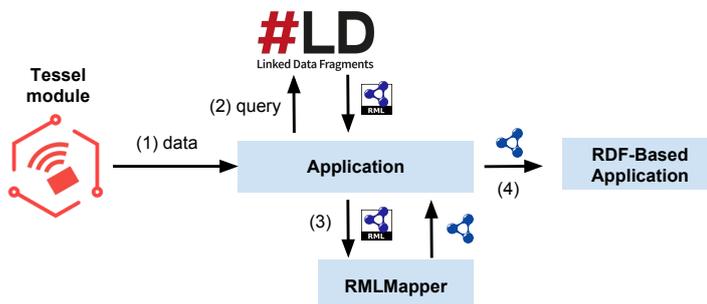


Fig. 1: (1) Sensor data is captured from the Tessel RFID module, as JSON objects; (2) the correct RML mappings are queried from the TPF server; (3) the RMLMapper is used to execute the mappings to generate RDF triples; (4) the application outputs the triples.

allows applications to dynamically query the server for sensor-specific mappings on-demand. Additionally, when the mappings are updated, these applications can query for the new version. Subsequently, when new RDF triples are generated, the new mappings are used. Furthermore, when users need to update or extend the mappings, they can do so without the need to define a completely new mapping that includes their changes, as opposed to the API provided by Patni et al. [5] which hides the mapping from the users. To update the mappings, first, users query the server for the required mapping. Next, they update or extend the mapping. Optionally, they can be published the mapping again, so they become reusable for others. Though, RML mappings can be published using any LDF server, we choose a TPF server instead of, e.g., a SPARQL endpoint, to ensure availability if we provide public access to these mappings [7].

3 Demo

Our demo consists of four steps, as shown in Fig. 1. First, the data of a Tessel RFID module is captured by the application as JSON objects:

```

1 { "device": "TM-00-04-f000da30-006d4744-20bc2586",
2   "module": "rfid-pm532",
3   "uid": "3c527c00" }
```

Each object contains the unique id of the RFID card that is tapped on the module. Second, the `ldf-server`³, an implementation of a TPF server, is dynamically queried for the module-specific mapping. This showcases the rapid development of applications as developers can reuse the existing mappings. Additionally, the mappings can be cached to improve performance. Besides mapping the sensor data to RDF triples, this mapping also maps the location of the sensor, which is available in an XML file:

³ <https://github.com/LinkedDataFragments/Server.js>

```

1 <sensors>
2   <sensor id="TM-00-04-f000da30-006d4744-20bc2586">
3     <lat>50</lat>
4     <long>41</long>
5   </sensor>
6 </sensors>

```

This showcases the integration of multiple heterogeneous (sensor) data sources to triples. Third, the mapping is executed by the RMLMapper⁴, which produces the triples. Finally, they are outputted to the users:

```

1 @prefix ex: <http://www.example.com/>.
2 @prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
3
4 ex:TM-00-04-f000da30-006d4744-20bc2586 a <https://w3id.org/saref#Sensor> ;
5   <http://www.schema.org/location> ex:loc/50_41 ;
6   <http://dbpedia.org/ontology/access> "3c527c00".
7
8 ex:loc/50_41 geo:lat "50"; geo:long "41".

```

During the demo, users will be able to see the original data stream in the JSON format, the mapped RDF triples, and the RML mappings, provided by the TPF server. Furthermore, users can update the mappings, and see the changes when new RDF triples are generated. A screencast of the demo can be found at <http://users.ugent.be/~pheyvaer/sensor2rdf/>. The open source code is available at <https://github.com/mmlab/demo-sensor2rdf>.

References

- [1] Melanie Swan. Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. *Journal of Sensor and Actuator Networks*, 1(3):217–253, 2012.
- [2] Oscar Corcho and Raúl García-Castro. Five challenges for the Semantic Sensor Web. *Semantic Web*, 1(1, 2):121–125, 2010.
- [3] Payam Barnaghi and Mirko Presser. Publishing Linked Sensor Data. In *Proceedings of the 3rd International Conference on Semantic Sensor Networks-Volume 668*, pages 1–16. CEUR-WS.org, 2010.
- [4] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Workshop on Linked Data on the Web*, 2014.
- [5] Harshal Patni, Cory Henson, and Amit Sheth. Linked Sensor Data. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pages 362–370. IEEE, 2010.
- [6] Lars Schor, Philipp Sommer, and Roger Wattenhofer. Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 31–36. ACM, 2009.
- [7] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web. *Journal of Web Semantics*, 37–38:184–206, 2016. ISSN 1570-8268. doi: 10.1016/j.websem.2016.03.003. URL <http://linkeddatafragments.org/publications/jws2016.pdf>.

⁴ <https://github.com/RMLio/RML-Mapper>