Sensor-based Linked Open Rules (S-LOR): An Automated Rule Discovery Approach for IoT Applications and its use in Smart Cities

Amelie Gyrard, Martin Serrano, Joao Bosco, Soumya Kanti Datta, Muhammad Intizar

HAL Id: emse-01644351
https://hal-emse.ccsd.cnrs.fr/emse-01644351

Submitted on 22 Nov 2017

Sensor-based Linked Open Rules (S-LOR): An Automated Rule Discovery Approach for IoT Applications and its use in Smart Cities

Amelie Gyrard*, Martin Serrano†, Joao Bosco Jares†, Soumya Kanti Datta**, Muhammad Intizar Ali†
*Univ Lyon, MINES Saint-Etienne, CNRS, Laboratoire Hubert Curien, France;
‡EURECOM, France; †Insight Center for Data Analytics, Ireland
*amelie.gyrard@emse.fr, **dattas@eurecom.fr, †{firstname.lastname}@insight-centre.org

ABSTRACT

This paper introduces an automated rule discovery approach for IoT device data *(S-LOR: Sensor-based Linked Open Rules)* and its use in smart cities. S-LOR is built following Linked Open Data (LOD) Standards and provides support for semantics-based mechanisms to share, reuse and execute logical rules for interpreting data produced by IoT systems. S-LOR follows LOD principles for data re-usability, semantics-based reasoning and interoperability. In this paper, S-LOR main capability is demonstrated in the context of enabling semantics-based reasoning mechanisms and tools according to application-demand and user requirements. S-LOR *(i) supports an automated interpretation of IoT data by executing rules, and (ii) allows an automated rule discovery interface.* The implemented S-LOR mechanism can automatically process and interpret data from IoT devices by using rule-based discovery paradigm. Its extension called Linked Open Reasoning (LOR) enables and encourages re-usability of reasoning mechanisms and tools for different IoT smart city applications. The use cases described in this paper fits in the domain of smart city applications within Internet of Things deployed systems.

Keywords

Semantic Web of Things, Internet of Things, Semantic Web Technologies, Reasoning, Knowledge.

1. INTRODUCTION

Cities around the world are deploying millions of devices to develop smart cities services. These devices (also called things) are being connected to Internet and the Web which sets the stepping stone for the Internet of Things (IoT) and Web of Things (WoT)[1] respectively. In the other side, Things are producing data which are sent to the Web for

----

[1] http://webofthings.org/

data analytics. Interpreting data is a time-consuming process and constantly redesigned in all smart city applications which leads to interoperability issues between applications. What is missing, is a methodology that enables reusability by sharing information and executing logical rules.

Web technologies such as Web Services have been used to process data. Recently, with the emergence of IoT, possibilities have been explored for using Semantic Web techniques to get data from devices, structure and unify data, and add value to data with Data Analytics and reasoning mechanisms to build innovative smart cities applications for example. Application Programming Interfaces (APIs) based on web technologies such as RESTful Web services built on top of the HTTP protocol can be used to exchange data. Web API functionalities have been exploited to connect the things to the web. For instance, a thing can be a NetAtmo[2] thermometer sensor sending data to the Web. Most of the time, the data is just an integer which is visualized in a user-friendly interface and interpreted by humans. Resolving the meaning of data is a challenging problem and without processing it the data is invaluable [12]. There is a necessity to interpret, analyze and understand sensor data to perform machine-to-machine communications in IoT [13]. The main challenge would be to enable machines to interpret data to build smart applications. The Semantic Web community aims to structure data and make it machine-understandable by using a set of vocabularies and Linked Data[3] mechanisms to associate data to other datasets to get meaningful information. Finally, the Data Analytics community aims to deduce meaningful information from datasets in order to build new services or applications by applying reasoning mechanisms. The main novelty of this paper is to investigate complementary communities which leads to an innovative approach exploiting Web technologies (Web Services, Semantic Web) and semantic-based reasoning approaches in the context of smart cities.

To enable analytics on IoT data, we are addressing the following research challenges:

Unifying data provided by different datasets. Semantic Web technologies are used for enabling: (1) Unification of data with a taxonomy, (2) Interpretation of data with a logic-based reasoning engine and rule datasets, and (3) Reusing domain knowledge. In the context of Web of Things (WoT), we have designed the M3 ontology, an extension of the W3C Semantic Sensor Networks ontology.

----

[2] https://www.netatmo.com/product/weather/
[3] http://linkeddata.org/

Domain knowledge reusability. There are several initiatives but most relevant ones are Linked Open Data and Linked Open Vocabularies (LOV)[4] approaches designed by the Semantic Web community and open source based approaches are encouraging the reuse of the knowledge already designed by experts. In order to enable the semantics-based data analytics approach applied to sensors, we designed Sensor-based Linked-Open Rules (S-LOR) to also encourage the reuse and interoperability of existing rules.

Unifying and reusing logical rules. The Jena framework[5] can be employed to develop semantic-based applications and it includes the Jena inference engine to execute Jena rules to interpret semantic sensor datasets. There is a need to design a rule pattern compliant with the taxonomy mentioned above. The rule pattern will be employed within the rule datasets.

IoT device data interpretation. A comprehensive review of the state of the art has been done to learn mechanisms to do analytics on data. We have studied six research fields regarding semantics-based reasoning approaches. All approaches have the common goal of enriching data: (1) Logical reasoning, (2) real-time and Linked Stream Processing approaches, (3) sharing and reusing methods, (4) semantics-based machine learning approaches, (5) semantics-based distributed approaches, and (6) semantics-based recommender systems. This analysis enables designing a high-level semantics-based reasoning service to encourage the reuse of existing tools with a set of reasoning mechanisms fitting different user's needs.

Development time optimization for designing smart city applications. Mechanisms to reduce the time used in designing applications need to/should be designed to share and reuse logical rules to deduce meaningful information from sensor data deployed within cities. The architecture design should be generic enough for other domains. This mechanism would reduce the development time of smart city applications.

In this paper, we introduce Sensor-based Linked Open Rules (S-LOR), an innovative approach for a logic-based reasoning engine based on Semantic Web technologies to deduce meaningful information from IoT/WoT data. S-LOR aims to share the ways of interpreting data in an interoperable way. Its extension, called Linked Open Reasoning (LOR), aims to encourage the reusability of semantics-based reasoning approaches. This paper extends previous work [6, 5, 14], by focusing on S-LOR architectures and its components.

The rest of the paper is structured as follows: Section 2 explains the open-source based approach, called Sensor-Based Linked Open Rules (SLOR): the architecture and its implementation. An extended version, called Linked Open Reasoning (LOR) is explained in Section 3. Section 4 presents different use cases of our approaches. Section 5 evaluates the S-LOR approach. Section 6 shares lessons learnt and potential extensions of the approach. Section 7 concludes the paper and highlights future work. Due to the lack of space, we do not have a specific section for the related work regarding data analytics combining both Semantic Web technologies and reasoning mechanisms. The most relevant work is mentioned in Section 3. The readers can refer to the semantics-based reasoning approach state of the art in [5, 14].

---

[4]http://lov.okfn.org/dataset/lov/
[5]https://jena.apache.org/

2. AN OPEN-SOURCE BASED APPROACH FOR SHARING AND REUSING RULES WITH S-LOR

Sensor-based Linked Open Rules (S-LOR) is an open-source and rule-based reasoning engine aiming to share, reuse and execute rules to interpret sensor data. In this section, we describe the architecture, its components and specifications for the implementation.

2.1 Architecture

The novelty contribution of this paper is the comprehensive architecture of S-LOR. Figure 1 is a high level architecture overview of S-LOR. S-LOR enables the interaction of users such as web-based application developers with a rule-based and semantic reasoning.
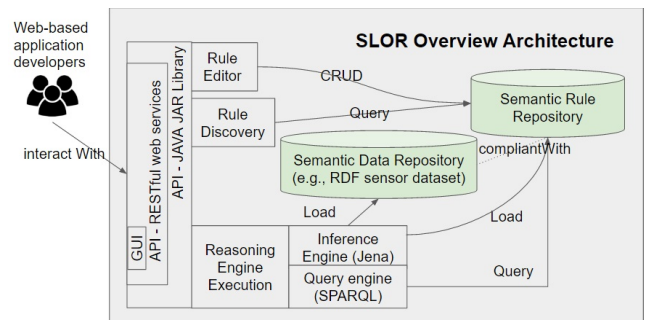

Figure 1: Reasoning Architecture Overview

S-LOR architecture comprises GUIs, APIs and the core components as follows. Graphical User Interfaces (GUIs) to enable users to interact with our web-based approach. Application Programming Interfaces (APIs) either used by the GUI or directly by developers and are designed according to RESTful principles. APIs can be released as libraries as well such as JAVA jars. The core components of the architecture are as follows: (1) The **Reasoning Engine Execution** can be done in two different ways, either with a rule-based engine such as the Jena inference engine or the SPARQL query engine by executing SPARQL CONSTRUCT rules, (2) the **Rule Discovery** enables the interaction with the Semantic Rule Repository to select a sub-set of relevant rules. This optimizes the reasoning engine process since less rules would be considered to be executed, (3) the **Rule Editor** enables to add new rules to the Semantic Rule Repository including the (Create, Read, Update, Delete) CRUD interactions, (4) the **Semantic Data Repository** store RDF datasets such as sensor datasets triplestores such as JenaTDB for a fast implementation or Virtuoso for addressing scalability issues, and (5) the **Semantic Rule Repository** stores the Jena rules to be later executed by the Jena inference engine.

2.1.1 Discovering rules

S-LOR provides a sensor discovery mechanism to retrieve specific rules classified according to sensor types. For instance, a developer is integrating a heartbeat sensor in his/her application, the sensor discovery mechanism enables retrieving all rules related to heartbeat sensors. Rules enable the interpretation of heartbeat measurements such as tachycardia, low heartbeat, etc.

Figure 2 explains the rule discovery mechanism. Rule consumers such as developers interact with the GUI or directly

with APIs, more precisely our RESTful Web services. In the web service URL, a parameter can be provided with the name of a specific sensor type. The web service executes the Jena ARQ query engine with the SPARQL query to retrieve all rules specific to this sensor type within the Semantic Rule Repository.
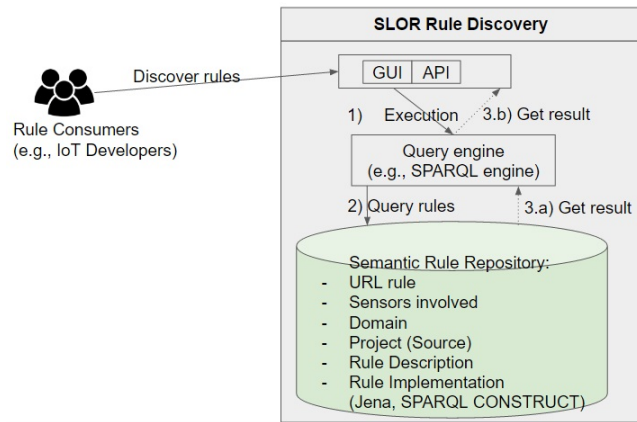

Figure 2: Discovering rules

Listing 1 shows an example of the RDF rule description stored in the Semantic Rule Repository. The rule contains a label, a comment, from which project the rule has been extracted (`m3:fromM2MApplication` property) and referenced within LOV4IoT, which sensors are associated to the rule (`m3:ruleUsingM2MDevice` property), and the URL of the Jena rule implementation (`m3:hasUrlRule` property). M3 refers to the namespace of the M3 ontology[6], an extension of the W3C Semantic Sensor Networks ontology[7] for unifying sensor metadata. M3 ontology can be seen as a dictionary to classify the sensor names, measurements names, applicative domains and units.

```
1  <rdf:RDF
2    xmlns:rdfs="http://www.w3.org/2000/01/rdf−schema#"
3    xmlns:m3="http://sensormeasurement.appspot.com/m3#"
4    xmlns:lorHealth="http://sensormeasurement.appspot.com
        /RULES/LinkedOpenRulesHealth.txt"
5    xmlns:lov4iot="http://sensormeasurement.appspot.com/
        dataset/lov4iot−dataset/"
6  >
7    <m3:Rule rdf:about="MildHypothermia">
8      <rdfs:label xml:lang="en">MildHypothermia</rdfs:
          label>
9      <rdfs:comment xml:lang="en">IF m3:
          BodyTemperature greaterThan 33 m3:
          DegreeCelsius
10         AND lessThan 35 m3:DegreeCelsius THEN
              MildHypothermia</rdfs:comment>
11     <m3:fromM2MApplication rdf:resource="&lov4iot;
          Jara"/>
12     <m3:ruleUsingM2MDevice rdf:resource="&m3;
          BodyThermometer"/>
13     <m3:hasUrlRule rdf:resource="&lorHealth;"/>
14   </m3:Rule>
```

Listing 1: RDF Rule description stored within the Semantic Rule Repository

---

[6] `http://sensormeasurement.appspot.com/m3#`
[7] `http://purl.oclc.org/NET/ssnx/ssn`

Listing 2 is the generic SPARQL query to retrieve all rule descriptions for a specific sensor type. The parameter `?m2mdevice` is automatically replaced according to the request of the user. The URI replacing the m2mdevice parameter should be compliant with he M3 ontology.

```
1  PREFIX rdfs: <http://www.w3.org/2000/01/rdf−schema
       #>
2  PREFIX m3: <http://sensormeasurement.appspot.com/
       m3#>
3
4  SELECT DISTINCT ?ruleLabel ?ruleComment ?
       m2mappli_label ?lorUrl
5
6  WHERE{
7  ?rule m3:ruleUsingM2MDevice ?m2mdevice .
8  #?m2mdevice is automatically replaced by http://
       sensormeasurement.appspot.com/m3#
       BodyThermometer
9  ?rule m3:fromM2MApplication ?m2mappli.
10
11 OPTIONAL{
12   ?rule rdfs:label ?ruleLabel.
13   ?rule rdfs:comment ?ruleComment.
14   OPTIONAL{ ?rule m3:hasUrlRule ?lorUrl }
15   FILTER(LANGMATCHES(LANG(?ruleLabel), "en
        ")).
16   FILTER(LANGMATCHES(LANG(?ruleComment),
        "en")).
17 }
```

Listing 2: SPARQL query to retrieve rules specific to a sensor type

2.1.2 Updating the Rule repository with new rules

Figure 3 explains the interactions between knowledge producers and the semantic rule Repository though either the GUIs or Web APIs.
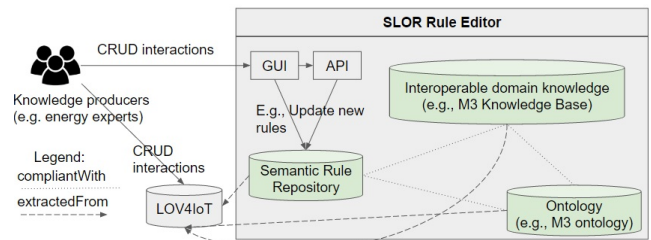

Figure 3: Updating the rule repository with new rules

To demonstrate the feasibility of the S-LOR approach, and due to unexpected interoperability issues (heterogeneous rule languages, rule engines and their implementations), we manually extracted rules from the domain knowledge referenced within Linked Open Vocabularies for Internet of Things (LOV4IoT)[8]. Figure 3 highlights the need of interoperability between the ontology repositories, the rule repositories and other knowledge bases.

2.2 Implementation: Web APIs and GUIs

A web service has already been implemented for the sensor discovery component, depicted in Listing 3. The `sensorType` variable needs to be compliant with the M3 ontology. The

---

[8] `http://sensormeasurement.appspot.com/?p=ontologies`

Web service returns the SPARQL result (shown in Listing 2) returned by the Jena Query engine with all rules specific to the sensorType as shown in Listing 3. Name of the rule is given within the `ruleLabel`, description of the rule within the `ruleComment`, and the access to the Jena rule implemented with `lorURL`.

```
1  #Web service to call:
2  #http://sensormeasurement.appspot.com/slor/rule/{
       sensorType}
3
4  #Example (sensorType is replaced by BodyThermometer
       compliant with the M3 taxonomy):
5  #http://sensormeasurement.appspot.com/slor/rule/
       BodyThermometer
6
7  #Result:
8  <result>
9  <binding name="ruleLabel">
10 <literal xml:lang="en">HighFever</literal>
11 </binding>
12 <binding name="ruleComment">
13 <literal xml:lang="en">
14 IF m3:BodyTemperature greaterThan 38 m3:DegreeCelsius
       AND lessThan 39 m3:DegreeCelsius THEN
       HighFever
15 </literal>
16 </binding>
17 <binding name="lorUrl">
18 <uri>
19 http://sensormeasurement.appspot.com/RULES/
       LinkedOpenRulesHealth.txt
20 </uri>
21 </binding>
22 </result>
23 <result>
```

Listing 3: SPARQL result example returned within the web APIs

Two GUIs have also been developed using this Web service. The first GUI enables retrieving all rules for a specific sensor type. The GUI is available online[9]. The second GUI enables the interaction with applicative domains. Once the domain is chosen by the developers, all sensors related to this domain are retrieved and then the developers select a specific device to get all the rules relevant for the sensor selected. The second GUI is accessible as well online[10].

3. TOWARDS THE LINKED OPEN REASONING ARCHITECTURE

We have done a deep analysis of the state of the art which leads to the classification of semantic-based reasoning mechanisms approaches and available tools [5, 14].

Based on this analysis, we extended the architecture to cover several kind of semantics-based reasoning mechanisms and tools by designing a single endpoint for users to do analytics on data as depicted in Figure 4. This endpoint would be a kind of recommender system, a suggestion tool to guide users in doing analytics on data. We refer the readers to the top layer within the architecture with the S-LOR component explained previously in Section 2.

Figure 4 shows a semantics-based reasoning architecture comprising 3 layers: The first layer shown at the bottom
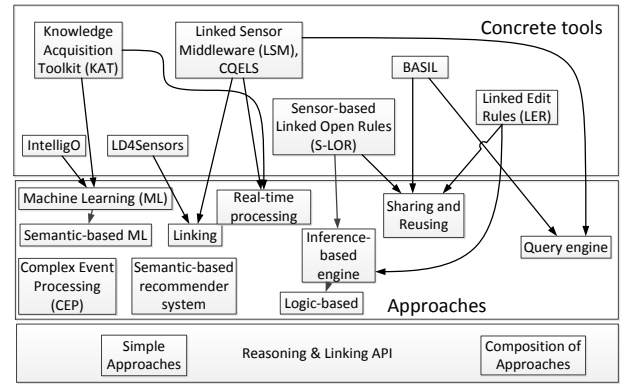
Figure 4: Reasoning Architecture API

provides API and Web services to access the reasoning approaches. This layer provides access to simple reasoning services or complex services which are a composition of existing services.

The second layer shows the a set of semantics-based reasoning approaches. Due to the lack of space, we refer our reader to previously published literature [5, 14]. We referenced, classified and analyzed 7 different reasoning methods, that we remind here. **Machine Learning** is a popular approach. However, it requires huge datasets for the training phase which are frequently unavailable at initial stages. Further, the choice of the algorithm is not easy for non experts. **Real-time** is an important feature when real-time aspect is required. **Linking** is an approach to enrich data with domain knowledge such as Linked Data. **Complex Event Processing (CEP)** is an approach for doing analytics on complex sensors such as accelerometer when simple reasoning mechanisms cannot be used. **Logic-based reasoning engine** such as rule-based engine enables deducing new information from IoT data with an easy learning curve compared to machine learning or CEP. **Sharing and reusing approaches** is an approach encouraging to share and reuse the way to do analytics on data. **Semantics-based distributed reasoning** is an approach to deal with scalability when big data analytics is required.

The third layer suggests concrete tools that can be reused to interpret data when the tools are available on the web. Each tool is associated to an approach introduced above. **Knowledge Acquisition Toolkit (KAT)** [4] which is a machine-learning approach dealing with real-time data. **Linked Sensor Middleware (LSM)** [10] deals with real-time data and enables linking between heterogeneous datasets. **IntelligO** [9] is a machine-learning approach using the Parsimonious Covering Theory (PCT). However, this tool cannot be tested since it is not shared on the web. **Linked Data for Sensors (LD4Sensors)** [11] enables linking of datasets. **Sensor-based Linked Open Rules (S-LOR)** is a logic-based reasoning engine presented in Section 2.
Open Reasoning for IoT-enabled Smart City Applications: Linked Open Reasoning Architecture can support for recommendation of right tools and techniques for the right analytical & reasoning job within smart city applications. Consider a real-time IoT stream analytical framework for smart cities (e.g. CityPulse[11], which provides support

for smart city applications using IoT stream processing and analytics). CityPulse provides a set of open source tools to acquire, process & analyze IoT data streams. We could benefit from LOR while designing travel planner and parking space finder scenarios of CityPulse project. LOR can consider application requirements such as, (i) support for real analytics, (ii) machine learning for event detection, (iii) complex event processing for integrating multi-modal datasets, and (iv) rule-based reasoning for re-usable application logic in several domains. Based on the given requirements, LOR could provide recommendations to use a combination of two concrete tools, (i) Knowledge Acquisition Toolkit (KAT), and (ii) Linked Sensor Middleware (LSM).

4. USE CASES

The Sensor-based Linked Open Rules (S-LOR) approach is applied to **three** projects: (1) S-LOR has been integrated within a framework to assist developers in designing Semantic Web of Things applications, (2) S-LOR has been integrated within Android-based powered devices, and (3) S-LOR architecture is relevant for the FIESTA-IoT project.

4.1 IoT development within M3 framework and SWoT generator

S-LOR has been integrated within the M3 framework, a framework to assist users in designing Semantic Web of Things applications [8]. M3 provides the workflow from data producers to data consumers and includes the Semantic web of Things generator component [7], which reuses the set of rules by using the rule discovery mechanisms explained in Section 2.1.1 for two different purposes: (1) rules for the semantic annotation, and (2) rules to deduce new knowledge from sensor measurements. Application templates can be used by developers, a tutorial is available online[12] done for the ISWC2016 Tutorial Hands-on session with the set of slides proving more explanations[13].

4.2 S-LOR Reasoning within Mobile Devices

A prototype of S-LOR research approach explained in Section 2 is running on a Google Cloud Platform™. Because of the tremendous advancement in mobile computing and the ongoing "mobile first" approach from the IoT industry, we have extended the same principle for mobile devices (e.g. Android powered devices). While designing a system that would utilize S-LOR and the M3 framework, we realized that porting the entire M3 framework on a mobile device is not practical due to memory, processing and power demands. Rather loading an application template to accomplish a cross domain application scenario is feasible as that is much lightweight. As a result, we engineered an operational framework as shown in Figure 5.

There are four operational phases in the software framework. The first phase performs a discovery of available internal (mobile device) and external (environmental) sensors from which IoT data can be collected. Then during the second phase, a sensor type and its operating domain is provisioned. The combined information is then forwarded to
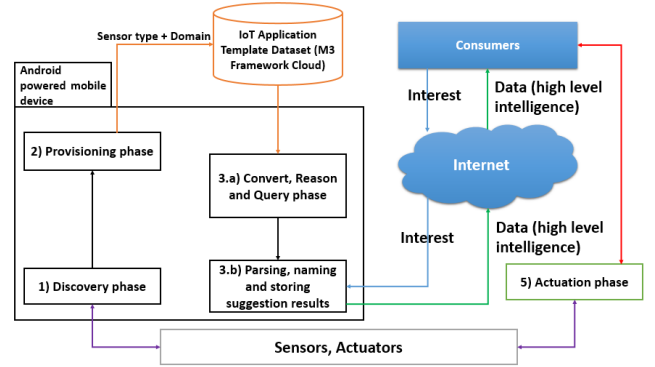

Figure 5: Framework for semantic reasoning in mobile devices

the M3 framework cloud which generates horizontal IoT application development templates involving the sensor type. Depending on the context and target cross domain application, one of the templates is downloaded on the mobile device. This template contains ontologies, datasets and rules necessary for semantic reasoning on the IoT data. Following the reasoning, high level abstraction of the data as well as related suggestions are communicated to the consumers. They can select a suggestion which triggers the final actuation phase. We developed an Android application that accomplishes the discovery, provisioning and reasoning phases with an Android powered device. Instead of using the Jena framework, its lightweight version for Android called AndroJena[14] has been used. During performance evaluation we noticed that the CPU load was within 20 percent and memory requirement within 20-25MB depending on the target scenario [3]. This scenario shows that this approach is efficient and feasible for mobile devices. Such generic framework can be applied to smart city [2], connected car [1] and more IoT domains.

4.3 Sensor Data Analytics within FIESTA-IoT

The S-LOR architecture presented in Section 2.1 can be integrated within the FIESTA-EU project[15] since the FIESTA-IoT ontology includes the M3-lite taxonomy, an extension of the M3 ontology briefly introduced in Section 2.1.1. In the project, we distinguish two kind of users called experimenters: Knowledge producers and Consumers. *Knowledge producers* update the semantic rule repository, a database of interoperable IF THEN ELSE rules explained in Section 2.1. Figure 6 enables experimenters adding more rules to the semantic rule repository. The rules should be compliant with the M3-lite taxonomy. Adding a new rule in the repository is easy. However, dealing with redundancy and overlapping rules is more complicated: correctness and completeness of rules are checked manually. *Consumers* take advantage of enriched data once the reasoning engine has been executed on sensor data stored in the Semantic Data Repository explained in Section 2.1.

Figure 6 shows the reasoning engine employed by experimenters/users. The architecture has been extended from the work published in [14]. FIESTA-IoT Reasoner loads data, a subset of rules from the Semantic Rule Repository and then executes the reasoning engine. The use of the Jena inference engine is easy to execute the Jena rules and de-

---

duce additional information. Then, FIESTA-IoT Reasoner updates the triplestore with additional triples (e.g., inferred data with higher level information) by executing the Jena rule-based engine with a sub-set of Jena rules. Then, experimenters retrieve inferred data stored within the Semantic Data Repository by executing the SPARQL query engine. The query result returns the inferred data to the FIESTA-IoT client (e.g. experimenters). The rule editor (presented in section 2.1) is being implemented within the FIESTA-IoT project.
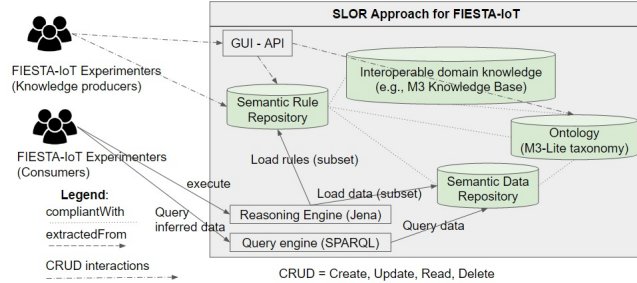

Figure 6: Reasoning within FIESTA-IoT

5. EVALUATION

The evaluation is focused on the validation of the S-LOR *Semantic Rule Repository* by looking at completeness and correctness of rules, that the rules are reliable to interpret IoT data. **Correctness** means that there is no incompatibility between rules. **Completeness** means that all sensor values are covered by a high level information. Table

M3 or SenML domain	M3 or SenML sensor/ measurement name	Description, other names (synonyms)	M3 or SenML Unit	M3 rules
Weather	HumiditySensor/ Humidity	Hygrometer, humidity sensor, moisture sensor, soil moisture probes	Percent	Correctness OK (Conflict resolved with [Kofler 2011] and [Rodriguez 2014]) + Complitude OK (5 rules [Staroch 2013])
Weather	WindDirectionSensor/ WindDirection	Wind direction	DegreeAngle	Complitude OK (5 rules [Staroch 2013]) + Correctness OK
Weather	CloudCoverSensor/ CloudCover	Cloud cover sensor	Okta	Complitude OK (5 rules [Staroch 2013]) + Correctness OK (even with [Kofler 2011])
Weather	SunPositionElevationSensor/ SunElevation	sun position elevation to detect (twilight, day, night, etc.)	DegreeAngle	8 rules [Staroch 2013] **Complitude NO + Correctness NO**

Table 1: Evaluating S-LOR with completeness & correctness

1 shows the M3 classification of some sensor devices within the weather domain. The last column is dedicated to completeness and correctness of rules available within the repository for each device. For instance, humidity rules cover all possible values (completeness) to deduce high level abstractions, and the overlapping (correctness) between different rules extracted is resolved. Regarding the sun position elevation sensor, correctness and completeness are not satisfied yet, but 8 rules are available within the dataset to interpret the measurements.

In the M3 nomenclature[16], for each sensor, we implement M3 rules if we can extract rules referenced within the LOV4IoT dataset. When we implemented the rules, we

checked manually completeness and correctness. A new rule should not overlap with the previous rules. For each sensor, we would like to have rules covering all possible values, to get high-level abstractions in all cases. Moreover, sometimes two different projects design non compatible rules related to the same device. In this case, we choose the work having the more rules related to a specific sensor, and delete the previous rules related to the same sensor. Indeed, if the work has more rule, we consider that the rules are more precise and we can differentiate more abstractions from sensor data.

6. EXTENSIONS AND FUTURE WORK

Future work is mainly about achieving the implementation of all components discussed in our architectures. Our reasoning approach will be applied to a new energy use case within the Smart Energy Aware Systems (SEAS) project[17] and ENGIE[18].

We have identified areas for extensions based on the lessons learnt from our experience:

- SPARQL CONSTRUCT rules instead of Jena rules can increase interoperability and reusability of the rule datasets. Listing 4 shows an example of SPARQL CONSTRUCT rule. This rule deduces the `Flu` concept from the body temperature measurement. The rule links sensor datasets to domain knowledge ontology and datasets such as naturopathy.

- Define associations between URIs to each particular rule. In the current implementation, we classified all rules by domain and then load all rules for a specific domain. For instance, we have a healthcare rule file storing all rules using sensors in the healthcare domain. The different domains related to IoT and WoT have been referenced within the M3 ontology. This improvement will optimize the execution of the reasoning engine, since it enables to load and execute less rules, e.g., the rules relevant for specific sensors deployed within the smart city application.

- The M3 ontology needs to be refactored, every concept and property related to the reasoning mechanism components should be moved to a dedicated reasoning ontology.

- The result of the logical rule (THEN part) is linked to domain ontologies or datasets. We need to define two new properties such as `isDeducing` and `isSuggesting` within the reasoning ontology to create the link between the measurements annotated with M3 and the domain knowledge. This would remove some inconsistencies that we realized once the implementation has been done.

1 **PREFIX** rdf: <http://www.w3.org/1999/02/22−rdf−syntax−ns#>
2 **PREFIX** m3: <http://sensormeasurement.appspot.com/m3#>
3 **PREFIX** nat: <http://sensormeasurement.appspot.com/naturopathy−dataset/>
4 **PREFIX** naturopathy: <http://sensormeasurement.appspot.com/naturopathy#>

```
5
6  #A SPARQL CONSTRUCT has the same behaviour
       than a Jena Rule: Retrieve all ontology URL from all
       projects in the
7  CONSTRUCT {?m naturopathy:hasDisease nat:Flu}
       WHERE {
8  ?m rdf:type m3:BodyTemperature.
9  ?m m3:hasValue ?v.
10 FILTER ( ?v > 38 ).
11 }
```

Listing 4: SPARQL CONSTUCT query to deduce more knowledge and linking ontology and datasets

7. CONCLUSIONS

The Sensor-based Linked Open Rules (S-LOR) approach has been designed to deduce meaningful information from devices sending data to the Web for processing. S-LOR follows Linked Open Data (LOD) design principles from the Semantic Web community.

S-LOR is a rule-based reasoning engine implemented to address interoperability and data analysis of sensor datasets. Linked Open Reasoning (LOR) approach has been designed to select the best data analytics approach according to the need of the data scientists. LOR can guide developers in choosing a specific semantics-based reasoning approach (e.g., machine learning, distributed approaches, real-time processing and recommender systems) and the available implementation.

A methodology can be extracted from the architectural design of S-LOR and LOR, the interaction from users with GUIs and developers with web services to access semantics-based components in a transparent way.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] S. K. Datta, R. P. F. D. Costa, J. Harri, and C. Bonnet. Integrating connected vehicles in Internet of Things ecosystems: Challenges and solutions. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016.

[2] S. K. Datta, R. P. F. da Costa, C. Bonnet, and J. Haerri. oneM2M architecture based IoT framework for mobile crowd sensing in smart cities. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 168–173, June 2016.

[3] S. K. Datta, A. Gyrard, C. Bonnet, and K. Boudaoud. oneM2M Architecture Based User Centric IoT Application Development. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 100–107, Aug 2015.

[4] F. Ganz, P. Barnaghi, and F. Carrez. Automated semantic knowledge acquisition from sensor data. *IEEE Systems Journal*, 2014.

[5] A. Gyrard. *Designing Cross-Domain Semantic Web of Things Applications*. PhD thesis, Telecom ParisTech, Eurecom, April 2015.

[6] A. Gyrard, C. Bonnet, and K. Boudaoud. Helping IoT Application Developers with Sensor-based Linked Open Rules. In *SSN 2014, 7th International Workshop on Semantic Sensor Networks in conjunction with the 13th International Semantic Web Conference (ISWC 2014), 19-23 October 2014, Riva Del Garda, Italy*, 2014.

[7] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano. Assisting IoT Projects and Developers in Designing Interoperable Semantic Web of Things Applications. In *IEEE International Conference on Internet of Things 2015 (iThings)*, 2015.

[8] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud. Cross-Domain Internet of Things Application development: M3 framework and Evaluation. In *FICLOUD 2015, 3rd International Conference on Future Internet of Things and Cloud, August 24-26, Rome, Italy*, 2015.

[9] C. Henson, A. Sheth, and K. Thirunarayan. Semantic Perception: Converting Sensory Observations to Abstractions. *Internet Computing, IEEE*, 2012.

[10] D. Le-Phuoc, H. Q. Nguyen-Mau, J. X. Parreira, and M. Hauswirth. A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012.

[11] M. Leggieri, A. Passant, and M. Hauswirth. inContext-Sensing: LOD augmented sensor data? In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*. Citeseer, 2011.

[12] M. Öspinar. *A Flexible Semantic Service Composition Framework for Pervasive Computing Environments*. PhD thesis, Moddle East Technical University, 2014.

[13] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context Aware Computing for The Internet of Things: A Survey. *Communications Surveys & Tutorials, IEEE*, 2014.

[14] M. Serrano and A. Gyrard. A Review of Tools for IoT Semantics and Data Streaming Analytics. In *The Building Blocks of IoT Analytics - Internet-of-Things Analytics*. River Publishers Series in Signal, Image and Speech Processing, November 2016.