Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation

Amelie Gyrard*, Soumya Kanti Datta†, Christian Bonnet†, Karima Boudaoud‡
*Insight @ NUIG, Galway, Ireland, Email: amelie.gyrard@insight-centre.org
†Eurecom, Sophia Antipolis, France, Email: {bonnet, dattas}@eurecom.fr
‡Laboratoire I3S-CNRS/UNSA, Sophia Antipolis, France, Email: karima@polytech.unice.fr

***Abstract*—Internet of Things (IoT) applications are becoming more and more popular but not interoperable with each other. In this paper, we propose the Machine-to-Machine Measurement (M3) framework to: (1) build IoT applications, (2) assist users in interpreting sensor measurements, and (3) combine domains with each other. The M3 framework is based on semantic web technologies to explicitly describe the meaning of sensor measurements in an unified way to ease the interpretation of sensor data and to combine domains.**

***Keywords*-Linked Open Data; Linked Open Rules; Linked Open Vocabularies; Internet of Things (IoT); Machine-to-Machine (M2M); Semantic Sensor Networks (SSN); Semantic Web of Things (SWoT), Applications.**

I. Introduction

Internet of Things (IoT) applications are becoming more and more popular. Machine-to-Machine [4] is a part of Internet of Things to automate the communications between machines. More and more Internet of Things (IoT) projects such as CityPulse[1], Spitfire[2] and READY4SmartCities[3] integrate semantics (i.e., ontology) to ease interoperability of sensor networks. Noy et al. [26] explained in the second step of their ontology development methodology that ontology designers should consider reusing existing ontologies. Although the above authors recommend it, it is not followed by domain experts in the real life. Domain experts could use semantic web tools indexing domain knowledge such as the Linked Open Vocabularies (LOV)[4] [34] catalogue for ontologies, the DataHub[5] project for datasets or semantic search engines such as Sindice[6], Watson[7] and Swoogle[8]. Most of the domain ontologies related to smart cities are not referenced on these tools since domain experts do not publish their ontologies online and do not follow semantic web guidelines. Most of the IoT applications do not semantically interpret M2M data. Some other limitations are that the M2M applications cannot be combined with each other since they are domain-specific and not interoperable as explained by ETSI M2M [23], Chen et al. [5] and Miorandi et al. [25]. Narang Kishor explained "every true IoT application or solution needs cross-domain expertise"[9] while the authors Gubbi et al. pointed out the necessity of novel fusion algorithms to infer high level abstractions of M2M data [11].

To build such interoperable cross-domain IoT applications, we have designed and developed the Machine-to-Machine Measurement (M3) framework. It assists IoT developers and end users in: (1) semantically annotating M2M data, (2) generating cross-domain IoT applications by combining M2M data from heterogeneous areas and (3) interpreting them. The end user receives a high-level abstraction of their sensor data. The contributions and novelties of the M3 framework are to explain in details and evaluate it as follows:

- Semantically annotate sensor measurements from heterogenous domains [14].
- Reasoning on sensor data with Sensor-based Linked Open Rules (S-LOR) [15] to infer high-level abstraction.
- The M3 common nomenclature to describe sensor measurement in a unified way [16], we introduced the idea of the M3 framework and generating IoT applications without detailed them.
- The Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset.
- Methods to improve the domain knowledge referenced in LOV4IoT according to the semantic web best practices and design it in an interoperable way.
- The generation of IoT application templates.
- Propose a deeper evaluation such as software performance and semantic web best practices.

The rest of the paper is structured as follows: section II presents the state of the art and clearly explains the limitations. Section III describes the M3 framework, section IV details the implementation of cross-domain IoT applications along with a use case. Section V is focused on the evaluation.

[1] http://www.ict-citypulse.eu/page/
[2] http://spitfire-project.eu/
[3] http://www.ready4smartcities.eu/
[4] http://lov.okfn.org/dataset/lov/
[5] http://datahub.io/
[6] http://sindice.com/
[7] http://watson.kmi.open.ac.uk/WatsonWUI/
[8] http://swoogle.umbc.edu/
[9] http://internetofthings.electronicsforu.com/2013/09/m2m-iot-embedded-narang-kishor-narnix/

Finally, we conclude the paper in section VI.

II. State of the Art

We present in this section existing works mentioning the idea to combine domains or to build tools to ease development application tasks and explain their limitations.

A. IoT related works

Recently, Chen et al. [5] introduce the need for intelligent processing for IoT/M2M data and explain the issue related to domain specific-applications. Moriandi et al. [25] describe the need for cross-domain applications, semantic interoperability and data management for exchanging and analyzing IoT/M2M data to infer useful information and to ensure interoperability among IoT applications and for reasoning. They clearly explain a lack of standardization related to ontologies and data formats but do not provide any solutions. Patel et al. [28] describe the challenge to ease application development dedicated to smart office and fire management IoT applications. They propose a tool to easily develop IoT applications, but the application developers still need to program the application logic layer and the authors do not explain the way to interpret sensor data. They explain the need of domain vocabularies but their approach is not based on semantic web technologies (e.g., ontologies). They take into consideration actuators too. Their evaluation is based on two Eclipse plug-in: Metrics 1.3.6 and EclEmma to show that their tool reduce the development time. No demonstration is available and they do not provide end-user interactions.

B. Semantic-based IoT

Sheth et al. [33] designed the concept 'Semantic Sensor Web' to semantically annotate sensors and their data and intoduce the need of domain ontologies without exposing the issues related to reuse these domain ontologies. The Spitfire [29] project combined semantic web and Internet of Things to create 'Semantic Web of Things'. Most of the existing works such as SemSOS [19], Sense2Web platform [2] and Semsor4grid4env [10] (Semantic Sensor Grids for Environmental Applications) semantically annotate sensor streams and just link them to the Linked Open Data and visualize them. As explained in the W3C Semantic Sensor Network (SSN) ontology [7] final report[10], SSN does not provide a basis for reasoning that can ease the development of advanced applications. Due to this limitation, we design the M3 ontology, an extension of the the W3C SSN ontology, to ease the reasoning and propose S-LOR [15], a set of unified rules based on the M3 ontology. Hachem et al. [17] explain the intervention of domain experts to interpret sensor data, which is costly and time-consuming. They do not propose to reuse the domain knowledge that has already been integrated and interpreted in existing projects. Recently, Paganelli et al. [27] propose a similar idea to build a framework to speed up development of Web of Things applications based on web services such as REST but not do not propose to interpret sensor data and link domains. Recently, Manate et al. [24] explained the need to employ domain-specific ontologies and ontologies matching and alignment tools to build IoT applications. They do not explicitly describe the issues encountered if we want to combine these domain ontologies. Firstly, ontology mapping tools are not enough mature for our domain specific ontologies. Secondly, the domain ontologies have not been designed in a unified and interoperable way even with the use of standardized semantic web languages.

C. Limitations

We encounter several shortcomings concerning the related works:

- They introduce the need of domain ontologies, but do not reference which ones.
- Do not explain the technical difficulties to reuse and combine these domain ontologies (e.g., various editor tools generating different syntaxes, different structure of ontologies).
- They introduce the need of combining domain knowledge, but did not try to apply ontology mapping tools on domain ontologies. Ontology mapping tools are not enough mature yet.
- Semantically annotating M2M data to explicitly describe their meaning.
- Inferring additional knowledge using logical reasoning in the context of IoT.
- Do not combine heterogeneous IoT domains to build promising applications.
- Do not reuse heterogeneous domain knowledge.
- Assisting developers and users to interpret the results provided by endpoints such as sensors and actuators by automatically generating IoT/M2M applications.

III. Machine-to-Machine Measurement (M3) framework

To solve the limitations stated above, we design the M3 framework as depicted in Figure 1 which is split into several layers as follows. The **perception layer** contains physical devices such as sensors, actuators and RFID tags. The **data acquisition layer** gets sensor data in SenML format [20] from M2M devices. Then, this layer converts them in a unified way (RDF/XML)[11] compliant with the Machine-to-Machine Measurement (M3) ontology. Resource Description Framework (RDF) [22] is a basic semantic web language to describe triples composed of subject-predicate-object. For

[10] http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/

[11] http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html

instance, 'summer is a season', 'summer' is the subject, 'is a' is a predicate, and 'season' the 'object'.

It is an extension of the W3C Semantic Sensor Network (SSN) ontology [7], more precisely, we extend the Observation Value concept to provide a basis for reasoning. The **persistence layer** stores M3 domain knowledge (ontologies, datasets) and semantic sensor data and inferred sensor data in a triple store. We also have datasets to retrieve domain knowledge to easily build IoT application template. A triple store is a database to store semantic sensor data. We also store M3 SPARQL queries and M3 rules in files. SPARQL [30] is a SQL-like language in semantic web to query semantic data. The **knowledge management layer** is responsible for finding, indexing, designing, reusing and combining domain-specific knowledge (e.g., smart home, intelligent transportation systems, etc.) such as ontologies and datasets to update M3 domain ontologies, M3 datasets and M3 rules which are structured in the same manner. Linked Open Vocabularies for IoT (LOV4IoT)[12] is a huge knowledge-base composed of domain ontologies, datasets and rules based on semantic web technologies which could be theoretically reused to design cross-domain applications. The **reasoning layer** infers high-level knowledge using reasoning engines and M3 rules extracted from Sensor-based Linked Open Rules (S-LOR) [15]. M3 rules are a set of rules compliant with the M3 ontology to infer new knowledge on sensor data. For instance, when the cloud cover is equal to 0 okta, M3 rules can deduce that the sky is blue. Okta is a unit of measurement used to describe the amount of cloud cover. The **knowledge query layer** executes SPARQL (a SQL-like language) queries on inferred sensor data. The **application layer** employs an application (running on smart devices) which parses and displays the results to end users. For instance, the M3 framework suggests activities according to the weather forecasting (e.g., catamaran when it is windy).

In the next section, we explain how these components are correlated with each other.

A. The M3 process

Tasks performed by M3 are displayed in Figure 2. The M3 framework is composed of several steps to automatically generate cross-domain IoT applications as follows. Firstly, the end user gives sensor used (e.g., LightSensor) and the context (e.g., Weather) and the M3 framework proposes IoT application templates fitting his needs. Secondly, the end user chooses the IoT application template. Thirdly the framework automatically generates the template to build the IoT application which contains the M3 domain ontologies, M3 datasets, M3 rules and M3 SPARQL queries. Then, the end user sends sensor data, the M3 converter annotates SenML [20] sensor data according to the M3 ontology. Then, the M3 framework runs the reasoning engine with the M3

[12]http://www.sensormeasurement.appspot.com/?p=ontologies

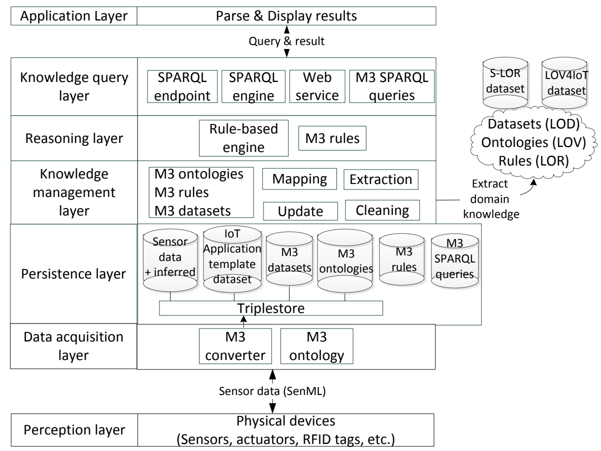

Figure 1: Architecture of the M3 framework

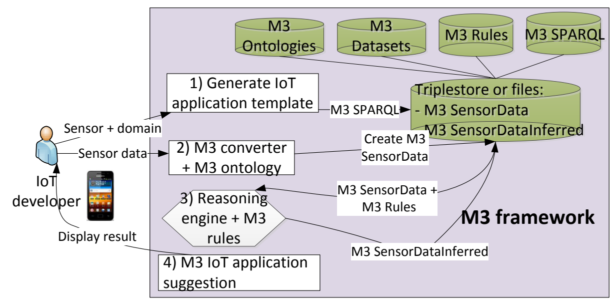

Figure 2: M3 process

rules provided in the template. Inferred sensor data are stored and updated in the triple store. The next step is to query sensor data using the M3 SPARQL (a SQL-like language) query generated by the framework. Finally, it is followed by parsing sensor data and displaying the suggested results to the end users.

B. Generating IoT application templates

The M3 framework generates IoT application templates according to the sensors and domains employed by the users. For instance, the user chooses a sensor and the domain (e.g., Thermometer and Health) and the M3 approach finds IoT application templates using the sensor and combined it with other domains. The sequence diagram is depicted in Figure 3. For the given example, the M3 framework proposes one cross-domain template "Body temperature, Symptoms and Home remedies" to suggest home remedies according to the body temperature and symptoms (e.g., fever) deduced by the M3 framework. By using the same sensor but in other domain (e.. Weather), the M3 framework proposes 4 other cross-domain templates. Once, the user chooses a template, the M3 framework will automatically generates the M3 domain ontologies, M3 datasets, M3 rules and M3 SPARQL queries needed to build the IoT application as depicted in Figure 4. The templates are defined in our application IoT

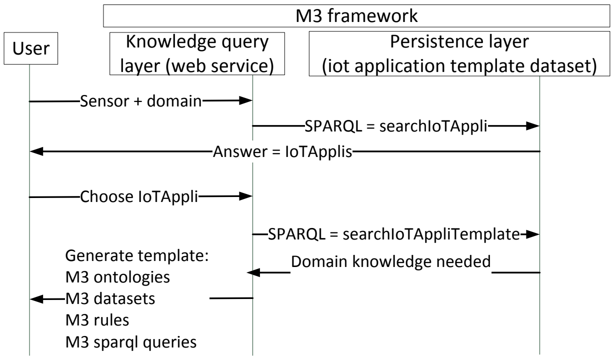

Figure 3: Generating IoT applications. Demo at: http://www.sensormeasurement.appspot.com/?p=m3api

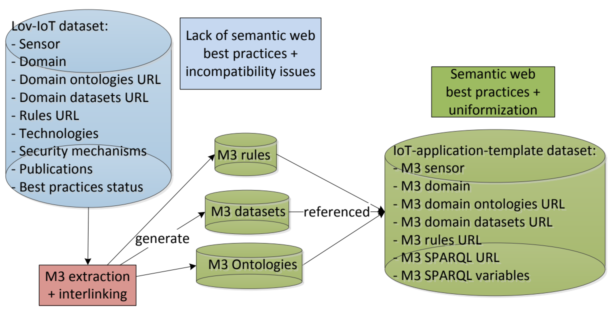

Figure 4: IoT application template dataset

template dataset, for each template we indicate sensor used, domains, M3 domain ontologies, datasets and rules relevant to build the IoT application template.

C. Converting sensor data

To interpret sensor data to infer high-level abstraction, we need to describe them in a unified way. Uniform descriptions of sensors, units, measurements and IoT domains are fundamental necessity to develop cross-domain applications and services. The entire M3 nomenclature is available here[13], implemented in our M3 ontology and explained in our previous publication [16].

D. Reusing domain knowledge with LOV4IoT

We pursued a deeper analysis of domain knowledge related to sensors, the research questions are as follows:

- What domains do sensors use?
- Which ontologies exist that cover each domain?
- What reasoning exit that cover each domain to interpret sensor data?
- Is the ontology publicly accessible e.g., downloadable from a website.
- Which technologies or tools are used to implement the ontology or rules?
- Does the ontology follow the semantic web best practices?

To facilitate IoT application development, we propose the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset which references more than 200 ontology-based works related to sensors in various domains such as health care, building automation, food, agriculture, tourism, security, transportation and smart city. More than 70 ontologies are now online thanks to our work and theoretically, could be easily reused. We discover, identify, study and reference these IoT projects since:

- Sensors and their measurements are described.
- They can be used to design cross-domain IoT applications (e.g., the naturopathy application to combine health, weather and smart kitchen).
- The projects are based on ontologies.
- The projects are rule-based systems.
- Domain experts published their works in conferences.
- They explained why they integrate semantics.
- They described how they evaluate ontologies.
- The domain ontologies, datasets or rules code could be reused to design our IoT application templates.

The LOV4IoT dataset is a synthesization and classification of these semantic-based works. For each work, we indicate the following information: domain, publications (authors, date, and titles), ontology, rule and dataset URL if we have them, technologies used, sensor used, ontology status (e.g., confidential, lost, published online soon, online, referenced in LOV when best practices are followed).

An intermediary step of this work enables the developer to surf on the LOV4IoT web page accessible online[14] to search domain ontologies according to a specific domain. For instance, if the user is looking for smart home ontologies then he goes to this section and finds more than 30 projects describing sensors and rules employed.

Unfortunately, to reuse the existing ontologies and link them is not so easy, there is a real need to popularize semantic web best practices as explained in W3C Web of Things [13], ETSI M2M and OneM2M [12] standardizations and redesign it in a unified way. To prove the feasibility of our approach and ease the interoperabiltiy, we design the M3 domain knowledge for several cross-domain scenarios.

E. Improving domain knowledge

To easily reuse and combine the domain knowledge referenced in LOV4IoT, there is a need to improve it and design it in a unified way to generate cross-domain IoT applications and interpret sensor values. Our M3 ontologies, M3 datasets and M3 rules have been extracted from the LOV4IoT dataset as follows (Figure 5):

- Improving the domain knowledge according to the semantic best practices.

[13]http://www.sensormeasurement.appspot.com/documentation/NomenclatureSensorData.pdf

[14]http://www.sensormeasurement.appspot.com/?p=ontologies

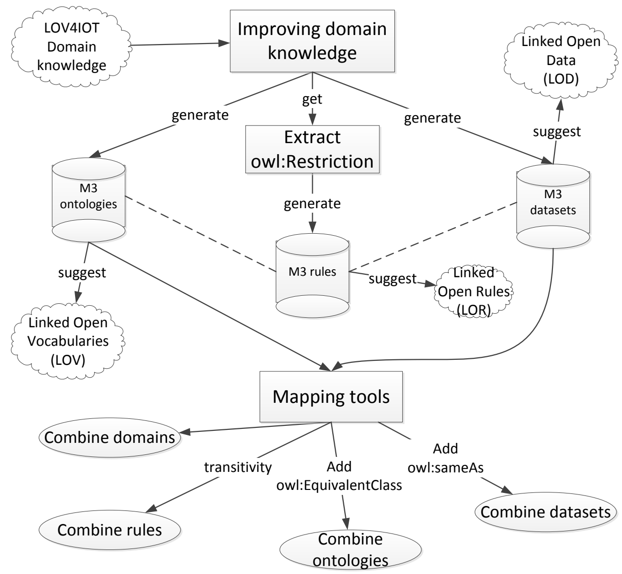

Figure 5: Generate and combine M3 domain knowledge

- Extracting owl:Restriction found in domain ontologies and convert them into M3 rules. Ontology Web Language (OWL) [1] is a semantic web language to design ontologies.
- Rewrite the domain ontologies to be compliant with our M3 framework. The M3 domain ontologies could be suggested to the Linked Open Vocabularies[15].
- Rewrite the domain datasets to be compliant with our M3 framework. The M3 domain datasets could be suggested to the Linked Open Data[16].
- Integrate ontology mapping tools to automatically align the domain knowledge to infer additional knowledge and build cross-domain scenarios.

We achieve this process manually for a first and rapid implementation. Since, the process is feasible, we try to find a way to automatically extract the domain knowledge and update it to be compatible with the M3 framework. In Figure 6, we show the need to: unify syntaxes and terms, add labels and comments useful for ontology matching tools, be compliant with the M3 framework to combine domain knowledge, etc.

F. Reusing rules with S-LOR

We present in this section our Sensor-based Linked Open Rules (S-LOR) [15] to share, reuse and combine sensor-based semantic rules to interpret sensor data and infer high-level abstraction. As explained in the previous section, we encounter numerous difficulties to automatically extract the domain knowledge from LOV4IoT, so, we redesigned manually our own M3 rules to build a knowledge base with more than 100 rules related to sensor measurements. The

[15]http://lov.okfn.org/dataset/lov/
[16]http://datahub.io/

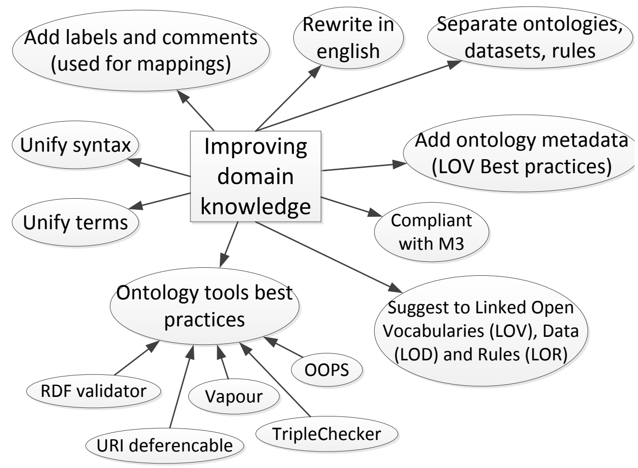

Figure 6: Improving domain knowledge

M3 rules are implemented in a unified way according to the Jena[17] framework, a framework to design semantic web applications.

IV. Cross-domain IoT application use case implementation

The M3 framework could be integrated in different components: cloud, mobile phones or gateways (e.g., Raspberry Pi). Most of the pieces of the M3 framework are implemented and tested in the cloud[18]. The M3 framework is also adapted to constrained devices such a mobile phones or tablets.

The user chooses one of the templates presented in section III-B and the M3 framework will automatically generates the M3 domain ontologies, M3 datasets, M3 rules and M3 SPARQL queries in a ZIP file as displayed in Figure 7 that we will use to build the application. Then, the wind speed or cloud cover sensor data (from end user) are sent to the M3 framework, which semantically annotates them and applies the M3 reasoning engine with the M3 rules provided in the template (ZIP file). Finally, M3 returns high-level cross-domain information to the user. Such scenarios have already been developed and are available at[19]. Here a cross domain application use case is mentioned that takes advantage of the ontologies for weather and tourism to suggest activities according to the weather. The M3 framework reused works designed by Kofler et al. [32] for the weather domain and Chien et al. [6] for tourism and design Semantic Web Rule Language (SWRL)[20] rules that could be reused in other IoT applications. Some of the design rules are IF m3:CloudCover = 0 m3:Okta THEN NoCloudCover [32] and IF NOT night

[17]http://jena.apache.org/
[18]http://www.sensormeasurement.appspot.com/
[19]http://www.sensormeasurement.appspot.com/?p=transport
[20]http://www.w3.org/Submission/SWRL/

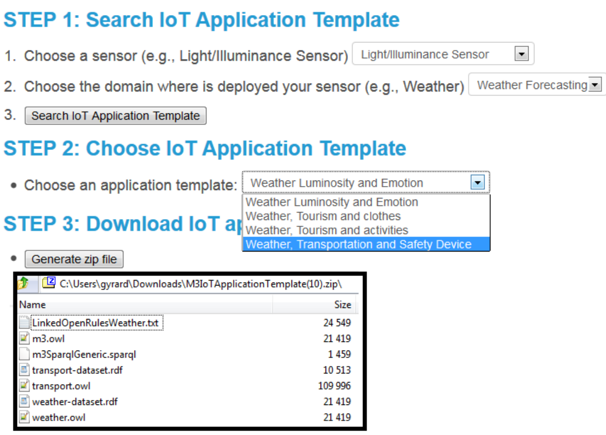

Figure 7: Generate IoT application templates
(http://www.sensormeasurement.appspot.com/?p=m3api)

AND (activity type = family journey AND water activity) then activity = boat [6].

The M3 approach and the scenarios have been tested in constrained devices too. We implemented and run the M3 approach (ontology, semantic sensor data, domain ontologies, datasets, reasoning engine and rules) on Android powered smartphones and tablets. AndroJena[21] framework has been used to implement it. AndroJena is a light version of Jena, a semantic web framework to build semantic web applications.

V. Evaluation

To evaluate our work, we propose three kind of evaluations. Firstly, we evaluate software performances for the M3 converter, reasoning engine, load sensor data and query sensor data. Secondly, we evaluate the M3 domain knowledge that we design with semantic web tools or methodologies. Finally, we test with end users the usefulness of the M3 framework by looking at the flow on our web site and user form results.

A. Software performance

The M3 framework has not been tested yet with very large datasets or real sensor data. In this work, we focus on showing the importance to combine the domains with each other and to reason on sensor data. We store data in a google database for a rapid prototype, our performances could be increased by storing sensor data in a triple store. We are working on the integration of the Jena TDB triple store and the Jena Fuseki SPARQL endpoint.

We evaluate the M3 converter by varying the size of sensor data to convert (between 4KB and 14KB) and compute the time needed in milliseconds as displayed in Figure 8. The device used for the evaluation has the processor: Intel(R) Core(TM) i7 CPU, X 920, 2GHZ, 4GB of RAM and the operating system is 64-bit Windows 7. We tested the M3

[21]http://code.google.com/p/androjena/

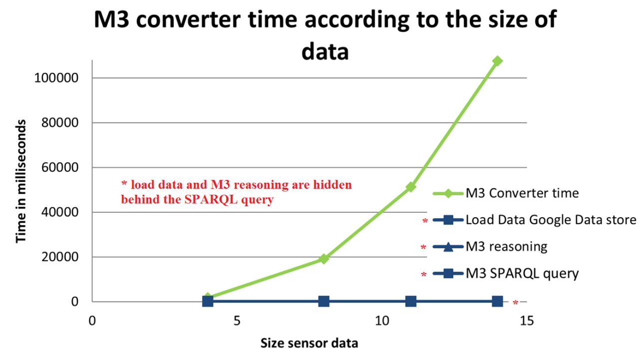

Figure 8: M3 converter time according to the size of data

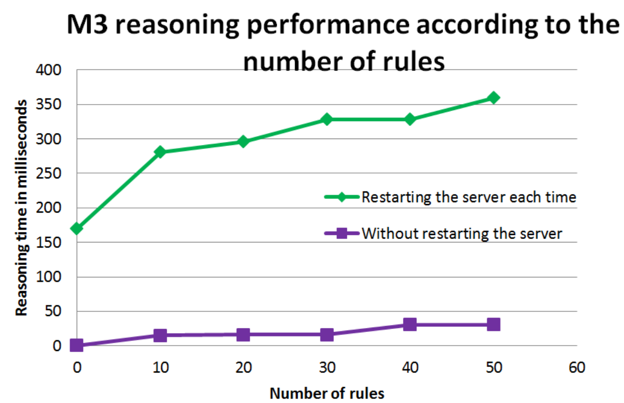

Figure 9: M3 reasoning performance according to the number of rules

converter with the transport scenario made of 55 M3 Jena weather rules. According to the graph, the M3 converter is not scalable for gigabytes of data, however, we are fast enough to interpret small quantity of data and further process it: the M3 reasoning takes between 24 and 28 ms, load data between 36 and 37 ms and the M3 SPARQL query between 6 or 7 ms.

Another evaluation is varying the number of M3 rules (between 10 and 55 rules) to evaluate the reasoning time in milliseconds. The results are displayed in Figure 9. The M3 reasoning part takes between 16 and 31 milliseconds for 4KB of sensor data, when we do not restart the server each time. At the time of this evaluation, we have 100 rules to interpret sensor values. The first optimization was to split the Linked Open Rules files into different files and classify M3 rules by domains to optimize performances. Another possible improvement is to select only the rules that we need for specific sensors to reduce the number of rules to run in the reasoning engine. The more we split the rules by domains, the better is the performance.

B. Semantic best practices

The knowledge acquisition and conceptualization of M3 domain ontologies and datasets has been done by reading research articles, synthesize, classify and compare works referenced in our LOV4IoT dataset. The implementation has been done with RDF, RDFS and OWL since they are W3C recommendations and following the best practices presented in [31], Methontology [9], Noy et al. [26] and books [18] [8] to build well-structured ontologies from scratch as follows. Specification and knowledge acquisition have been done by reading more than 200 ontology-based IoT projects publishing their works in conferences. In these steps, we considered reusing existing ontologies. Conceptualization has been done with an hybrid approach (top-down and bottom up) by defining the most important concepts first and they generalize them as much as possible. Formalization and implementation steps are for defining and creating the hierarchy and the properties of classes, included instances according to the ontology. We design the M3 domain ontologies and datasets. The evaluation step is to judge the quality of the ontology and is realized by building our cross-domain IoT applications. Documentation has been done using labels and comments inside the ontology. We also used the Parrot[22] tool. Finally, the maintenance is followed, since the ontology has evolved continuously through this work.

We use several semantic tools to evaluate M3 domain ontologies and datasets that we designed such as Oops, TripleChecker, RDF Validator, SSN Validator[23] [21] and Vapour. Oops[24] is used to detect some of the most common pitfalls appearing when developing ontologies, TripleChecker[25] to check that we use common namespace and ontologies and the appropriate concepts and properties. This tool finds typos and common errors in RDF data. RDF Validator[26] is used to check syntax, Vapour[27] [3] to check URI deferencable and test easily our ontologies on other semantic web tools. We suggested more than 27 domain ontologies to the Linked Open Vocabularies (LOV). Thanks to them we discover numerous bad practices, this is why we redesign our own ontologies and datasets to be compliant with best practices. Reasoner such as Hermit and Pellet under Protégé are used to check they are no incompleteness or inconsistencies. M3 rules have been designed according to the M3 ontologies and have been tested with the Jena inference engine in cross-domain IoT applications.

C. Evaluate our tools by end-users

We propose to look at the usefulness of the M3 framework designed. Firstly, since we implemented a prototype published online, we propose to look at visitors on our web site and ask to end users to test our M3 framework or some of the M3 components such as LOV4IOT, S-LOR or generating IoT applications.

Our web site shows most of the components of the M3 framework and has been visited a lot, more than 883 visits have been done since 5 December 2013 from 63 countries. We used Google Analytics since August 2014 to look at the web pages the most visited or other information. For instance, the LOV4IoT web page is the web page the most visited (135 pageviews since August 2014).

VI. Conclusion

In a nutshell, our contributions are: (1) identifying the current limitations, (2) interpreting M2M data and infer additional knowledge, (3) linking them to the Linked Open Data, Linked Open Vocabularies and the Linked Open Rules, (4) describing M2M data using a uniform nomenclature, (5) generating IoT application templates, (6) creating the Linked Open Vocabularies for Internet of Things (LOV4IoT) dataset, (7) improving the domain knowledge referenced in LOV4IoT according to the semantic web best practices and designing in an interoperable manner, and (8) evaluating M3.

A current version of this framework is being tested on Android platform using the AndroJena[28] framework. Future works are to automatically extract and combine the domain knowledge referenced in the LOV4IoT dataset to generate interoperable M3 ontologies, datasets and rules.

Acknowledgment

The authors would like to thank the semantic web experts (Ghislain Atemezing, Payam Barnaghi and Bernard Vatant) for their valuable feedback. This work is partially supported by the Com4Innov Platform of Pole SCS[29] and FIESTA-IOT[30] from the European Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943.

References

[1] Owl (ontology web language). http://www.w3.org/TR/owl-ref/.

[2] P. Barnaghi, M. Presser, and K. Moessner. Publishing linked sensor data. In *CEUR Workshop Proceedings: Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN), Organised in conjunction with the International Semantic Web Conference*, volume 668, 2010.

[3] D. Berrueta, S. Fernández, and I. Frade. Cooking http content negotiation with vapour. In *Proceedings of 4th Workshop on Scripting for the Semantic Web (SFSW2008)*. Citeseer, 2008.

[4] D. Boswarthick, O. Elloumi, and O. Hersent. *M2M communications: a systems approach*. Wiley, 2012.

[5] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. A vision of iot applications challenges and opportunities with china perspective. 2014.

[22] http://www.ontorule-project.eu/parrot/parrot
[23] http://iot3.ee.surrey.ac.uk/SSNValidation/
[24] http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp
[25] http://graphite.ecs.soton.ac.uk/checker/
[26] http://www.w3.org/RDF/Validator/
[27] http://validator.linkeddata.org/vapour
[28] http://code.google.com/p/androjena/
[29] http://www.pole-scs.org/
[30] http://www.fiesta-iot.eu/

[6] H.-Y. Chien, S.-K. Chen, C.-Y. Lin, J.-L. Yan, W.-C. Liao, H.-Y. Chu, K.-J. Chen, B.-F. Lai, and Y.-T. Chen. Design and implementation of zigbee-ontology-based exhibit guidance and recommendation system. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[7] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012. http://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

[8] L. R. e. M. H. David Wood, Marsha Zaidman. *Linked Data. Structured Data on the Web*. 2014.

[9] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[10] A. J. Gray, R. García-Castro, K. Kyzirakos, M. Karpathiotakis, J.-P. Calbimonte, K. Page, J. Sadler, A. Frazer, I. Galpin, A. A. Fernandes, et al. A semantically enabled service architecture for mashups over streaming and stored data. In *The Semanic Web: Research and Applications*, pages 300–314. Springer, 2011.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[12] A. Gyrard and C. Bonnet. Semantic Web best practices: Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things, 04 2014.

[13] A. Gyrard, C. Bonnet, and K. Boudaoud. Domain knowledge Interoperability to build the semantic web of things, 06 2014.

[14] A. Gyrard, C. Bonnet, and K. Boudaoud. Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In *WF-IOT 2014, World Forum on Internet of Things, 6-8 March 2014, Seoul, Korea*, Seoul, KOREA, REPUBLIC OF, 03 2014.

[15] 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*, Riva Del Garda, ITALY, 10 2014.

[16] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud. Standardizing generic cross-domain applications in Internet of Things. In *GLOBECOM 2014, 3rd IEEE Workshop on Telecommunication Standards: From Research to Standards, December 8, 2014, Austin, Texas, USA*, Austin, UNITED STATES, 12 2014.

[17] S. Hachem, T. Teixeira, and V. Issarny. Ontologies for the internet of things. In *Proceedings of the 8th Middleware Doctoral Symposium*, page 3. ACM, 2011.

[18] T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[19] C. Henson, J. Pschorr, A. Sheth, and K. Thirunarayan. Semsos: Semantic sensor observation service. In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pages 44–53. IEEE, 2009.

[20] C. Jennings, J. Arkko, and Z. Shelby. Media types for sensor markup language (senml). 2012.

[21] S. Kolozali, T. Elsaleh, and P. Barnaghi. A validation tool for the w3c ssn ontology based sensory semantic knowledge.

[22] O. Lassila and R. R. Swick. Resource description framework (rdf) model and syntax specification. 1999. http://www.w3.org/TR/REC-rdf-syntax/.

[23] E. M2M. Machine-to-Machine Communications (M2M); Study on Semantic support for M2M data, ETSI Techinal Report 101 584 v2.1.1 (2013-12), 2012.

[24] B. Manate, V. I. Munteanu, and T. F. Fortis. Towards a smarter internet of things: Semantic visions. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on*, pages 582–587, July 2014.

[25] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[26] N. F. Noy, D. L. McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[27] F. Paganelli, S. Turchi, and D. Giuli. A web of things framework for restful applications and its experimentation in a smart city.

[28] P. Patel, A. Pathak, D. Cassou, and V. Issarny. Enabling high-level application development in the internet of things. In M. Zuniga and G. Dini, editors, *Sensor Systems and Software*, volume 122 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 111–126. Springer International Publishing, 2013.

[29] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth, et al. Spitfire: toward a semantic web of things. *Communications Magazine, IEEE*, 49(11):40–48, 2011.

[30] E. Prud, A. Seaborne, et al. Sparql query language for rdf. http://www.w3.org/TR/rdf-sparql-query/.

[31] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web*, pages 63–81. Springer, 2004.

[32] C. Reinisch, M. Kofler, F. Iglesias, and W. Kastner. Thinkhome energy efficiency in future smart homes. *EURASIP Journal on Embedded Systems*, 2011:1, 2011.

[33] A. Sheth, C. Henson, and S. Sahoo. Semantic sensor web. *Internet Computing, IEEE*, 12(4):78–83, 2008.

[34] P.-Y. Vandenbussche and B. Vatant. Metadata recommendations for linked open data vocabularies. *Version*, 1:2011–12, 2011.