Assisting IoT Projects and Developers in Designing Interoperable Semantic Web of Things Applications

Amelie Gyrard*, Christian Bonnet†, Karima Boudaoud‡, Martin Serrano*

* Insight @ National University of Ireland, Galway, Ireland
† Eurecom, Sophia Antipolis, France
‡ Laboratoire I3S-CNRS/UNSA, Sophia Antipolis, France

Abstract — Internet of Things (IoT) is becoming more and more popular. Increasingly, European projects (CityPulse, IoTest, IoT-i and IERC), standard development organizations (ETSI M2M, oneM2M and W3C) and developers are involved in integrating Semantic Web technologies to Internet of Things. All of them design IoT application uses cases which are not necessarily interoperable with each other. The main innovative research challenge is providing a unified system to build interoperable semantic-based IoT applications. In this paper, to overcome this challenge, we design the Semantic Web of Things (SWoT) generator to assist IoT projects and developers in: (1) building interoperable Semantic Web of Things (SWoT) applications by providing interoperable semantic-based IoT application templates, (2) easily inferring high-level abstractions from sensor measurements thanks to the rules provided by the template, (3) designing domain-specific or inter-domain IoT applications thanks to the interoperable domain knowledge provided by the template, and (4) encouraging to reuse as much as possible the background knowledge already designed. We demonstrate the usefulness of our contribution through three use cases: (1) cloud-based IoT developers, (2) mobile application developers, and (3) assisting IoT projects. A proof-of-concept for providing Semantic Web of Things application templates is available at http://www.sensormeasurement.appspot.com/?p=m3api.

Keywords — Internet of Things (IoT) applications; Machine-to-Machine (M2M); Semantic Sensor Networks (SSN); Semantic Web of Things (SWoT); templates; Semantic Interoperability.

I. INTRODUCTION

Internet of Things (IoT) applications are becoming more and more popular. Machine-to-Machine [2] is a part of Internet of Things to automate the communications between machines without human involvement. Increasingly, Internet of Things (IoT) projects such as CityPulse1, Spitfire2, OpenIoT [34], READY4SmartCities3, IoT.es4, IoT-i5, IERC6 or Standard Development Organizations (SDO) such as ETSI M2M [21] or oneM2M [25] integrate semantics (i.e., ontology) to ease interoperability between heterogeneous sensor networks. More than 100 scenarios7 have been referenced by IoT projects such as continuous health care, smart home, smart orchard, detecting road conditions or air pollution countermeasures. Most of the existing IoT scenarios are similar, overlapping and not interoperable with each other as it has been highlighted by Serrano [31] and Barnaghi et al. [1]. They clearly explains the necessity to align data models and frameworks and the need of reasoning and interpretation of data to reduce human intervention. The necessity of novel fusion algorithms to infer high level abstractions from M2M data has been pointed out in [12]. Most of the existing approaches do not propose tools to easily interpret M2M/IoT 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 [21], Chen et al. [3] and Miorandi et al. [23]. Narang Kishor explained "every true IoT application or solution needs cross-domain expertise."8.

The main research challenge is providing a unified system to generate interoperable semantic-based IoT applications enabling: (1) unifying data, (2) interpreting data, and (3) combining applicative domains.

To build such interoperable Semantic Web of Things (SWoT) applications, we designed the SWoT generator which assists IoT developers in producing a template with all files required to build semantic-based IoT applications. The main novelty of this approach is to assist IoT developers in building semantic-based applications without having to (1) design their own models/ontologies, (2) design the rules to interpret data, (3) semantically annotate data. To the best of our knowledge, this is the first approach proposing such concrete approaches. The template generated will enable to the developers to: (1) semantically annotate M2M data, (2) infer high-level abstractions from sensor data, and (3) fa-
cilitate the development of domain-specific or cross-domain IoT applications. Moreover, in the long-term perspective, the SWoT generator will enable to build interoperable Semantic Web of Things applications.

The SWoT generator is a new contribution which has been integrated and combine to our previous contributions. It was previously introduced in [15] and [16]. The SWoT generator is integrated in the Machine to Machine Measurement (M3) framework [15] and reuses the workflow of enrichment of IoT data [13]. Further, it exploits the reasoning process to infer high-level abstractions from sensor data [14]. The main novelty of this paper compared to the previous published papers is to focus on the SWoT generator component and explain it in detail.

The rest of the paper is structured as follows: section II presents the state of the art and clearly explains the limitations. Section III emphasizes how the SWoT generator is integrated in the M3 framework and how the SWoT generator can assist IoT projects in speeding up the designing phase of Semantic Web of Things applications. Section IV provides three use cases. Section V explains the implementation of the generation of IoT application templates. Section VI is focused on the evaluation with different datasets and the interoperability of templates. Finally, we conclude the paper in Section VII.

II. STATE OF THE ART

In this section we explain existing works and their limitations since they have common goals: (1) inferring high-level abstraction from sensor data, (2) Semantic Web of Things related works introducing the need to reuse and combine the domain knowledge, and (3) applications and tools to ease development application tasks.

A. Inferring high-level abstractions

Henson’s [19] thesis explained the idea of “semantic perception” to interpret and reason on sensor data. Their approach is based on machine learning. Ganz [10] described in his thesis the need of: (1) new techniques to interpret "Cyber-Physical Data" and infer new knowledge, (2) taking into account context information, (3) a standardized model for meta-information, and (4) the meaning of data is dependent on both temporal and spatial attributes. He designed the Knowledge Acquisition Toolkit (KAT) to infer high-level information based on machine learning.

Such approaches enable interpreting IoT data, but not help IoT developer design their semantic-based applications.

B. Semantic-based IoT

Sheth et al. [32] designed the concept ‘Semantic Sensor Web’ to semantically annotate sensors and their data and introduce the need of domain ontologies without exposing the issues related to reuse these domain ontologies. The Spitfire [28] project combined Semantic Web and Internet of Things to create 'Semantic Web of Things'. They are focused on sensor discovery and not on interpreting sensor data values. Most of the existing works such as SemSOS [18], Sense2Web platform [8], Graph of Things [20] (previously Linked Sensor Middleware) and Sensorgrid4env [11] (Semantic Sensor Grids for Environmental Applications) provide sensor discovery and semantically annotate sensor streams, link them to the Linked Open Data and visualize them. The next step is to interpret sensor data to design interoperable IoT applications. As explained in the W3C Semantic Sensor Network (SSN) ontology [5] final report\(^9\), SSN does not provide a basis for reasoning that can ease the development of advanced applications.

C. Generating IoT applications

Patel et al. [27] 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, they do not explain the way to interpret sensor data. They explain the need of common domain vocabularies, but their approach is not based on semantic web technologies. No demonstration is available and they do not provide end-user interactions. Paganelli et al. [26] 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 do not propose to interpret sensor data and link domains. Ruta et al. [30] propose a Semantic Web of Things (SWoT) framework but not for reasoning on sensor data. Recently, Hachem [17] explained in her thesis the intervention of domain experts to interpret sensor data, which is costly and time-consuming. There is a real need to find approaches to share and reuse the way to interpret sensor data. Further, she explicitly describes as a long-term perspective the need to integrate inference mechanism to extract higher level knowledge from sensor data, since developers do not have the expertise for this task. Sivieri et al. [33] design the ERLIoT (ErLang for the Internet of Things) framework to assist developers in testing and debugging and verifying their code.

D. Limitations of these Works

We encounter several shortcomings concerning the related works:

- **Inferring high-level abstractions from sensor data.**
  Existing works semantically annotate M2M/IoT data to explicitly describe their meaning, but do not design simple approaches to easily share and reuse rules to interpret sensor data. Logical reasoning enables a simple and light reasoning compared to traditional approaches such as data mining. Further, logical reasoning will

\(^9\)http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/
A. Machine-to-Machine Measurement (M3) framework

The SWoT generator is the cornerstone component of the Machine to Machine Measurement (M3) framework [16]. This framework eases the task of developers in designing SWoT applications. The M3 framework comprises the following components as depicted in Figure 1.

In the first step, sensor used and the domain is given by the developer to the SWoT Generator which creates the SWoT template fitting his needs. The SWoT template is essential to build the IoT application. The template is composed of the interoperable domain knowledge (ontologies, datasets and rules) and SPARQL queries. In the second step, the developer semantically annotates his IoT data with the M3 converter and the M3 nomenclature [15] to get M3 data to ease the next step. In the third step, the developer gets the inferred M3 data with S-LOR (Sensor-based Linked Open Rules) [14] using the M3 template to infer high-level abstractions. Then, the developer executes the SPARQL query and loads the interoperable domain knowledge provided by the SWoT template. He will obtain the high-level abstractions of M3 data which can be some suggestions, notifications or instructions for actuators which will be parsed and displayed in a user-friendly interface.

The main novelty of the SWoT generator is to help IoT developers or project: (1) design semantic-based IoT applications with a little background in semantic web, and (2) easily interpret IoT data. The SWoT generator produces IoT application templates according to the sensors and domains employed by the developers. For instance, the developers choose a sensor and the domain (e.g., Thermometer and Health) and the SWoT generator finds IoT application templates using the sensor and combined it with other domains. The sequence diagram is depicted in Figure 2. For the given example, the SWoT generator 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). By using the same sensor but in other domain (e.g., Weather), the SWoT generator proposes 4 other cross-domain templates.

Once, the developers choose a template, the SWoT generator will automatically produces the interoperable domain ontologies, datasets, rules and SPARQL queries needed to build the semantic-based IoT application as depicted in Figure 3.

B. IoT application template dataset

The SWoT generator creates the SWoT template by querying the IoT application template dataset (see Figure 3). It references the domain knowledge needed to build different IoT scenarios. The domain knowledge is composed...
of interoperable domain ontologies, datasets and rules used either to semantically annotate IoT data, interpret them to infer high-level abstractions or to provide suggestions.

For each template, we indicate sensor used, domains, domain ontologies, datasets and rules required to build IoT applications. In Figure 4 is displayed an RDF extract of a template. The RDF dataset is available online10.

Figure 3: The SWoT generator is based on the IoT application template dataset

doing the entire nomenclature which is available here11. The nomenclature has been implemented in the ontology used to semantically annotate IoT data which is explained in our previous publication [15].

D. Designing, Developing and Running the Application

The SWoT template produced is used in three phases: (1) designing, (2) developing, and (3) running. In the designing phase, we assist the IoT projects and developers in choosing the application to develop by suggesting default semantic-based IoT applications by querying the IoT application template dataset through a simple SPARQL query. In the development phase, the developers will write the program to load the ontologies, datasets and rules to semantically enrich data and interpret data and then combine it with background knowledge to build the application.

In the running phase, the application gets real sensor data, and run the program previously mentioned to semantically annotate data and infer high-level abstractions, essential for the end-user application.

IV. USE CASES

In this section, we demonstrate three use cases which benefit from our proposed contribution: (1) cloud-based IoT developers, (2) mobile application developers, and (3) assisting IoT projects.

A. Cloud-Based Developers

Exploiting the M3 framework is really easy. Developers employed the template web services12 or the web user interface to generate SWoT templates. Firstly, developers look for the SWoT template they are interested in by choosing sensors and domains [16]. Secondly, they download the SWoT template composed of interoperable ontologies, datasets, rules and SPARQL queries. Thirdly, they use the

---

10 http://sensormeasurement.appspot.com/dataset/iot-application-template-dataset
12 http://www.sensormeasurement.appspot.com/?p=documentation
M3 converter\textsuperscript{13} to semantically annotate IoT data. The most difficult step is, being familiar with Jena\textsuperscript{14} in case developers never used such frameworks before. Jena is a framework to build semantic web applications. Fourthly, they just need to develop few lines of code to load M3 IoT data, run the Jena reasoner with the rules to generate inferred IoT data. Fifthly, they load ontologies, datasets, M3 inferred IoT data and execute the SPARQL query available in the M3 template. The main task of the developer is getting the results returned by the SPARQL query to display them as expected in a user-friendly interface. Developers build a cross-domain IoT application combining two domains: transportation and weather to suggest safety equipment (e.g., wipers) in the car according to the weather measurements (e.g., precipitation).

B. Mobile Application Developers

Android-based mobile developers use the SWoT generator API and web services explained above to download a template and then build the application on a mobile phone \cite{7}. Concrete applications have been done using the template generated by the SWoT generator in \cite{7} and \cite{6}.

C. Assisting IoT Projects

We updated our IoT application template dataset with new scenarios inspired by use cases mentioned in European projects such as CityPulse, IoT.est or even in standardizations such as ETSI M2M or oneM2M. The SWoT generator could assist such projects in building IoT applications since we generate templates with interoperable domain knowledge to semantically annotate sensor data and then infer high-level abstractions. Figure 5 explained how the M3 framework assists projects in generating interoperable IoT applications. IoT projects offer real and reliable sensor data, represented in SenML format, they can be enriched with our M3 framework to interpret them with the M3 reasoning called S-LOR. Then SWoT templates are generated according to the sensor data. For instance in the smart home domain, they propose to switch on/off the lights if nobody is in the room. In healthcare, IoT-i, CityPulse and IoT.est propose scenarios to interpret health measurements such as blood glucose, temperature, heart rate and send alerts if needed. We have SWoT templates to design such IoT applications. Our naturopathy scenario can assist health scenarios to just interpret health measurements or build smarter applications by providing cross-domain suggestions to remedy to the detected symptoms. Others scenarios are available in various domains such as smart home and transportation where SWoT templates can assist in developing such scenarios.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{The SWoT generator of IoT applications}
\end{figure}

1) \textit{FIESTA-IoT EU project:} This innovative work addressing semantic interoperability will be exploited and extended within the FIESTA-IoT EU project\textsuperscript{15}, which focuses on federation, unification and semantic interoperability applied to IoT. This innovative contribution will be extended by providing composition of templates by investigating semantic web services, web service composition and ‘Linked Open Services’ research fields to overcome this challenge.

V. IMPLEMENTATION

Our proof of concept has been implemented with the Jena framework\textsuperscript{16}, the Jena inference engine, Google App Engine, Java 1.7, Ajax, JavaScript, Bootstrap, HTML and CSS. The IoT application template dataset contains more than 30 SWoT templates: half of them are cross-domain applications, and the others are just to deduce high-level abstractions from sensor data without combining heterogeneous domains. These datasets have been employed to easily build several scenarios; (1) the naturopathy application that combines meteorology, healthcare, affective science and smart home domains to suggest home remedies based on their nutrients according to specific symptoms (e.g., fever) or suggest food according to the weather, (2) the tourism application to suggest clothes and activities according to the weather, and (3) the transportation application to suggest safety equipment in the car according to the weather. These cross-domain scenarios are accessible on our web site and have been developed with the following technologies: the Jena framework to design semantic web applications, Google App Engine to publish the prototype online, Ajax to query web services, HTML5, CSS3, Javascript and Bootstrap for the user interface.

The developer chooses one of the templates presented in section \textit{III} and the SWoT generator will automatically produces the SWoT template (e.g., ZIP file) composed of interoperable domain ontologies, datasets, rules and SPARQL

\begin{itemize}
\item http://www.sensormeasurement.appspot.com/?p=senml_converter
\item http://www.fiesta-iot.eu/
\item https://jena.apache.org/
\item https://jena.apache.org/\end{itemize}
queries that he will use to build the IoT application. Then, the wind speed or cloud cover IoT data are sent to the M3 converter, which semantically annotates them. Then, the developer executes the Jena reasoning engine with the rules provided by the SWoT template. Finally, the M3 framework returns high-level M3 cross-domain suggestions to the developer. 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 SWoT template contains works designed by Kofler et al. [29] for the weather domain and Chien et al. [4] for tourism to make them interoperable. Interoperable rules have been designed such as if the cloud cover is equal to 0 then the the weather is sunny and water activity can be suggested [4].

VI. EVALUATION

At the time of writing this paper, evaluating that the system can assist IoT developers in designing semantic-based IoT applications is really hard since we do not have enough IoT developers to evaluate this approach. This challenging evaluation will be done as a future work.

As a first step, we evaluate the SWoT generator, more precisely, (1) the templates produced are used with sensor datasets to build IoT applications and, (2) the quality of the domain knowledge provided in the template is checked with semantic web tools.

Firstly, we evaluate the SWoT generator with 7 datasets as depicted in Table I. In this work, we do have yet real datasets that we can reuse, as a first evaluation, we design our own datasets. For instance, the health dataset simulates heart beat, temperature, blood pressure, cholesterol and skin conductance measurements, whereas the weather dataset simulates luminosity, temperature, wind speed, humidity and precipitation measurements. Such datasets are used to exploit different SWoT templates which are comprised of interoperable domain ontologies, domain datasets and rules. The execution of reasoning engines with such templates enables semantically enriching sensor datasets and inferring such suggestions to remedy to the detected symptoms. SWoT templates can assist in designing other scenarios such as smart home and transportation.

Secondly, we evaluate the quality of the domain knowledge provided by the template as recommended by Ontology Development Guide [24]. The evaluation has been done with semantic web tools such as Oops[20], TripleChecker[21], RDF Validator[22], Vapour[23] and SSN Validator[24]. This is required to later easily automate tasks with ontology matching tools, download the domain knowledge, etc. The interoperable domain knowledge is referenced and exploited in SWoT templates to build different domain-specific or cross-domain scenarios.

The evaluation with different datasets shows that the SWoT generator is generic enough to deal with different scenarios and can assist IoT projects in designing semantic-based applications. Adding a new template is really simple and not too time consuming, it just implies adding a new instance into the RDF template dataset. It takes about one half-day to add a new template. The sensor datasets are based on the SenML format and the M3 nomenclature, we aim to support more heterogeneous formats as a future work. An important aspect would be to provide more complicated templates involving more sensors and domains and provide a way to combine templates with each others. Finally, an important aspect would be to adapt the SWoT generator to requirements expected by some IoT projects such as scalability and real-time.

VII. CONCLUSION

The main novelty and contribution of this paper is the Semantic Web of Things (SWoT) generator to assist IoT projects and developers in designing interoperable semantic-based IoT applications. Indeed, they do not need to design any ontologies, datasets, rules and SPARQL queries since it is provided by the template produced. We detailed the IoT application templates which are comprised of a cornerstone component: the interoperable domain knowledge.
IoT Applications (App)/M3 Suggestions:

App 6: Safety equipment in car according to the weather (Transport)

App 13: Home remedies according to the body temperature (Naturopathy)

App 5: Interpret weather measurements (Weather)

App 14: Deduce mood according to the external luminosity (Emotion)

App 19: Switch on/off light if nobody (Home, rule involving 2 sensors)

App 1: Deduce Snow (Weather, rule involving 2 sensors)

App 10: Interpret location measurements (Tourism)

App 12: Interpret health measurements (Health)

App 17: Interpret home measurements (Home)

App 21: Detect food in the kitchen (Smart Fridge)

App 7: Activities according to the weather (Tourism)

App 18: Interpret temperature or sound data (Home)

Web Service Composition [9] and ‘Linked Open Services’. To achieve such challenges, we automatically add new templates to the template dataset and integrated applications. Another main challenge would be to match our templates to the use cases provided by existing IoT-based projects. Moreover, we would like to work on the composition of such templates to design more sophisticated applications involving more complicated templates involving more sensors and domains. As future work, we will upgrade the SWoT generator by integrating more complicated templates involving more sensors and domains. Moreover, we would like to work on the composition of such templates to design more sophisticated applications. Another main challenge would be to automatically add new templates to the template dataset and build ‘universal templates’. To achieve such challenges, we intent to investigate Semantic Web Services [22], Semantic Web Service Composition [9] and ‘Linked Open Services’.

ACKNOWLEDGMENT

The authors would like to thank the semantic web experts (Ghislain Atemezing, Payam Barnaghi and Bernard Vatant) for their valuable feedback and colleagues, friends and students for fruitful discussions and help for the implementation. This work is partially supported by the Com4Innov Platform25 of Pole SCS26, DataTweet27 (ANR-13-INFR-0008) and by the European project “Federated Interoperable Semantic IoT/Cloud Testbeds and Applications (FIESTA-IOT)28” from the European Union’s Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943.

Table I: SWoT generator evaluated with 7 datasets to build various IoT applications

<table>
<thead>
<tr>
<th>Sensor measurements and M3 RDF dataset size</th>
<th>IoT Applications (App)/M3 Suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation + temperature (3 KB)</td>
<td>App 1: Deduce Snow (Weather, rule involving 2 sensors)</td>
</tr>
<tr>
<td></td>
<td>App 2: Safety equipment in car when snowy (Transport)</td>
</tr>
<tr>
<td></td>
<td>App 3: Activities when snowy (Tourism)</td>
</tr>
<tr>
<td></td>
<td>App 4: Clothes when snowy (Tourism)</td>
</tr>
<tr>
<td>Luminosity, wind speed, temperature, humidity, precipitation (8 KB)</td>
<td>App 5: Interpret weather measurements (Weather)</td>
</tr>
<tr>
<td></td>
<td>App 6: Safety equipment in car according to the weather (Transport)</td>
</tr>
<tr>
<td></td>
<td>App 7: Activities according to the weather (Tourism)</td>
</tr>
<tr>
<td></td>
<td>App 8: Clothes according to the weather (Tourism)</td>
</tr>
<tr>
<td></td>
<td>App 9: Food according to the outside temperature (Naturopathy)</td>
</tr>
<tr>
<td>Longitude + Latitude (3 KB)</td>
<td>App 10: Interpret location measurements (Tourism)</td>
</tr>
<tr>
<td></td>
<td>App 11: Find location information and suggest restaurant around (Tourism)</td>
</tr>
<tr>
<td>Blood pressure, body temperature, cholesterol, heartbeat, skin conductance (5 KB)</td>
<td>App 12: Interpret health measurements (Health)</td>
</tr>
<tr>
<td></td>
<td>App 13: Home remedies according to the body temperature (Naturopathy)</td>
</tr>
<tr>
<td></td>
<td>App 14: Deduce mood according to the external luminosity (Emotion)</td>
</tr>
<tr>
<td></td>
<td>App 15: Deduce mood from heartbeat, skin conductance and blood pressure (Emotion)</td>
</tr>
<tr>
<td></td>
<td>App 16: Deduce symptoms/diseases from heartbeat, skin conductance and blood pressure (Health)</td>
</tr>
<tr>
<td>Home dataset: Room temperature, sound (6 KB)</td>
<td>App 17: Interpret home measurements (Home)</td>
</tr>
<tr>
<td></td>
<td>App 18: Interpret temperature or sound data (Home)</td>
</tr>
<tr>
<td>Luminosity + presence (3 KB)</td>
<td>App 19: Switch on/off light if nobody (Home, rule involving 2 sensors)</td>
</tr>
<tr>
<td></td>
<td>App 20: Deduce if someone is in the room or not and switch on/off light (Home, actuation suggestion)</td>
</tr>
<tr>
<td>Food (3 KB)</td>
<td>App 21: Detect food in the kitchen (Smart Fridge)</td>
</tr>
<tr>
<td></td>
<td>App 22: Recipe according to the food available in your kitchen (Home)</td>
</tr>
</tbody>
</table>

REFERENCES


25http://www.com4innov.com/
26http://www.pole-scs.org/
28http://www.fiesta-iot.eu/


M. Serrano. Semantic interoperability, what is missing to desilo the use cases, 2014.

