SWoTSuite: A Toolkit for Prototyping Cross-domain Semantic Web of Things Applications

Pankesh Patel[1], Amelie Gyrard[2], Dhavalkumar Thakker[3], Amit Sheth[4], and Martin Serrano[2]

[1] ABB Corporate Research, India
pankesh.patel@in.abb.com
[2] Insight Center for Data Analytics, National University of Galway, Ireland
{amelie.gyrard,martin.serrano}@insight-centre.org
[3] University of Bradford, Bradford, United kingdom
d.thakker@bradford.ac.uk
[4] Kno.e.sis and Wright State University, USA
amit@knoesis.org

Abstract. Semantic Web of Things (SWoT) applications focus on providing a wide-scale interoperability that allows the sharing of IoT devices across domains and the reusing of available knowledge on the web. However, the application development is difficult because developers have to do various tasks such as designing an application, annotating IoT data, interpreting data, and combining application domains.
To address the above challenges, this paper demonstrates SWoTSuite, a toolkit for prototyping SWoT applications. It hides the use of semantic web technologies as much as possible to avoid the burden of designing SWoT applications that involves designing ontologies, annotating sensor data, and using reasoning mechanisms to enrich data. Taking inspiration from sharing and reuse approaches, SWoTSuite reuses data and vocabularies. It leverages existing technologies to build applications. We take a hello world naturopathy application as an example and demonstrate an application development process using SWoTSuite. The demo video is available at URL `http://tinyurl.com/zs9flrt`.

1 Introduction

In recent years, we have been witnessing a growing number of applications exploiting sensors are becoming increasingly popular. However, existing applications are largely specific to a domain and they lack methods to create innovative cross-domain applications. Some examples of cross-domain SWoT applications [1] include recommending food according to weather forecasts, home remedies according to health measurements, and safety equipment in a smart car according to the weather. To build these applications, developers have to perform various tasks such as designing an application, semantically annotating IoT data, and interpreting IoT data. This requires the learning of semantic web technologies and tools, which is a time consuming process.

The aim of our work is to reduce the development effort for prototyping SWoT applications [1]. Taking inspiration from Content Management Systems (e.g., Drupal, Wordpress) that enables a design of websites without learning web technologies, we reduce the gap of learning semantic web technologies by developing a framework that assists developers designing cross-domain SWoT applications. We design and implement SWoTSuite[5], a suite of tools for prototyping SWoT applications. We first present an architecture of SWoTSuite and then an application development process using it.

2 SWoTSuite architecture

In the following, we describe an architecture of SWoTSuite and its components.

Semantic annotator. It interacts with IoT devices that return data in heterogeneous formats. It converts sensors metadata in an unified description to provide further reasoning to overcome heterogeneity issues, similar to approach adopted by Semantic Sensor Observation Service (SemSOS) [2]. The sensor metadata is semantically annotated according to the M3 taxonomy [1, p. 93](an extension of W3C Semantic Sensor Network (SSN) ontology) that covers the limitations of SSN [3].

Persistent storage. It stores the M3 ontologies, M3 datasets [1], M3 rules [1], and unified sensor data received from the annotator. Moreover, it stores pre-written M3 compatible SPARQL queries to assist developers. The current implementation of persistent storage uses a triple store to store M3 ontologies, M3 datasets, unified sensor data received from the converter. The M3 rules and SPARQL queries are stored as files.

Knowledge manager. It is responsible for indexing, designing, reusing, and combining domain-specific knowledge that further update knowledge in the persistent storage. We have been building datasets to reference, classify, and reuse domain-specific knowledge that we call Linked Open Vocabularies for the Internet of Things (LOV4IoT)[6]. The LOVIoT provides domain ontologies, datasets, and rules that could be reused to design cross-domain SWoT applications.

Reasoning engine. It infers high-level knowledge using a Jena inference engine and M3 rules. The M3 rules are a set of rules that are extracted from LOV4IoT and redesigned to be compliant with the M3 taxonomy.

Query engine. It executes queries and provides smart suggestions. The query engine has been implemented using ARQ, a SPARQL processor for Jena. The query engine loads the M3 ontologies, M3 datasets, knowledge deduced from the reasoning engine, and executes SPARQL queries in order to provide suggestions.

[5] `http://sensormeasurement.appspot.com/`

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

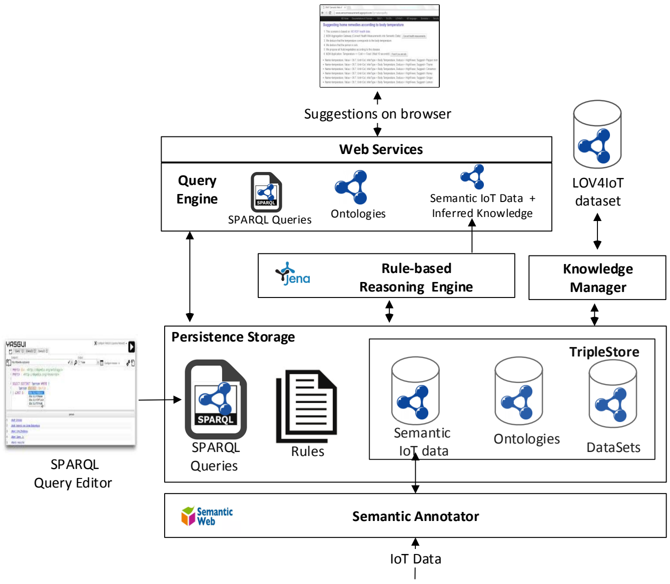

Fig. 1. SWoTSuite architecture

2.1 Demonstration

For demonstration purposes, we take an application that combines data produced by a body thermometer with other domains such as healthcare and food knowledge bases. It suggests some preventive measures, such as home remedies when a fever is detected (e.g., drink orange or lemon juice because they contain Vitamin C) or consulting a doctor immediately in case of severity, that can be acted upon by patients. Application development using SWoTSuite consists of the following steps. The steps in even more detail can be found at URL[7] and the recorded video is available at URL[8].

Step 1: Generating an IoT application template. It selects a pre-defined template from the persistent storage. The template consists of ontologies, datasets, rules, and SPARQL queries that are required to build a SWoT application. To select a template, developers provide a set of sensors (e.g., body thermometer) and the related application domain (e.g., healthcare) information. SWoTSuite queries the persistent storage with this information and returns templates to select. The user interface is available at URL[9].

Step 2: Semantically annotating data. Developers provide SenML data to the semantic annotator to get the RDF/XML data. We provide two options

[7] `http://sensormeasurement.appspot.com/?p=end_to_end_scenario`
[8] `http://tinyurl.com/zs9flrt`
[9] `http://sensormeasurement.appspot.com/?p=m3api`

for interacting with the semantic annotator[10]. Option 1 allows the developers to enter a URL of data source; in option 2, the developers enter SenML data manually and press the converter button to get annotated data. Currently, we have been working on different ways of annotating sensor data. One of ways is tools like Kino [4] does. The kino accepts documents via a special interface[11].

Step 3: Executing the reasoning engine. It deduces new knowledge from annotated IoT data. To perform this task, developers are guided through Linked Open Rules available in the persistent storage. An example of a rule is "if body temperature is between 37.5°C–39°C, then high fever". Moreover, the developers are guided through Java code[12] to load rules.

Step 4: Executing the query engine. It executes SPARQL queries and provides suggestions. We guide developers to use ARQ, a SPARQL query processor for Jena. The query engine loads the M3 ontologies, datasets, annotated data (Step 1), derived suggestions (Step 3), and SPARQL queries.

Step 5: Displaying results. It receives results from the query engine (Step 4) and displays derived knowledge and suggestions on a Web browser[13].

We have evaluated our framework on three aspects: the performance of the semantic annotator given a size of data [1, p. 85], the performance of the reasoner given a number of rules [1, p. 86], and the performance of different tasks according to a size of data [1, p. 87]. We are going to evaluate the usability of SWoTSuite with real users through a Google form at our ISWC 2016 tutorial [5]. The web link of this tutorial is available at URL[14].

References

1. Gyrard, A.: Designing cross-domain semantic Web of things applications. Theses, Télécom ParisTech (April 2015)
2. Henson, C.A., Pschorr, J.K., Sheth, A.P., Thirunarayan, K.: Semsos: Semantic sensor observation service. In: Collaborative Technologies and Systems, 2009. CTS '09. International Symposium on. (May 2009) 44–53
3. Gyrard, A., Datta, S.K., Bonnet, C., Boudaoud, K.: Standardizing generic cross-domain applications in internet of things. In: 2014 IEEE Globecom Workshops (GC Wkshps). (Dec 2014) 589–594
4. Ranabahu, A., Parikh, P., Panahiazar, M., Sheth, A., Logan-Klumpler, F.: Kino: A generic document management system for biologists using sa-rest and faceted search. In: Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on. (Sept 2011) 205–208
5. Gyrard, A., Patel, P., Datta, S., Ali, M.: Semantic web meets internet of things (iot) and web of things (wot). In: The 15th International Conference on Semantic Web (ISWC). (Oct 2016)

[10] `http://www.sensormeasurement.appspot.com/?p=senml_converter`

[11] `https://www.youtube.com/watch?v=12R81HrlAF8`

[12] `http://sensormeasurement.appspot.com/tutorial/M3ApplicationTutorial.zip`

[13] `http://www.sensormeasurement.appspot.com/?p=naturopathy`

[14] `http://sensormeasurement.appspot.com/?p=ISWC2016Tutorial`