WOTS2E: A Search Engine for a Semantic Web of Things

Andreas Kamilaris, Semih Yumusak and Muhammad Intizar Ali
Insight Centre for Data Analytics
National University of Ireland, Galway
Emails: andreas.kamilaris@insight-centre.org, semih.yumusak@karatay.edu.tr
ali.intizar@insight-centre.org

Abstract—A Semantic Web of Things (SWoT) brings together the Semantic Web and the Web of Things (WoT), associating semantically annotated information to web-enabled physical devices, services and their data, towards seamless data integration and better understanding of real-world information. A missing element in order to realize SWoT is a standardized, scalable and flexible way to globally discover in (near) real time web-connected embedded devices, as well as their semantic data. To address this gap, we propose WOT Semantic Search Engine (WOTS2E), which is a search engine for the SWoT, based on web crawling, being able to discover Linked Data endpoints and, through them, WoT-enabled devices and their services. In this paper, we describe the design, development and implementation of WOTS2E, as well as an evaluation procedure showing its operation and performance across the web.

Keywords—Web of Things; Semantic Web; Search Engine; Discovery.

I. INTRODUCTION

While the Internet of Things (IoT) [1] enables communication between heterogeneous devices at the network layer, the Web of Things (WoT) [2], [3], [4] envisions interoperability at the application layer, by reusing well-accepted and understood web standards. By using web technologies, protocols, description languages and formats such as REST, XML, JSON, MQTT, XMPP, Atom, WADL, OpenID and OAuth, the WoT has contributed in reducing the barriers for common understanding and smooth interplay among heterogeneous real-world devices, services and data. Although services offered by WoT devices can be described by using the REST architectural style [5], [6] (or WS-* services in some cases), microformats1 and description languages such as SensorML2, WADL3 and WSDL4, still the data provided by the WoT require some annotations to understand its semantics.

This need of understanding the data produced by real-world devices and services, led to the proposal of a Semantic Web of Things (SWoT) [7], [8], which is about bringing together the Semantic Web and the IoT/WoT. Its goal is to associate semantically rich and easily accessible information to real-world objects. Various efforts that have contributed in enabling the SWoT include ontologies for device and data annotation such as OntoSensor [9] and the SSN ontology for semantic sensor networks [10], as well as recent projects, like UBIWARE [11] and SPITFIRE [12]. These projects have combined Linked Open Data (LOD) and semantic technologies such as RDF5, RDF-S6, OWL7 and SPARQL8, to build complete frameworks for a semantically enabled IoT/WoT.

Although the aforementioned projects, technologies, description languages and formats have the potential to enable a true SWoT, dealing with all aspects related to device web enablement, interaction, service/data description, messaging, security, sharing and service orchestration (i.e. web and physical mashups [13]), there is a missing critical element, which is the "Achilles heel" of WoT [14]: the need for a standardized, scalable and flexible way to globally discover in real time web-connected embedded devices, as well as the semantic data produced by their monitoring activities, which is being continuously published on the web. The information and services discovered could then be used by human experts/programmers and machines, even by average users in the future, in order to create more advanced knowledge combining info from various distributed sources.

Aiming to address this gap, in this paper, we present WOT Semantic Search Engine (WOTS2E) which is a global (near) real-time search engine for the SWoT, being able to discover Linked Data endpoints, defined as online query gateways to semantically annotated Linked Data sources (i.e. SPARQL endpoints) and, through them, WoT-enabled devices and their services. Our contribution is thus to propose a novel method for discovering WoT devices/services and semantically annotated data on the web related to IoT/WoT, by using specialized web crawlers acting as agents for a discovery search engine on SWoT.

The remainder of the paper is organized as follows: Section II lists related work and Section III presents the methodology we followed to design and develop WOTS2E. Then, Section IV describes our implementation efforts while Section V evaluates WOTS2E as a global search engine for the WoT.

978-1-5090-4130-5/16/$31.00 ©2016 European Union;

2SensorML. http://www.opengeospatial.org/standards/sensorml
3WADL. https://www.w3.org/Submission/wadl/
4WSDL. https://www.w3.org/TR/wSDL
5RDF. https://www.w3.org/RDF/
6RDF-S. https://www.w3.org/TR/rdf-schema/
7OWL. https://www.w3.org/OWL/
8SPARQL. https://www.w3.org/TR/sparql11-query/
Finally, Section VI discusses the overall findings and Section VII concludes the paper.

II. RELATED WORK

The absence of standardized discovery methods for the WoT [14] led to the development of online, global sensor directories and collections, such as Xively\(^9\), SenseWeb [15], SemSOS [16] and the SWE discovery framework [17]. A key feature of these online directories/registries is that they provide open Web APIs supporting the development of third-party applications. The main drawback is that they are centralized, with a single point of failure.

Decentralized approaches have also been proposed, such as IrisNet [18], which uses a hierarchical architecture for a worldwide sensor Web. G-Sense [19] is a peer-to-peer (P2P) system for global sensing and monitoring. These approaches, although more robust and scalable, do not effectively solve the problem of sensor discovery as they still require sensor registration to dedicated gateways and servers, which need to maintain a hierarchical or P2P structure among them.

More recent approaches towards real-time discovery of physical entities include Snoogle [20] and Dyser [21], [22]. Snoogle is an information retrieval system for wireless sensor networks, but it is questionable whether it could scale for the world wide web. Dyser follows a different approach, focusing on entities rather than on sensor devices (e.g. whether a classroom is occupied or not). To achieve this, the authors exploit the periodic nature of people-centric sensors by using appropriate prediction models. However, these techniques are computationally-expensive and there are no guarantees for periodic behavior of entities. Besides, Dyser requires additional Internet infrastructure such as sensor gateways.

On the other hand, microformats suggest ways for making HTTP data available for indexing and searching. DiscoWoT [23] is a semantic discovery service based on microformats and RDFa\(^10\), proposing a web crawling discovery strategy. However, it does not include an implementation nor considers well-accepted Semantic Web technologies such as RDF, OWL and SPARQL for WoT data description.

Moreover, the utilization of the Domain Name System (DNS) as a scalable, pervasive, global meta-data repository for embedded devices, and its extension for supporting location-based discovery of Web-enabled physical entities was proposed in [24], [25]. However, this technique requires changes in the existing internet infrastructure.

Another limitation of related work is that most of the aforementioned approaches (except for [16], [17], [23]) do not focus on SWoT, Linked Data and Semantic Web technologies. These limitations encouraged us to examine the possibility of exploiting web crawling for discovery of Linked Data endpoints, and, through them, discovery of WoT devices and services, as well as acquisition of semantically annotated IoT/WoT data. Our technique relates to the discovery service referred in SPITFIRE [12], but WOTS2E scales to the whole web with a complete implementation and evaluation. SPITFIRE only suggested web crawling in its architecture, demonstrating a small prototype in an office building and parking space.

III. WOTS2E SEMANTIC SEARCH ENGINE

The operation of WOTS2E consists of the following steps:

A. Discovery of Linked Data endpoints, by means of web crawlers that crawl the world wide web.

B. Examination of each discovered Linked Data endpoint to understand whether it relates to an IoT/WoT project, deployment or infrastructure.

C. Analysis of the IoT/WoT Linked Data endpoints to acquire meta-data and information about the IoT/WoT devices and services available.

D. Recording of the IoT/WoT devices and services discovered, along with their service and data description information.

Each of these four main steps is described in the following subsections. Figure 1 depicts how WOTS2E works, illustrating the whole discovery procedure when web clients interact with WOTS2E for discovery information retrieval.

A. Discovery of Linked Data Endpoints

The first step requires the use of various web crawlers that continuously scan the web for discovery of Linked Data endpoints, with a frequency of one complete scan per day or per week (if per day is very ambitious). Metacrawling can be employed to increase scan frequency, by using current popular search engines (e.g. Bing, Yahoo, Google) to accelerate the discovery procedure. The initial seed list of the crawlers includes well-known repositories that regularly keep track of linked data resources on the Web. These are Datahub.io [26], Linking Open Data community project (LOD Cloud) [27], LODStats [28] and SPARQL Endpoint Status (SPARQLES) [29].

---

\(^9\)https://xively.com/

\(^10\)RDFa. https://rdfa.info/
B. Examination of Discovered Linked Data Endpoints

Each discovered endpoint needs to be examined in order to see whether it contains IoT/WoT datasets and ontologies, schemas and data. Popular ontologies used to describe real-world sensing services include OntoSensor [9], SSN [10], OpenIoT [30], IoT [30] and OGC Sensor Web Enablement\footnote{OGC SWE. http://www.opengis.net/ogc/markeets-technologies/swe} (SWE). Less popular ones involve SPITFIRE, SmartBuilding, SAREF, Mirabel and COSE. Hence, by querying these endpoints, checking whether they contain data from the aforementioned (or other relevant) ontologies, it can be decided if they relate to an IoT/WoT project, deployment or infrastructure.

Another possibility would be to explore well-known dataset descriptions, such as the Vocabulary of Interlinked Datasets\footnote{VoID. https://www.w3.org/TR/void/} (VoID) or the SPARQL Service Description\footnote{SPARQL-SD. https://www.w3.org/TR/sparql11-service-description/} (SPARQL-SD), for SPARQL endpoints. VoID and SPARQL-SD can provide indicators whether a given Linked Data endpoint contains WoT device/service details. However, this approach strongly depends on the provision of dataset description by data publishers. Both dataset owners and Linked Data endpoint service providers can be encouraged to provide a detailed description indicating whether their endpoints contain any WoT-related information.

C. Analysis of Linked Data Endpoints and WoT Device Discovery

This is an important step in the operation of WOTS2E, because it needs to understand which WoT devices and services are represented by the newly discovered endpoint. This can be achieved in one of the following ways:

- The VoID/SPARQL-SD files describe the WoT devices/services available (e.g. in LOD Cloud [27]).
- Information about WoT devices/services is provided through the endpoint’s datasets (i.e. RDF triples). In this case, the user needs to generate appropriate requests (i.e. SPARQL queries), to retrieve the discovery information (e.g. in SPITFIRE [12]).
- Use of open APIs, supported by the Linked Data endpoints, which allow the client (i.e. web crawlers) to get the discovery information via REST HTTP requests (e.g. in SemSOS [16]).

At each of the above cases, device information could include name, description, IP address, location etc. Description of services could be provided as a URL that leads to a service description language file, written in SensorML, WADL, WSDL etc. This would facilitate automatic machine-to-machine interaction. Service description could also be a link to some online API documentation.

D. Recording of WoT Devices and Services Discovered

In this final step, the information about IoT/WoT Linked Data endpoints, devices and their services, datasets and ontologies needs to become available to the public so that clients could discover relevant devices and directly interact with them (or retrieve historical data), and machines could locate devices and services required in various WoT applications, physical mashups [13] and urban mashups [31].

Hence, discovery of WoT devices/services needs to be organized in one of the following categories:

- **Service type**: Listing WoT devices based on the type of the service offered (e.g. monitoring traffic, air quality, temperature etc.).
- **Location**: Listing devices located in a particular area (e.g. floor, building, neighborhood, city, country etc.).
- **Time**: Listing devices producing data for a certain time period (e.g. during the last two days, one week etc.).
- **Features**: Listing devices based on particular properties (e.g. high reputation, accuracy, robustness etc.).
- **Interaction**: Listing devices according to the type of interaction they support (e.g. request-response, notifications on specified frequent time intervals, publish-subscribe etc.).

Another thing to consider are restrictions, as many WoT devices have certain security/usage constraints or they are located behind firewalls, configured to accept particular clients or a maximum number of requests. It is good to list those restrictions too, wherever possible.

IV. IMPLEMENTATION

In our implementation, we considered Linked Data endpoints to be represented as SPARQL endpoints, as SPARQL is currently the most popular way to retrieve and manipulate data stored in RDF format. In the future, we are willing to include other Linked Data endpoints too, if available and widely used. To understand how SPARQL endpoints are embedded in HTML code, we examined manually the majority of datasets listed in Datahub.io (ca. 250 datasets). The common patterns revealing presence of SPARQL endpoints, covering all the cases we examined, are listed in Table I (starting from the most common patterns encountered).

<table>
<thead>
<tr>
<th>Common Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;meta name=“Keywords” content=“OpenLink Virtuoso Sparql”&gt;</td>
</tr>
<tr>
<td>Virtuoso SPARQL Query Editor</td>
</tr>
<tr>
<td>OpenLink Software</td>
</tr>
<tr>
<td>&lt;a href=“<a href="http://www.openlinksw.com/virtuoso%E2%80%9D">http://www.openlinksw.com/virtuoso”</a> &gt;</td>
</tr>
<tr>
<td>&lt;label for=“sparql”&gt;&lt;SPARQL&gt;&lt;/a&gt;</td>
</tr>
<tr>
<td>&lt;label for=“query”&gt;Query text&lt;/label&gt;</td>
</tr>
<tr>
<td>OpenLink Virtuoso SPARQL Query</td>
</tr>
<tr>
<td>Virtuoso SPARQL Query Form</td>
</tr>
</tbody>
</table>

**TABLE I**

**MOST COMMONLY USED SPARQL ENDPOINT PATTERNS IN HTML**

These patterns were provided as an input to our web crawlers, in order to search the web for available SPARQL endpoints (see Section III-A). For web crawling, we used a metacrawling service [32] that utilizes HtmlUnit\footnote{HtmlUnit. http://htmlunit.sourceforge.net/}, for exploiting the search functionality available over popular search engines. The patterns listed in Table I were provided as input to
In order to analyze the URLs retrieved, the Jena Framework\textsuperscript{15} was used to send SPARQL queries to the candidate endpoints, checking whether they are valid SPARQL endpoints or not, such as the one below:

\begin{verbatim}
SELECT DISTINCT ?Concept
WHERE {[] a ?Concept} LIMIT 100
\end{verbatim}

In the next step, all valid SPARQL endpoints were examined whether they contain information related to IoT/WoT (see Section III-B). To achieve this, various queries were performed on the discovered endpoints, in order to understand whether they contained relevant ontologies (OntoSensor, SSN, OpenIoT, IoT, OGC SWE etc.). Every SPARQL endpoint returning valid results on these queries was recorded by our search engine as a valid resource of IoT/WoT content. An example query examining the presence of the SSN ontology is shown below:

\begin{verbatim}
SELECT * WHERE {{?s ?p ?o}
UNION {GRAPH ?g {?s ?p ?o}}
FILTER regex(?o, "/ssn").
FILTER isIRI(?o).}
\end{verbatim}

An easy alternative approach is to look into VoID description of a dataset assuming that a complete list of vocabularies used in a dataset are described using void:vocabulary and a query over VoID description can verify whether any of the listed vocabularies belongs to IoT/WoT-related ontologies.

After labeling some SPARQL endpoint as related to IoT/WoT, the next step was to analyze it, discovering which devices/services are available through it. From the three options listed in Section III-C, the first two were implemented. At the first, the VoID file was adapted to reveal information about WoT devices and services, as in the following example:

\begin{verbatim}
:ExampleDataset a void:Dataset;
void:subset :ExampleSensor .

:ExampleSensor a void:Dataset;
dcterm:title "WoT Example Sensor";
dcterm:description "http://../sens.wadl";
dcterm:contributor "Insight Centre";
dcterm:source "140.203.154.11" .
\end{verbatim}

A dataset in VoID is a set of RDF triples that are published and maintained by a single provider. We used the void:subset property to represent WoT devices assuming each subset represents a single WoT device. Since each void:subset itself is a void:Dataset, we used the Dublin Core terms such as dcterm:title for the device name, dcterm:description for the link to the device’s description language file (e.g. WADL, SensorML etc.), dcterm:contributor to list the owner of the device and dcterm:source to provide the URL or IP address for direct interaction with the device. SPARQL endpoints using SPARQL-SD for the endpoints description can be easily aligned with the VoID vocabulary, based on the fact that void:Dataset is a super class of sd:Dataset and of sd:Graph.

Therefore, any instance of these classes can be described just like any other VoID dataset.

A second approach for device/service discovery was to extend SSN, which is one of the most popular ontologies for sensors, in order to include discovery/description information through some ontology that describes web services. To achieve this, we linked the OWL-S ontology\textsuperscript{33} with SSN, as shown in Figure 2. In this case, a simple SPARQL query at the endpoint would then provide all the discovery information required.

\begin{verbatim}
Prefix iot: <http://purl.org/IoT/iot#>
Prefix ssn: <http://purl.oclc.org/..ssn#>
SELECT ?sm ?device ?service
WHERE{
  ?sm a iot:smart_entity
  ?sm iot:has_part_device ?device
  ?device ssn:observes ?service
  ?service a iot:Temperature
}
\end{verbatim}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Fig_2.png}
\caption{Information model for WoT device/service discovery/description.}
\end{figure}

In the final step (see Section III-D), all information about SPARQL endpoints, devices/services discovered and sensor meta-data information was stored as RDF triples on a Virtuoso RDF store, which was installed on WOTS2E. The most appropriate ontology to adopt in this case was the IoT ontology\textsuperscript{30}, as it allows to provide rich meta-data information about sensors, services and their measurements, by extending the SSN ontology. This information includes device and service type, location, features and interaction types. We argue that the iot:Resource class can be used to store the URL to the description language file (e.g. WADL, SensorML), while the iot:WoT could contain the domain or IP address of the WoT device. As an example, an appropriate SPARQL query to get all WoT devices measuring temperature would be:

\begin{verbatim}
Prefix iot: <http://purl.org/IoT/iot#>
Prefix ssn: <http://purl.oclc.org/..ssn#>
SELECT ?sm ?device ?service
WHERE{
  ?sm a iot:smart_entity
  ?sm iot:has_part_device ?device
  ?device ssn:observes ?service
  ?service a iot:Temperature
}
\end{verbatim}

V. Evaluation

To evaluate WOTS2E, its (meta-)crawling service ran for 24 hours, using the patterns listed in Table I, in order to get relevant URLs from the Bing, Yahoo, Google, Baidu, and Yandex search engines. The second row of Table II shows the
active and inactive endpoints discovered by WOTS2E. As of June 2016, totally 638 active endpoints have been discovered, which are 2.47 times more the endpoints available in Datahub (third row of Table II). This is an indication of the effectiveness of WOTS2E’s crawling service, in comparison to a popular and widely used, but still static repository.

### Table II

<table>
<thead>
<tr>
<th></th>
<th>Active</th>
<th>Inactive</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOTS2E</td>
<td>638</td>
<td>640</td>
<td>1278</td>
</tr>
<tr>
<td>Datahub</td>
<td>258</td>
<td>296</td>
<td>554</td>
</tr>
</tbody>
</table>

After collecting these 638 active SPARQL endpoints, they were examined one by one for relevance to IoT/WoT. From these 638 endpoints, Table III shows the relevant ones, according to the criteria listed in Section III-B.

Table III, in addition to WoT-related ontologies, also includes some more generic ontologies (i.e. DBPedia, RDFS, SKOS, PROV) because they contained IoT/WoT-related object types (e.g. dbr:Pressure_sensor, rdfs:sensoryUnit etc.).

### Table III

**IoT/WoT-Relevant SPARQL Endpoints Discovered by WOTS2E**

<table>
<thead>
<tr>
<th>Ontology</th>
<th>No. of endpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN</td>
<td>13</td>
</tr>
<tr>
<td>DBPedia</td>
<td>13</td>
</tr>
<tr>
<td>SmartBuilding</td>
<td>3</td>
</tr>
<tr>
<td>DogOnt</td>
<td>2</td>
</tr>
<tr>
<td>DUL</td>
<td>2</td>
</tr>
<tr>
<td>km4city</td>
<td>2</td>
</tr>
<tr>
<td>OpenEI</td>
<td>2</td>
</tr>
<tr>
<td>RDFS</td>
<td>2</td>
</tr>
<tr>
<td>SKOS</td>
<td>2</td>
</tr>
<tr>
<td>Fan Fpai</td>
<td>1</td>
</tr>
<tr>
<td>Fiemser</td>
<td>1</td>
</tr>
<tr>
<td>IoT</td>
<td>1</td>
</tr>
<tr>
<td>PROV</td>
<td>1</td>
</tr>
<tr>
<td>SAREF</td>
<td>1</td>
</tr>
</tbody>
</table>

In Table IV, we further collected IoT/WoT-specific triples from the endpoints which are directly related to IoT/WoT ontologies. Although there are several IoT/WoT ontologies created for common use, only a limited number of triples are available online. According to these results, the SSN ontology is mostly used in the domain. We have further analyzed the occurrence of different properties under the SSN ontology in Table V, with `featureofinterest` being the more popular one.

### Table IV

**Number of Triples Discovered using IoT/WoT ontologies**

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Number of Triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN</td>
<td>1,435,248</td>
</tr>
<tr>
<td>DUL</td>
<td>182</td>
</tr>
<tr>
<td>km4city</td>
<td>56</td>
</tr>
<tr>
<td>Fiemser</td>
<td>50</td>
</tr>
<tr>
<td>OpenIoT</td>
<td>44</td>
</tr>
<tr>
<td>SmartBuilding</td>
<td>36</td>
</tr>
<tr>
<td>DogOnt</td>
<td>24</td>
</tr>
<tr>
<td>SAREF</td>
<td>4</td>
</tr>
<tr>
<td>Fan Fpai</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table V

**Number of SSN properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>featureofinterest</td>
<td>182,434</td>
</tr>
<tr>
<td>observedproperty</td>
<td>178,638</td>
</tr>
<tr>
<td>observationvalue</td>
<td>178,620</td>
</tr>
<tr>
<td>observationresult</td>
<td>178,604</td>
</tr>
<tr>
<td>hasvalue</td>
<td>178,592</td>
</tr>
<tr>
<td>observationresulttime</td>
<td>176,082</td>
</tr>
<tr>
<td>observationsamplingtime</td>
<td>176,070</td>
</tr>
<tr>
<td>observationresulttime</td>
<td>176,060</td>
</tr>
<tr>
<td>isproducedby</td>
<td>2,564</td>
</tr>
<tr>
<td>madeobservation</td>
<td>2,532</td>
</tr>
<tr>
<td>starttime</td>
<td>182</td>
</tr>
<tr>
<td>endtime</td>
<td>168</td>
</tr>
<tr>
<td>property</td>
<td>72</td>
</tr>
<tr>
<td>hasproperty</td>
<td>42</td>
</tr>
<tr>
<td>observes</td>
<td>20</td>
</tr>
<tr>
<td>observedby</td>
<td>10</td>
</tr>
<tr>
<td>sensor</td>
<td>10</td>
</tr>
</tbody>
</table>

### VI. Discussion

Only a very small subset of available SPARQL endpoints (46, 7.2%) seem to relate to IoT/WoT based on our evaluation procedure. We believe that many more Linked Data endpoints are relevant, however they need to somehow reveal this information, either through the VoID/SPARQL-SDL files or by employing some well-known IoT/WoT ontologies, at least for describing the devices/services they support.

Our evaluation focused on global discovery of relevant endpoints, and not on their analysis for device/service discovery or the recording on WOTS2E of the WoT devices/services discovered on these endpoints. The reason is because our discovery technique and its guidelines have not yet been adopted by any of the producers of the SPARQL endpoints. Device and service discovery constitute an open issue of the WoT and standardized approaches do not exist. This issue becomes more complicated on the SWoT, where devices, services and data are behind Linked Data endpoints. Our method aims to suggest a solid proposal on how to achieve discovery on SWoT seamlessly and with minimum effort. This discovery service, provided as a middleware API, could facilitate the efforts of developers and machines to create advanced knowledge and complex services combining information from various web-based sources. As we plan to provide a user-friendly website to incrementally let users to access the discovered lists of services in a well-organized way, this search engine could be used even by average users in the near future.

A key problem will be to provide enough context to understand discovered services and data. Finding sensors/services measuring some IoT entity is not sufficient. For their information to be valuable, we need a very good understanding of their context and/or operation. WOTS2E can answer all queries as defined by the discovered ontologies, however, we still need good vocabularies/ontologies and descriptions for identifying relevant and appropriate services and information.

In our implementation, we used a metacrawling approach to accelerate the procedure of discovering relevant endpoints, al-
though our methodology suggests classical web crawling. Our aim is not to propose a discovery service heavily dependent on commercial search engines, but we only used metacrawling as a proof of concept, indicating the potential of scanning the whole web for SWoT context in less than 24 hours.

Finally, we note that local discovery of devices/services by their local gateways (or their Linked Data endpoint representatives) was not covered in this paper. This can be achieved by various IoT/WoT local discovery protocols available, such as UPnP, mDNS, DNS-SD, SLP, Sun’s Jini, Apple’s Bonjour, WS-Discovery etc.

VII. Conclusion and Future Work

In this paper, we have described a search engine for the Semantic Web of Things, based on web crawling, proposing a scalable and flexible way to globally discover in real time web-connected embedded devices, as well as their semantic data. We have presented the WOT Semantic Search Engine (WOTS2E), which is a search engine being able to discover Linked Data endpoints and, through them, WoT-enabled devices and their services. This paper explained the architecture and implementation of WOTS2E, showing its operation and performance across the web through an evaluation procedure.

As future work, we aim to contribute to the standardization efforts on the WoT (W3C WoT IG\textsuperscript{16}, OGC Sensor Web Interface for IoT SWG\textsuperscript{17}), promoting WOTS2E as a viable solution for a SWoT search engine. At the same time, we will work on improving WOTS2E in terms of better identifying relevant Linked Data endpoints, discovering IoT/WoT devices/services represented by the endpoints, as well as properly recording those devices/services for seamless lookups and search queries by web clients. Finally, a graphical user interface and a RESTful API is currently under development, in order to make the services of WOTS2E publicly available on the web in well-defined and presented structure/organization, easily accessible even for average users.

ACKNOWLEDGMENT

This research has been supported by Science Foundation Ireland (SFI) under grant No. SFI/12/RC/2289 and EU FP7 CityPulse Project under grant No.603095.

REFERENCES


\textsuperscript{16}W3C WoT IG. https://www.w3.org/2010/IG.

\textsuperscript{17}OGC Sensor Web interface for IoT SWG. http://opengeospatial.org/projects/groups/sweiotswg


