Demo Abstract: SemIoTic: Bridging the Semantic Gap in IoT Spaces

Sumaya Almanee, Georgios Bouloukakis, Daokun Jiang, Sameera Ghayyur, Dhrubajyoti Ghosh, Peeyush Gupta, Yiming Lin, Sharad Mehrotra, Primal Pappachan, Eun-Jeong Shin, et al.

Demo Abstract: SemIoTic: Bridging the Semantic Gap in IoT Spaces

Sumaya Almanee, Georgios Bouloukakis, Daokun Jiang, Sameera Ghayyur, Dhrubajyoti Ghosh, Peeyush Gupta, Yiming Lin, Sharad Mehrotra, Primal Pappachan, Eun-Jeong Shin, Nalini Venkatasubramanian, Guoxi Wang, Roberto Yus
UC Irvine,
{salmanee,gboulouk,daokunj,sghayyur,dhrubajg,peeyushg,yiminl18,primal,ejshin1,guoxiw1,ryuspeir}@uci.edu,
{sharad,nalini}@ics.uci.edu

ABSTRACT

This demonstration showcases the SemIoTic middleware [2] which provides inhabitants of an IoT space, as well as developers of applications, with a semantic view of the space. Participants will have an opportunity to see how useful IoT applications can be easily developed focusing on describing what information is needed without having to deal with the underlying IoT device infrastructure.

CCS CONCEPTS

• **Software and its engineering** → **Interoperability**; • **Hardware** → **Sensor applications and deployments**.

KEYWORDS

Internet-of-Things, Semantic interoperability, Open APIs

ACM Reference Format:
Sumaya Almanee, Georgios Bouloukakis, Daokun Jiang, Sameera Ghayyur, Dhrubajyoti Ghosh, Peeyush Gupta, Yiming Lin, Sharad Mehrotra, Primal Pappachan, Eun-Jeong Shin, Nalini Venkatasubramanian, Guoxi Wang, Roberto Yus . 2019. Demo Abstract: SemIoTic: Bridging the Semantic Gap in IoT Spaces. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3360322.3360996

1 INTRODUCTION

Emerging IoT *smart* ecosystems introduce new and useful automation capabilities that provide several interesting features such as security, wellness monitoring, zero-energy sustainable spaces both in homes, offices, and other buildings. Several challenges have to be addressed to support those applications. **(1) "Semantic Gap":** There exists a clear gap between the low-level data captured by IoT devices (i.e., sensors and actuators) and the high-level concepts that users are interested in (e.g., information about the status of spaces and their inhabitants). **(2) Reusability:** Developers would like to maximize the utilization of their applications across different spaces (e.g., at home, at work, in a public building, etc.) which would require space-agnostic implementations that can deal with different underlying device infrastructures. **(3) Interoperability:** Today's IoT devices are highly heterogeneous both in their capabilities as well as their interaction protocols (from Application Programming Interfaces –APIs– to messaging protocols –e.g., MQTT, CoAP–). **(4) Privacy:** The increasing legislative support for user privacy (e.g., the European General Data Protection Regulation –GDPR–) requires applications to provide a certain level of privacy guarantees (e.g., dealing with user-defined privacy policies about capture/retention/deletion of their data). For instance, understanding whether using a specific IoT device to capture information about an individual would be allowed w.r.t. her privacy expectations usually defined at a higher-level.

In summary, the developers of smart space applications bear the burden of understanding users requirements (both in terms of data and privacy) and deployment infrastructure, and translating between needs and IoT devices in a space-and-device-agnostic manner.

2 SEMIOTIC FRAMEWORK

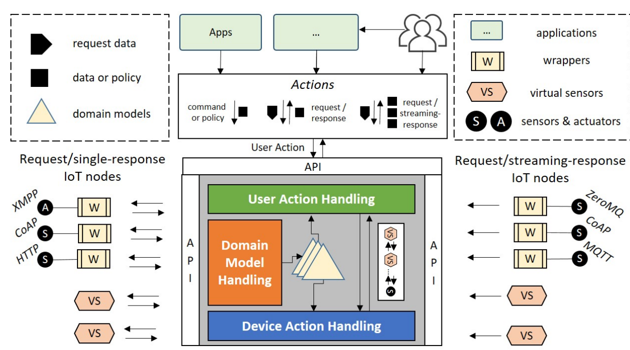

Figure 1: High-level architecture of SemIoTic.

The SemIoTic framework[1] presents a holistic end-to-end vision of smartspaces from applications to devices. SemIoTic [2] aims to facilitate the development and provisioning of IoT smartspace applications. With SemIoTic, users/developers can express their requirements at a high-level, by referencing concepts related to the space or its inhabitants, without having to deal with specific IoT devices. For instance, they can express their desire to "decrease the temperature of rooms where the occupancy is at least half of their

[1] http://tippersweb.ics.uci.edu/semiotic

BuildSys '19, November 13–14, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7005-9/19/11…$15.00
https://doi.org/10.1145/3360322.3360996

capacity and the temperature is above 85F". A SemIoTic instantiation deployed in a smart space handles these *user-defined actions* which include: (i) requests for dynamic or static information about the space (e.g., to obtain the current location of a person or to monitor the occupancy of a specific room every five minutes for the next two hours). (ii) commands related to such entities (e.g., to switch on the AC if the occupancy of the room is high) and (ii) privacy preferences/policies regarding the handling of information (e.g., to deny the capture of any information that can lead to disclosing the location of a person). SemIoTic translates user-defined actions into actions on the underlying device infrastructure defined for the specific space where the request for data takes place through the following three main components (see Fig. 1):

- *Model Handling* enables the definition of the smartspace in terms of types of *spaces*, *users*, and *devices*, as well as specific instances of those types. Semantic Web technologies (e.g., OWL and semantic reasoners) are utilized to develop this component.
- *User Action Handling* takes as input user actions, based on high-level concepts defined in the model, and translates them into an appropriate and feasible plan of actions on the devices deployed in the smartspace. We have implemented a number of ontology-based algorithms that take as input the devices of the space and generate possible execution plans.
- *Device Action Handling* accesses the devices assigned to execute the plan through *wrappers*, that encapsulate the interaction with specific devices, and/or *virtual sensors*, that process raw sensor data to produce semantically meaningful information. We devise abstract APIs that are used to design an extensible architecture of SemIoTic wrappers and virtual sensors.

3 DEMO OVERVIEW

In our demo, participants will play the roles of a smart application developer, wrapper/virtual sensor developer, and administrators of a SemIoTic-enabled space. As such, they will first experience the challenges in dealing with user requirements and IoT devices described in Section 1. Then, they will experience how SemIoTic handles those challenges.

Developing SemIoTic applications. In this part of the demo, participants will experience how SemIoTic facilitates the development of space-agnostic applications. We will showcase three different smart applications built using SemIoTic: (1) *Concierge* – a smart building assistant which provides users with information about the smart space. Among its functionalities, it helps users to locate others in real-time. (2) *Self-Awareness* – provides users with a detailed history of their interaction within a smart space: how much time was spent in specific rooms; with which users, etc. It also serves as a physical activity tracker which keeps count of the steps walked and floors climbed. (3) *Building Analytics* – offers end-users (e.g., space administrators) the ability to visualize information such as the occupancy levels and energy consumption of different parts of the space (e.g., room, floor, region) at various temporal granularities (minutes, hours, days).

During the demonstration, the participants will understand how these applications use SemIoTic interfaces to request such high-level information (e.g., location of people or occupancy of different spaces). They will also see how the same applications can be used, without code modifications, in two different SemIoTic instantiations with different underlying IoT device infrastructures.

Developing SemIoTic wrappers and virtual sensors. In this part of the demo, we will demonstrate how to build wrappers and virtual sensors to obtain data needed by the previous applications from IoT devices regardless of their data semantics, types, or exchange protocols (i.e., MQTT, CoAP, XMPP). Participants will learn how to develop these components by using SemIoTic's extensible wrapper architecture [2] in combination with the DeX API [1]. As sample wrappers, we will show the code developed to communicate with different real sensors: (1) *WiFi Access Point wrapper* – which receives SNMP traps (i.e., events that associate devices with WiFi APs when connected) in real-time; (2) *Bluetooth beacon wrapper* – which receives association events from bluetooth enabled devices that interact with them; (3) *Camera wrapper* – which collects images and/or video data by communicating with an IP camera; and (4) *WeMo outlet meter wrapper* – which connects to WeMo Insight Switches in the network and uses the UPnP protocol to extract energy consumption data from them.

We will also show how to develop virtual sensors for processing raw sensor data without having to deal directly with specific sensors. As sample virtual sensors, we will demonstrate: (1) *Presence detector* – which takes as input connectivity events (which can be obtained from both the WiFi AP and the beacon wrappers) and generates an observation specifying in which room a user is; and (2) *Occupancy counter* – which takes as input presence data within a time interval and generates occupancy values for each space.

Translating user requests. In the final part of the demonstration, the participants will be demonstrated how the high-level requests posed by the applications are automatically translated to the appropriate device requests given the ontology defined for each SemIoTic-enabled space and user policies. First, the participants will help in defining a new SemIoTic instantiation by creating the ontological description of the type of space and devices in it. Then, we will show them how an application request (e.g., to retrieve the occupancy of a space) gets appropriately and automatically translated by using the ontology defined by the attendees.

As privacy is an important concern in IoT based smart spaces, in this part of the demo, we will also showcase to the participants how SemIoTic uses the same translation algorithm to translate user defined privacy policies into IoT device actions. Participants will be able to define policies to highlight under which contextual parameters (e.g., location, time) a specific application requesting data can get access to it. For instance, they will be able to define a policy such as "do not share my location with Concierge on weekdays between 12-1pm" and see how the execution plan generated for a request by Concierge changes with and without the existence of the policy.

REFERENCES

[1] Georgios Bouloukakis, Nikolaos Georgantas, Patient Ntumba, and Valerie Issarny. 2019. Automated Synthesis of Mediators for Middleware-layer Protocol Interoperability in the IoT. *Future Generation Computer Systems* (2019).
[2] Roberto Yus, Georgios Bouloukakis, Sharad Mehrotra, and Nalini Venkatasubramanian. 2019. Abstracting Interactions with IoT Devices Towards a Semantic Vision of Smart Spaces. In *6th Int. Conf. on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys 2019), New York, NY, USA.*