Pers Ubiquit Comput (2014) 18:895–907
DOI 10.1007/s00779-013-0694-7

ORIGINAL ARTICLE

Building sustainable parking lots with the Web of Things

**Sujith Samuel Mathew · Yacine Atif ·
Quan Z. Sheng · Zakaria Maamar**

Received: 10 December 2012 / Accepted: 17 May 2013 / Published online: 27 June 2013

Abstract Peak-time traffic woes create considerable amount of stress and environmental pollution resulting in an economic loss. Research innovations in areas such as the *Web of Things* are able to curtail some of these issues by creating scalable and sustainable environments like parking lots, which provide motorists with access to convenient parking spots. We present a scalable parking lot network infrastructure that exposes parking management operations through a judicious mashup of physical things' services within a parking lot. Our system uses service-oriented architecture, allowing motorists to reserve parking spots in advance. In doing so, our proposed system leverages the use of HTTP and Wi-Fi for the Web enablement and interoperability of things within a parking spot and elevates it as a *Smart Parking Spot* on the Web. Our suggested semantic Web-based structure for representing things makes it possible to query physical things' states and services depending on their capabilities and other relevant parking-related parameters. Our performance evaluation reveals that a maximum of 40 % time is saved to find parking spots and also 40 % reduction in air pollution is observed.

S. S. Mathew (✉) · Q. Z. Sheng
School of Computer Science, University of Adelaide,
Adelaide, Australia
e-mail: sujith@adelaide.edu.au

Q. Z. Sheng
e-mail: quanzheng.sheng@adelaide.edu.au

Y. Atif
College of IT, UAE University, Al Ain, United Arab Emirates
e-mail: yacine.atif@uaeu.ac.ae

Z. Maamar
College of IT, Zayed University, Dubai, United Arab Emirates
e-mail: zakaria.maamar@zu.ac.ae

Keywords Smart parking · Sensor networks · Web of Things · Web services · Semantic Web

1 Introduction

Sustainability measures are surging with the increased awareness of the benefits and long-term implications these measures have on our planet. Sustainability efforts drive economic growth, greater prosperity, and new business opportunities [1]. In this paper, we tackle recurrent problems occurring in parking lots, where motorists spend considerable amount of time looking for parking spots, and in effect contribute to increasing environmental pollution. The financial impact of reducing the time to search for a parking spot was accounted to more than a billion Euros per year in a study done in France [2]. The report estimates about 70 million hours spent in a year by motorists in France looking for parking. Moreover, the number of reported auto-thefts indicates alarming statistics, for example in Canada, 93000 stolen cars were reported in 2010 [3]. With the increasing number of vehicles, these are issues that most modern cities grapple with. Road rage, accidents, abandoned trips, and fuel wastage are some of the other issues induced by the unproductive task of finding a secure parking spot. We approach these problems with the technology advances in the emerging *Web of Things* (WoT).

Recent advances in WoT research gained momentum because of successful research directions in the *Internet of Things* (IoT) domain [4, 5]. Today, the IoT promises a ubiquitous networking platform for representing everyday objects such as pacemakers, sidewalks, traffic lights, and daily commodities as identifiable, readable, addressable, and controllable objects on the Internet [4]. However, a networking infrastructure alone does not enable the success or usability of IoT. Business and industry depend on applications that are built on Web architectures, and on interoperable components (e.g., Web services) upon which these applications are built. The challenge of incorporating the Web as the application platform over IoT is termed as the Web of Things [5, 6]. The reducing cost, size, and technology advances in embedded systems and communication technology have made it possible for computing capabilities to disappear into our surroundings [7, 8]. Yet, with the heterogeneity and variations of real-world things, it is impractical to realize the WoT vision in a large scale, unless they are considered within scaled-down, homogeneous spaces of an application context. We noticed that real-world things are replicated in many spatial patterns, for example, a projector in every classroom, a patient monitor in every hospital room, or a parking sensor in every parking spot. We define *Ambient Spaces* (AS) to be the virtual representations of one or more Web-enabled things that is within such specific spatial contexts, and has information or operations to be represented on the Web [9]. The use of AS has made it possible to represent a large number of physical things and thereby enabling real-world things and people to seamlessly communicate [10, 11]. Novel applications are envisioned in these spaces within the scope of WoT extending accessibility to everyone and for everything from a Web browser.

Our first contribution in this paper is a semantic structure combining several data resources to represent things in a parking lot. The semantic structure that we propose is extendable and provides a generic framework to represent things in different AS instances. We use OWL to describe a capability-based classification of things and their relationships on the Web. We use URIs to represent things, RESTful services to access services provided by these things, and RDF to model the structure of information. The usability of things (such as parking spots) is better when they are classified and represented in a format that is machine-readable and well perceived by people. For example, if we need an application to locate a parking spot for the disabled that is available (*free*) and closest to a hospital, then we need to have things in the physical world linked to each other with a clearly defined semantic structure, so that such queries are made possible. Our proposed semantic structure provides such flexibility to query things based on their capabilities and relevant social properties to seamlessly interact with people and other things.

While it is important to describe a generic representation of real-world things in the virtual world, it is also important to describe how to use AS to create such systems. Our second contribution is the use of our proposed AS to develop a system for a parking lot (see Fig. 1) that

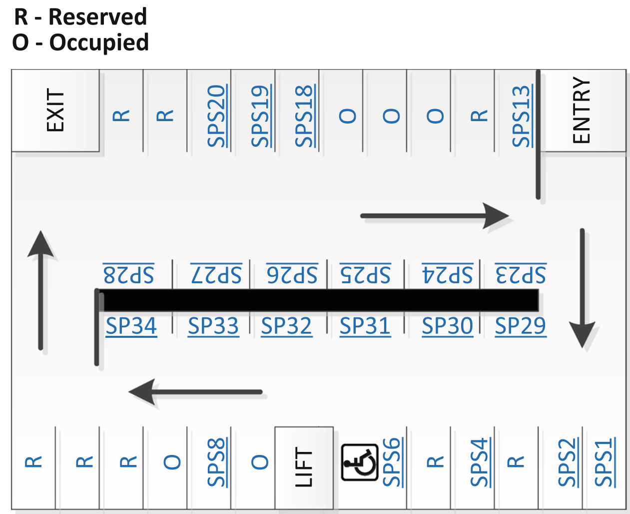


Fig. 1 An online view of a parking lot

abstracts parking spots as *Smart Parking Spots* (SPS) on the Web. Our system adopts a service-oriented architecture (SOA) approach to enable the interoperability of SPS and provides services for the access and control of SPS in a parking lot. We proposed the design of a single SPS earlier where RESTful services are employed to service-enable resource-constrained things such as sensors, displays, and actuators in a loosely coupled way [12]. In this paper, we expand our contribution by designing an ambient parking lot system with many SPSs, interacting with each other. We also illustrate a prototype of the proposed system and the preliminary evaluations conducted to verify the benefits of the system in terms of time saving and environment preservation. Instead of using relatively new protocols that are designed for resource-constrained device like 6LowPAN [13] or Constrained Application Protocol (CoAP), our system leverages the use of HTTP and Wi-Fi for the easy and universal Web enablement as well as interoperability of things within a parking spot. The use of small, low power Wi-Fi enabled Web server exposes a parking spot as an independent unit and reduces the use of wiring and cabling for creating the parking lot's networking infrastructure. This allows our system to scale well for any size of a parking lot, reducing the cost and expanding the design possibilities of parking spaces. Each SPS is considered as a Web resource which allows a remote reservation and an automatic allocation of a secured parking spot by binding that parking spot's services with a motorist's smartphone. The SPS online reservation module is poised to reduce the time spent to locate and occupy a parking spot, which also reduces the pollution within a parking lot. We have evaluated the use of SPS for reserving parking spots and found considerable reduction in carbon emission because of the reduced traffic time to find a spot.

The remaining sections of this paper are organized as follows. A motivation for our work and related works are further discussed in Sect. 2. We then describe the architecture of our system for ambient parking lots in Sect. 3. In Sect. 4, we reveal the design of the parking allocation system using the proposed architecture and present a prototype implementation. In Sect. 5, we analyze and evaluate our system and show the corresponding evaluation results. Finally, in Sect. 6, we summarize our findings and present some suggestions to extend this work to conclude the paper.

2 Motivation and related work

We present the challenges for realizing our system through a review of related works driving this realization, and we also illustrate the user experience through a motivational scenario. The related work survey crosses multiple research areas, and hence, we focus on relevant works that fall within the context of this paper.

2.1 Motivational scenario

Jon drives to his meeting on the twelfth floor of an office building. As he enters the parking lot of the building, his smartphone welcomes him to the parking lot and directs him to a prebooked spot, which he reserved online before starting the trip. Jon quickly locates the spot, and as he eases his car into it, his phone indicates the remaining parking time and announces the receipt of three messages. Jon reads the first one, which indicates that this is his fifth time at the spot within a month and the next visit would be free (as a bonus). The next message indicates that his friend Ben, who is heading to the same meeting as his, is in the parking lot as well, and the third message indicates a book sale at his favorite bookshop on the fifth floor. As it has become a habit, Jon uses his smartphone to trigger a washing service in the parking to clean and polish his car while he is in the building. He notices Ben, cruising around for a free spot and waves out as he walks to the elevator.

The above scenario illustrates the benefits of *Ambient Spaces* (AS) such as parking spots to plan daily tasks using WoT-based approaches. The seamless integration of physical parking spots into Web applications and the exposure of their services to mobile clients reinforce the vision of ambient spaces, where people and physical things interact to deliver a new genre of services that encompass real-world entities.

2.2 Related work

Improving traffic conditions has been an important research focus in recent years [14]. Existing guided parking frameworks [15] use wireless sensor networks (WSN) where each parking spot has a lamp that indicates if it is occupied (*red*) or not (*green*). This is realized through a sensor that detects the presence of a vehicle in the parking spot. Digital displays in the lot indicate the number of available parking spots on each level of the building or for each bifurcated parking area. However, during peak-time traffic situations, existing systems cannot guarantee a free parking spot when a vehicle reaches it, because there are many takers for a spot. Vehicles continue to move around in the parking lot for various lengths of time and in the process polluting the air with vehicular emissions.

Today, many parking lots employ an infrastructure of sensor networks to manage the available parking spots and provide safer parking. These networks are restricted by the lack of resources on the sensors, the heterogeneity of sensor nodes, and the difficulty to redesign or extend the network [16, 17]. Parking solutions based on WSN rely on a centralized storage of information because of sensors' resource restrictions [17]. Therefore, these frameworks have restrictions on scalability and are not flexible. They require cabling and related networking infrastructure, which makes it difficult for existing parking lots to adopt these solutions or extend the existing ones. Moreover, these solutions require various types of nodes for sensing, routing, and management. Our system is easy to deploy and scales well, as each node (parking spot) is equipped with a Wi-Fi module and a Web server connected to things such as parking sensors, displays, and lights. This approach encompasses each parking spot to be an independent hardware component and also virtually represented on the Web. Each parking spot is an independent Web resource with its state and services displayed online. The resources and respective representations are distributed across every parking spot instead of a centralized location and are accessed via Web services.

There are also many online applications that interface with parking lots such as *SFpark* (sfpark.org) in San Francisco and *Parkopedia* (parkopedia.com). SFpark tracks the availability of parking spaces and garages in some areas of San Francisco using sensors that detect free spaces. Though it is for a small area and expects to manage the traffic by varying the cost of parking, the solution provides accurate data about free parking spots. In contrast, our approach provides two other benefits. Firstly, our system couples a parking spot and a user's smartphone ensuring that the vehicle is retrieved only by the owner of the vehicle (i.e., the person who possess the smartphone used for registering the spot), and secondly, our system provides rich semantic data which is able to capture user centric information so as to provide user incentives as mentioned in our scenario in Sect. 2.1. Another popular solution in Europe is Parkopedia that provides parking solutions and also an online interface for users to submit reviews which are manually monitored for validity and acceptance. The system provides limited services, and the data are not openly available to people who would use the parking spots. In contrast, our system provides an open architecture where people access the parking spot directly through Web-based URLs. The semantic structure provides easy access to third party service providers to directly plug in their services, such as washing service as indicated in our scenario in Sect. 2.1.

The use of Web service for enhancing the scalability of WSN has been successfully tested [16]. In comparison with this approach, which uses SOAP, the use of REST architecture [18] for Web services has introduced a new paradigm for resource-constrained devices. This new approach enables the realization of the WoT as an interoperable and an open technology. Yet, the heterogeneous nature of things within a given space requires the use of gateways to bridge RESTful operations of things with the vendor-specific protocols [19, 20] of real-world things. Our system augments candidate things like sensors in a parking lot with Web capabilities, allowing parking services to be exposed as RESTful services over HTTP without the use of gateways.

Toward providing a semantic structure for classifying things, Kortuem et al. [21] address issues on modeling and representing smart objects in order to strike a balance between the objects and the infrastructure. This effort focuses on the design of industrial hardware. Instruments and tools in industrial scenarios are augmented with sensors, wireless communication capabilities, and display devices, to render them as smart. These tools are classified as activity, policy, or process-aware objects, based on their awareness, representation, and interaction. These types represent combinations of three dimensions with the aim to highlight the interdependence between design decisions and explore how these objects can cooperate to form IoT. However, this work is constrained to particular industrial devices and does not consider the vast majority of objects that could potentially be used to provide useful information. For example, objects that do not have sensing capabilities would not be classified as smart objects. In contrast, we propose a more comprehensive classification model where objects are abstracted into the Web based on factors such as their capabilities and location. Beigl et al. [22] define smart physical things as things augmented with computing and communication capabilities, which can be accessed by computer applications. Similarly, Friedemann [23] envisions smart things to be able to wirelessly communicate with people and other smart things, with the ability to perceive the presence of surrounding objects. Today, these definitions do not formally encompass all things that could be on the Web, for example, an RFID tagged chair or a personal digital assistant (PDA) both are accessible via the Internet. There is no significant work done so far to classify things based on their capabilities to verify whether a thing is capable of contributing to Web applications. Such a classification would facilitate the realization of a system to integrate things into the Web and also enable the systematic deployment of things into WoT on a large scale either as Web resources providing information or as Web services connecting to applications. Our proposed classification provides a semantic structure and novel taxonomy to clearly define a thing's capability to participate on the Web and also indicate the necessary capabilities to elevate real-world things as potential candidates for the WoT.

3 Ambient parking lot

As described earlier, AS is a virtual space represented as a mashup of one or more real-world things providing services that access or alters the state of the physical space. Here, we present the necessary representation of real-world things on the WoT and its application in creating a system for realizing an ambient parking lot.

3.1 A semantic representation of things on the web

To represent things in a parking spot as virtual things, it is necessary to understand the capabilities of candidate things, and query their representations, when required. Our proposed taxonomy of things uses an ontology based on OWL and SPARQL [9, 10], to facilitate the process by which a thing is termed smart. The ontology recognizes the four capability dimensions of real-world things for their participation in an AS to be Identity (ID), Processing, Communication, and Storage, referred to as the IPCS capability set. This capability-based classification of things lays a foundation to integrate different types of things into the WoT. We extend our classification to include the semantic structure for things that are to be represented on the Web and propose that things need to be Web Smart to participate on the WoT. Here, we introduce the Web Object Metadata (WOM) that describes the semantic information related to the capabilities of Web Smart things.

A Web Smart thing inherits the dimensions of smart thing, i.e., using an object-oriented connotation "a Web Smart thing is a Smart thing". Our ontology facilitates the process by which the IPCS capabilities of Web Smart things are satisfied by:

URL: A thing is uniquely identified with a URL. RESTful adaptations of the URL provide necessary semantics to access the state and functions of things.

Hence, a thing on the Web has a number of URLs identifying them with a unique namespace.

Web Server: A thing processes HTTP requests through a Web server either augmented directly onboard or connected externally. These requests are processed, and a respective representation is returned or converted to operations on the thing. Hence, a thing has a number of processes that enables its virtual access and control.

Web Services: Communication with a thing is defined by RESTful APIs with GET and POST methods. This enables the state and functions of things to be communicated using standard Web interfaces. Hence, a thing has a number of communications channels directly accessing its representation and state.

Storage: A thing must have capabilities to cache Web resources and status information. The storage could be onboard or remote such as a data center, or a private cloud. Hence, a thing has a number of storage options.

The role of the proposed semantic structure is to provide a unique vocabulary and description logic based on modeling things for rudimentary reasoning. The ontology consists of modules for the shared architectural knowledge layers and services of things. The WOM shown in Fig. 2 is a collection of different ontologies defining concepts such as capability, location, and friends of a thing to describe candidate WoT members. The representation is not exhaustive, but is indicative of the representation that is required for the application context that we consider. The WOM is extendable with specification of other relevant ontologies.

- *Friend of a friend* (http://xmlns.com/foaf/spec/) ontology is used to connect to other Web Smart things (WOM) and people (for example, the URL of a person on a social networking site) as shown in Fig. 2.
- *WOM-annotations* ontology provides rich semantic content to capture a thing's history, user experiences, and feedback [24]. For the annotations, we use Meaning of a Tag (MOAT) to represent tag details [25].
- *WOM-location* ontology abstracts a record of how a thing is traced from the virtual space to its physical whereabouts. Geospatial ontologies http://www.w3.org/2005/Incubator/geo/XGR-geo-ont-20071023/ provide sufficient location and geographical information for locating WoT resources. Also, relative location using indoor positioning provides possibilities to track things within environments such as parking lots [26].
- The *WOM-profile* hosts a summary description of a Web Smart thing's semantic information. The WOM components like that of capability and annotations contribute to the information in the WOM-Profile.

Real-world things are inherently dynamic and proprietary in nature, i.e., during the lifespan of a thing, it adorns various context values and also adapts to various ownership. Moreover, things also have various characteristics such as manufacturer's details, price, date of manufacturing, model number, user experiences, and ownership history. The ontologies listed above capture some of the properties of Web Smart things, and the properties are accessed from the WOM-Profile. The semantic representation of things in WOM-Profile has two sets of elements. The first set of elements is tagged with <wom:preset> which is a representation of all properties of things that do not change like capabilities and location. The second set of elements is tagged with <wom:dynamic> which is a representation of properties that may change for example, owner, price, discounts, and user experiences.

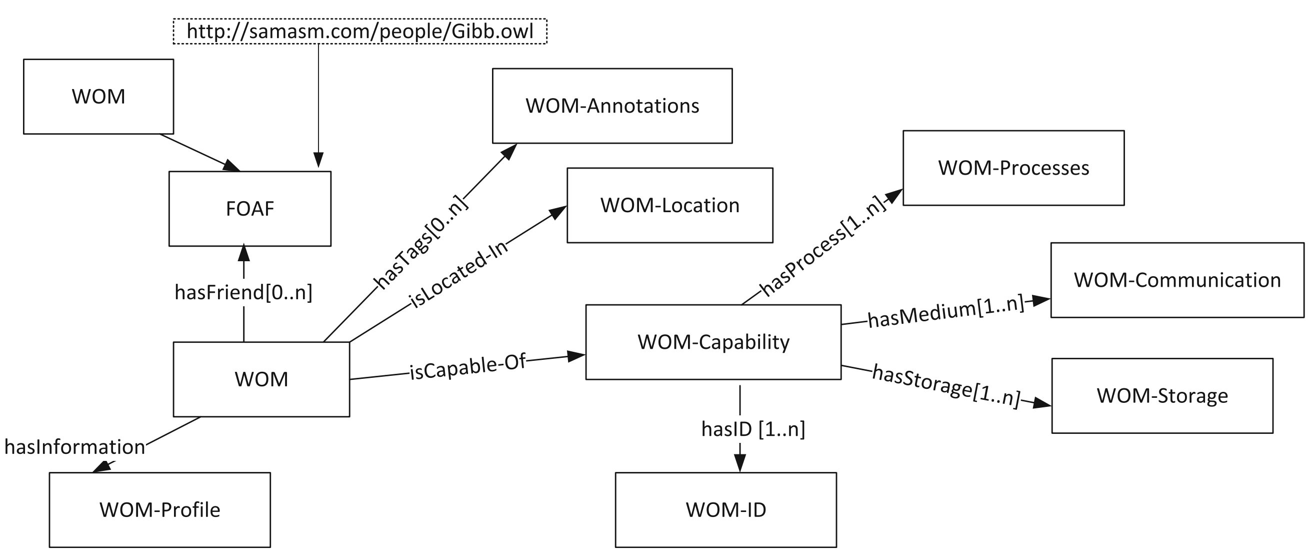


Fig. 2 Exposing information and control of a thing as Web services

Figure 3 illustrates the various namespaces and structure of a WOM-Profile, which would include the various properties described above separated into the preset and dynamic parts.

3.2 Enabling things in an ambient space

To represent things such as parking sensors and digital displays on the Web, it must be made available as a Web object that is identified by a URL. The minimal requirement for representing a thing on the Web is shown in Fig. 4 where a Web object exposes the information and control of a thing as RESTful services accessible over HTTP. The Web object handler receives and responds to the requests for services. The representation of Web objects must reflect real-time scenarios and also be retrievable and updatable. Hence, the representation formats of Web objects must ensure three criteria: (1) it must be understandable to other Web objects, (2) it must be understandable to people, and (3) it must be light-weight. The representation of states and functions of real-world objects in XML ensures easy interoperability between Web objects, and their representation in HTML enhances human perception of real-world objects. The dynamic context of real-world objects is reflected in the corresponding XML document, which is used to update the Web object HTML presentation in real time.

To represent things in a parking spot as Web objects, it is necessary to understand the capabilities of involved things and enhance their presentation, when required. Earlier, we proposed a taxonomy of things based on OWL and SPARQL [27], to facilitate the process by which a thing is termed *Smart.* This is achieved by augmenting a thing with additional capabilities. The requirement of additional capabilities recognizes four fundamental dimensions of candidate elements, to be Identity, Processing, Communication, and Storage, referred to as the IPCS set. With these additional capabilities, a parking spot becomes a smart parking spot (SPS). The integration of many SPS creates an ambient parking lot. We discuss in the next section the various components that realize an ambient parking lot.

3.3 System architecture

Parking lots are organized spaces for retaining vehicles that are not moving for a period of time. An ambient parking lot system creates a mashup of SPS operations to create a virtual space on the Web where motorists and things such as smartphones, parking sensors, and vehicles interoperate to provide secure and timely parking services in the real world. The multilayer architecture of the ambient parking lot system is illustrated in Fig. 5.

The *Smart Parking Client* (SPC) layer enables user interactions and feedback from the ambient parking lot applications. The SPC is essentially a mobile application that resides on any Web-enabled device like a smartphone that is used for reserving a parking spot. The SPC also verifies the user within the parking lot when it is coupled with a reserved spot. Once inside the parking lot, the SPC becomes part of the parking lot network infrastructure and seamlessly communicates with other entities in the parking lot.

The *Smart Parking Spot* (SPS) layer represents things such as sensors, lights, and displays within a parking spot. Each SPS is wirelessly integrated into the network within the parking lot using Wi-Fi. This has two benefits: the amount of wiring and cabling is reduced, and it also extends the parking span to multiple floors or to a larger area easily by scaling the number of access points. When an SPS is coupled with an SPC (e.g., smartphone of a motorist), it is in an *Occupied* state and otherwise it is either *Free* or *Reserved*. The state is reflected online and also displayed onsite. The coupling and decoupling of SPS and SPC goes through a verification process which ensures the security of the vehicle in the parking lot.

Fig. 3 Semantic structure for representing things described in a WOM-Profile

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf= ... xmlns:moat= ... xmlns:foaf= ... xmlns:wom= ...
xmlns:sps= ... >
<rdf:Description rdf:about= ... >
<wom:profile>
        <wom:preset>
                <!-- Fixed details of a Parking spot -->
        </wom:preset>
        <wom:dynamic>
                <!-- Changing details of a Parking spot -->
        </wom:dynamic>
</wom:profile>
</rdf:Description>
</rdf:RDF>
```

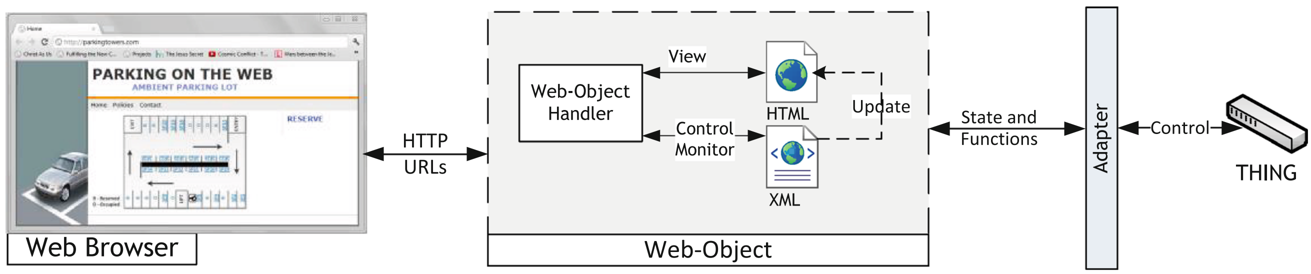
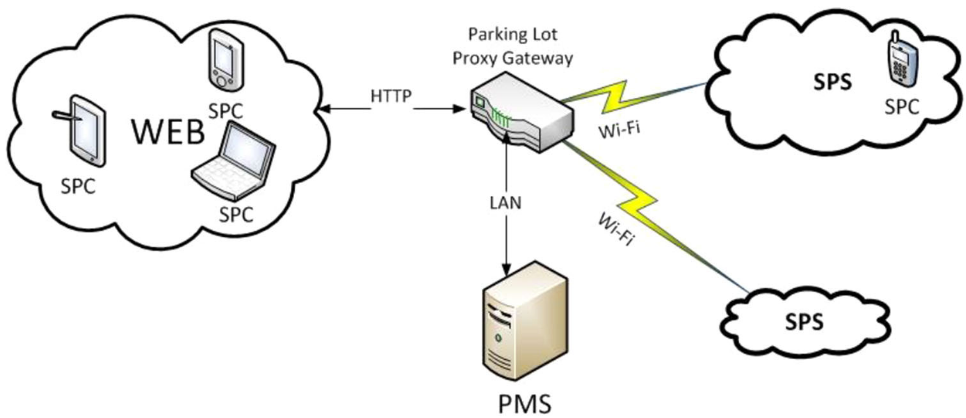


Fig. 4 Exposing information and control of a thing as Web services

Fig. 5 Layers of an ambient parking lot system



The *Parking Management System* (PMS) hosts a Web-based application that allows motorists to reserve a parking spot over the Web. The reservation facility is purchased through online payment for a period of time selected by the motorist. A database connected to the PMS indexes the URLs of the RESTful services of all SPS and corresponding things within them enabling the easy access and search for SPS.

4 System design and prototype implementation

An ambient parking lot with the *PMS*, three *SPS* in different states and an *SPC* is illustrated in Fig. 6. The *Data Center* hosts information of the parking reservations, vehicular services, and the various points of interest within the parking lot. Here, we reveal the implementation of these modules to realize our proposed system.

4.1 Parking management system

The process of parking is initiated by a motorist who requires a parking spot for a period of time. The motorist invokes the parking lot URL using a Web-enabled smartphone (SPC). The Web page hosted by PMS provides the options for *automatic* or *manual* selection of a parking spot. The automatic option selects the first available spot (e.g., SPS1) for the motorist as shown in Fig. 7. The manual option displays the parking lot allows the motorist to select a spot or click on a point of interest like the *Lift* or the *Exit*. On selecting a particular point of interest, a list of parking spots closest to the selected one is displayed to the motorist.

The ambient parking lot is represented by a URL (e.g., http://parkingtowers.com), which represents the general namespace for accessing any SPS within the lot. For example, the URL, http://parkingtowers.com/floor1/SPS1?start=1301&stop=1351, reserves SPS1 for 50 min from 13:01. HTTP responses like "*200 OK*" or "*303 See Other*" indicate the results of accessing the URLs. Once the online payment is completed, the state of the selected SPS is set to *Reserved* (R). The spots that are *Occupied* (O) cannot be selected at the given point in time since there is a possibility that the time may be extended by the motorist that has occupied the spot. Reserving an SPS in advance reduces the time to find a parking spot like Jon did in our scenario.

While a database stores the URLs of RESTful services and their relations to each SPS, the information relevant to each parking spot such as dimension, type, state, and location is stored in the SPS as XML. Hence, each SPS is a Web resource and also a data store. This creates an efficient and scalable model for distributed storage of information accessed using RESTful services.

4.2 Smart parking spot (SPS)

Smart parking spot (SPS) is an ambient space that integrates things like lights, sensors, displays, and a motorist's

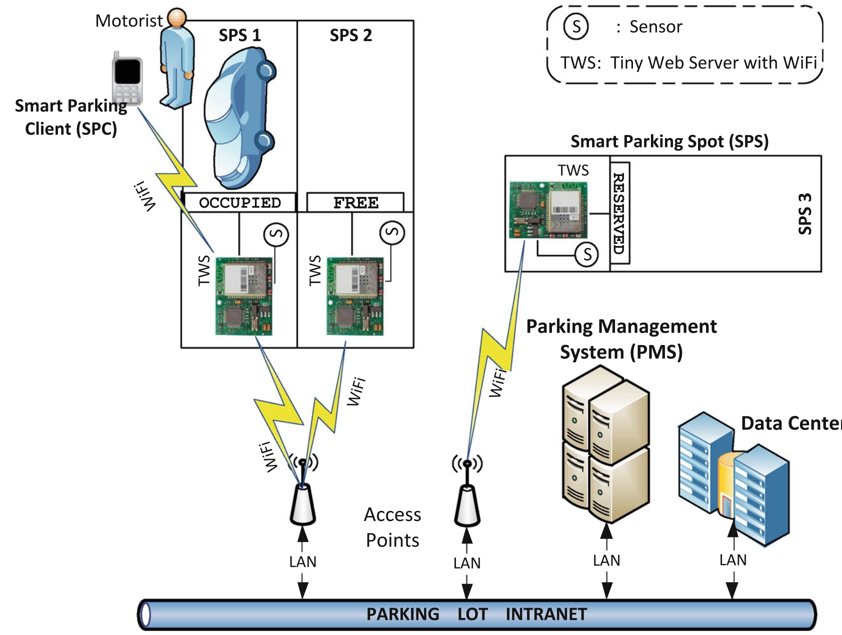


Fig. 6 Design of ambient parking lot with three SPS

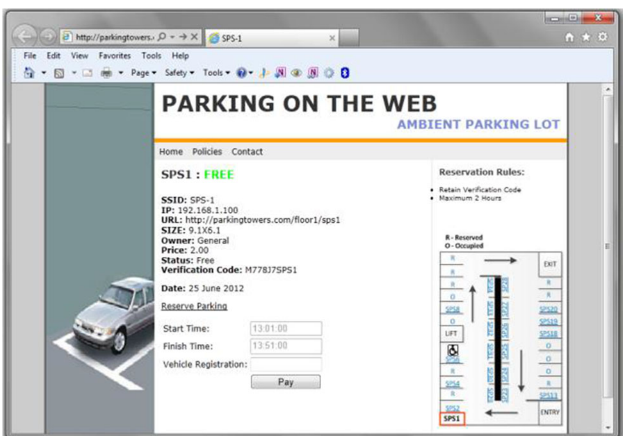


Fig. 7 Online display of the layout of parking spots and reservation page

phone into the ambient parking lot [12]. An SPS is realized by augmenting it with a Wi-Fi enabled Tiny Web Server (TWS), which communicates over TCP/IP, and has storage space. Figure 8 illustrates the SPS design and the essential hardware components. The TWS is connected to an ultrasound sensor and a digital display, to sense the presence of a vehicle in the spot and display the state information, respectively. The SPS is Wi-Fi enabled and functions as an access point providing wireless access to the parking lot Intranet. Each SPS has a unique SSID (Service Set Identification), which enables a motorist's smartphone (SPC) to uniquely identify a spot. This encapsulates each SPS into an independent hardware unit, which enhances the scalability of our system and enables flexible design of parking lots. This also makes it possible for existing parking lots to easily adapt our system.

Identifying each resource uniquely within a given namespace is necessary for building RESTful services. An SPS is uniquely identified by appending the parking lot's URL with a "/<ssid>" making it a unique resource within the namespace. The parameters of URLs are parsed to decide on corresponding operations such as, reserving a spot, verifying motorists, or extending time. An SPS hosts an XML representation of the various parameters such as location, dimension, ownership, cost, state, driving direction of the aisle, or if it is a disabled spot. The TWS provides services to query and update these parameters.

Once a user has reserved an SPS, the *state* parameter of the SPS is updated to *Reserved*. When a vehicle occupies an SPS which is in a *Free* or *Reserved* state, the sensor senses the presence of the vehicle and a timer (e.g., 60 s) is started. The SPC (e.g., a smartphone) must verify itself and be paired with the SPS before the timer expires, failing which an alarm is raised. Once an SPC and an SPS are paired, the state of the SPS is updated to *Occupied*, as shown in Fig. 8a. On departure from the SPS, the sensor again triggers a timer (e.g., 60 s), before which the SPC must identify itself. If the SPC identification fails, an alarm is raised as it could indicate that the vehicle is being retrieved by someone who did not reserve the spot. If the SPC successfully identifies itself, once the vehicle leaves, the state of the SPS is updated to *Free* and it is available for reservation.

HTTP is generally used with pull technology for accessing information, which in this case would require the PMS to continuously poll all SPS to check their *state* to see if vehicles are occupying or leaving them. This is not efficient, and hence, we use HTTP callback to indicate SPS

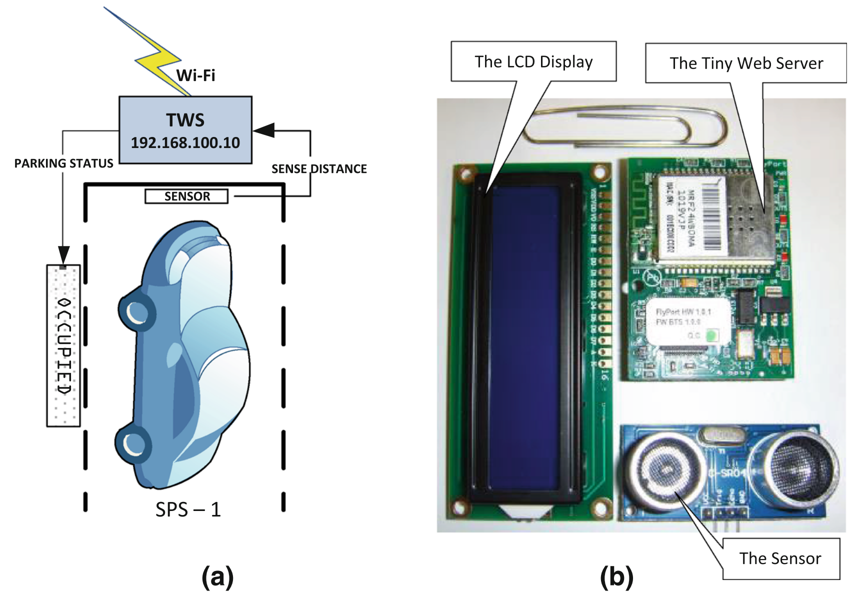


Fig. 8 **a** Design of smart parking spot (SPS). **b** Hardware components

state when an event occurs. For example, when a vehicle occupies an SPS, an HTTP POST message is sent to the PMS using a predefined URL and with relevant information in the message body. This improves the efficiency of the PMS to handle requests only when specific event occur, such as when a car occupies or leaves an SPS.

An SPS provides services, which are composed together to enable efficient parking management. An instance of service composition is illustrated in Fig. 9, where events such as booking a parking spot triggers several services, including (1) the *SpotBookingWS* where the parameters are verified, (2) the *SpotSensorWS* that triggers the sensor to verify the availability of the spot, and (3) if available, the *SpotDisplayWS* that indicates on the digital display that the spot is “Reserved.” When a car parks at the spot, the sensor triggers a parking event that invokes the *SpotVerifyWS* with a verification code from the occupant. On successful verification, the *SpotDisplayWS* is invoked to change spot state to “free” on the digital display, or if the verification fails, the *SpotAlarmWS* is invoked.

4.3 Smart parking client

The SPC is a mobile application that retains the parameters of the parking reservation and also communicates with SPS. Once the reservation is complete, the reservation parameters such as the SSID of the reserved spot, the relative location inside the parking lot, and a verification code are stored by the SPC. The application also allows the user to manually enter the reservation parameters, in case the reservation was done from another device like a PC. On arrival at the SPS, the SPC identifies and connects to the SPS based on the reservation parameters that are stored, and then, the *verification code* is exchanged. This identifies the user as the one who made the reservation. On departure, the SPC connects again to the SPS and exchanges the verification code. This ensures that only a person with the smartphone (SPC) that made the reservation can retrieve the vehicle.

4.4 Prototype implementation

The hardware components used to design SPS are shown in Fig. 8b. We used a TWS module, FlyPort [28], which is 35X48 mm in dimension with an integrated 802.11 Wi-Fi interface module and a 16-bit processor. The internal flash of 256 KB is sufficient for the intended Web application. Dynamic Web pages access inputs and output ports, which enable the manipulation of a thing from the Web. We configured the TWS with a unique SSID to serve as a

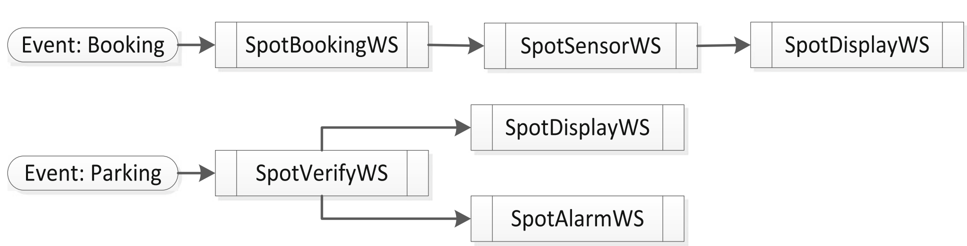


Fig. 9 Composition of parking services

Wi-Fi access point and as a part of the parking lot IP network infrastructure. The Web application parses the incoming RESTful URLs to perform corresponding operations. An *LCD display* and an *ultrasound sensor* are connected to the ports of the TWS to display the state and determine the presence or absence of a vehicle, respectively. Ultrasonic sensors such as the HC-SR04 are cheaper, simpler, and stably detect echoes from barriers from 2 to 500 cm as compared to vision sensors, laser sensors, and image sensors [29]. The TWS hosts an HTML page and an XML page to represent the parking spot. The HTML provides the online view of the parking spot, and the XML has well-defined semantic parameters.

The PMS implementation involves a standard Web application that is built using JSP, Servlets, and MySQL database. The Web application allows users to reserve a parking spot for a selected time period on a day. All reservations expire by the end of the day. The SPC is developed on the Android 2.3.1 platform using *android.net.wifi.WifiManager* methods to identify an SPS with a particular SSID. The SPC searches for a particular SSID and connects to the SPS. The verification code is exchanged, and the state of the SPS is updated using the reservation parameters. This couples the SPC and the SPS and also identifies the SPC as an integral part of the parking lot Intranet.

5 Analysis and preliminary evaluation

Here, we evaluate the extent to which the use of *Smart Parking Spots* (SPS) in the proposed ambient parking lot system reduces the time to find a free spot and preserve a sustainable environment. For evaluation purposes, we consider Ben and Jon, going to attend a meeting during peak-time traffic conditions. While Ben chooses to drive directly to the meeting, Jon decides to use his smartphone to reserve an SPS from the Web using the *automatic* option and then drives to the venue. We assume that both, use the same route, and reach the parking lot at the same time. This scenario assumes that Jon will find a free parking spot and that Ben may not find one. Also, the time spent for Jon to reserve a parking spot is typically less than the time spent by Ben to find a free parking spot. Despite the advantage that Jon has in this scenario, we wanted to measure if there was a significant amount of time saved when SPS is used. Moreover, we also wanted to use our findings to measure the amount of reduction in vehicular pollution, to see if there was any major impact.

The performance metrics of our evaluation are *cruising time*, *total time,* and *emission rates*. We define *cruising time* as the time spent in the parking lot to search for a free spot and occupy it. The *total time* includes the time to reserve an SPS, time to drive to the parking lot and cruising time. Within parking lots, vehicles assume various rates of acceleration to reach a desired spot, i.e., they slow down or accelerate while looking out for a parking spot. Considering the various rates of acceleration for different vehicles, the average *emission rates* were NO (nitrogen oxides), HC (hydrocarbons), CO (carbon monoxides), and CO_2 determined to be 1.44, 0.76, 10.51 mg/s, and 3.22 g/s, respectively [30].

Table 1 Traveling events to the parking lot for Ben and Jon

#	Ben's travel	Jon's travel
1		Online reservation of SPS
2	Drive to the parking lot for 30 min	Drive to the parking lot for 30 min
3	Cruise for a free parking spot	Cruise to the prebooked SPS
4	Occupy the parking spot	Occupy the SPS

Based on the traveling events of Ben and Jon as shown in Table 1, we are interested in the following questions for the duration of the peak-time: (1) How much *cruising time* is spent to reach a parking spot? (2) What is the *total time* taken for reaching the parking spot? (3) What is the amount of carbon emission?

The possibility of finding a free parking spot at a given point of time is dependent on two factors: (1) the number of cars arriving and occupying the free spots, and (2) the time of arrival at the parking lot, i.e., the later they arrive into the peak-time lesser would be the chance of finding a free spot. Hence, the chance to get a free spot reduces as time advances into the peak-time. We also consider maximum cruising time, beyond which the motorist leaves the parking lot without finding a free spot.

To complete an exhaustive evaluation, we would require Ben and Jon to arrive at the parking lot at various time slots incrementally for the duration of peak-time traffic (120 min). For example, what would be the results if they arrive 35 min into the peak-time or 42 or 56 min into the peak-time and so on? This is expensive and cumbersome to evaluate in real life, and hence, we simulate the scenario, where Ben and Jon arrive at the parking lot at time $\boldsymbol{t}$ which is incremented for every minute of peak-time traffic duration.

For our simulation, we consider the total number of parking spots to be $\boldsymbol{N}$ (800 spots), and the simulation is run for the duration of peak-time traffic (120 min). The time to reach the parking lot for both Ben and Jon is $\boldsymbol{T}$ minutes, which is a constant value (30 min). A random number of cars $\boldsymbol{C_t}$ arrive at the parking spot at time $\boldsymbol{t}$ following a Poisson distribution, based on which the number of available parking spots reduces. Cruising time depends on the available spots at a given time and also the maximum

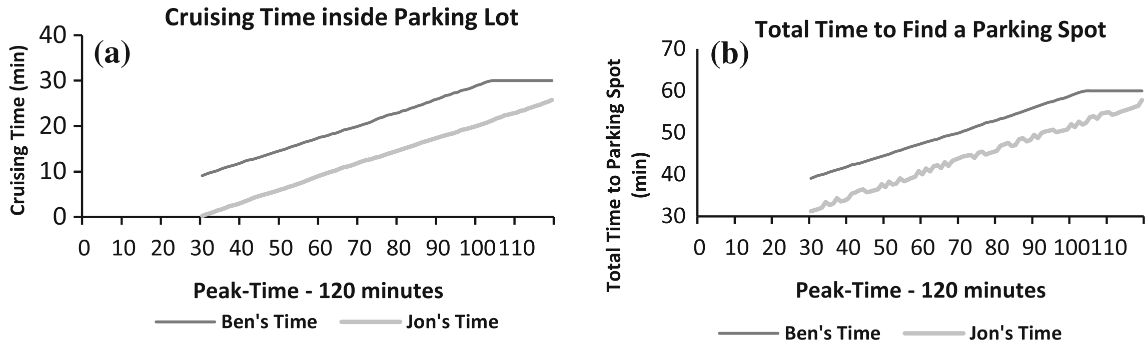


Fig. 10 Comparison of time spent in finding a parking spot, **a** cruising time of Ben and Jon. **b** Total time consumed by Ben and Jon

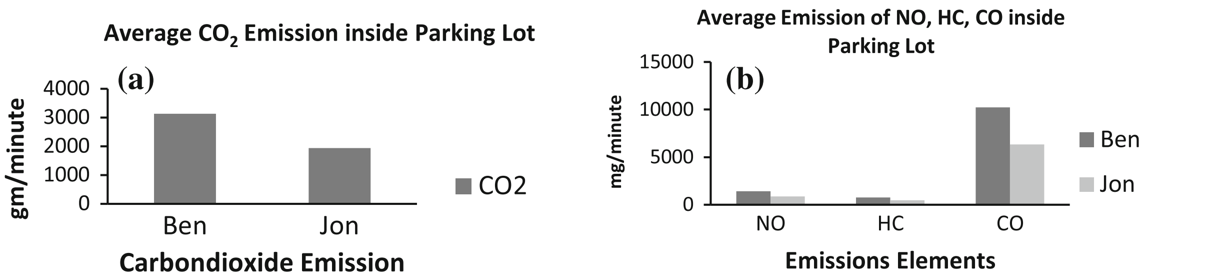


Fig. 11 The emission rates for SPS and unreserved parking spots for **a** carbon dioxide (CO_2) and **b** other pollution elements NO, HC, and CO

cruising time $\boldsymbol{X}$. The maximum cruising time is a constant value (30 min), beyond which the motorist leaves without finding a free spot.

Let $\boldsymbol{A_t = A_{t-1} - C_t}$ be the available parking spots at time $\boldsymbol{t}$, and then, Ben's cruising time to an unreserved spot $\boldsymbol{UC_t}$ and Jon's cruising time to an SPS $\boldsymbol{RC_t}$ are formulated as follows:

- UC_t at time t is $(N - A_t)/N * X$
- RC_t at time t is $(N - A_{t-T})/N * X$, where A_{t-T} is available spots at time $t - T$, because the spot was reserved T minutes earlier.

The simulation compares Ben's and Jon's time to find a parking considering they would reach the parking lot at varying degree of peak-time traffic. Graphs of the simulation results are shown in Fig. 10, which are based on average results collected after 200 iterations. In Fig. 10a, b for both Ben and Jon, the *x-axis* coordinates of the graphs indicate their arrival time at the parking lot, i.e., the number of minutes into peak-time after a 30 min drive. The *y-axis* coordinates indicate cruising time and total time in Fig. 10a, b, respectively.

With peak-time increasing, we observe that there is an increasing gradient in the time taken to find a free spot with Ben taking longer time to find a spot. As the graph illustrates, toward the end of the peak-time Ben leaves without finding a parking spot while Jon continues to occupy SPS. The results in Fig. 10a show an approximate average of 40 % reduction in cruising time when using an SPS, compared to an unreserved parking spot. Similarly, the results in Fig. 10b show the total time to reach an SPS inclusive of travel time, and the online reservation time is better than using an unreserved spot. After 200 iterations of the simulation, it was noticed that there is an approximate average of 32 % reduction in total time to reach an SPS when compared to unreserved parking spot.

The contrasts between the rates of emission for NO, HC, CO, and CO_2 are shown in Fig. 11, for Ben and Jon. The graphs indicate considerable reduction in emission components when the time to find a parking is reduced. Collectively with the use of SPS, there is an average of about 38 % reduction in emission.

6 Conclusion

Abstracting parking spots into WoT as ambient spaces creates a flexible model for parking lots. Our proposed semantic structure for representing things captures relevant information of parking spots that are *preset* and *dynamic*. The preset information provides relevant details that are predefined for a parking spot. The dynamic information provides user-related information, maintains history, and this enables various possibilities to provide user-specific services. We described in detail the architecture of an ambient parking lot, which is a mashup of many smart parking spots. Our prototype implementation and subsequent evaluation indicate substantial direct benefits to users in terms of time saved to find free parking spots. Indirectly, the users benefit from an environment where there is reduced vehicular emissions and pollution. Other, benefits for the user is the simplicity of the Web interface, the automated verification process that provides a level of security for the vehicle, and the possibility to directly associate people with occupied parking spots.

With the WoT technology which encompasses distributed data, the use of RESTful services, and wireless access, our proposed system enhances the scalability and flexibility of a parking lot network infrastructure and related information. This in effect reduces the total travel time and cruising time for motorists and also reduces the rate of pollution, creating more sustainable parking environments. Using our system, we plan to design Web applications for searching, locating, and reserving choice parking spots, to reduce overall traffic congestions and provide safe parking. We plan to study the impact and efficiency when many parking lots within a city adopt our system. We also continue to explore the options of optimizing the system by working toward measuring the performance when one TWS is used for multiple parking spots.

References

1. Bapna M (2012) How business can drive sustainable growth at Rio and beyond, Forbes, GREEN TECH. http://www.forbes.com/sites/manishbapna/2012/05/14/how-business-can-drive-sustainable-growth-at-rio-and-beyond/
2. Le Fauconnier A, Gantelet E (2006) The time looking for a parking space: strategies, associated nuisances and stakes of parking management in France. In: The proceedings of the European transport conference. Strasbourg, France
3. Shannon B, Dauvergne M (2013) Police-reported crime statistics in Canada, 2010. Statistics Canada Catalogue no. 85-002-X. Retrieved on 23 Mar 2013 from http://www.statcan.gc.ca/pub/85-002-x/2011001/article/11523-eng.htm
4. Mulligan G (2010) The internet of things: here now and coming soon. IEEE Internet Comput 14(1):35–36
5. Guinard D, Trifa V, Karnouskos S, Spiess P, Savio D (2010) Interacting with the SOA-based internet of things: discovery, query, selection, and on-demand provisioning of Web services. IEEE Trans Serv Comput 3:223–235
6. Christophe B, Boussard M, Lu M, Pastor A, Toubiana V (2011) The Web of Things vision: things as a service and interaction patterns. Bell Labs Tech J 16(1);55–61
7. Shelby Z (2010) Embedded Web services. Wirel Commun IEEE 17(6):5257
8. Ostermaier B, Kovatsch M, Santini S (2011) Connecting things to the Web using programmable low-power wifi modules. In: The proceedings of the second international workshop on Web of Things 2011, p 2
9. Mathew SS, Atif Y, Sheng QZ, Maamar Z (2010) Ambient things on the Web. J Ubiquitous Syst Pervasive Netw (JUSPN) 1(1):1–8
10. Mathew SS (2011) Managing things in an ambient space. In: The proceedings of the 9th international conference on service oriented computing (ICSOC). Paphos, Cyprus
11. Mathew SS, Atif Y, Sheng Q, Maamar Z (2013) The Web of Things—challenges and enabling technologies. In: Bessis N, Xhafa F, Varvarigou D, Hill R, Li M (eds) Internet of things and inter-cooperative computational technologies for collective intelligence, vol 460. Springer, Berlin, pp 1–23
12. Mathew SS, Atif Y, Sheng QZ, Maamar Z (2013) A new parking system based on the Web of Things. In: The proceedings of the workshop on computing and networking for internet of things (ComNet-IoT), in conjunction with international conference on distributed computing and networking (ICDCN)
13. Hui JW, Culler DE (2008) Extending IP to low-power, wireless personal area network. Intern Comput IEEE 12(4):37–45
14. Riener A, Zia K, Ferscha A, Ruiz CB, Rubio JJM (2013) Traffic flow harmonization in expressway merging. Pers Ubiquit Comput 17(3):519–532
15. Lu R, Lin X, Zhu H, Shen X (2009) SPARK: a new VANET-based smart parking scheme for large parking lots. In: The proceedings of the 28th conference on computer communications (IEEE INFOCOM). Rio de Janeiro, Brazil, pp 1413–1421
16. Priyantha NB, Kansal A, Goraczko M, Zhao F (2008) Tiny Web services: design and implementation of interoperable and evolvable sensor networks. In: The proceedings of the 6th ACM conference on embedded network sensor systems (SenSys'08). ACM, New York, NY, USA, pp 253–266
17. Chen M, Chang T (2011) A parking guidance and information system based on wireless sensor network. In: The proceedings of the IEEE international conference on information and automation (ICIA). Shenzhen, China, pp 601–605
18. Fielding RT (2000) Architectural styles and the design of network-based software architectures, PhD thesis. University of California, Irvine
19. Guinard D, Trifa V (2009) Towards the Web of Things: Web mashups for embedded devices, workshop on mashups, enterprise mashups and lightweight composition on the Web (MEM 2009), held in conjunction with the 18th international. World wide Web conference (WWW'09). Madrid, Spain
20. Wilde E (2007) Putting things to REST, TECHNICAL REPORT UCB iSchool Report 2007-015. School of Information, UC Berkeley, California, USA
21. Kortuem G, Kawsar F, Fitton D, Sundramoorthy V (2010) Smart objects as building blocks for the internet of things. Intern Comput, IEEE 14(1):44–51
22. Beigl M, Gellersen H, Schmidt A (2001) Mediacups: experience with design and use of computer-augmented everyday artefacts. Comput Netw 35(4):401–409
23. Mattern F (2003) From smart devices to smart everyday objects, In: Proceedings of smart objects conference
24. Mitrokotsa A, Sheng QZ, Maamar Z (2012) User-driven RFID applications and challenges. Pers Ubiquit Comput 16(3):223–224
25. Passant A, Laublet P (2008) Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. In: The proceedings of the WWW 2008 workshop linked data on the Web (LDOW2008). Beijing, China, p 48
26. Vera R, Ochoa SF, Aldunate RG (2011) EDIPS: an easy to deploy indoor positioning system to support loosely coupled mobile work. Pers Ubiquit Comput 15(4):365–376
27. Mathew SS, Atif Y, Sheng QZ, Maamar Z (2011) Web of Things: description, discovery and integration. In: International conference on internet of things and cyber, physical and social computing (iThings/CPSCom), pp 9–15

28. FlyPort Wi-Fi module, openPICUS. Retrieved on 2013.03.19 from http://www.openpicus.com/site/products
29. Dong-he Y, Xiang L (2007) Research of intelligent mobile robot's obstacle avoidance based on ultrasonic sensor. Comput Eng Des 28(15):3659–3661
30. Frey HC, Unal A, Rouphail NM, Colyar JD (2003) On-road measurement of vehicle tailpipe emissions using a portable instrument. J Air Waste Manag Assoc 53:992–1002