Modeling and Simulating the Web of Things from an Information Retrieval Perspective

CRISTYAN MANTA-CARO, University of Granada
JUAN M. FERNÁNDEZ-LUNA, University of Granada

Internet and Web technologies have changed our lives in ways we are not yet fully aware of. In the near future, Internet will interconnect more than fifty billion things in the real world, nodes will sense billions of features and properties of interest, and things will be represented by web-based, bi-directional services with highly dynamic content and real-time data. This is the new era of the *Internet and the Web of Things*. Since the emergence of such paradigms implies the evolution and integration of the systems with which they interact, it is essential to develop abstract models for representing and simulating the Web of Things in order to establish new approaches. This paper describes a Web of Things model based on a structured XML representation. We also present a simulator whose ultimate goal is to encapsulate the expected dynamics of the Web of Things for the future development of information retrieval (IR) systems. The simulator generates a real-time collection of XML documents containing spatio-temporal contexts and textual and sensed information of highly dynamic dimensions. The simulator is characterized by its flexibility and versatility for representing real-world scenarios and offers a unique perspective for information retrieval. In this paper, we evaluate and test the simulator in terms of its performance variables for computing resource consumption and present our experimentation with the simulator on three real scenarios by considering the generation variables for the IR document collection.

CCS Concepts: •**Information systems** → **Document collection models;** *Spatio-temporal systems;* •**Computing methodologies** → **Discrete-event simulation;**

General Terms: Modeling, Simulation, Perspective

Additional Key Words and Phrases: Discrete-event Systems, Information Retrieval, Simulation, Web of Things

ACM Reference Format:
Cristyan Manta-Caro, Juan M. Fernández-Luna, 2017. Modeling and Simulating the Web of Things from an Information Retrieval Perspective *ACM Trans. Web* 0, 0, Article 0 (2017), 28 pages.
DOI: 0000000.0000000

1. INTRODUCTION

At this moment in time, the number of things connected to Internet exceeds the current world population. If we bear in mind that only one percent of real-world objects are Internet-connected and that the number of connected devices is expected to reach around fifty billion by 2020 [Hodges et al. 2013], the possibilities are unimaginable. This new paradigm is referred to as the *Internet of Things* (IoT) and describes the technologies and research disciplines enabling the Internet to adopt intelligence and to venture into the real world of interconnected physical objects [Feki et al. 2013]. Furthermore, if we enable advanced Web access through virtual elements that are abstract representations of real-world things, we can create smart, intelligent spaces which appear as the new paradigm: the *Web of Things* (WoT) [Christophe et al. 2011]. The WoT

This work has been funded by the Spanish Ministerio de Economia y Competitividad under Project TIN2013-42741-P.
Authors' address: C. Manta-Caro and J. M. Fernández-Luna Department of Computer Science and Artificial Intelligence, University of Granada, Granada, 18071, Spain.

DOI: 0000000.0000000

can refer not only to those parts of the web comprising special web technologies and application systems connected to the real world via sensors and actuators [Diaconescu and Wagner 2014] but also to the combination or evolution of paradigms such as the Semantic Web and the Ubiquitous Web, where Web 3.0 is for things [He et al. 2012] "*"The Semantic Web enables human knowledge to be machine-readable and the Ubiquitous Web allows Web services to serve any thing, forming a bridge between the virtual world and the real world*"".

The term Web of Things was first used in 2007 by David Raggett at the UWE's Web Developers Conference in Bristol (UK). The novel paradigms of the IoT and WoT are closely related and both impose new requirements and constraints, bringing new perspectives and challenges to current systems with which they interact. Some research has focused on architecting the WoT [Uckelmann et al. 2011], [Guinard 2011] and the applications for integrating the real world in the Web. The WoT architecture has been proposed to use RESTful principles as a way to follow the pillars of scalability and modularity of the traditional Web [Guinard et al. 2010]. Other research, meanwhile, has highlighted information retrieval (IR) systems in the form of search engines with different levels of sophistication [Manta-Caro and Fernández-Luna 2014b]. Until now, however, no simulation model, research or development has focused on a scalable WoT simulator. The expected dynamism of the WoT involves billions of real-world things which will constantly produce a vast amount of information and the exponential growth in the number of attached sensors will in turn obtain a vast amount of frequently changing data due to the nature of the IoT. The design, development, implementation and adaptation of new, real-time search engines that allow things to be found and provide information about the variables of these things (as well as the features and services they offer) represent a crucial research topic [Christophe et al. 2011] given the new WoT environment. Within this IR context [Croft et al. 2009], the new WoT paradigms introduce highly dynamic factors that should be carefully considered.

This strong dynamism has not been well explored or evaluated in conventional IR systems. The WoT can be characterized by a colossal number of documents in a highly dynamic collection. In order to address the new requirements and constraints imposed by the WoT on IR systems, a test collection is required for evaluation purposes. At the moment, however, there is no evidence that such a collection exists. Therefore, the main motivation of our research work is to create a real-time IR test collection that mimics the complex behavior of the WoT to further evaluate IR systems. Accordingly, the best way to generate this collection is to simulate IoT/WoT dynamics. The IoT/WoT will completely change the way we interact with the world and our simulator can be used to show real scenarios which lead to this future ecosystem with a flexible and versatile approach by taking into account information retrieval foundations and perspectives. We shall use these IR test collections to propose new research lines in the IR field by first evaluating different XML indexing strategies, real-time techniques and the most suitable data structures.

In order to describe the WoT simulator in detail, this paper is structured as follows: Section 1 presents an introduction to the WoT and discusses its dynamism and impact on current IR systems; Section 2 describes our WoT model from an IR perspective, which is the basis for the subsequent sections; Section 3 illustrates our simulator for the WoT approach, examines similar related work in the context of sensor networks, wireless sensor networks (WSN) and sensor web systems as the foundation for the design, development and implementation of our approach, and compares the main simulation, emulation and prototype environments used in IR systems for the IoT and WoT; Section 4 details the discrete-event system mechanism behind the simulation engine; Section 5 presents three experimentation scenarios using the WoT Model and

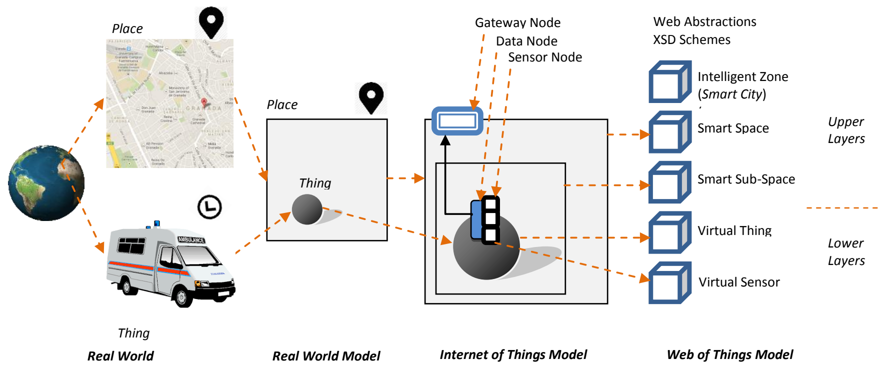


Fig. 1. Proposed WoT multi-level model based on [Manta-Caro and Fernández-Luna 2014b]

Simulator; Section 6 describes the tests and results of the conducted experiments; and finally, Section 7 outlines our conclusions and future lines of work.

2. WEB OF THINGS MODEL

Our vision of the WoT begins with an abstract, real-world representation. In [Manta-Caro and Fernández-Luna 2014b], we proposed a WoT model and presented a preliminary conceptualization of the abstract model and WoT representation which was semantically enriched using XML/URI elements and compared with other related lines of work. We refer to this as the basic model and it is aware of the spatial context and mainly focuses on hierarchy to present a balanced model between lower and upper layers represented at XSD schema levels in Figure 1.

The basic model arises from a real-world model with two essential elements:

— Things (either tangible or intangible)
— Places (the spaces where the things are or which they are related to)

Our IoT model is represented by the underlying infrastructure linking these two real-world elements with the Web. The IoT mainly consists of a sensor layer to obtain real-time information about the properties of things or spaces in the real world, and a data layer where information from the real things is processed and stored. This data layer provides Internet connectivity to data nodes, or gateway nodes, which would perform protocol conversion functions. Our WoT model comprises five levels of abstraction involving the entire universe of elements that we consider relevant. Compared to other proposals, our model achieves completeness and balance by considering the spatial context on three levels, together with formal models of a virtual sensor and a virtual thing.

The abstract model is supplemented with a structured WoT XML representation. In our model, the five multi-level abstract representations of the real world are:

— Intelligent zones (IZ)
— Smart spaces (SS)
— Smart subspaces (SSS)
— Virtual things (VT)
— Virtual sensors (VS)

The model encapsulates the spatio-temporal context, takes elements and is directed toward considering the efforts of organizations such as the OpenGIS, W3C, ISO for standardizing technologies that point to real-world interconnection.

2.1. Description of the Multi-level WoT Extended Model

The main and final motivation behind WoT modeling is to establish the information structure that follows the collected data from smart things to provide advanced web services as a basis for developing future IR systems or applications in the form of search engines, which allow real-world things to be found through their virtual counterparts. Our work proposes the representation of the WoT based on the basic model [Manta-Caro and Fernández-Luna 2014b] using its components listed above but extending and adding to it with a hierarchical relation of membership between model elements with explicit XML tags. In order to provide the representation with flexibility, the basic model contains three spatial components: the possibility of a space comprising subspaces and the ability to federate smart spaces into so-called intelligent zones. The proposed representation simplifies the lower layers, allowing users to focus on application extension and WoT composition layers.

The IoT sensor layer is associated in the WoT model with an abstraction called the *virtual sensor* which is designed to enable Web representation, to perform feature composition and to provide high-level information fusion. We propose that the model be balanced by considering the spatial context and by creating a web abstraction of the sensor level. In his article [Mayer 2012], Mayer presents an example of an unbalanced model in the spatial search for smart things, where the model does not include a sensor level although it does possess a well-defined, spatial-oriented model. In our basic model, each sensor as an abstraction model element has a URI that identifies its dynamic XML document, containing its description, properties and data. Tangible and intangible things in the real world are modeled by the abstract Web component called the *virtual thing*. Not only do virtual things consolidate the information available at the virtual sensors linked to them but they also contain features, functionalities or services that can be made available in the virtual or physical world through their web abstractions. In the same way, each model element (in this case the virtual thing) is uniquely identified by a URI, which is related to a dynamic XML document containing real-time information.

Like their real counterparts, virtual things are confined in smart spaces that correspond to abstractions of real-world places and sites that have been endowed with intelligence. Thanks to their virtual location sensors, virtual things are able to change not only their state but also the smart space where they are. In this way, the movement of virtual things to other smart spaces will update the hyperlinks between documents and the membership of virtual things to a place. Real-world environments, sites and places are modeled using an abstract component called the *smart space*, which condenses the features of the environment where they are located and which contains one or more virtual things. Another model [Romer 2010] proposes a state-based search of entities with a stochastic sensor model and various space considerations. It limits the possible search results to the level of entities (in our approach called things), where sensor information is not considered relevant for the search, and the constrained search for spaces is not considered. In our WoT basic model, the possible results can be in any level: data, sensors, things, and/or spaces. We propose that the basic model be extended to enable a wider range so that we can search for smart spaces, things or types of sensors or data/states that meet certain restrictions and/or contain certain components by adjusting and optimizing all the elements and representations at the XML level and by considering the discrete-event paradigm. For example, let us model an ambulance as a smart virtual thing somewhere in a smart city (an intelligent zone in our model). As Figure 1 illustrates, the smart ambulance is connected to the Internet by means of an IoT infrastructure. Real-time information such as location, availability and facilities can be collected with the sensor layer and published to the

Web abstraction for further access. Since the model includes the possibility of a smart space comprising one or more smart subspaces, we could model the smart city as a set of smart neighborhoods and include smart hospitals. A real-time search engine for the WoT can respond to the query to find the closest, free operating room to the ambulance's current location. In order to meet this type of criterion, the extended model considers special XML tags grouped according to purpose: properties, membership and states.

In similar work, the search for entities does not consider the sensor level nor the spatial context [Pfisterer 2011], [Christophe et al. 2011]. However [Guinard 2011] and [Guinard et al. 2010] present a balanced model that includes both levels and this is similar to our approach. These articles explore the semantic enrichment by means of ontologies. Our proposed, extended Web representation model is presented in the following section. Our main motivation is to feed an IR system with a collection of dynamic documents originating from the abstract extended model in order to retrieve relevant documents given a query. Although the representation of elements can be as simple as using the associated metadata of the physical objects, or as complex as using ontologies and semantic profiles (like our WoT philosophy of reusing web technologies), our proposal is firstly to employ XML as a simple, structured, semantically-enriched vehicle containing the information about the items that can be retrieved in the model, and secondly, to consider progress in representing XML-based sensor networks. In line with our WoT vision, new Web technologies have been proposed that can form a part of this paradigm such as the innovative Web protocols based on binary representation for addressing the resource constraint on developing and deploying WoT applications in microdevices and nanodevices [Kyusakov et al. 2014]. It also presents a novel method for generating efficient XML interchange (EXI) grammars based on XML schema definitions. One major result was described by Kyusakov [Kyusakov et al. 2014] who evaluated the use of binary protocols for embedded Web programming and proposes a framework comprising a highly efficient EXI processor, with a Constrained Application Protocol (CoAP)/EXI/XHTML Web page engine, which reinforces our decision to use XSD schemas, XML documents and definitions to model the WoT.

2.2. Structured WoT Extended Representation

Each of the proposed components corresponds to an XSD schema which together with the data and refined XSD tags define a dynamic WoT XML document collection which results in an IR test collection as illustrated in Figure 2.

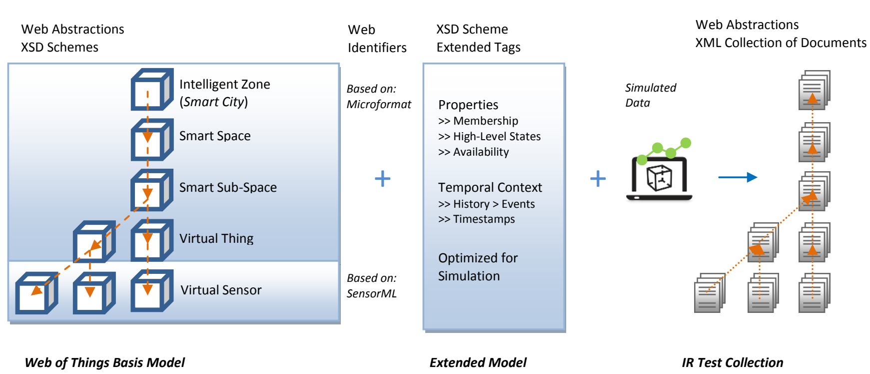


Fig. 2. WoT basic model, WoT extended model and dynamic IR test collection for XML documents

The proposed virtual sensor XSD schema comprises a group of general information tags containing keyword, identification and classification elements. A group of references with contact elements can link and specify role-based documentation elements in a similar way to the XML implementation of the Observations and Measurements (O&M) model on SensorML (see Figure 3).

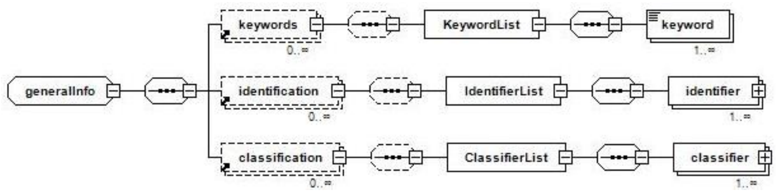


Fig. 3. General information in the XSD virtual sensor schema

The next section of the XML schema contains the group of properties which enables characterization of the virtual sensor (see Figure 4) and description of the sensor capabilities, high-level state and a membership element for associating the virtual sensor with the virtual thing sensed.

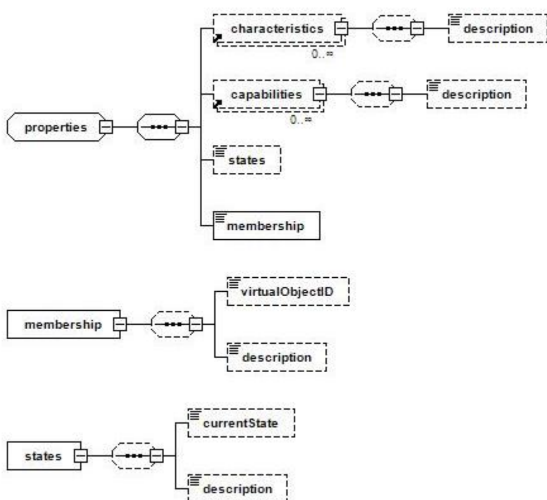


Fig. 4. Properties and state in the XSD virtual sensor schema

There are also fields for storing history and events as shown in Figure 5, and Observing elements of the virtual sensor XML schema on the basis of the OpenGIS O&M schema. This group contains XML elements for storing sampling time, result time, feature of interest and the result as illustrated in Figure 6.

Since the virtual thing component should capture information about the observed physical phenomenon, we propose the use of SWE scheme elements. The proposed scheme follows the structure of the XML representation of the virtual sensor using groups and elements: general information, references, properties and history. The

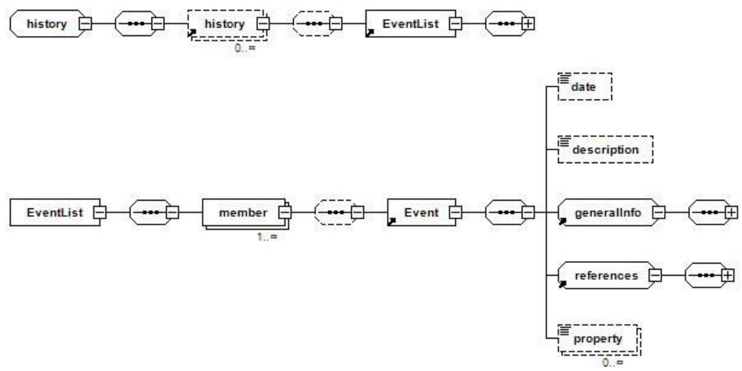


Fig. 5. Events and history in the XSD virtual sensor schema

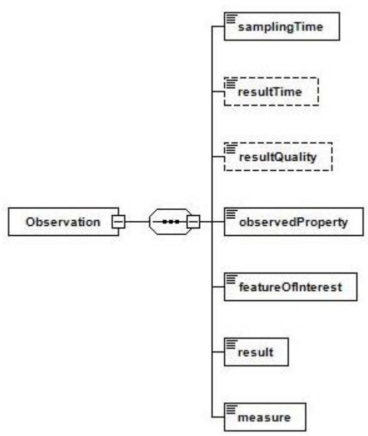


Fig. 6. Observation field in the XSD virtual sensor schema

property group has been enriched with an object availability element and the membership field is associated with a higher, hierarchical level, indicating the smart space containing the virtual things. An element location and a list of attached sensors are added. The XML schema of the smart space component is built based on the microdata Place schema in (http://schema.org/Place). This contains similar elements to the previous components with a list of virtual things and subspaces. The intelligent zone follows a similar schema with Web domain identification and the list of smart spaces comprising it.

3. SIMULATOR SYSTEM ARCHITECTURE AND RELATED WORK

The proposed WoT simulation architecture considers the multi-layer approach described in the previous section by considering the WoT model upper layers to encapsulate the spacial context, and the lower layers to encapsulate the temporal context and WoT dynamics. A preliminary high level design (HLD) for the WoT architecture and model is described in [Manta-Caro and Fernández-Luna 2014a]. The following subsection presents a more detailed description of the subsystem architecture and a comparison with other work with certain additional considerations in the form of an

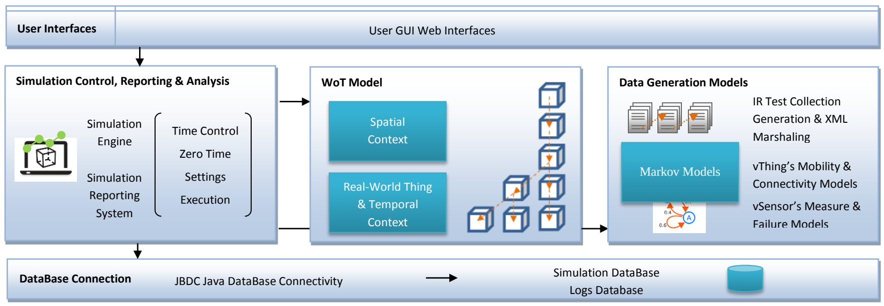


Fig. 7. Simulator conceptual architecture and modular decomposition

evaluation framework taxonomy. The low level design (LLD) in terms of the discrete-event mechanism behind the simulator is described in the following section.

The simulator architecture has been designed based on three core modular components which are illustrated in Figure 7 and which together comprise the functional process logic of the simulator. The three-tier architecture contains a persistence layer and a presentation layer. The core modules in the functional process logic are:

— Web of Things simulation model subsystem
— Simulation data generation subsystem
— Simulation control and simulation report subsystem

3.1. Simulation Entities in the Model Sub-system

The simulator model subsystem is responsible for creating the abstractions of the simulation entities. It is mainly based on the WoT extended model. The simulation entities for the spatial context come from the three upper layers which are built based on XSD schemes of intelligent zone, smart space and smart subspace. These simulation entities are similar in structure and have similar spatial attributes. The intelligent zone entity is identified by ID and a Web domain, in conjunction with classification, description and characteristic information: location name, elevation, latitude, longitude, etc. In the case of smart and subsmart spaces, membership information tags identify the intelligent zone, in particular, and the tree structure, in general, of these spaces.

The two lower layers encapsulate the temporal context and dynamics in terms of history, events and timestamps within XSD schemes, corresponding to the core simulation entities of virtual things and virtual sensors. The virtual thing entity contains identification, description and categorization attributes. With the purpose of following the WoT paradigm, virtual things include property labels and information about their features. The construction of its XSD scheme is a simplistic perspective and was inspired by the microdata (http://schema.org/Thing). The temporal context is encapsulated in the so-called events attribute of the entity, where each state change is logged with the corresponding timestamp. The virtual thing can be in one of several states exhibited in its behavior model following the discrete-event rules described in the following section. The virtual sensor simulation entity uses the events attribute in the same way as the virtual thing does (see Figure 5). The virtual sensor is associated with a measurement model, which follows a probability density function selected by the user It is also associated with a failure model and the mechanisms for this are described in the following section. The motivation behind the construction of the simulator model subsystem is to encapsulate the main factors that produce high WoT dynamism.

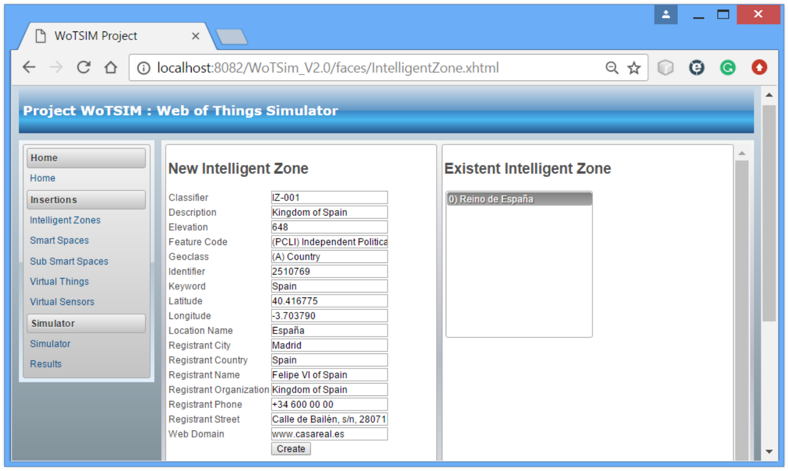


Fig. 8. Simulator web interface

3.2. Data Generation Subsystem and Behavior Models

The data generation subsystem is responsible for logically building the real-time IR test collection of XML documents containing data for the WoT entities (see Figure 7) and contains methods for performing the XML marshaling. The data generation subsystems, meanwhile, use the behavior models of virtual things and virtual sensors to produce spatio-temporal data. For example, the mobility model provides the current location of a virtual thing and its trajectory, and the connectivity model provides data about its current state. In the case of the virtual sensor, the measurement model provides stochastic data following the probability density function of the feature of interests, and the failure model provides the state of operation of this entity.

3.3. Control and Reports Sub-system

The simulator provides a Web-based GUI interface for a user to build a WoT simulation and this consists of the following steps as illustrated in Figure 8 and Figure 9:

- Create simulation entities: intelligent zones, smart spaces, smart subspaces, virtual things and sensor things.
- Set main parameters: spatial context and temporal context.
- Configure simulated observations in terms of the probability density function (PDF) of the variables of interest (see Figure 9).
- Configure the simulation engine system.
  - Set the stop simulation parameters.
  - Set the stochastic generation models: virtual things and sensor things.
- Run the simulation.

The source code can easily be modified in the simulator to customize data availability and the virtual thing mobility model and it also possible to modify the virtual sensor failure model. The focus of the simulation is to represent the WoT dynamic behavior in terms of information generation, its distributed architecture and the information availability of the Web XML representations of real-world objects and their associated sensors. At zero simulation time, multiple XML documents are created based on the simulation entities entered by the user, following a tree-structured directory.

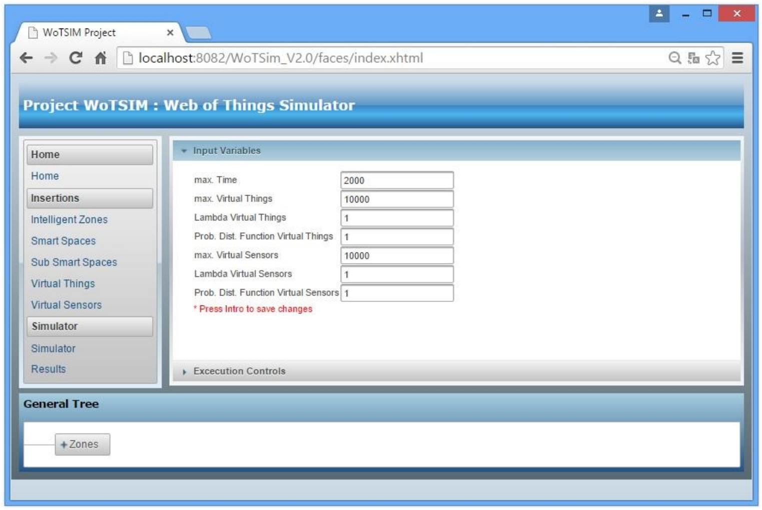


Fig. 9. Simulator environmental variables web interface

The simulation control subsystems are responsible for the simulation engine itself, which in turn manages the discrete-event mechanism, sets the zero simulation time and controls the the passage of time by events. Finally, it determines the end of the simulation given the stopping event. The control subsytem can be seen as the implementation of the discrete-event mechanism described in the following section. Additionally, it gathers and logs information for some of the fundamental variables and statistics.

3.4. Related Work and Evaluation Environments of the Web of Things

The WoT simulator is built using a discrete-event approach and considering the contributions and related work from the following dimensions:

— *Evaluation centricity*: IoT-centric, Web of Objects-centric, WoT-centric
— *Evaluation mechanism*: simulation, emulation, prototyping, hybrid
— *Evaluation scalability*: small, medium, large, super-large
— *Evaluation awareness*: spatial, temporal, spatio-temporal

These dimensions are used to perform a comparison framework of the related approaches. In [Manta-Caro and Fernández-Luna 2014a] we presented a survey of the simulation, emulation and prototype environments for the WoT, which we refer to and group as evaluation environments in this article. We also highlight the main contributions and present a taxonomy of these evaluation frameworks. From the centricity perspective, we identify three types of approaches that we denote as IoT-centric, Web of Object-centric, WoT-centric (see Figure 10).

Firstly, we distinguish within the IoT-centric category the general purpose network simulators (i.e. ns-2 and ns-3) that provide strong support for the design and implementation of new protocols and are ready-to-go. It is possible to add special extensions such as, for example, wireless sensor networks (WSN) as described in [Geyik 2013], [Samaras 2013] or [Cheng et al. 2011] for e-textile nets including behavior models or [Song 2012] for new radio models and LTE/WiFi coexistence analysis in simulation smart home scenarios. However, the existence of a strong module inter-dependence hampers the addition of new features or functionality. Secondly, although the specific purpose simulators provide greater control and versatility and can constitute a sound basis for building a WoT simulator, they focus on physical, data and network layers, and therefore fail to address the upper application layers. For example, [Jeong et al. 2015] presents a specific multi-WSN simulator in order to efficiently sense the IoT and this is achieved by collecting sensor log data to compile node information and predict the lifetime in terms of battery consumption. In the performance evaluation, the log file comprises information from 100 sensors gathered during a 10-day period. It is worth mentioning that a maximum of 1024 sensor nodes simulation can be conducted using 1024 threads. There is no consideration of Web technologies to enable a WoT paradigm and the focus of the contribution is on the sensor layer providing analysis functions to examine IoT infrastructure problems.

The IoT massive proposals then represent a group and [Looga 2012] found that simulation and emulation of IoT solutions is mostly suitable for small- and medium-scale testing but not for large-scale testing with millions of nodes running concurrently and so the large-scale IoT emulator called MAMMotH was proposed. The target of the emulation platform is to run up to 20 million nodes on a large cluster. A similar proposal is VisibleSim, a discrete-event core simulator with a discrete time functionality. It is built to simulate intelligent communicating objects placed in a real 3D world and to handle various platforms such as Blinky Blocks. Simulations can scale up in numbers and can accurately and smoothly simulate two million nodes on a simple laptop [Dhoutaut 2013]. The proposal focuses on distributed sensing, and control environments even when there are no associated Web technologies.

We then distinguish the Web of Objects-centric category. Firstly, in the Web of Sensors, various papers focus on describing the approach and estimating data volumes to perform resource analysis [Mekni and Graniero 2010] or building prediction models for the weather, for example [Seablom 2008] and [Talabac 2010]. The Sensor Web paradigm in terms of modeling and data assimilation has evolved remarkably since its first conception in 2005, and it can be a useful basis for a WoT simulator but adding a vision focusing on real-world things rather than physical phenomena. Although all of the approaches focus on the sensor level, some for example to improve positioning accuracy [Ju-Min 2013], or applying a high-level sensor simulator and sensor web standards to an industrial environment [Gimenez 2013], generating realistic and real-time data from mobile/fixed sensors that can be accessed via HTTP, none has a real-world or virtual thing model. In another direction, we can identify various lines of research which focus on devices such as [Han 2014] which presents a simulation toolkit with app prototyping capabilities based on the device profile for web services (DPWS) and which considers spaces, devices, operations and events to be core components. This work can be considered as a Web of Objects initiative since there is no real-world or virtual thing model and it focuses on the device infrastructure. As a major contribution, it represents the dynamics of the IoT/WoO with operations and events. It was used to support the development of an incident management prototype in order to test the collaboration between objects from a contextual perspective.

Next, we can identify the WoT-centric category. Firstly, there is a group of approaches along the lines of semantic enrichment by means of ontologies or other strategies. For example, [Diaconescu and Wagner 2014] proposes a general framework for simulating WoT systems based on an ontology approach. Furthermore, it introduces actuators as part of the WoT paradigm, adding the possibility of controlling the real world. In order to enrich the semantics of the model, it uses a unified foundational ontology built by the authors which is based on the robotics and automation IEEE SUMO approach. The test case presented uses one space, one single data node comprising a Samsung Galaxy S with an IOIO-OTG board, and three types of sensors for temperature, light

Table I. Comparison of simulation, emulation and prototype environments

Authors	Name	Centricity	Eval. Type
[Manta-Fernandez2015]	DE-WoTSIM	WoT	SIM
[Han 2014]	DPWSim	WoT	SIM
[Corredor 2014]	WoTOP	WoT	SIM
[Truong 2012]	*Truong-Romer*	WoT	SIM
[Diaconescu and Wagner 2014]	WoTCO	WoT	SIM
[Mayer 2012]	*Mayer-Trifa*	WoT	PROTO
[Christophe et al. 2011]	*Christophe-Verdot*	WoT	PROTO
[Pfisterer 2011]	SPITFIRE	WoT	PROTO
[Thebault et al. 2013]	EnvB	WoT	PROTO
[Garcia-Macias 2011]	UbiVisor	WoT	PROTO
[Michel et al. 2012]	Gander	WoT	PROTO / SIM
[Elahi 2009]	*Elahi-Kellerer*	WoT	PROTO / SIM
[Romer 2010]	Dyser	WoO	PROTO / SIM
[Looga 2012]	MAMMoTH	IoT	EMU
[Jeong et al. 2015]	WSL3	IoT	SIM
[Song 2012]	*Song-Han*	IoT	SIM
[Dhoutaut 2013]	VisibleSIM	IoT	SIM
[Jin 2011]	ISE	IoT	PROTO
[Ding 2012]	IoT-SVK	IoT	PROTO
[Albakour 2012]	SMART	IoT	PROTO

intensity and custom soil moisture. In the same group, [Mayer et al. 2014] proposes the usage of abstract sensing and actuation primitives, expanding the WoT paradigm to control real-world smart things. Mayer also introduces atomic interactive device components and captures the semantics of these interactions. In a similar way to this last approach, [Christophe et al. 2011] presents a search process for the WoT based on clustering semantic profiles of objects according to similarities, predicting context with a score calculation to select a particular algorithm.

A second group can be identified as resource-oriented, and [Corredor 2014] proposes an open platform called the micro Web of Things Open Platform (WoTOP) to enhance the re-usability of context data to deliver and create health, wellness and ambient assisted living (AAL) care services. It is based on a resource-oriented architecture using RESTful interfaces focusing on the integration, interoperability and re-usability of the model yet customized to healthcare environments. It tested the proposal in a simulated medium-size residential care home, executed by taking into account the number of subscriptions. Performance evaluation focuses on the delay time.

We have also included and presented a survey of the major evaluation frameworks used for IR development in recent years, and highlight the major advantages and characteristics of our approach. This paper presents a more in-depth comparison of the simulation, emulation and/or prototype environments used to develop IR systems in similar contexts to IoT and/or WoT as illustrated by Table I, Table II and Figure 10.

We have considered the following comparison dimensions:

— Evaluation type: PROTO stands for prototype, SIM for simulator and EMU for emulator
— Scalability: given the number of simulation/emulation entities which can be represented or abstracted
— IR scope: sensor-level only, thing-level only, space-level only, sensor-thing-level, all-levels
— Context awareness: spatial, temporal, states-at-query, spatio-temporal

From the **scalability dimension**, we can highlight two contributions which focus on achieving super-large emulation [Looga 2012] and simulation [Dhoutaut 2013]. Both lines of research concur that it is fundamental to test and understand the issues of

Table II. Comparison of simulation, emulation and prototype environments II

Name	Scalability	IR Scope	Awareness	Models
DE-WoTSIM	Large	All-Levels	Spatio-Temporal	Behavior
DPWSim	Medium	None	Temporal	Operations
WoTOP	Medium	None	Spatial	Resource
Truong-Romer	Small	Sensor-Level	States at Query	Similarity
WoTCO	Small-Medium	Sensor-Level	Spatial	Actuator
Mayer-Trifa	Medium	Thing-Level	Spatial	None
Christophe-Verdot	Small	Thing-Level	States at Query	Prediction
SPITFIRE	Small	Sensor / Thing	States at Query	Auto-Annotation
EnvB	Not Given	Thing-Level	Spatial	Interaction
UbiVisor	Not Given	Thing-Level	Spatial	Augmented-Reality
Gander	Medium	Thing-Level	Spatio-Temporal	Here and Now
Elahi-Kellerer	Medium	Sensor-Level	States at Query	Prediction
Dyser	Large	Thing-Level	Spatial	IR Space
MAMMoTH	Super-Large	Sensor-Level	Temporal	Node
WSL3	Medium	Sensor-Level	Spatial	Prediction
Song-Han	Small	Sensor-Level	Spatial	Radio-layer
VisibleSIM	Super-Large	Sensor-Level	Spatial	Mobility
ISE	Small	Thing-Level	Spatial	Indexing
IoT-SVK	Not Given	Sensor-Level	Spatio-Temporal	Indexing
SMART	Not Given	Sensor-Level	Spatial	Social Streams

scalability on networking and systems themselves. The emulator is suitable for testing super-large-scale scenarios with a focus on sensor networks. [Dhoutaut 2013] presents an interesting architecture which is able to simulate four environments, where the simulation core comprises multiple targets, a networking model and a physical engine to perform calculations to determine mobile object movement by applying Newton's second law. It does not consider the fact that these nodes can be geographically distributed and hierarchically organized or attached to a real-world thing. In this respect, our approach involves spatial context awareness with a hierarchical perspective adding scalability and the possibility of representing an entire smart world.

From the **IR scope dimension**, we can highlight an evolution of approaches starting with the possibility of a *sensor-level search*. For example, [Elahi 2009] focuses on predicting models for the sensor search, introducing a ranking to enable an efficient search with a certain output state at the time of the query. [Ding 2012] focuses on indexing techniques and search algorithms and presents the IoT-SVK search engine which is based on spatio-temporal, value-based, and keyword-based conditions for real-time retrieval of massive sensor data. [Albakour 2012] focuses on a framework for community-based sensors, while also considering social interactions as a source of social sensor information. [Truong 2012] focuses on a fuzzy sensor search and a similarity search which are extremely accurate.

At the *thing-level search*, [Romer 2010] proposes a constraint-based entity search tested on the Bicing system in Barcelona, describing the problem statement in greater detail and the design space in terms of IR. [Mayer 2012] uses a scenario with multiple sensors within six spaces and its main contribution is treating the location as a very important property of things. In terms of semantic enrichment, [Pfisterer 2011] focuses on a semantic search using a cluster of annotated sensors and automatic inference of the annotation of new sensors and adding semantics to the things. The main difference between our contribution and [Pfisterer 2011] is the semantic enrichment component and we have decided to use an XML representation with tags that follow the microdata and OGC3 SensorML proposals; Pfisterer, on the other hand, uses RDF and LOD. We all agree, however, on the usefulness of semantic enrichment. [Garcia-Macias 2011] presents a sentient visor approach for browsing the IoT and analyzes the characteristics that distinguish it from the IoT. We could also highlight the framework introduced by [Console et al. 2013] for creating rich environments, whereby augmented reality is a main feature as in [Garcia-Macias 2011], and the social networking web merges with the paradigm of the Web of Things to establish a social web of intelligent things and people.

From **context awareness**, we can highlight [Michel et al. 2012] who tested query performance on a real data-set by introducing a distributed processing mechanism with an innovative here-and-now vision. This approach broadly introduces the importance of the *temporal context* for IR systems on the WoT. It is, however, restricted to nearby personal networks. Similarly, another work [Jin 2011] presents the IoT search engine called ISE, which focuses on searching for nearby terms with RFID. In this regard, a considerable number of approaches emphasize the *spatial context* such as [Thebault et al. 2013] and [Thebault 2011] who present a mobile browser prototype for the WoT, with a user interface approach to facilitate the thing's interaction and [Mayer et al. 2014] who presents a model-based interface description scheme to enable automatic, modality-independent user interface generation which can act as a ubiquitous controller.

4. DISCRETE-EVENT SIMULATION MECHANISM

This section details the low-level design (LLD) of our discrete-event simulation mechanism. Figure 11 illustrates the macro components that comprise the DES mechanism and the logic of the entire simulation engine of the proposed WoT simulator. We begin by listing the following concepts:

— System: collection of entities (e.g. intelligent zones, smart spaces and subspaces, virtual things and virtual sensors) which interact together over time to represent the physical world
— WoT model: an abstract representation of a system, containing structural, logical or mathematical relationships in terms of state, entities and their attributes, sets, processes, events, activities and delays
— System state: collection of variables that contain all the information to describe the system at any time
— Entity: any object or component in the system that requires explicit representation in the model
— Attributes: the properties of a given entity
— Entity list: a collection of associated entities

Discrete-event simulation is stochastic, dynamic and discrete. Inter-arrival times are random variables and have density and cumulative distribution functions.

4.1. Stopping Event

Simulation has a stopping event (SE) that defines how long the simulation will run. We have decided to use the following two mechanisms to stop the simulation:

— At time zero, we schedule a stop simulation event at a specified future time (TE, Time of End). We therefore know that the simulation will run during the time interval [0,TE].

— TE is determined by simulation and is the time of occurrence when the number of entities vThings or vSensors reaches the maximum number defined by the user.

4.2. Events in the WoT Simulation

We consider an event to be an instantaneous occurrence that changes the state of a system. It is created with a Java object where the main attributes are event type and

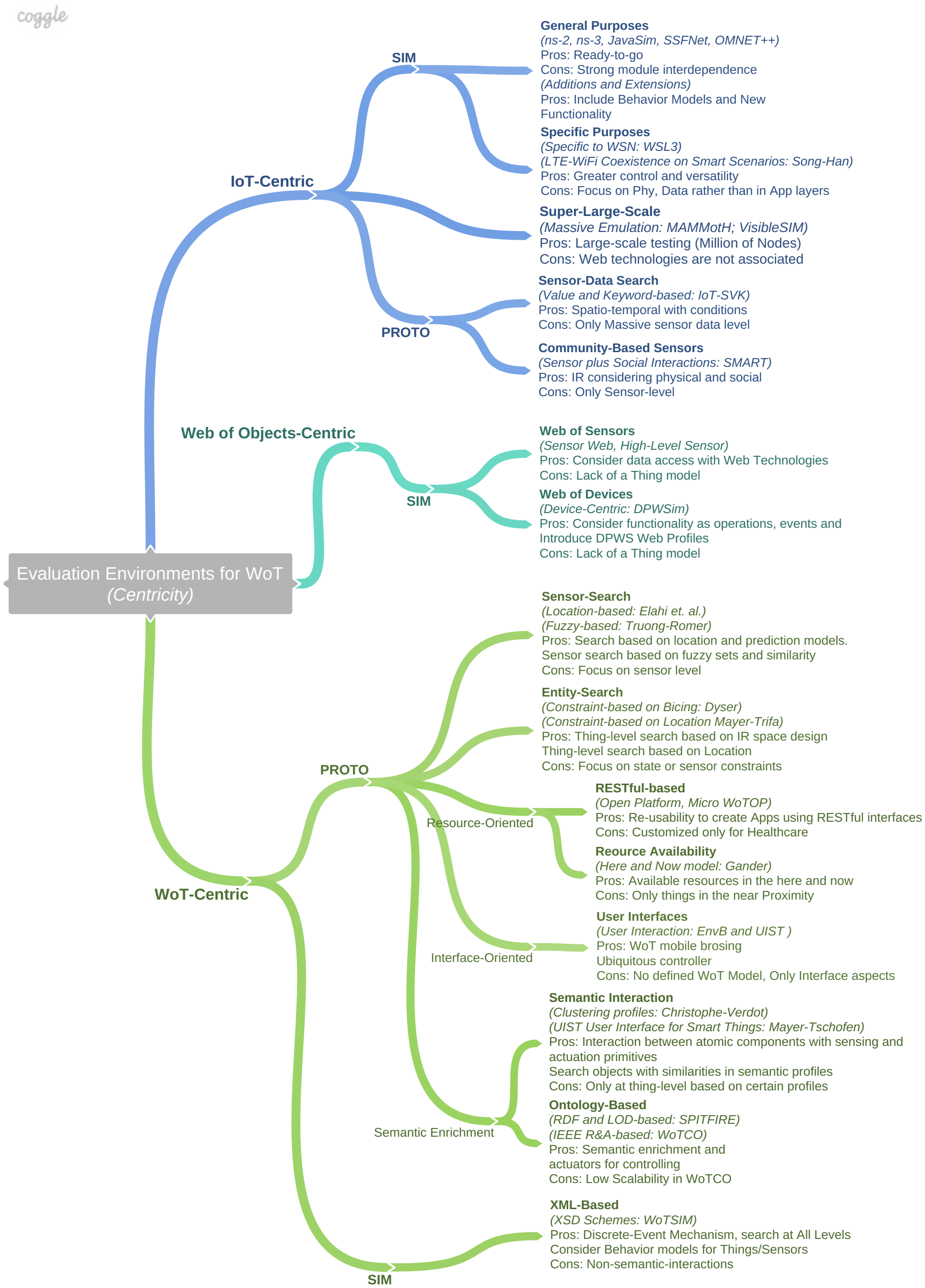


Fig. 10. Evaluation framework taxonomy

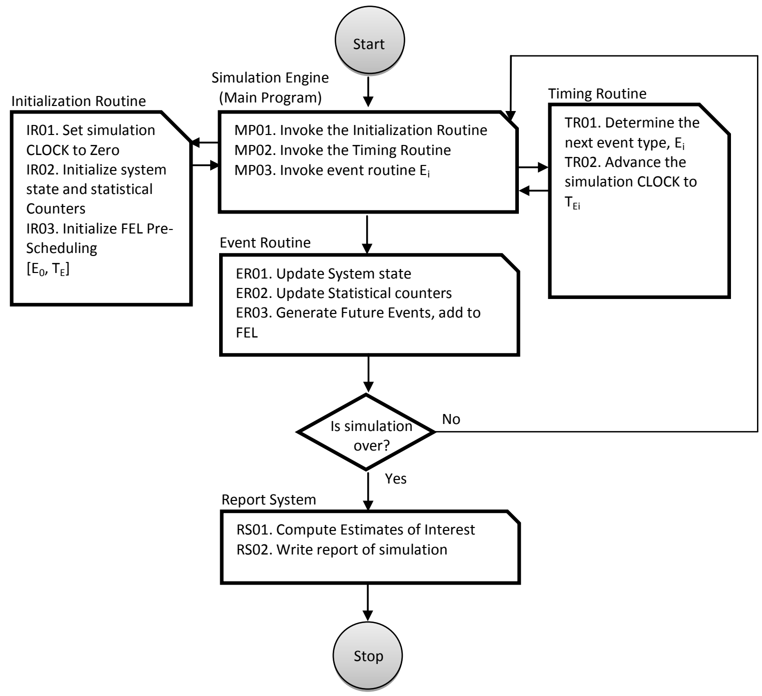


Fig. 11. Macro components and logic of the WoT simulation engine

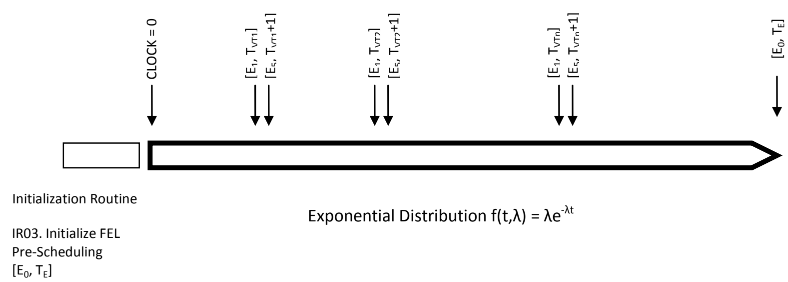


Fig. 12. Pre-scheduling of vThing arrivals and vSensor samples with inter-arrival exponential distributions

event time. Each event notice is recorded together with its current or future time occurrence attributes and any associated data necessary for execution. After initialization of the simulation system, a future event list (FEL), which includes the event notices for future events ordered according to time of occurrence, is also initialized and a first pre-scheduling is performed with the stopping event. Table III illustrates the event types in the WoT simulation with a brief description.

The stopping event has been established as Type 0 and is scheduled in the initialization routine at IR03. At time zero, we also pre-schedule Type 1 event notices for every vThing in the entity list, and Type 5 events for every vSensor associated to these vThings. The inter-arrival vThing time follows an exponential distribution and we as-

Table III. Type of events in the WoT simulation

Type	Description
0	Stopping event of simulation
1	Arrival of a new vThing
2	Departure of a vThing
3	Disconnection of a vThing
4	Reconnection of a vThing
5	Sampling of a vSensor
6	Failure of a vSensor
7	Repair of a vSensor

sume that the vSensors take their first sample 1 time unit after the associated vThing reaches the simulation. Figure 12 shows the chronological result of IR03 on the FEL.

The event-scheduling and time-advance algorithm is implemented by the synchronization routine as follows (see Figure 11):

— Step TR01. Remove the event notice for the imminent event from the FEL.
— Step TR02. Advance CLOCK to the imminent event time.

After the CLOCK advances to Ti, a Type 1 event is executed and the event routine is performed.

— Step ER01 Execute imminent nth event: update system state, change entity attributes and set membership as needed.
— Step ER02. Update cumulative statistics and counters.
— Step ER03. Generate future events and place their event notices on FEL ranked by event time.

Type 1 event: *Arrival of a new vThing*. A smart arrival subspace is assigned having been randomly selected with uniform distribution among all the smart subspaces in the simulation. Figure 14 illustrates the transition diagram of events, in which for example a new vThing belongs to a smart subspace which has been selected with a probability of 1/(total number of subspaces).

With regard to the system state update and cumulative statistics and counters, we assume the new vThing to be in a stationary online state. Smart subspace available seats are decreased by one. This attribute is used to limit the number of vThings that can be members or can be contained by the smart subspace. The vThings arrival counter is increased by one.

The Type 1 event can generate the following future events:

— Type 2 event within a stationary time following an exponential distribution from the current CLOCK.
— Type 1 event in case the available seats are equal to zero in the space, with a relocation time following an exponential distribution from the current CLOCK.

Type 2 event: *Departure of a vThing*. In this event, the state of the vThing is changed to moving online, increasing the number of available seats in the smart subspace. The counter of vThing movements is increased by one. The vThing could become disconnected from the network because of external factors with a certain probability and if this occurs, a future Type 3 event is scheduled with an average disconnection time during movement. Otherwise, a Type 1 event is scheduled with an average moving time and both temporal times are characterized by an exponential distribution.

Type 3 event: *Disconnection of a vThing*. The vThing state changes to offline and consequently real-time information about the vThing and associated vSensors is not available to read in the WoT. The offline vThing counter is increased by one. Like Type

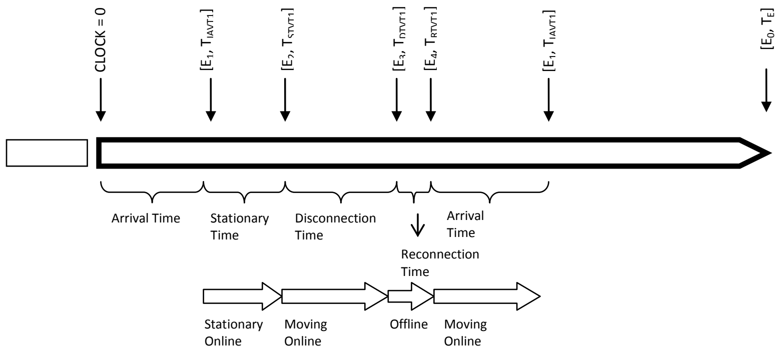


Fig. 13. Example of future events for a specific vThing

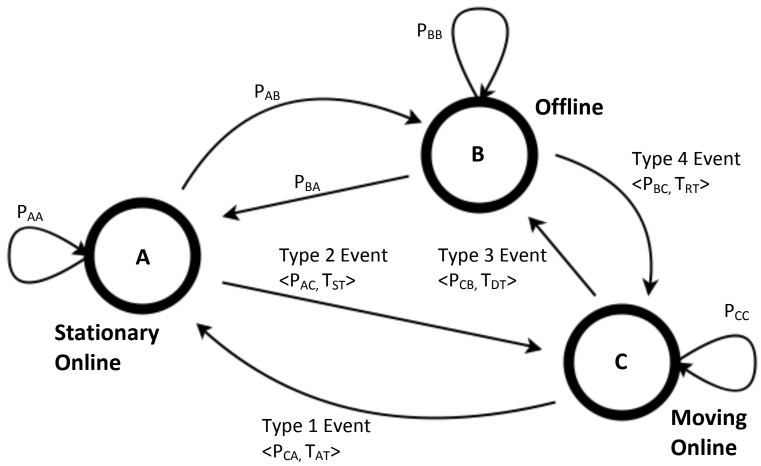


Fig. 14. vThing transition diagram

2 events, Type 3 events are based on a three-state Markov chain and the future state of the vThing is computed with a certain probability given by the transition matrix. The vThing could therefore reconnect at some future point whilst moving or as a result of reaching a smart subspace when the vThing connects to Internet. As a result, we schedule either a Type 4 or Type 1 event with an average connection time from the CLOCK.

Type 4 event: *Reconnection of a vThing*. In this event, the vThing state changes to moving online. A future Type 1 event is scheduled.

Figure 13 shows a sequence of events scheduled on the FEL during simulation and details the service times between these events and the transition in vThing states during simulation. For this same example, a transition diagram is shown in Figure 14.

Type 5 event: *Sampling of a vSensor*. As mentioned previously, a Type 5 event is scheduled for each vSensor one simulation time interval after a new vThing arrives at the simulation. When a vThing reaches a smart subspace, its associated vSensors start to measure and sample the variable of interest. We assume that the initial vSensor state is active. The vSensor generates a sampling value with a certain probability

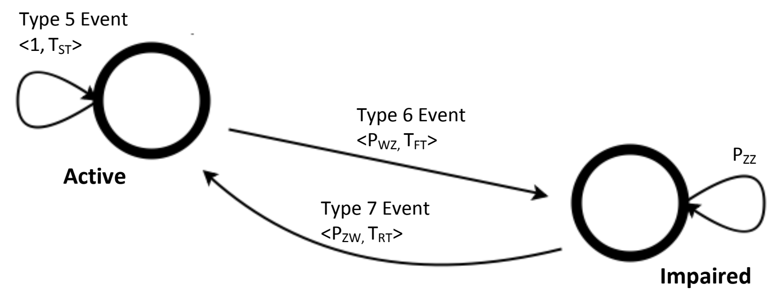


Fig. 15. vSensor transition diagram

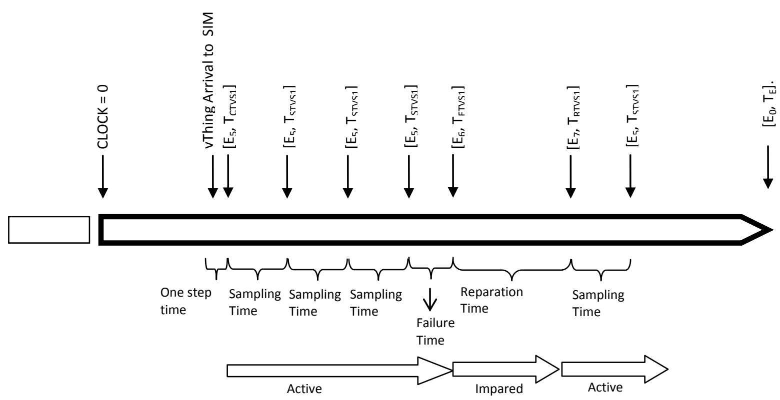


Fig. 16. Example of future events for a specific vSensor

distribution function defined by the user. The random variable is also characterized by a maximum and a minimum value.

In the possible case of the vSensor failing in the future with a given probability, a Type 6 event is scheduled with a failure time and with Weibull or exponential distribution. Otherwise, the vSensor continues operating normally and a further, periodic Type 5 event is re-scheduled at x seconds from the actual CLOCK.

Type 6 event: *Failure of a vSensor*. If this occurs, the vSensor enters the impaired state. Consequently, there will be no available information from its variable of interest. We assume the vSensor can be repaired in the future, so its reactivation is scheduled with a Type 7 event, within a repair time.

Type 7 event: *Repair of a vSensor*. The vSensor returns to the active state and a Type 5 event will occur in a sampling time. Figure 15 illustrates the transition diagram for a vSensor. Figure 16 shows a sequence of events scheduled on the FEL during simulation and the times between these, and the state transition for a single vSensor.

5. SIMULATION OF WEB OF THINGS REAL SCENARIOS

In this article, we evaluate the simulator engine in terms of scalability and diversity of the scenarios to simulate. We chose the following real scenarios to illustrate the flexibility of our model, and present various experiments and simulation results. As we showed in the previous section, we have included a specific set of behavior models in the discrete-event mechanism related to the virtual thing and virtual sensor entities. However, the mechanism could be extended to include other models not covered in this work. In the first subsection, we list the characteristics of our proposed simulator and other features that the approach cannot currently perform but which might be included in the future (see Table IV), and describe alternative scenarios in which the simulation approach could be used. We also explain how to obtain the inputs and data sources for feeding the simulator, and the main parameters associated with things/sensors which are in fact the main input/output data sources for our WoT approach.

We present three scenarios which were selected to exhibit different attributes. Firstly, Scenario A has a single sensor type, a single virtual thing and a single smart space in an intelligent zone. Secondly, Scenario B has multiple sensor types, one virtual thing type, two smart spaces in an intelligent zone. Finally, Scenario C has multiple sensor types, multiple virtual thing types and multiple smart spaces in an intelligent zone.

5.1. WoT Simulator Usage, Boundaries and Limitations

Generally speaking, WoTSim could be used by researchers to:

— Simulate the WoT dynamics of a system, considering the behavior models of things/sensors
— Extend thing/sensor model behavior to include other WoT features and characteristics
— Support the design of model-driven functional systems such as Web-based forecasting systems
— Test extension to existing models in order to add new functionality and gather data to optimize transport systems
— Explore the industrial IoT/WoT application (this would require the inclusion of aspects relating to availability, reliability and resilience)
— Help in the characterization of real-world dynamic systems to build next-generation control systems

From the **IR perspective:**

— Design and test crawling and indexing modules of an IR system in terms of indexing delay, query response time
— Design and test ranking and retrieval modules of an IR system in terms of accuracy, precision and recall
— In particular, by way of future work, we plan to evaluate different XML indexing strategies, add real-time techniques and select most suitable data structures for an IR system for the WoT.

When a simulation is being prepared, the following steps are important to gather and provide the parameters to the simulator:

(1) Map and adapt the WoT model to a real-world scenario (Insertions Menu Figure 8)
   (a) Preset spatial context: all three space levels contain the same tag information and remain as a constant.
   (b) Gather spatial information: geographical information can be extracted from http://www.geonames.org/ through a DB query or web service. For example, Kingdom of Spain is characterized by the geocode PCLI Independent Political Entity, geoclass A Country, and coordinates. On the other hand, an intelligent zone or smart space requires an internal classifier and authorizing registration information provided by the user.

Table IV. Boundaries of the proposed WoT simulator

Feature	DOs	DONTs (*Feasible Extensions*)
Structure	Nested and Extended WoT model	vSensors directly attached to Spaces
Entity	Uniform distribution for vThing's mobility	Complex vThings mobility and trajectories
Capabilities	vThings markovian states based on connectivity	Thing IoT interconnection M2M protocols
Geographic	Spatial awareness with geo-referenced data	Multi-layered GIS using vectorial sytems
Prediction	Stochastic data created from models	Prediction-based data reduction at vSensor
Retrieval	IR Perspective and Synthetic Collection at Disk	Web-Service API for consuming data

   (c) Gather thing information: descriptive information about the real-world thing is entered in textual fields. For real scenarios such as Scenario A, the information can be extracted from data-sheets of specification documents, in this case https://www.bicing.cat/.
   (d) Gather sensor information: descriptive information about the real-world thing is entered in textual fields. Three main fields relate to the intrinsic nature of the sensor and the observed property: the probability density function, which could be based on previous estimations for the observed property, e.g. Gaussian, Beta distributions for properties such as temperature, humidity; the min-max values or operation range of the sensor, e.g. 0°C - 200°C in a thermocouple; and the unit of measurement for the observations, e.g. Celsius in the example case.
(2) Environmental variables of the discrete-event mechanism (Simulator Menu Figure 9)
   (a) Max time: stop criteria to stop the simulation, as a function of the discrete-event mechanism time (Time of End).
   (b) Max vThings/vSensors: alternative stop criteria based on max vThings/vSensors in the simulation.
   (c) Lambda vThings/vSensors: Lambda is the mean arrival rate per unit time following an inter-arrival time of exponential distribution.
   (d) Probability distribution function PDF vThings/vSensor: 1 for exponential. These environmental variables must be set according to the size of the WoT system to be simulated and dynamics.

5.2. Scenario A: Bicing system

We consider the Bicing system www.bicing.cat to be a WoT scenario. Bicing is a bicycle-sharing system in Barcelona and its main purpose is to cover short and medium-length daily routes within the city. Our first step in terms of the simulation is to map this real WoT scenario to our WoT model. For the upper layers, the Kingdom of Spain is considered to be our intelligent zone with the ability to federate Spanish smart spaces, and directly associated to .es and .cat WoT domains. In this respect, the City of Barcelona is modeled as a smart space containing more than 421 smart subspaces each representing one of the Bicing stations in the system.

We create 6000 virtual things corresponding to the 6000 bicycles in the system, each with an attached virtual sensor. There is a considerable difference in the lower layers between the real scenario and our WoT vision and model. The physical sensors, part of the IoT infrastructure, are physically attached to the Bicing stations collecting a bike's availability in a slot. Although we consider the spatial context to be extremely important, by definition the WoT is thing-centered. In our proposal, therefore, the expected dynamics can escalate in size by adding, for example, GPS, accelerometers, user heart rate monitor sensors to the bicycles. Furthermore, the WoT vision encompasses bestowing intelligence on things through their Web representation. As we have already mentioned, virtual things are associated with the smart space where they are located, and in our simple mobility model bicycles move from one station to another with a uniform probability. We assume that the bicycles are connected to Internet. Whenever an entity is created, an XML document collection is generated at zero simulation time and there is one XML document for each entity in the simulation. In other words, we build an IR test collection comprising 12,424 XML documents. From this collection, 3.4 percent are static and 96.6 percent are dynamic XML documents with real-time information.

While the simulation is running, every bicycle movement is logged with a timestamp in its XML document with an update of its current station. The XML bicycle representation stores changes in Internet connection and general virtual thing state changes. In a lower layer, information about the virtual sensors attached to the bicycle is kept in XML documents, recording whether the bicycle is available or not to rent in that subspace and this experimentation could be modeled with a binomial probability distribution.

5.3. Scenario B: Lübeck system

In the previous example, we show a scenario with a simple type of sensor and the single smart space of Barcelona. The next scenario is the Belgrade facility in the SmartSantander project www.smartsantander.eu/. The EkoBus system has been implemented in the cities of Belgrade and Pancevo and public transport vehicles monitor a series of environmental parameters (CO, CO2, NO2, temperature, humidity) over a large area and provide additional information for the end-user such as bus location and estimated arrival times at the bus stops. In order to simulate this scenario, we select Serbia to be the intelligent zone for the model. Two smart spaces are contained in this zone: the cities of Belgrade and Pancevo. Each of the Ekobus system path routes is characterized as a smart subspace. We also modeled every bus stop as a smart subspace. Each Ekobus is represented by a virtual thing and there are five in Belgrade and sixty in Pancevo. Each virtual thing contains six types of sensors: NO2, CO, CO2, temperature, humidity and GPS location.

During the WoT simulation, Ekobus movement is logged with a timestamp in its XML document and its current stop and path route are updated. The XML representation of each Ekobus stores changes in Internet connection and general virtual thing state changes. In a lower layer, XML documents store information about every virtual sensor attached to the Ekobus, such as whether the Ekobus is at a particular bus stop. In this scenario, at zero simulation time, there is an IR test collection comprising 638 XML documents. This collection represents one intelligent zone, two smart cities, sixty smart path subspaces and one hundred and twenty smart bus stop subspaces, sixty-five virtual things and three hundred and ninety virtual sensors.

5.4. Scenario C: WoT-based Ambient Assisted Living System

In the previous scenarios, we described how the proposed WoT model can be used to represent real-world and IoT/WoT systems. Both employ the WoT model in a similar approach and both have only one type of thing, although the second model uses different types of sensors. In this example, we introduce a scenario with multiple types of virtual things and sensors.

Ambient Assisted Living (AAL) aims to allow older adults to live actively and independently at home. We can model the house with AAL facilities such as an intelligent zone comprising smartspaces, areas and subareas e.g. a shelf. Moreover, medical equipment can be modeled as a virtual thing, e.g. a portable ECG or a glucometer. It is worth mentioning that the number of Internet-connected, portable medical mobile devices is expected to double over the next five years. Even medicine such as insulin can be modeled as a virtual thing with some kind of connectivity or smart identification in order to log its location, quantity or expiration date. In this scenario, a patient or a relative might be interested to knowing in real-time where to find the glucometer or insulin,

Table V. Experimental results

Result Variable	**Scenario A**	**Scenario B**	**Scenario C**
IR Collection Size (Initial in MB):	15.5	0.8	0.2
IR Collection Size (Final in MB):	59.2	40.3	18.7
IR Collection Growth Rate in KB/m:	30.3	27.4	12.84
IR Collection Changes per Second:	107	14	9
IR Collection Number of Files:	12424	638	121
IR Collection Number of Folders:	6424	248	61

or how much insulin is left. During the simulation, we build an IR test collection comprising XML documents and these represent one intelligent zone, ten smart spaces, thirty smart subspaces, twenty virtual things and sixty virtual sensors.

6. EXPERIMENTATION AND RESULT ANALYSIS

In regard to the design of the experimentation, we decided to consider the following fundamental variables:

For the **IR test collection**: collection size, number of changes in the collection, growth rate For the **simulator performance**: CPU consumption, memory consumption, disk usage For the **WoT dynamics**: number of events in time for the virtual things and number of events in time for the virtual sensors

Table V illustrates the results of an experiment conducted over a 24-hour period. We show the growth of the collection size in relation to the number of virtual things and sensors, which can also refer to the number of files and folders in the simulation.

The initial size of the IR test collection is expressed in MB and depends on the number of files and folders in the simulation scenario. We estimate the growth rate by considering the final size after the simulation has stopped. We calculated an average number of changes per minute from the log files that the simulation keeps during the time period for each event.

In Figure 17 we show the CPU result and the memory consumed for the experiment. We monitor the CPU and memory resources allocated to the main Java process on the PC hosting the simulation engine. The computer has the following specifications:

— OS: Windows 8 Pro x64
— Processor: AMD A6-3500 2.10 GHz
— RAM: 8.00 GB

When idle, we estimate 0.18% of average CPU usage and 586KB allocated to the Java process. Once the simulation is running, the average CPU increases according to the number of simulation entities or complexity of the scenario and also the physical memory used by the process. We track a maximum of 4.25% of CPU consumption on average by the simulation engine with about 12,000 simulation entities running.

Finally, Figure 18 represents the dynamics of the WoT in terms of the number of events over time in the lower layers of our model. In Scenario A we can see that model dynamics are high at both levels due to the number of entities exceeding the number of sensor types. In this case, we can see the impact of a Type 5 event (sampling of a vSensor) since each of the 6000 sensors in the simulation generates a measure per minute in this simulation setup. The dynamics of the vThing level are quite similar in every scenario, and we have used the same mobility and connection model for each of the three scenarios so the dynamics are the same. The dynamics in events in the vThing and vSensor layers directly affect the changes per second that the IR test collection assumes.

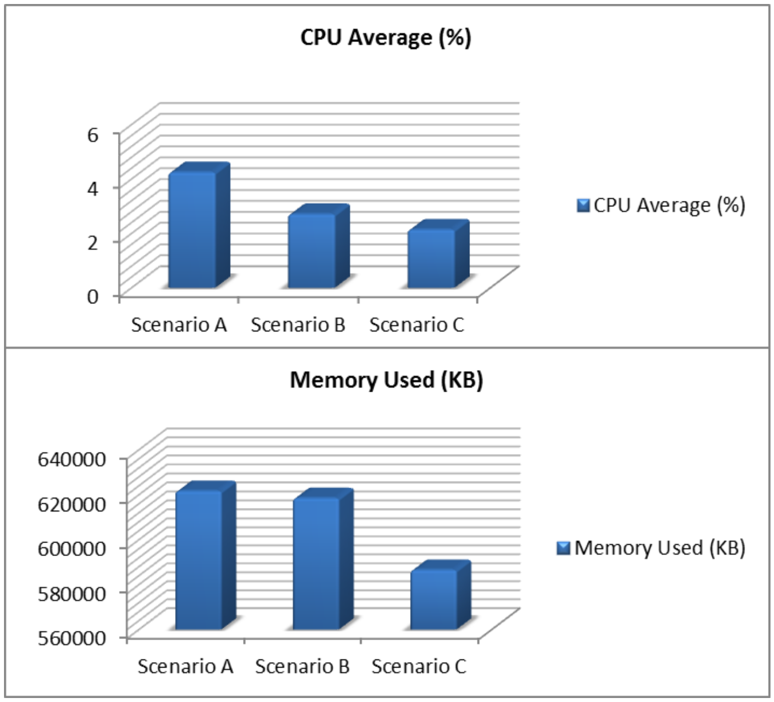
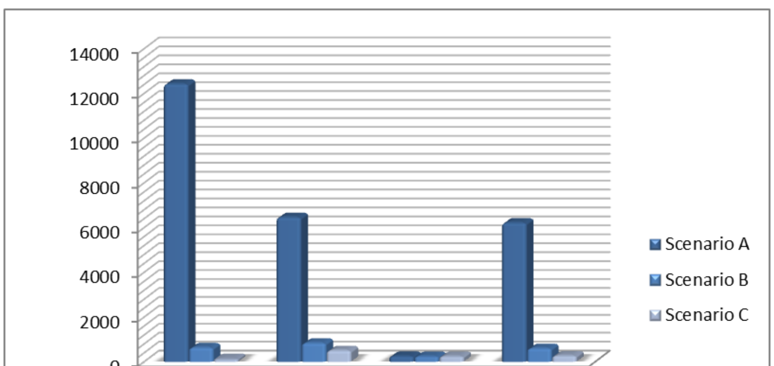


Fig. 17. Graph of average CPU, memory used



	Files	Changes / sec	Events at VT Level / min	Events at VS Level / min
Scenario A	12424	6462	246	6216
Scenario B	638	842	238	604
Scenario C	121	509	241	268

Fig. 18. Graph of WoT dynamics

7. CONCLUSIONS AND FUTURE WORK

The Web of Things has a different dynamism from the conventional web and it is essential to bear this in mind when designing and developing applications and information retrieval systems. This dynamism stems from changes in the location of real-world things, in the collection of documents storing information about new sensors or objects, in the connectivity status, in the lifetime of these documents data and their removal from the document collection, in addition to the periodic, real-time updating of data collected by the sensors associated with real world things. Most WoT models focus on the sensor layer which is overlaid with real-world abstraction layers according to the perspective and purpose of each research project, such as for example an abstraction layer of real-world things or entities. Different alternatives exist for describing and representing each layer in the different WoT models by using both non-Web and Web technologies such as metadata, microdata or ontologies. In view of their flexibility and simplicity, XML standards are used to describe and represent the WoT model layers in our proposal. Our proposed abstract WoT model with its dynamic representation takes into account the efforts of organizations for technological standardization such as OpenGIS, W3C, ISO for interconnection with the real world by considering a real-world view that highlights the importance of the spatial context with the addition of the relations between things and spaces. The temporal context is added via historical event elements. By way of future work, we plan to study different conventional and semi-structured indexing information methods with an XML focus to assess their performance and suitability for real-time WoT indexing. The model is being used to build a discrete-event WoT simulation with XSD schemas as inputs and to marshal an extended collection of XML documents.

In terms of scalability, we have built a flexible, versatile, large-scale simulator which is able to mimic an entire smart world and which contains entities that encapsulate spatial and temporal contexts. As in other research, in our work we consider location and spatial context to be important factors that should be considered during WoT simulation. For semantic enrichment, we decided to use an XML representation with tags following microdata and OGC3 SensorML proposals, although there are other alternatives such as RDF, OWL, LOD. One important contribution of the discrete-event simulator presented is the possibility of encapsulating expected WoT dynamics. It is also possible to modify and improve the behaviour models for virtual things and sensors in terms of mobility, connectivity and failure or even to include new behaviour models for these entities.

In the future, the spatial context can be improved by means of a multi-layered GIS and spatial subdivision approach similar to [Mekni 2013], where the GIS representation includes layers with information about land usage, elevation, streets and blocks, and cadastral information. The GIS data is then deconstructed and combined to generate *virtual geographic environments* (VGE). Further lines of future work could also be followed in other ways to add models to the sensor data generation component by incorporating sensor behavior models with mathematical approaches to real data in specific environments as in [Seablom 2008] or models based on real sensor node implementations as in [Dhoutaut 2013]. In future research, it is important to study and evaluate the prediction models in the sensor layer since these could have a significant effect and impact on IR system design.

ACKNOWLEDGMENTS

This work has been funded by the Spanish Ministerio de Economia y Competitividad under the project TIN2013-42741-P and TIN2016-77902-C3-2-P. We would like to express our special thanks to the Iberoamerican University Association for Postgraduate Studies AUIP www.auip.org for their support for the implementation, progress and development of this work.

REFERENCES

Macdonald C.-Ounis I. Pnevmatikakis P. Soldatos J. Albakour, M. 2012. SMART: An open source framework for searching the physical world. In *Proceedings of the Workshop on Open Source Information Retrieval (SIGIR '12)*. IEEE, Portland, Oregon, USA, 1–4.

Wei Cheng, Nenggan Zheng, Man Lin, and L.T. Yang. 2011. ENST: A Simulation Toolbox Based on Simulink for E-Textile Networks. In *Proceedings of the IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*. IEEE, Dalian, 131–138. DOI:http://dx.doi.org/10.1109/iThings/CPSCom.2011.79

B. Christophe, V. Verdot, and V. Toubiana. 2011. Searching the 'Web of Things'. In *Proceedings of the Fifth IEEE International Conference on Semantic Computing ((ICSC) 2011)*. IEEE, Los Alamitos, CA, 308–315. DOI:http://dx.doi.org/10.1109/ICSC.2011.69

Luca Console, Fabrizio Antonelli, Giulia Biamino, Francesca Carmagnola, Federica Cena, Elisa Chiabrando, Vincenzo Cuciti, Matteo Demichelis, Franco Fassio, Fabrizio Franceschi, Roberto Furnari, Cristina Gena, Marina Geymonat, Piercarlo Grimaldi, Pierluige Grillo, Silvia Likavec, Ilaria Lombardi, Dario Mana, Alessandro Marcengo, Michele Mioli, Mario Mirabelli, Monica Perrero, Claudia Picardi, Federica Protti, Amon Rapp, Rossana Simeoni, Daniele Theseider Dupré, Ilaria Torre, Andrea Toso, Fabio Torta, and Fabiana Vernero. 2013. Interacting with Social Networks of Intelligent Things and People in the World of Gastronomy. *ACM Trans. Interact. Intell. Syst.* 3, 1, Article 4 (April 2013), 38 pages. DOI:http://dx.doi.org/10.1145/2448116.2448120

Metola E.-Bernardos A.M Tarrio P. Casar J.R. Corredor, I. 2014. A Lightweight Web of Things Open Platform to Facilitate Context Data Management and Personalized Healthcare Services Creation. *Int J Environ Res Public Health* 11, 5 (2014), 4676–4713. DOI:http://dx.doi.org/10.3390/ijerph110504676

Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

Piranda B.-Bourgeois J. Dhoutaut, D. 2013. Efficient simulation of distributed sensing and control environments. In *Proceedings of the IEEE International Conference on Green Computing and Communications*. IEEE, Beijin, 452–459. DOI:http://dx.doi.org/10.1109/GreenCom-iThings-CPSCom.2013.93

Ion-Mircea Diaconescu and Gerd Wagner. 2014. Towards a General Framework for Modeling, Simulating and Building Sensor/Actuator Systems and Robots for the Web of Things. In *Proceedings of the First Workshop on Model-Driven Robot Software Engineering ((MORSE), co-located with STAF14)*, Uwe A and Gerd Wagner (Eds.). University of York, UK, 27–38. http://ceur-ws.org/Vol-1319/#morse14_paper_03

Dai J.-Gao X. Yang Q. Ding, Z. 2012. A Hybrid Search Engine Framework for the Internet of Things. In *Proceedings of the ninth Web Information Systems and Applications Conference*. IEEE, Haikou, 57–60. DOI:http://dx.doi.org/10.1109/WISA.2012.11

Romer K.-Ostermaier B. Fahrmair M. Kellerer W. Elahi, B. 2009. Sensor ranking: A primitive for efficient content-based sensor search. In *Proceedings of the Intl. Conference on Information Processing in Sensor Networks*. IEEE, San Francisco, CA, 217–228.

M. A. Feki, F. Kawsara, M. Boussard, and L. Trappeniers. 2013. The Internet of Things: The Next Tehnological Revolution. *IEEE Computer* 46, 2 (2013), 24–25.

Alvarez-Lozano J. Estrada-Martinez P. Aviles-Lopez E. Garcia-Macias, J. 2011. Browsing the Internet of Things with Sentient Visors. *IEEE Computer* 44, 5 (2011), 46–52. DOI:http://dx.doi.org/10.1109/MC.2011.128

Szymanski-B. K. Zerfos-P. Geyik, S. C. 2013. Robust Dynamic Service Composition in Sensor Networks. *IEEE Transactions on Services Computing* 6, 4 (2013), 560–572.

Molina B.-Palau C. Esteve-M. Gimenez, P. 2013. SWE Simulation and Testing for the IoT. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, Manchester, 356–361. DOI:http://dx.doi.org/10.1109/SMC.2013.67

Dominique Guinard. 2011. *A Web of Things Application Architecture – Integrating the Real-World into the Web*. Ph.D. Dissertation. ETH Zurich, Zurich, Switzerland.

D. Guinard, V. Trifa, and E. Wilde. 2010. A resource oriented architecture for the Web of Things. In *2010 Internet of Things (IOT)*. 1–8. DOI:http://dx.doi.org/10.1109/IOT.2010.5678452

Lee G.M.-Crespi N. Van Luong N. Han, S.N. 2014. DPWSim: A simulation toolkit for IoT applications using devices profile for web services. In *Proceedings of the IEEE World Forum on the Internet of Things ((WF-IoT))*. IEEE, Seoul, Korea, 544–547. DOI:http://dx.doi.org/10.1109/WF-IoT.2014.6803226

Jing He, Yanchun Zhang, Guangyan Huang, and Jinli Cao. 2012. A Smart Web Service Based on the Context of Things. *ACM Trans. Internet Technol.* 11, 3, Article 13 (Feb. 2012), 23 pages. DOI:http://dx.doi.org/10.1145/2078316.2078321

S. Hodges, S. Taylor, N. Villar, J. Scott, D. Bial, and P. Fischer. 2013. Prototyping Connected Devices for the Internet of Things. *IEEE Computer* 46, 2 (2013), 26–84.

Young-Sik Jeong, Hyun-Woo Kim, Neil Y. Yen, and Jong Hyuk Park. 2015. Multi-WSN Simulator with Log Data for Efficient Sensing on Internet of Things. *International Journal of Distributed Sensor Networks* 2015 (2015), 1–11. DOI:http://dx.doi.org/10.1155/2015/348682

Zhang D.-Zou Q. Ji G. Qian X Jin, X. 2011. Where Searching Will Go in Internet of Things?. In *Proceedings of Wireless Days (WD)*. IEEE, Niagara Falls, ON, 1–3. DOI:http://dx.doi.org/10.1109/WD.2011.6098173

Wen-Xiu A. Deng-Ao L. Dong-Dong Z. Yuan-Yuan C. Ju-Min, Z. 2013. Effective Algorithms for WSN With Weight Principle in Web of Things. *IEEE Sensors J.* 14, 1 (2013), 228–233. DOI:http://dx.doi.org/10.1109/JSEN.2013.2276619

Rumen Kyusakov, Pablo Puñal Pereira, Jens Eliasson, and Jerker Delsing. 2014. EXIP: A Framework for Embedded Web Development. *ACM Trans. Web* 8, 4, Article 23 (Nov. 2014), 29 pages. DOI:http://dx.doi.org/10.1145/2665068

Zhonghong O.-Deng Y. Yla-Jaaski A. Looga, V. 2012. MAMMOTH: A massive-scale emulation platform for Internet of Things. In *Proceedings of the IEEE 2nd International Conference on Cloud Computing and Intelligent Systems*. IEEE, Hangzhou, 1235–1239. DOI:http://dx.doi.org/10.1109/CCIS.2012.6664581

C. Manta-Caro and J.M. Fernández-Luna. 2014a. A discrete-event simulator for the web of things from an information retrieval perspective. In *Proceedings of IEEE Latin-America Conference on Communications ((LATINCOM))*. IEEE, Cartagena de Indias, 1–6. DOI:http://dx.doi.org/10.1109/LATINCOM.2014.7041851

C. Manta-Caro and J.M. Fernández-Luna. 2014b. Modeling the Web of Things from an IR approach. In *Proceedings of ECIR14 Information Access in Smart Cities Workshop ((i-ASC 2014))*, Iadh Ounis-Charles L. A. Clarke M-Dyaa Albakour, Craig Macdonald and Veli Bicer (Eds.). University of Glasgow, Amsterdam, the Netherlands, 7–10. http://dcs.gla.ac.uk/workshops/iASC2014/papers/iasc2014_caro.pdf

Guinard D. Trifa V. Mayer, S. 2012. Searching in a web-based infrastructure for smart things. In *Proceedings of the 3rd International Conference on the Internet of Things ((IOT))*. IEEE, Wuxi, 119–126. DOI:http://dx.doi.org/10.1109/IOT.2012.6402313

Simon Mayer, Andreas Tschofen, Anind K. Dey, and Friedemann Mattern. 2014. User Interfaces for Smart Things – A Generative Approach with Semantic Interaction Descriptions. *ACM Trans. Comput.-Hum. Interact.* 21, 2, Article 12 (Feb. 2014), 25 pages. DOI:http://dx.doi.org/10.1145/2584670

Mehdi Mekni. 2013. Holonic Virtual Geographic Environments. *International Journal of Remote Sensing Applications* 3, 3 (2013), 117–126. http://www.ijrsa.org/paperInfo.aspx?ID=4826

Mehdi Mekni and Phil Graniero. 2010. A Multiagent Geosimulation Approach for Intelligent Sensor Web Management. *International Journal of Distributed Sensor Networks* 2010 (2010), 1–16. DOI:http://dx.doi.org/10.1155/2010/846820

Jonas Michel, Christine Julien, Jamie Payton, and Gruia-Catalin Roman. 2012. Gander: Personalizing Search of the Here and Now. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Alessandro Puiatti and Tao Gu (Eds.). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 104. Springer Berlin Heidelberg, 88–100. DOI:http://dx.doi.org/10.1007/978-3-642-30973-1_8

Römer K. Bimschas D. Kleine-O. Mietz R. Truong C. Hasemann-H. Kröller A. Pagel M. Hauswirth M. Karnstedt M. Leggieri M. Passant A. Pfisterer, D. 2011. SPITFIRE: Toward a Semantic Web of Things. *IEEE Communications Magazine* 49, 11 (2011), 40–48.

Ostermaier B. Mattern F. Fahrmair-M. Kellerer W. Romer, K. 2010. Real-Time Search for Real-World Entities: A Survey. In *Proceedings of the IEEE*, Vol. 98:11. IEEE, Los Alamitos, CA, 1887–1902. DOI:http://dx.doi.org/10.1109/JPROC.2010.2062470

Hassapis G. D.; Gialelis-J. V. Samaras, I. K. 2013. Modified DPWS Protocol Stack for 6LoWPAN-Based Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics* 9, 1 (2013), 209–217.

Talabac S.J. Ardizzone J. Terry-J. Seablom, M.S. 2008. A Sensor Web Simulator for Design of New Earth Science Observing Systems. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*. IEEE, Boston, MA, 298–301. DOI:http://dx.doi.org/10.1109/IGARSS.2008.4780087

Han B. Zhang X. Yang D. Song, Y. 2012. Modeling and simulation of Smart Home scenarios based on Internet of Things. In *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, Beijin, 596–600. DOI:http://dx.doi.org/10.1109/ICNIDC.2012.6418824

Seablom M. Emmitt G. D. Wood S. Atlas R. Ardizzone-J. Burns R. Kemp E. Talabac, S. J. 2010. End-to-End Design and Objective Evaluation of Sensor Web Modeling and Data Assimilation System Architectures:

Phase II. In *Proceedings of the Earth Science Technology Forum*. NASA's Earth science endeavors, Arlington, 1–8.

Boussard M. Lu M. Mivielle C. Richir S. Thebault, P. 2011. EnvB: An Environment-based Mobile Browser for the Web of Things. In *Proceedings of the 15th international conference on Intelligent user interfaces, Workshop on interacting with smart objects*. 1–4.

Pierrick Thebault, Dominique Decotter, Mathieu Boussard, and Monique Lu. 2013. Embodying Services into Physical Places: Toward the Design of a Mobile Environment Browser. *ACM Trans. Interact. Intell. Syst.* 3, 2, Article 8 (Aug. 2013), 34 pages. DOI:http://dx.doi.org/10.1145/2499474.2499477

Romer K. Chen K. Truong, C. 2012. Fuzzy-based sensor search in the Web of Things. In *Proceedings of the 3rd International Conference on the Internet of Things (IOT)*. IEEE, Wuxi, 127–134. DOI:http://dx.doi.org/10.1109/IOT.2012.6402314

Dieter Uckelmann, Mark Harrison, and Floria Michahelles. 2011. *Architecting the Internet of Things* (1st ed.). Springer-Verlag Berlin Heidelberg. DOI:http://dx.doi.org/10.1007/978-3-642-19157-2

Received January 2016; revised May 2017; accepted