

ORIGINAL SUBMISSION

Open Access

Federated semantic nodes to scale search process in the IoT

Benoit Christophe

Correspondence: benoit.christophe@alcatel-lucent.com
Bell Labs Research, Alcatel-Lucent,
Route de Villejust, 91620 Nozay,
France

Abstract

The Internet of Things (IoT) refers to extending the Internet to heterogeneous components such as actuators, sensors or other smart devices. In this vision, the IoT foresees novel usage of such devices through applications enabling new device-device or device-people associations. Recent interest in providing digital extensions for these *things* has led to billions of connected devices offering their services or data through different platforms, some of them wrapped with semantic descriptions to realize aforementioned associations through accurate search processes. However, the ubiquitous aspect of the IoT and the potential mobility of the devices that compose it, prevent from adopting a centralized approach to search and manage these associations for scalability reasons. As devices composing the IoT are geolocalized, we believe that designing a framework composed of geographically distributed and cooperating nodes with local reasoning capabilities is a much more scalable approach to realize the IoT vision. We describe our approach of such vision by creating a federated network composed of such nodes that declare their location, process semantic descriptions of devices and share deduced associations with other peers that are selected based on their location nearness.

Introduction

The Internet of Things (IoT) is a recent paradigm that gained momentum by proposing to extend the Internet to a variety of *things* – such as Radio Frequency Identification (RFID) tags, sensors, actuators or other smart technological components (e.g. smartphones) – which allows to foresee novel usages and applications where these *things* cooperate together with people, to achieve a common goal [1]. Similar vision in Ubiquitous computing led early thinkers [2] to propose the use of Semantic Web technologies in order to automate this process by enhancing interoperability between these *things*. Hence few projects revolved around this idea. In particular, Chen et al. [3] as well as Katasonov et al. [4] coupled the use of ontologies with agents to allow heterogeneous devices to cooperate. Singh et al. [5] proposed a framework enabling the collaboration of different pervasive computing environments (called AS, standing for Autonomous Systems) by using Semantic Web technologies. More recently Pfisterer et al. [6] proposed an architecture aiming to open the access to sensors, allowing enhancing integration of data and services of heterogeneous sensors and facilitating novel applications. However, the recent interest

in providing digital extensions for these *things* has led to billions of connected devices offering their services or data through different platforms such as *Cosm*^a, *Nimbits*^b, *ThinkSpeak*^c or *Thingworx*^d. Considering that such connected devices may be mobile or may become unavailable, the underlying huge amount of heterogeneous and continuously changing data or services offered by these distributed connected devices, bring scalability issues.

We believe that the use of Semantic Web technologies must be thought with deployment considerations in mind i.e. integrating the distributed and ubiquitous aspect of the IoT, in particular when involved *things* are dispatched in an indoor environment. In this paper, we propose the vision of a distributed framework composed of nodes capable of processing Semantic Web descriptions of connected devices and organized in a federated architecture. We also propose that knowledge acquired by each node, be shared with geographically nearby peers, considering location as a very important criterion when searching or associating devices. Our expectation is that finding devices or their associations in the context of the IoT will refer highly to a given location that distributed nodes of our framework will better manage. The remainder of the paper presents the overall approach of federated semantic-enabled nodes sharing their knowledge in an indoor environment, a first implementation leading to early results assessing our approach and concludes on future works.

Federation of semantically described nodes

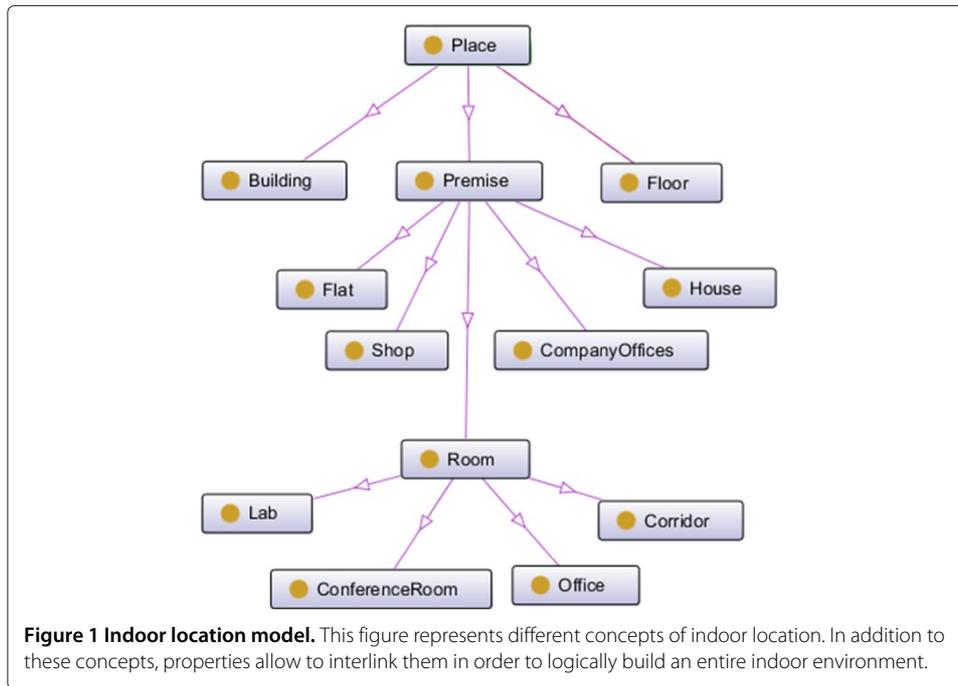
The federation as we envision it relies on creating geographically distributed nodes managing a pool of semantically described entities (connected devices or persons^e) in an indoor environment. Each node composing such federation is mapped on a place e.g. room, corridor, etc. which it processes the description to efficiently share knowledge with “nearby” nodes.

Describing indoor location node

Describing nodes mapped onto places of a given indoor environment requires to capture semantics associated to indoor location concepts as in this case, GPS coordinates of different rooms may not be obtained nor be sufficient to discriminate proximity between different nodes. We address such requirement by using the Web Ontology Language^f (OWL) to define a model representing structures of buildings, rooms or other premises. In this model (Figure 1), the *Place* concept represents a space delimited by physical boundaries (such as walls) and can be narrowed to *Building*, *Floor*, *Premise* or other kind of structures this model defines. Conjointly to the definition of a *Place*, some OWL properties (mentioned in Table 1) allow different places to be interlinked, resulting to *Places* being completely described and localized relatively to others.

Merging location of all nodes

Sharing knowledge between different nodes requires for a node to know its neighbors (i.e. the other nodes mapped onto *Places* near the *Place* represented by this one). Hence, assuming nodes have been described following aforementioned model, the federation uses a double cascading process to 1) merge and infer on all location data in the top-node of the federation and 2) spread such processed information back to all nodes (Figure 2).

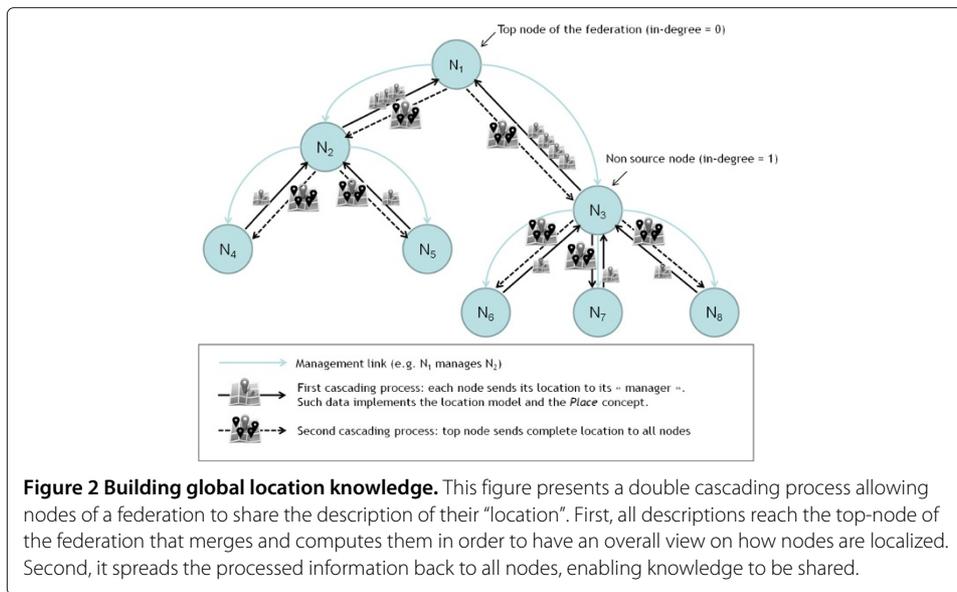


Result of this process is a global knowledge on nodes and their “neighbors”. As based on a set of nodes composing the federation at a given time, the process can be replayed to update such knowledge if nodes are added or removed from the federation.

Table 1 OWL Properties interlinking Places

Property Name	Domain	Range
<i>Description</i>		
contains	Place	Place
<i>Allows a Place to contains other places (e.g. a floor containing some rooms)</i>		
isAdjacentTo	Place	Place
<i>Models that two places are separated by some boudaries</i>		
inEast	Place	Place
inWest	Place	Place
inNorth	Place	Place
inSouth	Place	Place
<i>Refinement of isAdjacentTo</i>		
givesAccess	Premise and not House	Premise and not House
<i>Means that someone can go from one place to the other</i>		
hasAnnex	Building or House	Buidling or House
<i>Means that a Building or House can have annexes</i>		
hasAddress	Building or Premise	Address
<i>Allows to attach a postal address to some places</i>		
isIncludedIn	Place	Place
<i>Means that places can include other places</i>		
isPrivate	Place	Boolean
isPublic	Place	Boolean
isSemiPrivate	Place	Boolean
<i>Allows to know if a place can be searched or not</i>		

These properties allow different Places, Buildings, Rooms, Corridors, etc., to be interlinked together e.g. “Room A gives access to Room B”.

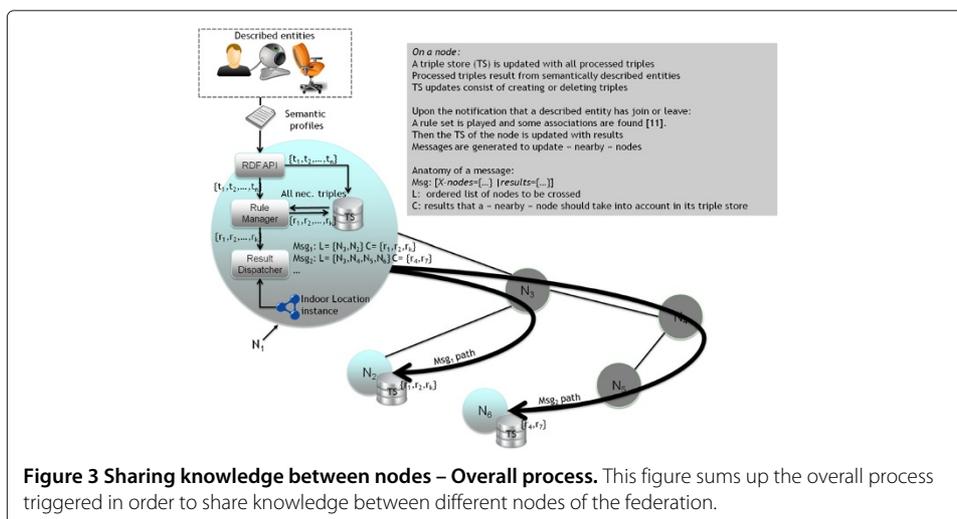


Sharing knowledge between federated nodes

To share knowledge between nodes, we apply a two steps process displayed in Figure 3. First, we execute a set of rules against the aggregated location data in order to select a set of nearby peers. Then, for each fact a node has learned (new association, new semantic entity having joined or left a *Place*, etc.), a message is created from this node to all selected recipients that can update their respective triple stores (recall that we process semantic descriptions, composed of RDF triples usually pushed in a triple store) dedicated to storing both semantic descriptions and such shared fact.

Customized Semantic Web rules

To share knowledge about a set of managed devices or their associations, processable rules are triggered on each node. Role of such rules is to determine a set of peers interested by the knowledge of a given node in order to initiate a mechanism throwing



messages that will convey the knowledge between the node and its determined peers. Rules can be added, removed and may depend on the strategy chosen to share knowledge between nodes. An example of such rule could be to notify all the places accessible from a given place P about a device having reached P . Some messages would then be generated between P and the places found by the execution of the rule.

Notification message

Upon the execution of a rule, a node has to send a message containing 1) *a body* composed of the results to share and 2) *a header* composed of the appropriate route to follow in order to reach a selected peer.

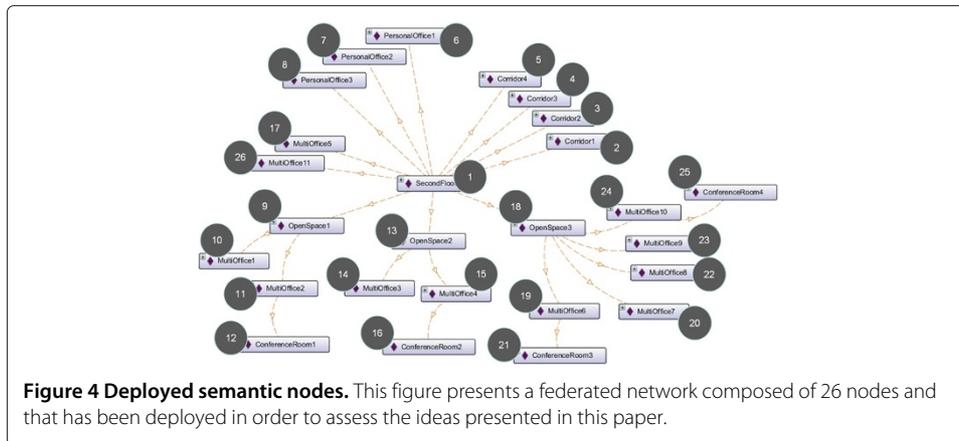
As results to convey are RDF triples, the *body* is simply composed of these triples.

Determining the path between two nodes is obtained by processing the gathered and inferred location of all nodes. It involves the anonymous property “inverse of contains” (with *contains* – a defined property – and its inverse provided by a semantic engine) to find the ancestors of both the issuer and the recipient nodes. Hence, with this property two sub-graphs are built, compared and merged if a common node is found. Applying Dijkstra [7] algorithm on this merged graph gives the shortest – and only – path between both nodes. Due to the particular nature of our federated infrastructure (Directed Acyclic Graph with no undirected cycles), we are sure that this process converges to a unique solution.

Implemented framework and early results

We have implemented the concept of a semantic node using different Semantic Web technologies. To store and retrieve RDF descriptions of connected devices (or persons), we used OWLDB [8] together with the OWL API [9]. We used Pellet [10] coupled with customized SWRL^g built-ins, to execute OWL-based rules enabling knowledge sharing process. We fed both Pellet and OWLDB with semantic profiles of entities respecting models detailed in [11]. To determine the path between two nodes willing to share knowledge, we used the *JGraphT* open source library, providing features to build and process graphs. Built-ins allowing messages to be generated and sent between nodes were developed in Java, and registered in Pellet. Messages generated by these built-ins were composed of one customized HTTP Request header (*X-nodes*) containing the ordered list of nodes retrieved by the aforementioned algorithm and a set of triples to push in the triple store of the recipient of such information. All mentioned components were wrapped in a Web application allowing the concept of semantic node to be deployed on a servlet container such as Tomcat^h.

We tested our approach by generating a federated network composed of 26 nodes (Figure 4) with up to four management levels (i.e. the maximum number of nodes between a leaf of the federation and the top-node is equal to four). To assess our approach with a sufficient amount of semantic profiles to be shared and searched, we generated 100000 semantic profiles resulting in approximately 1 million of triples. We measured that the time to exchange one triple between two nodes of this federation was under a second (see Table 2), confirming process feasibility (assuming that only few triples would be exchanged upon the discovery of associations or devices). In parallel, we measured that



the time required to discover a connected device was growing exponentially with the number of semantic profiles inserted in the queried triple store (see Table 3). These assessments strengthened then our concept of smartly cooperating semantic nodes, to search for connected devices and their associations.

Conclusions

We propose a distributed framework composed of nodes capable of processing Semantic Web descriptions and organized in a federated architecture, in order to allow scalable search and management mechanisms for the IoT. We consider a particular deployment infrastructure, where each node of such infrastructure is mapped to a physical environment (e.g. buildings, rooms, etc.). Knowledge acquired by nodes is then shared through a notification message mechanism, based on the execution of rules. A first prototype of

Table 2 Measuring time to exchange knowledge

Number of triples exchanged	Time taken to update store (in seconds)	Meaning
10	less than 1	Knowledge of 1 connected device exchanged
100	less than 1	Knowledge of 10 connected devices exchanged
1000	1	Knowledge of 100 connected devices exchanged
10000	2	Knowledge of 1000 connected devices exchanged
100000	27	Knowledge of 10000 connected devices exchanged

Table 2 presents the measurements that were obtained when sharing knowledge in a federated network composed of 26 nodes. Until one hundred thousand triples were exchanged between nodes, to help in assessing our approach.

Table 3 Measuring time to discover a connected device

Number of triples contained in the store	Time required to query the store (in seconds)
10	0,0006
100	0,0009
1000	0,0042
10000	0,6959
100000	19,3377

Table 3 presents the measurements that were obtained when discovering connected devices of which semantic descriptions were stored in a triple store. The request consisted of finding connected devices having a specific attribute (e.g. *has Capability* set to "light"). We increased the semantic descriptions managed by the store from ten triples to one hundred thousand.

semantic node and an experimentation based on a simulated building have been realized to assess our concept. Future works include analyzing how the federated architecture behaves when nodes are dynamically added or removed as well as the number of requests that can be processed in parallel, compared to a centralized approach.

Endnotes

^aCosm platform. <https://cosm.com/>

^bNimbits is a service you can use to record and share data on the cloud, <http://www.nimbits.com/index.html>

^cThinkSpeak an open application platform designed to enable meaningful connections between things and people, <https://www.thingspeak.com/>

^dThingWorx enables businesses to rapidly develop applications that connect people, systems, and the intelligent devices, <http://www.thingworx.com>

^eAssuming that persons have an associated profile such as FOAF, <http://www.foaf-project.org/>

^fOWL2 Web Ontology language overview, <http://www.w3.org/TR/owl2-overview/>

^gSWRL A Semantic Web Rule Language combining OWL and RuleML, <http://www.w3.org/Submission/SWRL/>

^hTomcat server, <http://tomcat.apache.org/>

Acknowledgements

This work was carried out within the Seventh Framework Programme under the *IoT-a: Internet of Things Architecture* (<http://www.iota.eu/public>) Integrated Project (IP), and is partly supported by the European Union under contract number: 257521.

Received: 9 August 2012 Accepted: 13 August 2012

Published: 3 October 2012

References

1. Atzori L, Iera A, Morabito G (2010) The Internet of Things: A survey. *Comput Netw* 54(15): 2787–2805. <http://dx.doi.org/10.1016/j.comnet.2010.05.010>
2. Lassila O (2005) Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help. In: in the proceedings of the Semantic Web and Policy Workshop, 4th International Semantic Web Conference
3. Chen H (2003) An Intelligent Broker for Context-Aware Systems. In: *Adjunct Proceedings of UbiComp*, pp 183–184
4. Katsanov A, Kaykova O, Khriyenko O, Nikitin S, Terziyan VY (2008) Smart Semantic Middleware for the Internet of Things. In: Filipe J, Andrade-Cetto J, Ferrier JL (eds) *ICINCO-ICSO*. INSTICC Press, pp 169–178
5. Singh S, Puradkar S, Lee Y (2006) Ubiquitous computing: connecting Pervasive computing through Semantic Web. *Inf Syst. E-Business Management* 4(4): 421–439
6. Pfisterer D, Römer K, Bimschas D, Kleine O, Mietz R, Truong C, Hasemann H, Kröller A, Pagel M, Hauswirth M, Karnstedt M, Leggieri M, Passant A, Richardson R (2011) SPITFIRE: toward a semantic web of things. *IEEE Commun Mag* 49(11): 40–48
7. Dijkstra EW (1971) A short introduction to the art of programming. <http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD316.PDF>
8. Henß J, Kleb J, Grimm S, Bock J (2009) A Database Backend for OWL. In: Hoekstra R, Patel-Schneider PF, <http://ceur-ws.org>: CEUR Workshop Proceedings 2009 (eds) Proceedings of the 6th International Workshop on OWL: Experiences and Directions (OWLED 2009), Volume 529. http://www.webont.org/owlled/2009/papers/owlled2009_submission_3.pdf
9. Horridge M, Bechhofer S (2009) The OWL API: A Java API for Working with OWL 2 Ontologies. In: Hoekstra R, Patel-Schneider PF, *CEUR-WS.org 2009* (eds) OWLED, Volume 529 of CEUR Workshop Proceedings. http://ceur-ws.org/Vol-529/owlled2009_submission_29.pdf
10. Sirin E, Parsia B, Grau B, Kalyanpur A, Katz Y (2007) Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2): 51–53. <http://dx.doi.org/10.1016/j.websem.2007.03.004>
11. De S, Barnaghi P, Bauer M, Meissner S (2011) Service modelling for the Internet of Things. In: *Computer Science and Information Systems (FedCSIS)*, 2011 Federated Conference on, pp 949–955

doi:10.1186/2192-1121-1-8

Cite this article as: Christophe: Federated semantic nodes to scale search process in the IoT. *Communications in Mobile Computing* 2012 **1**:8.