



HAL
open science

A QoC-aware discovery service for the Internet of Things

Porfírio Gomes, Everton Cavalcante, Thais Batista, Chantal Taconet, Sophie Chabridon, Denis Conan, Flavia C. Delicato, Paulo F. Pires

► To cite this version:

Porfírio Gomes, Everton Cavalcante, Thais Batista, Chantal Taconet, Sophie Chabridon, et al.. A QoC-aware discovery service for the Internet of Things. UCAMI 2016 : 10th International Conference on Ubiquitous Computing & Ambient Intelligence, Nov 2016, Las Palmas, Gran Canaria, Spain. pp.344 - 355, 10.1007/978-3-319-48799-1_39 . hal-01405742

HAL Id: hal-01405742

<https://hal.science/hal-01405742>

Submitted on 6 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A QoC-Aware Discovery Service for the Internet of Things

Porfirio Gomes¹, Everton Cavalcante¹, Thais Batista¹,
Chantal Taconet², Sophie Chabridon², Denis Conan²,
Flavia C. Delicato³, Paulo F. Pires³

¹ DIMAp, Federal University of Rio Grande do Norte, Natal, Brazil
porfiriodantas@gmail.com, everton@dimap.ufrn.br, thais@ufrnet.br

² SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Évry, France
{chantal.taconet, sophie.chabridon, denis.conan}@telecom-sudparis.eu

³ PPGI/DCC, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
{fdelicato, paulo.f.pires}@gmail.com

Abstract. The Internet of Things (IoT) is an emergent paradigm characterized by a plethora of smart objects connected to the Internet. An inherent characteristic of IoT is the high heterogeneity and the wide distribution of objects, thereby calling for ways to describe in an unambiguous and machine-interpretable way the resources provided by objects, their properties, and the services they offer. In this context, discovery services play a significant role as they allow clients (middleware platforms, end-users, applications) to retrieve available resources based on appropriate search criteria, such as resource type, capabilities, location, and Quality of Context (QoC) parameters. To cope with these concerns, we introduce *QoDisco*, a QoC-aware discovery service relying on multiple-attribute searches, range queries, and synchronous/asynchronous operations. *QoDisco* also comprises an ontology-based information model for semantically describing resources, services, and QoC-related information. In this paper, we describe the *QoDisco* architecture and information model as well as an evaluation of the search procedure in an urban air pollution monitoring scenario.

1 Introduction

The *Internet of Things* (IoT) [5] is an emergent paradigm that has rapidly evolved in recent years as an extension of the current Internet with seamlessly interconnected physical entities (a.k.a. *things*) towards providing value-added information and functionalities for end-users and/or applications. Devices can be attached to or embedded into these things or be deployed to monitor the surrounding environment. Two types of devices are of particular relevance in IoT, namely *sensors* and *actuators* [16]. Sensors typically obtain data from entities while actuators are capable of actuating on an entity or its environment. Both sensors and actuators comprise *resources* often hosted in these devices, allowing clients (middleware platforms, users, applications) to interact with them

through the Internet. Since implementations of resources can be highly dependent on the underlying device hardware, *services* provide high-level interfaces exposing the functionalities of these resources. Resources can be described in terms of metadata such as resource type, capabilities, location, etc. [7], as well as the information required to interact with the exposed services.

An inherent characteristic of IoT is the high heterogeneity of available resources, a consequence of the widely distributed plethora of smart things with different capabilities/functionalities and network protocols. This makes finding, selecting, and using smart things in a fast, user-friendly way a hard task for both machines and human users [22]. In this context, *discovery services* play a significant role because they enable clients to retrieve unknown available resources based on given search criteria. These mechanisms also allow obtaining up-to-date information about functional capabilities offered by devices in the environment and/or the *observations* obtained by them [23]. Although discovery services represent a well-studied topic in distributed systems, traditional approaches are not suitable to this context mainly due to the high dynamicity of IoT resources.

A discovery process in the IoT can highly benefit from considering *Quality of Context* (QoC) information associated with data provided by devices, e.g. precision, accuracy, up-to-dateness, etc. [9] Using such an additional information is relevant because it can augment the richness of the discovery procedure and better support decision-making actions, besides dealing with the inherently imperfect, inconsistent, and noisy nature of data provided by devices [19]. Therefore, QoC arises as way of filtering out these data and ranking which information should be used [10].

Aiming to tackle the aforementioned concerns, this paper presents *QoDisco*, a QoC-aware discovery service structured upon a set of repositories storing resource descriptions and QoC-related information. *QoDisco* complies with important requirements for discovering resources in the IoT context, e.g. multi-attribute and range queries [23], and support to both synchronous and asynchronous searches. Furthermore, our proposal encompasses an ontology-based information model with concepts related to resources and services, relationships among them, and QoC information. This information model takes advantage of well-established, standardized ontologies to semantically describe resources and services, namely the Semantic Actuator Network (SAN) ontology [28], the Semantic Sensor Network (SSN) ontology [6], and the Ontology Web Language for Services (OWL-S) [21]. In addition, we have incorporated part of the Quality of Context Information Meta-model (QoCIM) [20] to our information model aiming at dealing with QoC criteria in the search procedure.

The remainder of this paper is organized as follows. Section 2 elicits important requirements to be addressed by a discovery service in IoT. Section 3 describes *QoDisco*, its architecture, and the information model used to describe resources, services, and QoC information. Section 4 presents an evaluation of the search procedure performed by *QoDisco* in an urban pollution monitoring scenario. Section 5 briefly discusses related works. Finally, Section 6 contains some concluding remarks and directions for future work.

2 Requirements for a Discovery Service in IoT

There are some important concerns to be considered by discovery services targeting IoT. Paganelli and Parlanti [23] highlight five of these requirements, in particular the identification of resources, the ability to perform multi-attribute and range queries, dealing with the existence of multiple providers for publishing information about resources, and APIs for managing resources. In addition to those requirements, we have considered three other important concerns, namely (i) semantic description of resources, (ii) modeling of QoC criteria, and (iii) support for both synchronous and asynchronous interactions. Table 1 outlines these requirements.

Table 1. Important requirements for a discovery service in IoT

Requirement	Description
Flexible identification schema	Transparency regarding the adopted identification schema (e.g. URIs, IP addresses, etc.) used to address resources, services, and observation data
Multi-attribute query	Capability of finding resources by performing queries based on an exact matching of one or more attributes
Range query	Capability of finding resources based on queries specifying bounds on single or multiple attributes
Dealing with multiple publishers	Several entities can produce and publish information about a given resource
Management APIs	Capability of managing information associated with a given resource
Semantic description of resources and services	Use of ontologies to semantically model resources and exposed services
Modeling QoC criteria	Modeling QoC indicators associated with context data
Synchronous and asynchronous interactions	Capability of handling both synchronous and asynchronous interactions with clients

Chun et al. [12] argue that an information model should be semantically modeled in terms of ontologies defining common terms used in IoT to model resources and exposed services. Adding semantics to an information model can promote interoperability among resources, information models, data providers, and service consumers, as well as it can make data access, service integration, semantic interpretation, and knowledge extraction easier. In addition, we argue that the imperfect and uncertain nature of data in IoT requires considering associated QoC indicators.

Due to the dynamicity of resources, a discovery service in IoT should also be able to handle both synchronous and asynchronous interactions. When receiving

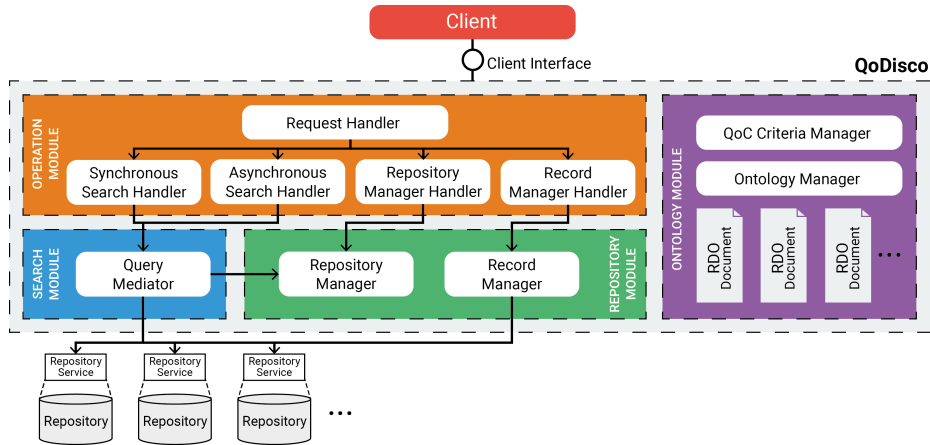


Fig. 1. *QoDisco* architecture

a synchronous request, the discovery service sends a unique response to the client. When registering asynchronous searches, clients of the discovery service are notified in case of removal, insertion or update of resources.

3 *QoDisco*

This section presents *QoDisco*, our QoC-aware discovery service for IoT. Section 3.1 details the *QoDisco* architecture whereas Section 3.2 describes the information model used to describe IoT resources, services, and QoC information.

3.1 Architecture

QoDisco relies on a collection of repositories to perform data and service discovery tasks. This approach provides several advantages over a single repository approach, in particular high scalability, fault-tolerance, independent control of repositories, and data distribution among repositories [27]. In the context of this work, we envision clients such as IoT middleware platforms, applications, and users storing resource descriptions (a.k.a. records) in repositories and registering them in *QoDisco*. When receiving a discovery request, *QoDisco* performs a search for resource descriptions and/or data stored in the available repositories.

The *QoDisco* architecture comprises four modules, each one responsible for a set of functionalities provided by the platform. Fig. 1 illustrates these modules, which are briefly described in the following. The current prototype of *QoDisco* is available at <http://consiste.dimap.ufrn.br/projects/qodisco>.

The *Ontology Module* consists of repository domain ontology (RDO) documents, which describe the concepts composing the information model of *QoDisco*. In this module, the *Ontology Manager* offers operations for adding, removing, and modifying RDO documents, while the *QoC Criteria Manager* is responsible

for adding, removing, and modifying QoC criteria (also defined into RDO documents). New RDO documents can be defined by extending available ones in order to encompass concepts used in a particular domain, and RDO documents describing QoC criteria follow the concepts defined in the *QoDisco* information model. A given QoC criterion represents an attribute that qualifies a context information (e.g. accuracy, up-to-dateness, etc.) and it can be calculated in different ways according to the situation in which it is used.

The main functionality of the *Repository Module* is to manage repositories. In this module, the *Repository Manager* is responsible for managing repositories and mapping each one to a (group of) specific RDO documents in the *Ontology Module*. This implies that a repository joining *QoDisco* has to be mapped to at least one domain specified by a RDO document. In turn, the *Record Manager* is responsible for adding, removing, and modifying records in repositories.

The *Search Module* encompasses the *Query Mediator*, which forwards queries to the repositories belonging to *QoDisco*. To perform the search, this component specifies an RDO document name (i.e. domain name) to the *Repository Manager*, which provides the IP addresses and port numbers of the repositories mapped to such a domain. For instance, if the *Query Mediator* makes a request for repositories belonging to the health care domain, the *Repository Manager* returns the respective addresses and port numbers of these repositories.

The *Operation Module* is composed of the five following components:

Request Handler: It is the entry point of the *Operation Module*, and receives both synchronous and asynchronous requests for searching and managing repositories and records.

Synchronous Search Handler: It performs synchronous queries by querying the *Query Mediator*.

Asynchronous Search Handler: Using the publish-subscribe paradigm [13], it notifies clients (i) about the discovery of a new resource matching the search query and (ii) about the modification or removal of a resource description.

Repository Manager Handler: It interacts with the *Repository Manager* to add or remove repositories.

Record Manager Handler: It interacts with the *Repository Manager* to add, remove or modify records.

The functionalities of the *Operation Module* and the *Ontology Module* are provided through the *Client Interface*, which complies with the Representational State Transfer (REST) architectural style [15]. The operations of the *Client Interface* are related to both synchronous and asynchronous search, as well as to the management (insertion, removal, update) of records, repositories, RDO documents, and QoC criteria. More information about the *Client Interface* is available at <http://consiste.dimap.ufrn.br/projects/qodisco>.

3.2 Information Model

Fig. 2 depicts the information model adopted by *QoDisco*. As previously mentioned, this information model comprises an ontology-based vocabulary of con-

cepts related to IoT resources and services, relationships among them, and QoC-related information. In order to semantically describe resources, the *QoDisco* information model takes advantage of the SAN ontology [28], an extension of the W3C’s SSN ontology [6] that provides concepts, attributes, and properties to model both sensors and actuators. Additionally, we have incorporated part of the SOUPA ontology [11] in order to include location-related concepts to describe spatial locations of entities (*PhysicalStructure*) in terms of latitude, longitude, altitude, distance, and surface, as well as symbolic representations of space and spatial relationships, e.g. a room situated at a given floor of a building. In our information model, location-related information can be associated with both resources and context data (observations).

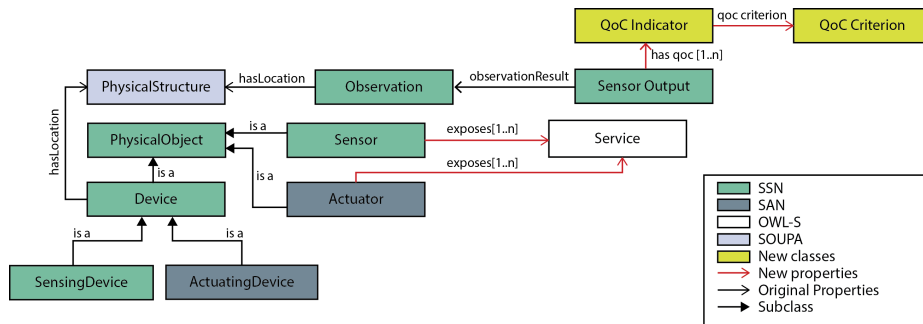


Fig. 2. Information model adopted by *QoDisco*

In spite of their comprehensiveness for describing sensors and actuators, both SSN and SAN ontologies do not provide elements for modeling services exposed by resources. For this reason, the *QoDisco* information model also comprises the OWL-S ontology [21], which has been used to semantically model Web services. In OWL-S, a service is described in terms of three sub-ontologies, namely *ServiceProfile*, *ServiceModel*, and *ServiceGrounding*. A *ServiceProfile* describes a service in terms of its inputs, outputs, preconditions, and effects. A *ServiceModel* describes the operations performed by a service and allows invoking, composing, and monitoring it. A *ServiceGrounding* describes how a service is invoked and provides information about data types used in its operations. In order to represent the relationship between resources and services, we have created the *exposes* property, which links the *Sensor* and *Actuator* concepts (representing resources) to the *Service* concept (representing services), as shown in Fig. 2.

To cope with QoC-related concerns, we have incorporated part of the QoCIM meta-model [20] by describing its concepts as an ontology. The *QoC Criterion* concept can be used to represent a QoC parameter associated with context data, e.g. accuracy, precision, completeness, up-to-dateness. In turn, the *QoC Indicator* expresses the QoC level of an observation (measured data) made by a resource. In the *QoDisco* information model, the *has_qoc* property associates a QoC level (*QoC Indicator* concept) to an observation made by a resource (*SensorOutput*

concept) while the *qoc_criterion* property indicates that a QoC indicator must be associated with a QoC criterion. Our information model follows the same flexible ideology of QoCIM, i.e. defining a basis to design and represent any QoC criterion instead of providing a predefined list of supported QoC criteria. A given QoC criterion can be built upon other primitive or composed QoC criteria.

4 Evaluation

This section presents an evaluation of the scalability of *QoDisco* by assessing the performance of the search for resources and observations. Such an assessment was carried out to evaluate both synchronous and asynchronous search processes. Section 4.1 describes the scenario considered in our evaluation. Section 4.2 outlines the configuration used to perform the evaluation. Finally, Section 4.3 shows the results of the computational experiments.

4.1 Scenario

We consider a scenario related to air pollution monitoring in a urban area where public buses are equipped with air pollution monitoring sensors and a GPS receiver. In addition, bus stations are equipped with more sophisticated air pollution monitoring sensors capable of providing data with better QoC. The air pollution monitoring sensors are responsible for measuring air pollutants such as ozone, carbon monoxide, sulfur dioxide, and nitrous oxide. Four QoC criteria are used to qualify pollution measurements as well as the GPS location: (i) *margin of error*, which quantifies doubt about the result of a measurement and is expressed in the same unit of the measurement; (ii) *freshness*, the time elapsed from the collect of observations and its delivery to a requester; (iii) *precision*, which qualifies how close or how repeatable the results from a measurement are and is typically expressed as a percentage; and (iv) *spatial resolution*, which measures the precision with which a physical area is expressed.

The location of the buses and bus stations is qualified with spatial resolution. Pollution measurements provided by buses are qualified with both precision and freshness, while the ones provided by sensors are qualified with margin of error and freshness. In this scenario, *QoDisco* can be used by clients to search for and select sensors and respective measurements according to their capabilities, location, QoC indicator parameters, etc. All of this information is described in conformance with the *QoDisco* information model.

4.2 Experimental Setup

Aiming at minimizing network influence, all experiments were performed using a single computational node with Intel® Core™ i5 processor, 8 GB of RAM, and Mac OS 10.11.3 64-bit operating system. This computational node was used to deploy the *QoDisco* repositories in a virtual machine with 1 GB of RAM and running Linux Ubuntu 14.04.3 Server OS 64-bit. Each repository was deployed

at a Linux container using the Docker technology [2]. In order to ensure the reliability of the results, the experiments were run 20 times.

To discover resources and observations by using their location, type, or any of their properties in *QoDisco*, queries can be created using the SPARQL Protocol and RDF Query Language [4], a W3C standard for performing semantic searches, or using the Web interface of the platform. In all experiments, we have calculated the time spent by *QoDisco* for a SPARQL query² that searches for all the observations related with ozone pollution levels registered in *QoDisco* with a QoC indicator of precision greater than or equal to 95%.

```
SELECT * WHERE {
?observation a ssn:Observation ;
  ssn:observedProperty pol:Ozone ;
  ssn:observationResult ?observationResult .
?observationResult qodisco:has_qoc ?qoc .
?qoc qodisco:has_qoc_criterion context:Precision .
?qoc qodisco:has_qoc_value ?value
FILTER (?value >= 95)
}
```

This query returns a response in the XML format containing (i) an URI to the observation itself, (ii) an URI referring to the sensor output, (iii) a reference to the QoC indicator, and (iv) the QoC value regarding the observation. It is worth mentioning that additional attributes could be considered in addition to the ozone pollution levels by adding other observed properties (`observedProperty` element). Other QoC criteria can be also considered by including logical operators in the `FILTER` clause of the SPARQL query.

4.3 Results

Synchronous search. To assess the performance of *QoDisco* when performing a synchronous search, we have inserted a set of dummy observations on ozone pollution level in five repositories associated to the air pollution domain. Next, we have measured the time spent by *QoDisco* to respond to a client query as the number of observations increases. Fig. 3 (left) shows the average time (in milliseconds) to respond to the synchronous search query with an increasing number of observations. Such a response time is directly influenced by the sequential search performed by *QoDisco* on the available repositories. Improvements might be achieved by parallelizing the search procedure when it is performed on multiple repositories. Nonetheless, different SPARQL queries could influence the obtained assessments.

Asynchronous search. To assess the performance of *QoDisco* when performing an asynchronous search, we have considered five repositories associated to the air pollution domain, each one with 1,000 dummy observations on ozone

² Due to space constraints, prefixes referring to the *QoDisco* ontologies were omitted.

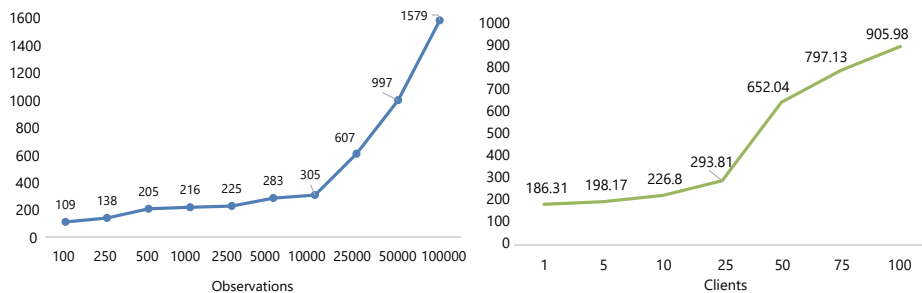


Fig. 3. Average response time to respond to synchronous (left) and asynchronous (right) searches

pollution level. Next, we have measured the time spent by *QoDisco* to respond to searches performed by an increasing number of clients interested on notifications upon new ozone pollution level measures with precision greater than or equal to 95%. Such simultaneous requests were simulated by making use of Apache JMeter [1], a Java-based open-source tool for analyzing and measuring the performance of Web services. Fig. 3 (right) shows the average time (in milliseconds) to notify clients upon an asynchronous search, with an increasing number of client requests. It is important to highlight that such a response time highly depends on the publish-subscribe broker responsible for the management of the exchanged messages. In the current prototype of *QoDisco*, we use Mosquitto [3], a lightweight publish-subscribe message broker service for IoT. *QoDisco* only sends a message to Mosquitto, which handles the requests and responses from/to clients (subscribers) interested on the notifications.

5 Related Work

Bellavista et al. [8] review context distribution solutions for ubiquitous computing and investigate in particular how they perform context source selection. Some solutions take into account QoC, but with some restrictions. COSINE [17] and CMF [29] perform automatic context source selection depending on the QoC ensured by context sources, but selection criteria are limited and are not reconfigurable which limits their dynamicity. MiddleWhere [26] adopts a dynamic approach and let services apply QoC constraints on the received context data, but it is dedicated to location data.

Focusing on IoT, Perera et al. [25] analyze fifty context-aware projects and identifies context discovery as one of the six challenges where new solutions are required. Context discovery encompasses understanding sensor data produced by context sources, and relating them to high-level context information automatically. This might require to annotate context data, but in an automated way in order to tackle the scale and the dynamicity of the IoT. According to the authors, semantic technology shows promising ways to encode domain knowledge. In this direction, not much effort has been dedicated to resource discovery based on

QoC-related information. One of the few existing proposals is CASSARAM [24], a framework that uses sensor characteristics and context information such as reliability, accuracy, and battery life registered at a single, centralized registry to automatically search for, select, and rank appropriate sensors among a large set of available devices according to user-defined criteria and priorities. Similarly to *QoDisco*, CASSARAM is built upon an ontology-based information model comprising the SSN ontology and the SPARQL language is used in the search process. In turn, the discovery mechanism of the LOCA framework [18] relies on a distributed peer-to-peer architecture and considers context information modeled by means of a lightweight and completely novel ontology.

An important aspect to be considered with respect to the aforementioned works regards the use of centralized and decentralized approaches. A significant limitation of CASSARAM is the use of a single centralized registry, which is not suitable for IoT due to well-known drawbacks related to scalability, fault-tolerance, and security threats. As stated in the comparative study conducted by Evdokimov et al. [14], decentralized architectures such as the ones used in LOCA are more suitable to IoT since they well fulfill important requirements such as fault-tolerance and scalability. *QoDisco* benefits from these requirements as it is structured upon a collection of repositories, each one mapped to one or more domains. Despite these works take into account context information, they do not consider in addition QoC criteria in the discovery process, which is valuable towards providing more accurate results according to application/user requirements. Indeed, we have not found any proposal in the literature addressing the use of QoC criteria for discovery services in IoT, thereby constituting a significant contribution of our work.

6 Conclusion

The discovery of suitable resources among the plethora of existing available entities connected to the Internet poses an important challenge to the development of IoT systems as these resources are highly dynamic and heterogeneous. Therefore, it is important to provide clients (e.g. middleware platforms, end-users, and applications) with means of easily retrieving available resources and describing them in an unambiguous way. Due to the imperfect nature of data provided by IoT devices, it is valuable complementing observations with QoC criteria to augment the richness of the search procedure and the support of decision-making.

In this paper, we address these concerns by proposing *QoDisco*, a QoC-aware discovery service for IoT. *QoDisco* supports resource discovery based on multiple attributes, range queries, synchronous/asynchronous interactions, and QoC criteria. Additionally, *QoDisco* encompasses an information model that takes advantage of well-established, standardized ontologies to semantically describe resources (sensors and actuators), the services that they expose, and QoC-related information. Unlike the approaches commonly found in the literature, the collection of repositories of *QoDisco* allows coping with important requirements in the IoT scenario. Moreover, the definition of a consistent information model is

highly relevant as semantic annotations can be used to describe resources and specify search queries for discovering resources. Obtained experimental results have shown that *QoDisco* is able to handle discovery requests quite efficiently, even with a significant number of registered data and simultaneous clients.

This work can be continued in several ways. As ongoing work, we are carrying out additional experiments considering network environments closer to the ones observed in IoT, with low bandwidth, delays, package losses, etc. We also need to consider other important facets in discovery services for IoT, such as dynamism and security/privacy concerns regarding resources and their metadata. Finally, we intend to implement strategies aimed to reduce response times, in particular by parallelizing the search on repositories (instead of the current sequential search) and using optimized semantic search algorithms.

References

1. Apache JMeter. <http://jmeter.apache.org/>
2. Docker. <https://www.docker.com/>
3. Mosquitto: An open source MQTT Broker. <http://mosquitto.org/>
4. SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>
5. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. *Computer Networks* 54(15), 2787–2805 (Oct 2010)
6. Barnaghi, P., et al.: Semantic Sensor Network XG Final Report. <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/> (Jun 2011)
7. Bassi, A., et al. (eds.): Enabling things to talk: Designing IoT solutions with the IoT Architectural Reference Model. Springer Berlin Heidelberg, Germany (2013)
8. Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys* 44(4) (Sep 2012)
9. Buchholz, T., Küpper, A., Schiffers, M.: Quality of Context: What it is and why we need it. In: Proceedings of the 10th International Workshop of the OpenView University Association (2003)
10. Chabridon, S., Laborde, R., Desprats, T., Oglaza, A., Marie, P., Marquez, S.M.: A survey on addressing privacy together with quality of context for context management in the Internet of Things. *Annals of Telecommunications* 69(1), 47–62 (Feb 2014)
11. Chen, H., Finin, T., Joshi, A.: The SOUPA ontology for Pervasive Computing. In: Tamma, V., Cranefield, S., Finin, T.W., Willmott, S. (eds.) *Ontologies for agents: Theory and experiences*, pp. 233–258. *Whitestein Series in Software Agent Technologies*, Birkhäuser Basel, Switzerland (2005)
12. Chun, S., Seo, S., Oh, B., Lee, K.H.: Semantic description, discovery and integration for the Internet of Things. In: Proceedings of the 2015 IEEE International Conference on Semantic Computing. pp. 272–275. IEEE, USA (2015)
13. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. *ACM Computing Surveys* 35(2), 114–131 (Jun 2003)
14. Evdokimov, S., Fabian, B., Kunz, S., Schoenemann, N.: Comparison of discovery service architectures for the Internet of Things. In: Proceedings of the 2010 IEEE International Sensor Networks, Ubiquitous, and Trustworthy Computing. pp. 237–244. IEEE, USA (2010)

15. Fielding, R.T.: Architectural styles and the design of network-based software architectures. PhD Dissertation, University of California-Irvine, USA (2000)
16. Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., Boyle, D.: From machine-to-machine to the Internet of Things: Introduction to a new age of intelligence. Academic Press, Oxford, United Kingdom (2014)
17. Juszczak, L., Psaiar, H., Manzoor, A., Dustdar, S.: Adaptive query routing on distributed context - The COSINE Framework. In: Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware. pp. 588–593. IEEE Computer Society, USA (2009)
18. Li, J., Zaman, N., Li, H.: A decentralized locality-preserving context-aware service discovery framework for the Internet of Things. In: Proceedings of the 2015 IEEE International Conference on Services Computing. pp. 317–323. IEEE Computer Society, USA (2015)
19. Marie, P., Desprats, T., Chabridon, S., Sibilla, M.: Extending Ambient Intelligence to the Internet of Things: New challenges for QoC management. In: Hervás, R., Lee, S., Nugent, C., Bravo, J. (eds.) Proceedings of the 8th International Conference on Ubiquitous Computing and Ambient Intelligence, Lecture Notes in Computer Science, vol. 8867, pp. 224–231. Springer International Publishing, Switzerland (2014)
20. Marie, P., Desprats, T., Chabridon, S., Sibilla, M.: The QoCIM Framework: Concepts and tools for Quality of Context management. In: Brézillon, P., Gonzalez, A.J. (eds.) Context in Computing: A cross-disciplinary approach for modeling the real world, pp. 155–172. Springer New York, USA (2014)
21. Martin, D., et al.: Bringing semantics to Web services: The OWL-S approach. In: Cardoso, J., Sheth, A. (eds.) Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, Lecture Notes in Computer Science, vol. 3387, pp. 26–42. Springer Berlin Heidelberg, Germany (2005)
22. Mayer, S., Guinard, D.: An extensible discovery service for smart things. In: Proceedings of the Second International Workshop on Web of Things. ACM, USA (2011)
23. Paganelli, F., Parlanti, D.: A DHT-based discovery service for the Internet of Things. *Journal of Computer Networks and Communications* 2012, 1–11 (2012)
24. Perera, C., Zaslavsky, A., Christen, P., Compton, M., Georgakopoulos, D.: Context-aware sensor search, selection and ranking model for Internet of Things middleware. In: Proceedings of the 14th IEEE International Conference on Mobile Data Management (2013)
25. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials* 16(1), 414–454 (2014)
26. Ranganathan, A., Al-Muhtadi, J., Chetan, S., Campbell, R., Mickunas, M.D.: MiddleWhere: A middleware for location awareness in Ubiquitous Computing applications. In: Jacobsen, H.A. (ed.) Proceedings of the 2004 ACM/IFIP/USENIX International Middleware Conference, Lecture Notes in Computer Science, vol. 3231, pp. 397–416. Springer Berlin Heidelberg, Germany (2004)
27. Schmidt, C., Parashar, M.: A peer-to-peer approach to Web service discovery. *World Wide Web* 7(2), 211–229 (Jun 2004)
28. Spalazzi, L., Taccari, G., Bernardini, A.: An Internet of Things ontology for earthquake emergency evaluation and response. In: Proceedings of the 2014 International Conference on Collaboration Technologies and Systems. pp. 528–534. IEEE, USA (2014)

29. van Kranenburg, H., Bargh, M.S., Iacob, S., Peddemors, A.: A context management framework for supporting context-aware distributed applications. *IEEE Communications Magazine* 44(8), 67–74 (Aug 2006)