Context-based Framework to Discovery, Search, and Selection of Computing Devices in the Internet of Things

Everton de Matos[1], **Willian T. Lunardi**[1], **Leonardo Amaral**[1],
Ramão Tiburski[1], **Fabiano Hessel**[1], **Sabrina Marczak**[1]

[1]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

{everton.matos.001, willian.lunardi, leonardo.amaral}@acad.pucrs.br,

ramao.tiburski@acad.pucrs.br,{fabiano.hessel, sabrina.marczak}@pucrs.br

Abstract. *With the development and proliferation of the Internet of Things (IoT), many application areas have started to make use of this new computing paradigm. Moreover, the number of active computing devices are growing at a rapid pace around the world. Consequently, a mechanism able to manage this different devices has become necessary. IoT middleware solutions have been developed to supply this need. However, device discovery, search and selection remain a critical challenge. In this paper we present COBASEN, a work-in-progress framework that has been designed to address the challenges regarding the discovery, search and selection of devices when large number of devices with overlapping and sometimes redundant functionality are available.*

1. Introduction

The term Internet of Things (IoT) was coined in 1998 [Kevin 2009] and defined as the computing paradigm that allows people and things (computing devices) to be connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service [Perera et al. 2014]. Connectivity between devices is a critical functionality that is required to comply with the vision of the IoT. In this sense, there are current market statistics and predictions that show rapid growth in device deployments related to IoT environments.

The IoT vision is in everywhere and can be applied in many areas. Therefore, it is fundamental in the Brazilian scenario to leverage solutions related to the major computing challenges of the country. Looking ahead, with the evolution of technology, the amount of IoT devices will only grow. By 2020, it is estimated that there will be 50 to 100 billion devices connected to the Internet [Sundmaeker et al. 2010]. In this sense, manually selecting and configuration of these devices together with the choice of what devices are relevant to users and applications are considered a required task [Perera et al. 2012]. The challenge in this work is to develop a system that can search middleware devices based on their context characteristics and provide to the user an easy and transparent method of discover these devices.

In this paper, we present COBASEN, a work-in-progress framework that has been developed to facilitate the discovery, search, and selection of devices in IoT environments. COBASEN allows to capture the users needs through a query to discover and select devices. Depending on user preferences, and to aggregate devices, in order to facilitate the use of these devices.

The remainder of this paper is organized as follows: In Section 2 we present some background definitions and describe COMPaaS middleware architecture. Our proposed solution, COBASEN, is presented in detail in Section 3. In Section 4 we provide implementation details. Related Work are presented in Section 5. Finally, we present conclusions and prospects for future work in Section 6.

2. IoT Middleware and Context of Things

An IoT ecosystem usually is based on a layered architecture style and use this view to abstract the integration of objects/things and to provide services solutions to applications. Middleware system is a software layer interposed between the infrastructure of devices and applications, and is responsible for providing services according to devices functionality. In a prior work we have described COMPaaS (Cooperative Middleware Platform as a Service) [Amaral et al. 2015], an IoT middleware composed of three main systems: API, Middleware and Logical Device. API is the system that has the methods to be used by applications that want to use the middleware services. Middleware is the system responsible for abstracting the interactions between applications and devices and hides all the complexity involved in these activities. Logical Device is the system responsible for hiding all the complexity of physical devices and abstracts their functionalities. Although COMPaaS have all these features, it does not have a context-aware functionality to show its devices directory, which could assist in the process of selection, data fusion, and management in COMPaaS.

Context is considered any information that can be used to characterize the situation of an entity. Entity is a person, place, or object/thing that is relevant to the interaction between an user and an application, including the user and the application themselves[Abowd and Mynatt 2000]. Context information about devices/things needs to be acquired and stored with annotations that will make easy to retrieve it later. Abowd and Mynatt [Abowd and Mynatt 2000] identified the five W's (Who, What, Where, When, and Why) as the minimum annotations that are necessary to understand context. Once context can be obtained through the devices, it is possible to create a semantic version of devices information and characteristics, which can be useful to guide users and even systems in the identification and management of relevant devices.

3. COBASEN Framework

Our goal with this work is to allow users to discover, search, select and aggregate devices that best suit their application requirements, making the middleware patterns transparent to the user. COBASEN (Context-based Search Engine) is a framework which uses context to allows user to search and select devices. Moreover, COBASEN is able to auto generate the specification of aggregation of the desired devices based on a set of parameters, submit the specification to the middleware, and send a feedback to user.

COBASEN framework can be divided in two components, Context Module and Search Engine (Figure 1 (A)). Context Module is responsible for acquiring the device context from the middleware and sending it to the Search Engine. The Search Engine is responsible for indexing the obtained device context and answering queries using the index. The steps proposed for COBASEN are presented in Figure 1 (B). The framework allows users to search and select devices according to users needs (steps 1 and 2). After the aggregation (selecting a set of devices), the user fills in a specification form (step 3). Next, the Search Engine will be responsible for creating the aggregation specification according to the middleware standards and submit it to the middleware (steps 4 and 5).

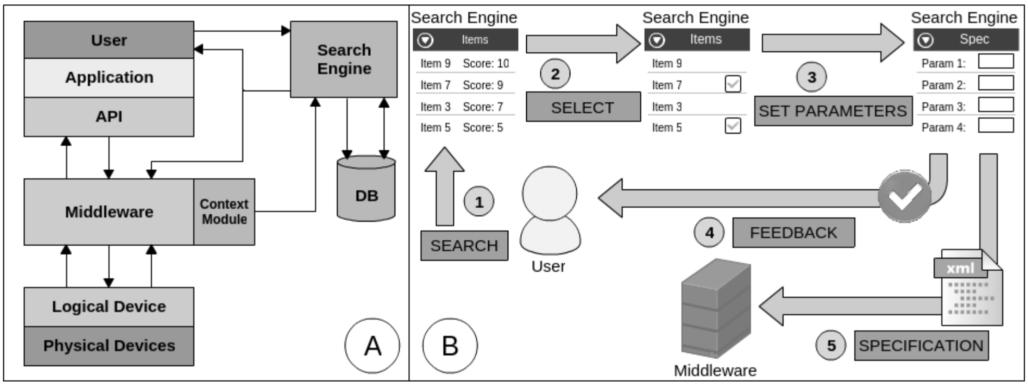


Figure 1. (A) shows the modules of COBASEN interacting with COMPaaS. (B) shows an overview of COBASEN framework.

3.1. Context Module

The objective of the Context Module is to assemble a structure of information with data of each device and pass this information to the Search Engine, allowing the engine be aware of devices context. The Context Module is connected to the middleware (Figure 1 (A)), having access to some features of the middleware, such as the repository of registered devices. The way that information is modeled can delimit the use of techniques of filtering and indexing. For example, we can opt to define separately only the essential information (URI, name) to control register and updates of devices, and concatenate the other context information into a single string. Another strategy could be to keep all the information as the original structure.

The first step to start the contextualization is to obtain the information of the devices. Context Module monitors the middleware, and each time a device is registered or updated, the Context Module updates the Search Engine device list. The acquisition of the Context Module is made by accessing the information repository. Following a pre-established pattern, the Context Module mounts a file in XML format with the information of the devices, following the standard of the five W's [Abowd and Mynatt 2000]. Finally, this XML file is sent to Search Engine that will index the device in a database with its information.

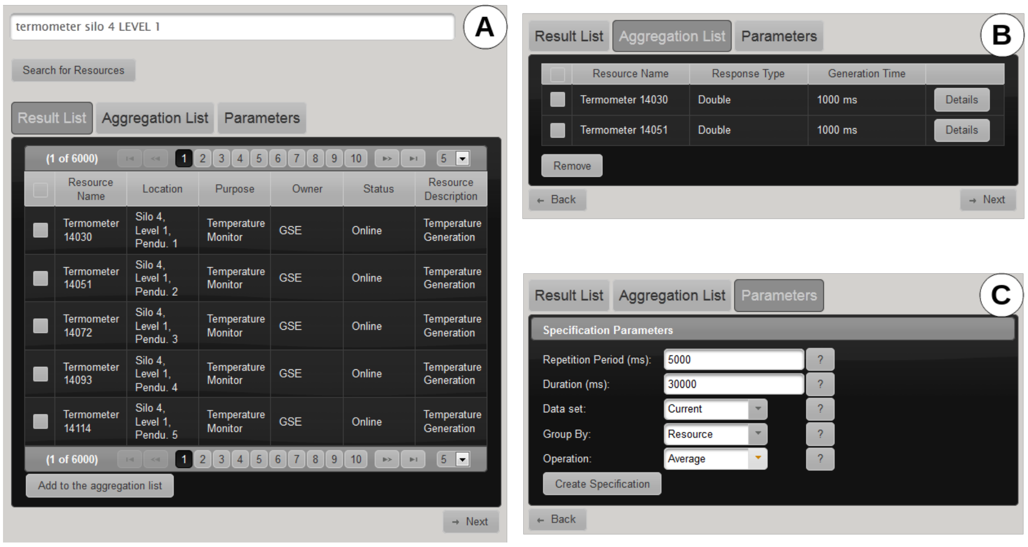
3.2. Search Engine

Search Engine is a system that aims to create an index of objects and use this index to respond queries from users. An index is a data structure that makes it efficient to retrieve objects given the value of one or more elements of the objects [Rajaraman and Ullman 2011]. The most common situation is where the objects are records, and the index is on one of the fields of that record (e.g. the records are book and the field is his title).

In COBASEN, the Search Engine is responsible for enable the search of devices through a query with the utilization of analyzer functions, enable selection of one or more devices (aggregation), generate the specification of the respective aggregation created by the user, send the specification to middleware and send the feedback to user.

4. Implementation and Discussion

The aim of this section is to present a brief analysis of the performance of COBASEN applied in different scenarios which we developed based on real world requirements. The idea is to test our system in ease of use (Figure 2) and performance Figure 3.



termometer silo 4 LEVEL 1

□	Resource Name	Location	Purpose	Owner	Status	Resource Description
□	Termometer 14030	Silo 4, Level 1, Pendu. 1	Temperature Monitor	GSE	Online	Temperature Generation
□	Termometer 14051	Silo 4, Level 1, Pendu. 2	Temperature Monitor	GSE	Online	Temperature Generation
□	Termometer 14072	Silo 4, Level 1, Pendu. 3	Temperature Monitor	GSE	Online	Temperature Generation
□	Termometer 14093	Silo 4, Level 1, Pendu. 4	Temperature Monitor	GSE	Online	Temperature Generation
□	Termometer 14114	Silo 4, Level 1, Pendu. 5	Temperature Monitor	GSE	Online	Temperature Generation

□	Resource Name	Response Type	Generation Time	
□	Termometer 14030	Double	1000 ms	Details
□	Termometer 14051	Double	1000 ms	Details

Specification Parameters

Repetition Period (ms): 5000
Duration (ms): 30000
Data set: Current
Group By: Resource
Operation: Average

Figure 2. Interfaces of the prototype tool

The proposed framework was analyzed and evaluated using a prototype which we developed using Java. The data was stored in PostgreSQL database. Figure 2 represents the interface that the end user will interact to use COBASEN. The steps to interact with COBASEN are: (A) search and selection of devices, (B) aggregation of devices, and (C) creation of the specification that will be sent to the middleware.

We used a computer with AMD FX-8350 (8 core) 4.0GHz and 16GB RAM to evaluate our proposed framework. Since we use a search engine to manage device context and related data, we employed the open source Apache Lucene API [Apache Software Fundation] to index data. We conducted each experiment 50 times and averages were considered. We generate a synthetic data set which comprises device descriptions and context information (e.g. name, URI, location) for one million of devices.

The intent of the experiment show in Figure 3 was to evaluate COBASEN performance. First, in Figure 3 (A), we testing how much time the Search Engine take to search for devices based on user request, and return a feedback to user with found devices. Figure 3 (B) shows how much time COBASEN take to make the indexing process of devices.

In Figure 3 (A), the query processing time is acceptable related to the fact that we are not using pagination (restrict the number of returned devices per page). Even in an extreme case where an environment that has 1 million of devices and a user who makes a query that return up to 20000 results, the processing time did not exceed 500

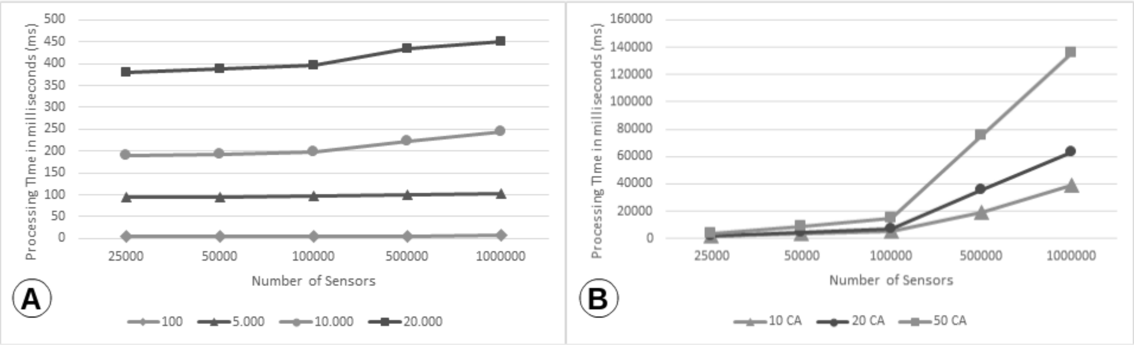


Figure 3. (A) shows processing time taken when the number of devices and the number of results per page gets increased, (B) shows processing time taken by indexing process when number of context characteristics and numbers of device get increased.

milliseconds. As depicted in Figure 3 (B), reducing the number of context characteristics allows to performs COBASEN faster. Processing time starts to get increased after 100000 devices.

Inevitably, in the future, applications for IoT will be presented in several areas related to major computing challenges in Brazil, for example, education, health, finance, energy/oil/gas and mobility. COBASEN presents itself as a great tool in improving the IoT applications development. In IoT environments is very unusual to come across the fact that those who implant middleware are not who develop the applications. The deployment of a middleware in IoT environments that has characteristics with our framework enables users to develop applications for those environments, without having to learn the middleware patterns and/or understand the devices modeling, in an efficient way.

5. Related Work

An IoT directory must support search in order to discover heterogeneous objects [Perera et al. 2014]. Search enables the retrieval of derived information based on syntactic, semantic and structural information contained in data. COBASEN is a context-based search engine attached in the COMPaaS IoT middleware that makes possible the search in order to discovery and selection of devices.

In terms of IoT middleware solutions to provide searching functionalities, Linked Sensor Middleware (LSM) [Digital Enterprise Research Institute] provides limited functionalities such as selecting devices based on location. Nevertheless, all the searching need to be done using SPARQL query language which is not intuitive. Global Sensor Network (GSN) [GSN Team] is another IoT middleware. In short, GSN lists all devices available in a combo-box which is used by the user to select the desired device. Xively [LogMeIn] is another approach, but also offers only keyword search.

As can be seen in the related work, they do not offer many features that facilitate the search for devices. On the other hand, the work being performed with the COBASEN aims to mitigate this gap through contextualization of devices, fast indexing methods and semantic search for devices.

6. Conclusions and Future Work

With the advances in IoT devices technology will increase the number of devices available around us. This means that we will have access to multiple devices with similar characteristics. The problem is that user will need to decide what device are more important than others with similar characteristics. Trying to fill this gap we present COBASEN, that allows middleware users who are not aware of the domain of devices to be able to discovery, search, and select devices of an IoT middleware to develop their applications. Our experimental results demonstrated that the proposed framework is efficient in help users to the development of applications to IoT environments.

In future, we plan to improving both Context Module and the Search Engine. In the Context Module we will explore techniques to analyze and categorize devices. A challenge that we plan to address is to have a lightweight reasoning method in the Context Module. In the Search Engine we will explore the possibilities of developing functions to improve the querying (e.g. pagination, faceting, and filtering) and the security (authentication and access control). The pagination technique will improve the search engine response time.

References

Abowd, G. D. and Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58.

Amaral, L., Tiburski, R., Matos, E., and Hessel, F. (2015). Cooperative middleware platform as a service for internet of things applications. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (to be published)*, SAC '15. ACM.

Apache Software Fundation. Apache Lucene. `https://lucene.apache.org/`. Accessed: 2015-04-04.

Digital Enterprise Research Institute. Linked sensor middleware (lsm). `http://lsm.deri.ie/`. Accessed: 2015-03-26.

GSN Team. Global Sensor Network. `http://sourceforge.net/apps/trac/gsn/`. Accessed: 2015-03-11.

Kevin, A. (2009). That 'internet of things' thing, in the real world things matter more than ideas. *RFID Journal*, 22.

LogMeIn. Xively. `http://xively.com/`. Accessed: 2015-04-10.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2012). Ca4iot: Context awareness for internet of things. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 775–782. IEEE.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):414–454.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things.