# Multi-Dimensional Contexts for Querying IoT Networks

Claudia Diamantini[1], Antonino Nocera[2], Domenico Potena[1], Emanuele Storti[1], and Domenico Ursino[1]

[1] DII, Polytechnic University of Marche
[2] DIII, University of Pavia



**Abstract.** The pervasiveness of smart objects in people daily life is increasing, as the capabilities of objects are becoming more and more sophisticated. Objects participate to the Internet of Thing (IoT) with changing contexts and scopes, thus resulting in the rise of multiple networks linked to each other to form a new paradigm, called Multi-IoTs (MIoT). Of course, cooperation strategies among objects must follow this innovative trend as classical strategies based on the concept of coexistence appear no more adequate. In this scenario, this paper proposes a contribution by introducing a complex model for devices and contexts that follows a knowledge representation approach. It adopts dimension hierarchies in the multidimensional perspective typical of OLAP systems to represent roll-up relationships between admissible members of the considered dimensions, enabling the retrieval of relevant objects through a supervised algorithm.

## 1 Introduction

Nowadays devices and sensors are smart enough to have connection capabilities forming a network also known as Internet of Things (IoT). Furthermore, the variety of types of objects, contexts, and scopes for which devices participate to the IoT push towards the definition of multiple networks linked to each other to form the notion of Multi-IoTs (MIoT). In these scenarios it is of crucial importance the development of techniques for the retrieval of relevant devices and data conveyed by them. As a matter of fact, some contexts, e.g. outdoor applications like intelligent transportation, have largely casual, non pre-determined goals. Moreover, objects move and meet other ones, so that cooperation strategies must be dynamically defined. Some approaches in the literature define some simple criteria to decide about the creation of the network [2]: (a) proximity (objects connect to each other if they are spatially close for a sufficiently long time interval); (b) homogeneity (objects belonging to the same brand and of the same kind, e.g. Samsung community); (c) ownership (objects belonging to the same user); (d) friendship (objects belonging to users that are connected to each other in some social networks). These criteria do not take into account usage and goal context in order to decide for the actual usefulness of the contact. Approaches for the definition of context in IoT have been developed in the literature. This concept has been often intertwined with that of empowering the IoT of social features. Indeed, the idea of studying object contexts has raised from the need of improving the quality of interactions among them. For instance, [13] is the first work in which the idea of filtering contacts on the basis of proximity contexts has been developed. The authors of this work actually suggest to combine the classical notion of proximity with other metrics, such as movements patterns, thus defining a context proximity notion. Several other studies have, then, refined this notion by bringing the notion of context and context matching under the spotlight as a mean to confer social feature to objects in the IoT [6,11].

In this paper we propose an approach to identify and query relevant devices inside a MIoT, that is based on the notion of device's "context", describing its semantics in terms of its customary usage, the interest and behavior (in terms of activities done with the object itself) of the owner, and so forth. Interactions among objects can be then driven by the degree of similarity among their respective contexts. As usually done in the framework of IoT research, a context is defined as a set of couples $(p, v)$, where $p$ is some property, and $v$ represents an admissible value of the property. Differing from other approaches, we adopt a richer model based on a hierarchy of admissible values for each property, in the spirit of the multi-dimensional model typical of OLAP systems. This allows us to introduce different types of relationships among contexts, namely identity, inclusion (similar to the notion of roll-up among members of each context dimension) and a fuzzier relationship of compatibility. On the basis of this model, this paper presents an algorithm to determine the subset of devices that are more likely able to provide relevant answers to a user's query.

Knowledge representation approaches to context modeling have been extensively investigated in the literature [7,4,15] with the aim to accurately describe complex entities, relationships, and constraints [12] forming a context. Many of them adopt logical languages and ontologies to this end [1,8,18,16]. However, the expressiveness of ontologies has complexity as a shortcoming, which can be critical for many applications, like those of the IoT scenario where computing abilities are limited [4]. Furthermore, none of the cited models considers a hierarchical organization of context properties. The Context Dimension Tree, implemented in the PerLa context language for designing and managing wireless sensor network applications [17], considers a tree structure where a *dimension* node is the parent of a set of *concept* nodes. Nesting of more than one dimension level is possible, allowing the representation of structured concepts (namely admissible members). In contrast, the model proposed in this paper adopts dimension hierarchies in the multidimensional perspective typical of OLAP systems, representing roll-up relationships between admissible members of the dimensions.

The above literature focuses on "how" context information can be represented. Another issue is related to "which" information should be included in context. As already noted, this greatly varies from application to application. Authors in [5] recognize a limit inherent in the adoption of knowledge representation methodologies for context modelling, namely the difficulty for a limited group of people to enumerate all the possible concepts and relationships that may be used in a practical mobile or desktop context-aware application. We believe that the multidimensional perspective proposed in this paper offers a more "general-purpose", "application-independent" perspective: in practice, if a dimension is not relevant for a given application, we model it as if any possible admissible value was acceptable, corresponding to a "roll-up all" operation.

The object's network(s) created in this way can be compared to a person's list of contacts in a social network. This view is strongly supported by the research community and, indeed, a number of works have tried to adopt the paradigm of social network in the IoT realm and to combine these worlds improving their usability and the offered services [9,14,3,10]. In [11], the authors propose a social infrastructure allowing access to both humans and objects. In this system, humans can interact with each other offering services through their own objects, or they can interact directly with objects by using the infrastructure as a communication channel. An important step towards the definition of a unified framework for including things in the virtual human communities is described in [3]. This paper provides contribution in both the definition of policies managing the social interaction among objects, so that the result is an accessible and usable social network, and the formalization of an architecture for the IoT, so that objects include all the features needed for an easy and effective integration in existing social sites. As a result of this study the authors propose a new paradigm called Social Internet of Things (SIoT). Very recently, other studies have focused on the social factors that foster the rise of new and stable communication in the IoT. For instance, the work described in [10] first proposes a study on which social aspect improves the quality of object-to-object communication and, after that, provides insights on how to include those aspects in a new framework for creating an IoT-only social community.

Differing from a social network, where peers are not classified according to their profile, or only very naïve manual classification in a predefined set of categories can be done, the proposed model allows a very rich and flexible definition of contexts, and a graded association of profiles to contexts. Furthermore, interaction mechanisms in social networks are typically limited to information sharing by broadcasting (posting information that reaches all node's contacts) or to one-to-one messaging (directly contacting a node via its name/address). In the present paper we also present an algorithm to determine the best set of nodes that can be queried according to given information needs or preferences and contexts, hence producing a projection, or view, of the original overall network, that can be actually exploited for the goal at hand.

This paper is structured as follows: Section 2 presents the model for device contexts in a Multi-IoT scenario, on the top of which a supervised algorithm for device retrieval is presented in Section 3. Experiments to assess the effectiveness of our approach are reported in Section 4, whereas Section 5 ends this paper.

## 2 A model for devices in a multi-IoT network

This section is devoted to introduce the model used to represent devices and their relations in the context of a multi-IoT network. We define a device $\Delta_i$ as a set of sensors $\Delta_i = \{S_1, \ldots, S_n\}$. We refer to the term *context* as a scenario in which the devices operated in the past, are currently operating or may operate in the future. A set $P_i$ of properties are associated with a device, each of which represented as a pair $(p, v)$ where $p$ is a property (e.g., *owner*, *brand*, *model*, *cost*, *reference measure*) and $v$ is the corresponding value. We discuss in more detail contexts and properties in the following.

### 2.1 Context of a device

In order to specify a context, we refer to a set of dimensions $\mathcal{D} = \{D_1, \ldots, D_m\}$, which are to be intended as the dimensions in the multidimensional model (e.g., or instance *time*, *space*, *goal*). Therefore, from a data perspective, a context may be compared to a fact in a multidimensional model. In this work, we do not assume that all contexts for all devices are defined by the same set of dimensions. According to the information provided by its sensors, each device will specify a context according to available information.

Analogously to the multidimensional model, given a dimension $D_j \in \mathcal{D}$, it is possible to define a hierarchy of levels $L_j = \{l_{j_1}, \ldots, l_{j_m}\}$ such that $l_{j_1} \rightarrow l_{j_2} \rightarrow \cdots \rightarrow l_{j_m}$. The notation $l_{j_p} \rightarrow l_{j_q}$ implies that a functional dependency exists from $l_{j_p}$ to $l_{j_q}$. To make an example, given the dimension *space*, the following relation holds: *street* $\rightarrow$ *district* $\rightarrow$ *city* $\rightarrow \cdots \rightarrow$ *country*. Given a level $l_{j_i}$ of a dimension $D_j$, its members $\{\iota_{j_1}, \ldots, \iota_{j_i}\}$ are named instances, e.g. level *city* can include instances *Rome, Venice, Milan.*

The schema of a context $C$ for a device is a set of dimensions $\mathcal{D}$. A context instance $c \in C$ is a tuple $c = (\iota_1, \iota_2, \cdots, \iota_m)$. To make an example, considering a dimensional schema including $\{$ *space*, *time*, *goal* $\}$, a possible instance can take the following values: $\iota_{space}$= *Grosseto*, $\iota_{time}$= *June* $16^{th}$ *2019, 11:00-12:00*, $\iota_{goal}$ = *Running.*

Given a dimension $D_j \in \mathcal{D}$, let $\iota_{j_p}, \iota_{j_q} \in D_j$ be two instances of $D_j$. The following relations are defined between $\iota_{j_p}$ and $\iota_{j_q}$:

- $id(\iota_{j_p}, \iota_{j_q})$: it means that $\iota_{j_p}$ and $\iota_{j_q}$ are identical instances, e.g. *id*(*Montecitorio square*, *Montecitorio Sq.*).
- $inc(\iota_{j_p}, \iota_{j_q})$: it means that $\iota_{j_p}$ is included in $\iota_{j_q}$, e.g. *inc*(*Montecitorio square*, *Rome*). In order for this property to hold, it must be that *square* $\rightarrow$ *city* and *Montecitorio square* is in Rome. Moreover, given that *square* $\rightarrow$ *city* and, in turn, *city* $\rightarrow$ *nation*, it also holds that *inc*(*Montecitorio square*, *Italy*). In the following we refer to $dist(\iota_{j_p}, \iota_{j_q})$ as the distance between $\iota_{j_p}$ and $\iota_{j_q}$, that is the number of steps in the dimensional hierarchy necessary to move from the former to the latter.
- $cpt(\iota_{j_p}, \iota_{j_q})$: it means that $inc(\iota_{j_p}, \iota_{j_r})$, $inc(\iota_{j_q}, \iota_{j_r})$, where $l_{j_p} \rightarrow l_{j_r}$, $l_{j_q} \rightarrow l_{j_r}$ and $l_{j_p}$ (resp., $l_{j_q}$, $l_{j_r}$) represents the level of $\iota_{j_p}$ (resp., $\iota_{j_q}$, $\iota_{j_r}$). This means

that $\iota_{j_p}$ and $\iota_{j_p}$ are siblings, e.g. *cpt*(*Rome*,*Milan*), *cpt*(*Italy*, *France*), whereas it does not hold that *cpt*(*Rome*, *Paris*).

Let $c = (\iota_1, \iota_2, \cdots, \iota_m)$ and $c' = (\iota'_1, \iota'_2, \cdots, \iota'_m)$ be two contexts. The following relations are defined among them:

- $id_C(c, c')$, if $id(\iota_i, \iota'_i)$, $1 \leq i \leq m$; for instance $c$=(*Montecitorio square*, [10:00-11:00], *Running*) $c'$=(*Montecitorio sq.*, [10:00-11:00], *Running*).
- $inc_C(c, c')$, if $id(\iota_i, \iota'_i)$ or if $inc(\iota_i, \iota'_i)$, $1 \leq i \leq m$ and $id_C(c, c')$ does not hold; for instance $c$=(*Montecitorio square*, [10:00-11:00], *Running*), $c'$=(*Montecitorio sq.*, *Morning*, *Running*). We define $dist(c, c')$ as a measure of distance between $c$ and $c'$, measured as $\sum_{i=1}^{m} dist(\iota_i, \iota'_i)$, where $dist(\iota_i, \iota'_i) = 0$ if $id(\iota_i, \iota'_i)$.
- $cpt_C(c, c')$, if either $id(\iota_i, \iota'_i)$ or $inc(\iota_i, \iota'_i)$ or $cpt(\iota_i, \iota'_i)$, $1 \leq i \leq m$ and neither $id_C(c, c')$ nor $inc_C(c, c')$ holds; for instance, $c$=(*Montecitorio square*, [09:00-10:00], *Running*) $c$=(*Montecitorio square*, [10:00-11:00], *Running*)

In the following, we define an operator *match*($c$), which returns the set of devices that are matching with the context $c$. In order for a device $\Delta_i$ to match with $c$, the context provided by $\Delta_i$ must be a superset of $c$, i.e it must provide at least the same dimensions included in $c$, with the same members.

**Definition 1.** *(Match operator)*
Given a net $N = \{\Delta_1, \ldots, \Delta_n\}$ including a set of $n$ devices, each with a corresponding context $c_i$, and given a context $c = (i_1^{D_1}, \ldots, i_m^{D_m})$, *match*($c$)=$\{\Delta_i \in N : c \subseteq c_i\}$.

### 2.2 Properties of devices

Let $(p_i, v_i)$ and $(p_j, v_j)$ be two property-value pairs. We define the following relations between them:

- $id_P(p_i, p_j)$, if $p_i = p_j$ and $v_i = v_j$;
- $cpt_P(p_i, p_j)$, if $p_i = p_j$ and $cpt_V(v_i, v_j)$;
- $cpt_V(v_i, v_j)$, to state an explicit compatibility relation among them. For instance, a device measuring temperature in Farenheit is compatible with another device referring to Celsius.

Let us define $P(d)$ as the set of properties for a device $d$, i.e. $P = \{(p_1, v_1), (p_2, v_2), \cdots, (p_n, v_n)\}$. We introduce a Jaccard modified operator $J^*(P, P')$ which works on two sets of properties $P$ and $P'$ and returns a float between 0 ed 1 representing the ratio of $P$ that are identical or compatible with $P'$. It is computed through the following steps:

- the set $I = \{(p, p') | p \in P, p' \in P', id_P(p, p')\}$ is computed, which includes all properties that are identical;
- the set $C = \{(p, p') | p \in P, p' \in P', cpt_P(p, p')\}$ is computed, including those properties that are compatible each other;
- finally, the result of $J^*(P, P')$ is computed as $J^*(P, P') = \frac{|I| + \alpha \cdot |C|}{|P|}$, where $\alpha \in [0, 1]$ allows the contribution of the second factor to be weighted.

## 3 Retrieval of devices in a multi-IoT network

Given a user and its network, this section presents a supervised algorithm aimed to retrieve the subset of devices satisfying some requirements, that are expressed through a query.

A query $q$ is represented as $q = \langle c, Z \rangle$, where $c$ is the context of interest and $Z = \{(p_1, v_1), (p_2, v_2), \cdots, (p_n, v_n)\}$ represents the properties that the devices must satisfy, expressed as a set of pairs $(p, v)$ indicating the required value $v$ for each property $p$. The pseudocode is reported in Algorithm 1.

To provide an answer to a query $q$, the supervised algorithm first executes a $SEARCH$ function, computing all contexts that are respectively identical, included or compatible with $c$, checking whether there are devices that belong to such contexts. In case there is no answer, the algorithm *rewrites* the query by finding one or more contexts that include $c$, i.e. that are more general than $c$ (formally the set $\{c' \in C : inc_C(c, c')\}$), and then the $SEARCH$ is performed again. The whole procedure iterates until to: (1) a solution is found, (2) the number of rewritings is above a defined threshold (using the function *dist* discussed in Section 2.1), (3) there are no more general contexts. The pseudocode for the $SEARCH$ function is reported in Algorithm 1, and operates as follows:

1. The Algorithm searches for devices that have contexts identical to or included in $c$ by calling the corresponding function. If there are such devices (line 4) then their properties $P(s)$ are matched against the vector $Z$ to verify if, and to what extent, they satisfy the user query in terms of property values (line 6). Finally, each retrieved device, ranked according to the Jaccard function $J^*$ (discussed in Section 2.2), is added to the output list $S_O$ (line 7).
2. In case no device is found in the previous step, then the algorithm searches for devices having a context compatible with $c$, through the corresponding function. Similarly to the previous case, for each retrieved device a rank $r$ is computed and the pair $\langle device, rank \rangle$ is added to the output list $S_O$ (line 14). To take into account that such devices have contexts that are compatible, but neither identical to, nor included in, $c$, the rank is weighted by a parameter $\alpha_{cpt} < 1$.

Finally, the list $S_O$ is returned as output (line 18).

## 4 Experiments

The supervised algorithm discussed in the previous section has been evaluated. In particular, hereafter, we compare our approach with two baseline algorithms:

- $Baseline_1$ (no contexts): this approach does not consider contexts. Therefore, the query is expressed only as $q = \langle \emptyset, Z \rangle$.
- $Baseline_2$ (only identical): this approach considers contexts but it does not include a knowledge base. Therefore, the supervised approach can only match identical contexts but no reasoning on inclusion or compatibility can be done.

**Algorithm 1** Pseudocode for the SEARCH function

**Input:**
The set $S$ of devices in the user net.
A query q=$\langle c, Z\rangle$ over $S$, where $c \in C$ is a context and $Z = \{(p_1, v_1), (p_2, v_2), \cdots, (p_n, v_n)\}$ is a set of pairs (property value)
**Output:**
a ranked set of devices $S_O = \{\langle s_1, r_1\rangle, \ldots, \langle s_n, r_n\rangle\}$, where $s_i \in S$ and $r_i \in [0, 1]$

**function** SEARCH($q$)
    $S_O \leftarrow []$
    $S' \leftarrow$ FIND_DEVICES_ID(c,S)
    **if** $S' \neq \emptyset$ **then**
        **for each** $s \in S'$ **do**
            $r \leftarrow J^*(Z, P(s))$
            $S_O \leftarrow \langle s, r\rangle$
        **end for**
    **else**
        $S' \leftarrow$ FIND_DEVICES_CPT(c,S)
        **if** $S' \neq \emptyset$ **then**
            **for each** $s \in S'$ **do**
                $r \leftarrow \alpha_{cpt} * J^*(Z, P(s))$
                $S_O \leftarrow \langle s, r\rangle$
            **end for**
        **end if**
    **end if**
    **return** $S_O$
**end function**

**function** FIND_DEVICES_ID($c$, $S$)
    $C' \leftarrow IC(c) \cup INC^-(c) = \{c' \in C | id_C(c, c') \vee inc_C(c', c)\}$
    $S' \leftarrow \{s \in S | s \in match(c'), c' \in C'\}$
    **return** $S'$
**end function**

**function** FIND_DEVICES_CPT($c$, $S$)
    $C' \leftarrow CPT(c) = \{c' \in C | cpt_C(c, c') \vee inc_C(c', c)\}$
    $S' \leftarrow \{s \in S | s \in match(c'), c' \in C'\}$
    **return** $S'$
**end function**

Firstly, the dataset, the knowledge base and the experimental settings are introduced. Then, we present results aimed to evaluate three different measures: (1) the probability to obtain a response from a query, (2) how the query rewriting approach discussed in Section 3 affects retrieval and (3) how it affects precision of the supervised algorithm. Hereby, we refer to the precision as the ratio $\frac{relevantcontexts}{retrievedcontexts}$, where *retrieved contexts* stands for the contexts belonging to the retrieved sensors, while *relevant contexts* refer to those contexts that are relevant to the query.

**Dataset and model** In this set of experiments we refer to a dataset that includes 5000 devices, each with a single sensor. Each device belongs to one or more IoT networks, the largest of which includes 500 devices. These are described in terms of their properties, although, in these tests, we consider a single property, namely the "measure" retrieved by the device (e.g., temperature, humidity or pressure), for the sake of simplicity.

The model taken into consideration for these tests includes 3 dimensions, 3 hierarchical levels for each dimension and a branching factor of the dimensional tree equal to 3. In particular, the dimensional tree will include 1 root element at level 1, 3 elements at level 2 and 9 elements at level 3 (i.e. 13 elements in total). The overall number of possible contexts is, therefore, given by all the possible combinations of elements for each dimension, e.g. $|C| = 13^3 = 2197$ in this case. Specific compatibility relationships among members are defined for each dimension. In our experimental setting, each member is defined to be compatible with 5% of the other members in the same level (e.g., given the level "street" of dimension space, each specific member will have 5% of chances to be defined as compatible with any other street).

**Experimental settings** The initialization includes the following steps:

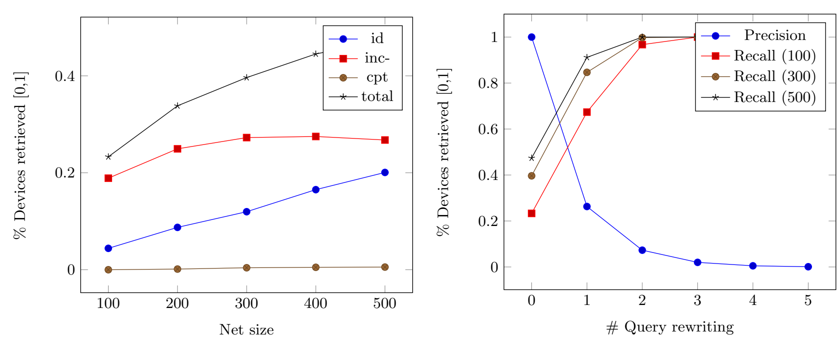
- Device initialization: a device is picked from the dataset, and the set $S$ of all the devices belonging to its net are extracted. In our test, the following procedure is repeated with 5 devices belonging to nets with sizes $S_1, \ldots, S_5$ from $|S_1|$ =100 to $|S_5|$=500.
- For each device s $\in S_i$: a context $c$ is randomly assigned to $s$. The assignment is done according to a strategy following a power law: we randomly assign a context from a small subset of $C$ (i.e., 20% its size) to the large majority of devices (i.e., 80% in our tests), which hereafter we name $C_s$. We aim to model in such a way as the probable context similarity of devices belonging to the same network. The procedure assigns a random context from the remaining set $C - C_s$ (i.e., in this case corresponding to the remaining 80% of $C$) to the rest (i.e., the remaining 20% of devices).
- A query $q = \langle c, Z \rangle$ is defined by assigning a context $c$ from $C_s$ and a property value from the list of possible property values seen in the database.
- The query $q$ is launched on the network.

The procedure has been repeated for a number of queries equal to $|C|$, in order to perform a comprehensive evaluation. Results are, then, averaged.

**Results** Results are summarized in Figure 1. Figure 1(a) shows the performances of the supervised algorithm with increasing net sizes, in terms of percentage of queries with an answer. On the other hand, Figure 1(b) shows how the number of rewritings affects precision (as defined above) and how it affects recall in terms of the increase in the percentage of queries answered when one or more rewritings occur (see Section 3).

The most relevant insights are summarized as follows: the increase in the number of devices in a net has a positive impact, on average, on results. In particular, the larger the size, the greater the ratio of queries with an answer. By rewriting a query $c$ as a query $c'$ (which includes the former), the likelihood of obtaining an answer increases. Indeed, the number of contexts included or compatible in $c'$ is obviously larger than in the former case. A single rewriting is enough to significantly increase the chance to find at least one solution to the query. However, this is achieved at the price of a relaxation of the original query specification and, as a consequence, of a decrease of the precision of results.

A direct comparison with $Baseline_1$ is straightforward, as the number of retrieved devices is purely based on property information. Therefore, with $Baseline_1$ the result set is always larger, but the whole set of sensors in the network must be analysed and matched against the properties, with no guarantee on the correctness of the results. In other terms, recall is not higher whereas precision is much lower. On the other hand, by comparing these results with the $Baseline_2$ approach, it is possible to conclude that our approach allows the retrieval of a much larger number of relevant results. In this respect, $Baseline_2$ is basically equivalent to the evaluation of only identical contexts. Therefore, precision is high, whereas recall is typically much lower. On the other hand, by exploiting the knowledge base, and specifically inclusion relations among contexts, more contexts can be obtained. Moreover, contexts that are compatible are added to the result with a lower rank, meaning that their relevance is lower than the others but can be still useful, depending on the specific application case. Finally, by exploiting inclusion relationships and the notion of rewriting, our approach can optionally lower precision in order to increase the probability to answer a query.




(a)

(b)

Fig. 1: (a) Percentage of retrieved devices with networks of difference size, (b) precision and recall against the number of query rewritings.

## 5 Conclusion

This paper has discussed an approach to querying a Multi-IoT network of devices, which relies on a multi-dimensional model for device's contexts that takes also into account its customary usage, user interest and behavior. On top of that, we proposed a supervised algorithm to determine the best set of nodes that can be queried according to given information needs or preferences and contexts, experimentally evaluating its effectiveness. Future works include: (1) the extension of the model with more expressive relations between dimensions members, (2) an unsupervised algorithm that starts with no initial specification of desiderata and automatically generates a query on the basis of its current context and device's properties, and (3) a more comprehensive experimentation.

## Acknowledgement

This work was partially funded by the Department of Information Engineering at the Polytechnic University of Marche under the projects "A network-based approach to uniformly extract knowledge and support decision making in heterogeneous application contexts" and "Novel and Optimal Solutions for Metering and Enforcing Security and Privacy in Cloud Systems" (RSA-B 2018).

## References

1. Alessandra Agostini, Claudio Bettini, and Daniele Riboni. Hybrid reasoning in the care middleware for context awareness. *International Journal of Web Engineering and Technology*, 5(1):3–23, 2009.
2. Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.
3. Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608, 2012.
4. Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.
5. Preeti Bhargava, Shivsubramani Krishnamoorthy, and Ashok Agrawala. Rocomo: A generic ontology for context modeling, representation and reasoning in a context-aware middleware. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 584–585. ACM, 2012.
6. Julian Bleecker. A manifesto for networked objects—cohabiting with pigeons, arphids and aibos in the internet of things. 2006.
7. Oscar Cabrera, Xavier Franch, and Jordi Marco. Ontology-based context modeling in service-oriented computing: A systematic mapping. *Data & Knowledge Engineering*, 110:24–53, 2017.
8. Harry Chen, Tim Finin, and Anupam Joshi. Soupa: standard ontology for ubiquitous and pervasive applications. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004*, 2004.
9. Lianhong Ding, Peng Shi, and Bingwu Liu. The clustering of internet, internet of things and social network. In *2010 Third International Symposium on Knowledge Acquisition and Modeling*, pages 417–420. IEEE, 2010.
10. Qinghe Du, Houbing Song, and Xuejie Zhu. Social-feature enabled communications among devices toward the smart iot community. *IEEE Communications Magazine*, 57(1):130–137, 2019.

11. Dominique Guinard, Mathias Fischer, and Vlad Trifa. Sharing using social networks in a composable web of things. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 702–707. IEEE, 2010.
12. Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, 2006.
13. Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *international conference on Ubiquitous Computing*, pages 116–122. Springer, 2001.
14. Huansheng Ning and Ziou Wang. Future internet of things architecture: like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4):461–463, 2011.
15. C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1):414–454, 2014.
16. Anand Ranganathan, Robert E McGrath, Roy H. Campbell, and M. Dennis Mickunas. Use of ontologies in a pervasive computing environment. *Knowledge Engineering Review*, 18(3):209–220, 2003.
17. Fabio A. Schreiber, Letizia Tanca, Romolo Camplani, and Diego Viganò. Pushing context-awareness down to the core: more flexibility for the perla language. In *Electronic Proc. PersDB 2012 Workshop (Co-located with VLDB 2012)*, pages 1–6, 2012.
18. Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. Cool: A context ontology language to enable contextual interoperability. In Jean-Bernard Stefani, Isabelle Demeure, and Daniel Hagimont, editors, *Distributed Applications and Interoperable Systems*, pages 236–247. Springer Berlin Heidelberg, 2003.