|  |  |
| --- | --- |
|  |  |

LABORATOIRE



INFORMATIQUE, SIGNAUX ET SYSTEMES

DE SOPHIA ANTIPOLIS

UMR 6070

SEMANTIC-BASED SERVICES FOR DEVICE SELECTION:

DYNAMIC KNOWLEDGE BASE MANAGEMENT

*Gérald Rocher, Rahma Daikhi, Jean-Yves Tigli*

Rapport de recherche

ISRN I3S/RR-2015-03-FR

Mars 2015



Laboratoire d’Informatique, de Signaux et Systèmes de Sophia Antipolis – UNSA-CNRS

2000, rte.des Lucioles – Les Algorithmes – Bat Euclide B – B.P. 121 – 06903 Sophia Antipolis Cedex – France

Tél : 33(0)4 92 94 27 01 – Fax : 33 (0)4 92 94 28 98 – [www.i3s.unice.fr](http://www.i3s.unice.fr)

UMR 6070

|  |
| --- |
| RESUME :  *Dans les systèmes informatiques ambiants, les applications logicielles sont composées à partir d'une sélection de services logiciels intégrés à des dispositifs et des objets de notre vie quotidienne qui, devenant communicants (Internet of Things), peuvent dès lors, être observés et contrôlés. Ces objets et ces dispositifs, fixes ou mobiles, sont soumis aux phénomènes physiques de l’environnement réel dans lequel ils sont placés ce qui impliquent une évolution de leur disponibilité dans le temps et dans l'espace. Il est donc primordial que le mécanisme de sélection de services, au-delà des fonctionnalités offertes par les services, tienne compte également de leur dynamique physique inhérente à leur intégration dans le monde réel. Dans ce cadre, l'utilisation des standards du web sémantique (Web of Things) est étudiée pour obtenir, à partir d’annotations sémantiques formelles sur les dispositifs et les services, une représentation dynamique et incrémentale des connaissances fonctionnelles et contextuelles qui leur est associée et permettre la gestion de leur évolution dans le temps et dans l'espace****.*** |
| MOTS CLES :  Web sémantique, représentation et gestion de la connaissance, Contexte, Informatique ambiante, Internet des objets |

ABSTRACT:

*In ambient computing systems, software applications are composed from a selection of services integrated in devices and everyday life objects. Communicating, they can therefore be observed and controlled (Internet of Things, IoT). These devices and objects, immobile or mobile, are placed in a highly dynamic physical environment where their availability changes over time and space. It is therefore essential that the selection mechanism, beyond the capabilities offered by the services, also takes account of their multidimensional dynamicity, inherent to their integration in the real world. In this context, the use of the semantic web standards (Web of Things, WoT) is studied in order to, from formal semantic annotations, obtain a dynamic and incremental representation of the functional and contextual knowledge about devices and services, enabling the management of their evolution and availability in time and space.*

KEY WORDS:

Semantic Web, knowledge representation and management, Context, Ambient computing, Web of Things

SEMANTIC-BASED SERVICES FOR DEVICE SELECTION:

DYNAMIC KNOWLEDGE BASE MANAGEMENT

*Gérald Rocher, Rahma Daikhi, Jean-Yves Tigli*

**Abstract**

*In ambient computing systems, software applications are composed from a selection of services integrated in devices and everyday life objects. Communicating, they can therefore be observed and controlled (Internet of Things, IoT). These devices and objects, immobile or mobile, are placed in a highly dynamic physical environment where their availability changes over time and space. It is therefore essential that the selection mechanism, beyond the capabilities offered by the services, also takes account of their multidimensional dynamicity, inherent to their integration in the real world. In this context, the use of the semantic web standards (Web of Things, WoT) is studied in order to, from formal semantic annotations, obtain a dynamic and incremental representation of the functional and contextual knowledge about devices and services, enabling the management of their evolution and availability in time and space.*

**Index Terms**

Semantic Web, knowledge representation and management, Context, Ambient computing, Web of Things

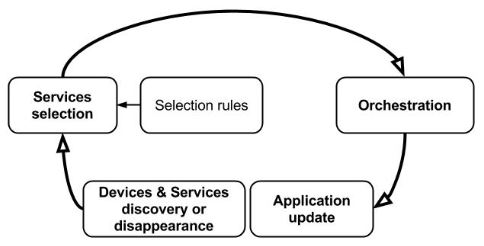
# INTRODUCTION

Ambient computing [1], also known as ubiquitous or pervasive computing, is a term referring to the integration of the real physical world into the digital world of internet. This applies to all static or dynamic objects of our daily life (chair, table, lamp, etc…) or physical environments (city, building, vehicle, medicalized space, etc…), which become, from software applications standpoint of view, observable and controllable [2]. This is made possible through communicating *devices* embedded in objects or placed in the environment [3]. These devices implement *resources* interacting with objects (actuator) and/or gathering information (ID, descriptions, sensor observations, etc…) about themselves, the objects or the environment they relate to. Access to these resources is achieved through software *services* exposing their interfaces and allowing communication with the digital world. These services are then able to participate in the development of complex software applications that have to, among all the available services, *select* and *orchestrate* those allowing to reach the functionality that meets user’s needs.

Devices, objects and physical environments dynamic nature implies that the software applications, in order to maintain their functionality over the time (service continuity), must dynamically adapt themselves to constant changes occurring in the real world. Applications are then said « context-aware». Thus, a context-aware software application is the result of a services composition within a dynamic self-adaptive loop (*Figure 1)*.

Within this dynamic self-adaptation loop, the selected services relevance, based on annotations on devices and services, is essential. Unfortunately, these annotations are often static and uncorrelated from the dynamicity of the real physical environment the associated devices and services are interacting with.

**How to improve then the relevance of the selected services and dynamically increase the knowledge by still maintaining its consistency with the real environment?** The environment is indeed the theater of physical phenomena, often independently modeled from each other (temperature, quality of service, etc…) but subject to evolutionary principles involving a common temporality. To do so we propose an approach based on semantic web standards (SWoT) [4] and associated technologies [5].   
Each device, through formal semantic annotations composed of ontological fragments, brings the conceptual knowledge of its own domain. Aggregated together, all these fragments allow to dynamically (devices discovery/disappearance) increase the terminological content of a centralized knowledge base (KB). Dynamically enriched with properties gathered from sensors placed in the environment or embedded on the objects, these annotations also allow the KB to keep knowledge coherency with the environment and its evolution over the time.



**Figure 1:** dynamic self-adaptation loop

# RELATED WORK

## On the knowledge representation and management using the semantic web standards technologies

Within the dynamic self-adaptation loop, the service selection mechanism must be able to *interpret*, on the one hand, the functionalities offered by the devices and the services and, on the other hand, the context from which it is able to *reason* about their availability over the time, the space or any other relevant dimension in the context of physical objects and devices (temperature, quality of service, consumption, etc…). In this context, proven semantic web technologies are well appropriated for representing the knowledge : RDF/RDFS (Resource Description Framework/Schema) [6] languages, subset of first order logic with binary predicates, OWL DL (Web Ontology Language) [7], on top of RDF/RDFS for defining ontologies, based on description logic adding higher expressivity [10], allow knowledge representation, access (SPARQL [8]) and reasoning to infer new knowledge.

### Ontology

An *ontology* [9] is a formal and explicit knowledge representation model hierarchically structuring the concepts of a domain. A set of concepts allows the expression of the complete domain knowledge. An ontology is composed of three main element types:

1. ***Classes*** (or concepts) and sub-classes hierarchically organized according to a taxonomy,
2. P***roperties*** allowing to define ***facts*** or ***relations*** between classes. There are two property types:
   1. Object property that defines a relationship between two instances of a class or between classes,
   2. Data types properties as a relation between a literal value and a class instance.
3. Class ***instances*** (class individual) which may take the characteristics defined by the properties.

An ontology is used to share a common understanding about a given domain. In the context of IoT it is unlikely that a single ontology can express all the concepts implicated in the real world. Building a large ontology can be done by integrating several existing ontologies describing portions of a large domain [52],[53],[54].

### Knowledge base and reasoning

An ontology can be seen as a meta-system for a KB as a description of the knowledge representation they contain. Based on description logic, KB includes facts and individuals on all the defined concepts from which a reasoning engine is used to derive implicit knowledge from explicit knowledge. Knowledge is structured in two description levels, *ABox* and *TBox*, respectively defining *assertions* on the instances and individuals, and the general concepts *terminologies* [11].

### Multidimensional knowledge representation

A multidimensional knowledge representation allows reasoning about the devices and services availability over the time, the space or any other relevant dimension in the context of physical objects or devices (temperature, quality of service, consumption, etc…). From a semantic standpoint of view this is equivalent to state that a relationship is true only under a given context, the context being any measure characterizing a physical concept.

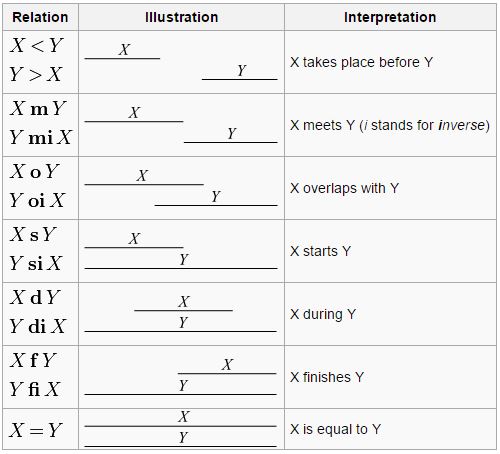
In the field of semantic web, knowledge representation language such as OWL expresses relations as binary relations that do not allow the representation of contextualized relations (which would need ternary relationships) [12]. Several mechanisms have been developed that mainly handle temporal and/or spatial dimensions in ontologies. In the *temporal RDF* approach, a temporal label is added on RDF triples indicating the time interval during which they are valid [13]. The ontology *versioning* approach [14] suggests that the ontology has different versions over the time. At each modification, a new version of the ontology is created. The *named graphs* approach [15] allows knowledge contextualization [16], [17]. However, with this approach, reasoning capabilities only applies to elements in the graph [18]. It is therefore not possible to create relationships between different graphs (contexts). The *context slices* approach [19] (generalization of the *4D-Fluents* approach [20]), in addition to allow the contextualization of a given relationship, has the advantage, on the one hand, of allowing inferences computations on all the elements in the ontology and, on the other hand, of allowing any individuals to be used as an assertion’s context. This approach is the one we have adopted for multidimensional knowledge representation.

## On the time and space semantics

As stated earlier, the services selection mechanism has to reason about the devices and services availability over the time and space from information provided by the formal semantic annotations. This approach involves to provide a semantic description for temporal and spatial dimensions along with associated topological relationships allowing to compare individuals’ temporal or spatial properties.

### Temporal semantics

The temporal structure introduced with ITL (Interval Temporal Logic) described by Allen [29], allows time dimension linear modeling from *intervals* and *instants***.** Thus, from two time intervals are deduced the 13 Allen’s topological relationships. These relationships allow to reason about objects and devices temporal dimension. OWL-Time ontology incorporates the definition of instants, time intervals along with Allen’s relationships [30], [31].



**Figure 2** : Allen's temporal relationships

### Spatial semantics

Provides a description of the space dimension with *points* and *regions* that define qualitative topological relationships (Region Connection Calculus RCC-8 [32], 9-Intersections model [33]), **quantitative** (A 10km, etc…), and **directions** (N, S, E, O, NO, NE, SE, SO).

|  |  |  |
| --- | --- | --- |
| https://lh5.googleusercontent.com/ht1cg2xRFvwQHPcwwq-4AG-fPTzgbKwdiAMKNE0VMMLHefXNh7q7HmWGU0E2QespAvrBX6EyZU6jIY6OEH3uzPZEn6ExlR_nbvBJ4ivGguffXcPPHqrOuoQCzG5MFzR9Dvt3KGw  **Figure 3 :** RCC-8 spatial relationships | DC(x; y)  PO(x; y)  EC(x; y)  EQ(x; y)  TPP(x; y)  NTPP(x; y) | → x is disconnected from y,  → x partially overlaps y,  → x is externally connected with y,  → x is identical with y,  → x is a tangential proper part of y,  → x is a no tangential proper part of y. |

However, topological relationships representation such as Allen or RCC-8, although feasible, are not trivial from OWL-DL [34] and the authors present a translation of the RCC-8 topological relationships to OWL-DL. Although the spatial and temporal concepts relationships representation are achievable from OWL-DL, inference and query engines capabilities have to be improved with *ad-hoc* rules and operators. Thus, in [35], the authors, based on [34], developed custom reasoning and query engines(Pellet Spatial). SWRL (Semantic Web Rule Language) is used to define inference rules from Allen’s [36], [37], [38] or RCC-8 [36] topological relationships. This approach has been implemented in several specialized query engines like TOQL [35], SQWRL [39], SOWLQL [36], GeoSPARQL [40], etc...

## On the context representation

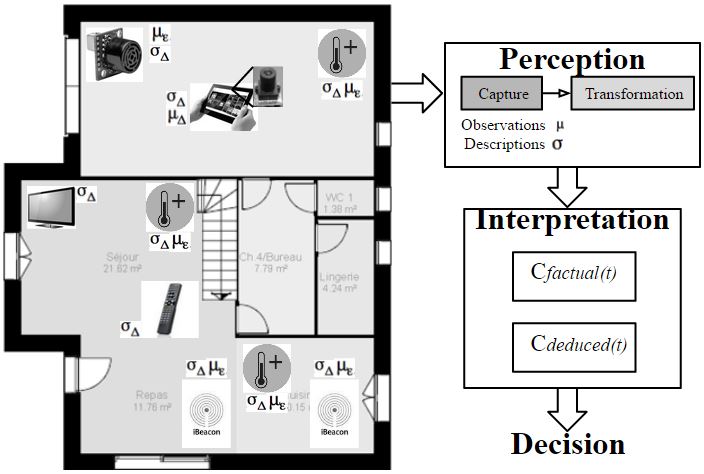
The notion of context in ambient computing is widely debated. Among all available definitions we have selected this one [27] :

*"The context is the set of the application external parameters that can influence its behavior by defining new views of data and functionality. These parameters have a dynamic aspect allowing them to evolve during execution time".*

This definition is interesting because it considers not only the application external parameters but also their dynamicity (inherent to real physical environments the software applications are dived in).

A.Dey [28], stated that a context-aware software application must implement three distinct mechanisms:

1. ***Perception*** of the software application context change,
2. Contextual data***Interpretation***,
3. ***Decision*** on the adjustments to trigger.



**Figure 4**: Context extraction

## On the semantic annotations usage in WoT

Our approach is based on formal semantic annotations distributed on each device providing a knowledge representation about the available services functionalities and about their contextual properties.

In [22] the authors present an approach proposing a solution to the problem of the devices and services functional knowledge representation from semantic annotations, based on REST (Representational State Transfer) architecture. This approach do not handle devices and objects dynamicity [21] : the set of available services and their location are statically encoded either in a database [23] or in their annotations. Also, the knowledge contextualization is limited to devices localization only.

# METHODOLOGY

## Context model

Since there is no consensual definition for the context, we try to give a model applicable to IoT. Let’s consider devices ∆ placed in the real physical environment ε. Their *state* and the set of proposed services ϛ, described from the information they provide (that is, the description of the device and services as well as observations of objects or the environment they observe from sensors) allow to formally describe the context C(t) that may affect the software application behavior.

Then:

(1)

(2)

(3)

From this context knowledge representation one can define, at each time t, the usability and the relevance of the available services:

(4)

(5)

(6)

**Usability** (4) denotes the fact that the device state permits its associated services to be used for the software composition. The usability then depends on the physical state of the device and its environment and is therefore dynamic. This parameter requires sensors deployment (observers) sensing the environment (exteroceptive sensors), or the device’s intrinsic physical state (proprioceptive sensors).

**Relevance** (5) indicates if a service matches a desired functionality.

The combination of this two parameters (6) allows the selection of the services being valid candidates for the composition of the software application.

Let see now the interest of considering the devices usability in addition to the relevance with two relevant scenarios.

## SCENARIOS

A services selection mechanism based only on the services functionalities relevance does not ensure the application coherency with its execution context. At each instant, devices and services usability has to be taken into consideration.

### Scenario 1: Temporal availability

A washing machine, for the purpose of saving energy, can be used only within a particular time interval. A services selection mechanism based only on the services relevance, can grant a software application to control the washing machine at any time (…even one minute before the end of the services availability time period). A services selection mechanism based on AND will allow to reason about the services availability and will not grant the software application to control the washing machine at any time.

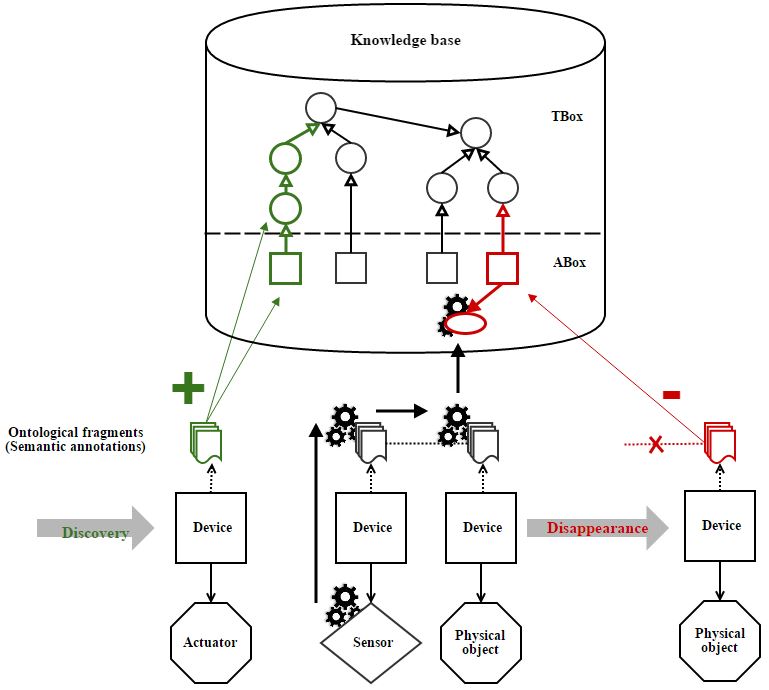
### Scenario 2: Spatial availability

Spatial localization devices are of great importance in the process of adapting the software application with its execution context. Let’s take the example of an agent controlling a water distribution network. He hold a remote control for opening and closing valves. At some points, it is possible that several identical services (allowing to open/close the valves) are simultaneously available. A services selection mechanism based only on the relevance of the services will consider controlling all available valves around the agent. A services selection mechanism based on AND will allow to reason about the availability of the services based on the valves positions compared to the agent’s position and will grant controlling the valve closest to the agent.

## SWoT dynamicity

Conceptually, we denote three levels of dynamicity for SWoT (Figure 5):

1. The *terminological level* where a new discovered device leads to increase the KB content (*TBox*),
2. The *assertion level*, where a new device discovery or a device disappearance lead to respectively increase and decrease the KB content (*ABox*),
3. The *semantic annotation level* where devices coupling leads to have sensors properties propagated across all coupled devices annotations.



Sélection de services

Requêtes

Réponses

**Figure 5:** SWoT dynamicity levels

## Formal semantic annotations

The calculation of first order predicate logic (equations 4, 5 and 6) involves linking a Boolean value to atomic formulas. **They express the properties of objects to which we must give a meaning and thus a semantics**. This semantics is important as it will determine the devices and services usability and relevance, and thus, contribute to the coherence of the application with its execution context.

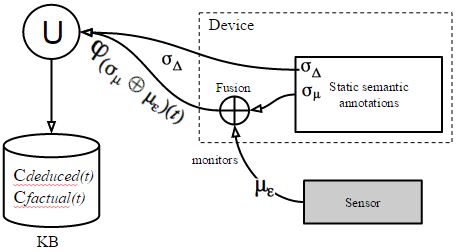
In this context, semantic web standards are used to provide, from formal semantic annotations, a dynamic representation of the functional and contextual knowledge of the devices, the services and the environment they interact with. These annotations enable then to build a dynamic context semantic representation enriching a centralized knowledge base from which are applied reasoning and querying.

The devices’ dynamic nature (specifically the objects or the environment they are attached to) involves defining two scopes regarding the ontological elements defined in the formal semantic annotations:

1. ***Static*** :
   1. Denoted, these ontological elements describes :
      1. The devices and the services **domain’s concepts** and static properties (*TBox*),
      2. The devices and services instances (*ABox*).
   2. Denoted, these elements brings a **semantic description for sensors measures**.
2. ***Dynamic***: Assertions generated by the *fusion* operator ⊕ from and. Denoted and, it provides a semantic for :
   1. Individuals or contextual properties varying over the time (*ABox*),
   2. Individuals’ instances gathered from measures (*ABox*).

is a knowledge representation of the device’s state φ as described in equations (1) and (4). At each time *t*, the semantic annotation associated to the device’s state φ is the combination of the semantic description and observations. These annotations are used to enrich the KB.

**is a knowledge representation of the contextual properties** in such a way it could be inherited by other devices.



**Figure 6 :** Dynamic and static semantic annotations

|  |  |
| --- | --- |
|  |  |

**Figure 7** : Semantic annotations example

So far, thanks to the formal semantic annotations, at each time *t,* the KB’s content is a snapshot view of the devices, the services and the environment. When queries are issued, the usability of the services and KB’s coherency with the context is considered as *implicit* and acquired. This knowledge representation does not include the real world’s objects physical characteristics that would allow reasoning about their usability over the time and space. With this knowledge representation addition, the devices and services’ usability, as well as the KB’s coherency with the context would be considered as *explicit*. A multidimensional devices and services’ knowledge representation involves defining the scope « contextualized » for the ontological elements defined in the formal semantic annotations. Denoted, these ontological elements describes the properties 𝛾Δ1 valid only in the context 𝜇Δ2. As stated in the related works, we rely on Chris Welty’s work where he defined a contextualized knowledge representation pattern, the ‘*Context slices*’ [19], based on semantic web standards.

The context representation 𝜇Δ2, can be gathered from discrete or continuous measurements. In the case of continuous measures, a discretization component may be necessary to reduce the number of possible contexts. For instance, in the case where 𝜇Δ2 is gathered from a temperature sensor, the discretization component can partition the measures into three categories: cold, warm, hot. In [41], the author introduces the « contexteur » mechanism used to capture, compute and distribute dynamic contextual annotations. Such a mechanism can be adapted and used for continuous environmental values discretization.

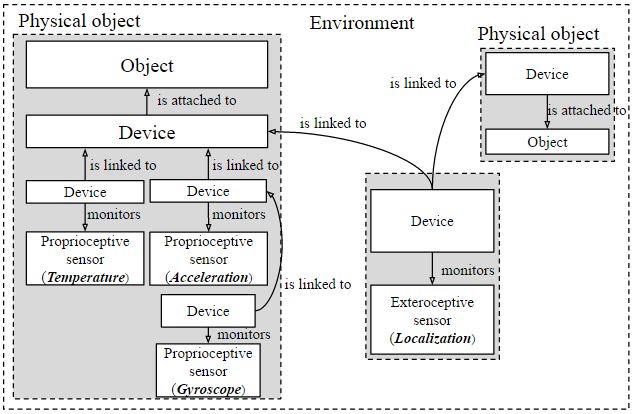
## Formal semantic annotations interoperability

So far we have described devices:

1. Using local ontological elements σΔ, semantically describing offered services and related concepts (devices associated with objects),
2. Using ontological elements σμ and measures μΔ, semantically describing sensors’ measures at each time t (devices associated with sensors).

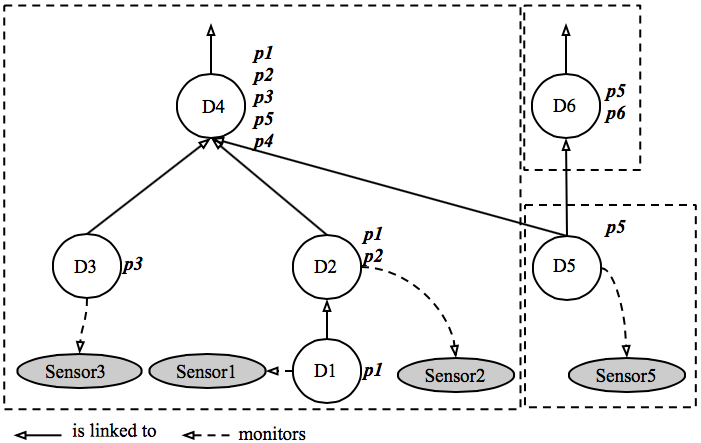
For the time being, the state φ of a device associated with a sensor only describes itself. As we explained earlier, exteroceptive sensors measure environment’s physical phenomena and therefore provide contextual properties that may be relevant to all devices placed in this environment (i.e.: the temperature of a room, a location, etc. ...). Proprioceptive sensors, ‘embedded ' on the objects, produce objects’ intrinsic contextual properties. The overall state 𝜑 of a device is then characterized by the measurements gathered from proprioceptive and exteroceptive sensors.

In the example in ***Figure 8***, the device’s state φ is characterized by some contextual properties (temperature, acceleration and orientation) gathered from proprioceptive sensors and a contextual property gathered from an exteroceptive sensor (location).



**Figure 8** : Devices and sensors configuration example

The device’s semantic annotations should allow the representation and the description of this global state. Furthermore, each of the devices associated to an exteroceptive sensor can be used several times to participate in the description of different devices in the same environment. But the way a device is going to be coupled cannot be anticipated. We must therefore allow a strong interoperability between devices and therefore, between the semantic annotations. The semantic annotations’ interoperability can be viewed at two levels: syntactic and semantic [52]. The syntactic level ensures consistent exchanged data representation. We base the semantic annotations’ representation on XML/RDF format. OWL language usage provides a generic solution to the problem of semantic interoperability. Thus, a device instance is able to inherit properties from exteroceptive sensors monitored by devices coupled with the relationship 'is linked to'. In doing so, we obtain a structural directed graph G = (Δ, predicate) expressing the contextual properties inheritance.



**Figure 9**: Structural graph of contextual properties inheritance

In the example in Figure *9*, the contextual properties of the device D4 are the result of the aggregation of contextual properties of devices D1 , D2, D3 (proprioceptive) and D5 (exteroceptive). From a semantic standpoint of view it makes sense: if D1, D2, D3 and D5 respectively measure the acceleration, the orientation, the temperature and the location, then D4 has all of these contextual properties. Note also that the proprioceptive sensors’ contextual properties are local to the object and are not supposed to be inherited by devices outside of the object (For instance, the acceleration property of a device embedded in an object is intrinsic to this object and cannot be inherited by other objects).

There is no property value inheritance mechanism with OWL language, which, although allowing the description of classes, subclasses and properties is not behaving like object-oriented languages [42]. Thus, each of the contextual properties is duplicated in the devices’ formal semantic annotations that inherit it through the contextual properties inheritance graph G = (Δ, predicate).

## Formal semantic annotations coupling

The overall device’s state φ knowledge representation and its coherency with the context depend on the contextual properties inheritance graph. In this graph, the structural connections are conditioned by the predicate "is linked to". These structural links represent a physical reality implicit at the KB level that has no knowledge about the way the overall state φ of a device Δ is established:

1. Devices are physically embedded on an object. The appropriateness of observers usage in the device’s semantics is implicit and defined ‘by design’,
2. Devices are placed in the environment and observe objects in their scope. The appropriateness of observers usage in the device’s semantics can be:
   1. Explicit and user defined. But the user does not think at everything and has an incomplete knowledge of the environment which, in addition, dynamically evolves over the time.
   2. Inferred by a reasoner from devices’ common properties. The contextual properties inheritance graph representation, defined with an ontology, would allow to add a structural semantic to the context representation. In that case, we would not only infer on the domain’s assertions but also on the whole device set. For instance, let’s consider three devices D3, D4 and D5 (*Figure 7*) respectively measuring the exteroceptive properties *ta*, *l* and *i*. These devices are coupled to three devices D1, D2 and D3. By transitivity, D3 inherits properties *l* and *i*, D5 inherits properties *ta* and *i*. D4 inherits properties *ta* and *l* and D6 inherits properties *l* and *i*. The new created links implies new inferences. At the end of the process, all devices bringing exteroceptive measurements form complete subgraphs.

This leads to distinguish two main environment types [43]:

1. Environments with fixed structure said “bounded”. The network, software and hardware infrastructures are fixed, the position of the several sensors available in the environment is known at the deployment/development time. The fixed environment topology greatly facilitates the elaboration of the contextual properties inheritance graph.
2. Mobile environments where devices are not linked to a given physical environment but are mobile. Each entity, device or sensor has a proper dynamic. These environments are characterized by:
   1. A highly dynamic and unpredictable topology infrastructure,
   2. Unpredictability and partial accessibility due to potentially nonexistent sensors,
   3. Unbound character.

**Figure 10:** Inferences on exteroceptive measurements

In these environments, applications are built following a bottom-up approach and are described through a set of independent rules. The elaboration of the contextual properties inheritance graph could also be described from rules. However, unlike the construction of an application from rules based on functionalities semantics orchestrated to reach a global user need, devices in the contextual properties inheritance graph cannot be semantically coupled together based only on their measures semantic. A common referential is needed for these links to make sense. For instance, if we consider two devices embedding a GPS (the common referential), the semantic link can be automatically achieved if, for instance, the distance in between devices is less than a given value.

# PRELIMINARY RESULTS

## Upper level ontology



An ontology is used to share a common understanding about a given domain. In the context of IoT it is unlikely that a single ontology can express all the concepts implicated in the real world, by definition open and highly heterogeneous, not only from a technical standpoint of view but also from a semantic standpoint of view (semantic diversity). For instance, a device with a particular semantic can be deployed in several entities in conjunction with some other devices. When put all together these devices bring a new semantic which, with regards to all possible associations, is hard to anticipate in a global ontology.

Based on this, we propose an approach where each device exposes its complete domain’s ontology through formal semantic annotations, namely the concepts (classes), the properties and the instances. An upper level ontology is used to express knowledge applicable to all domains. This ontology is, for the time being, composed of several sub ontologies covering the following domains (Figure 12):

|  |  |
| --- | --- |
| **Domain** | **Sub-ontology** |
| Devices and services | OWL-S [46] |
| Knowledge contextualization | Context-slices [19] |
| Temporal dimension (Intervals, instants, Allen’s topological relations) | OWL-Time [31] |
| Spatial dimension (Points, areas and RCC-8 topological relations) | SOWL spatial [36] |

**Table 1** : Upper ontology content

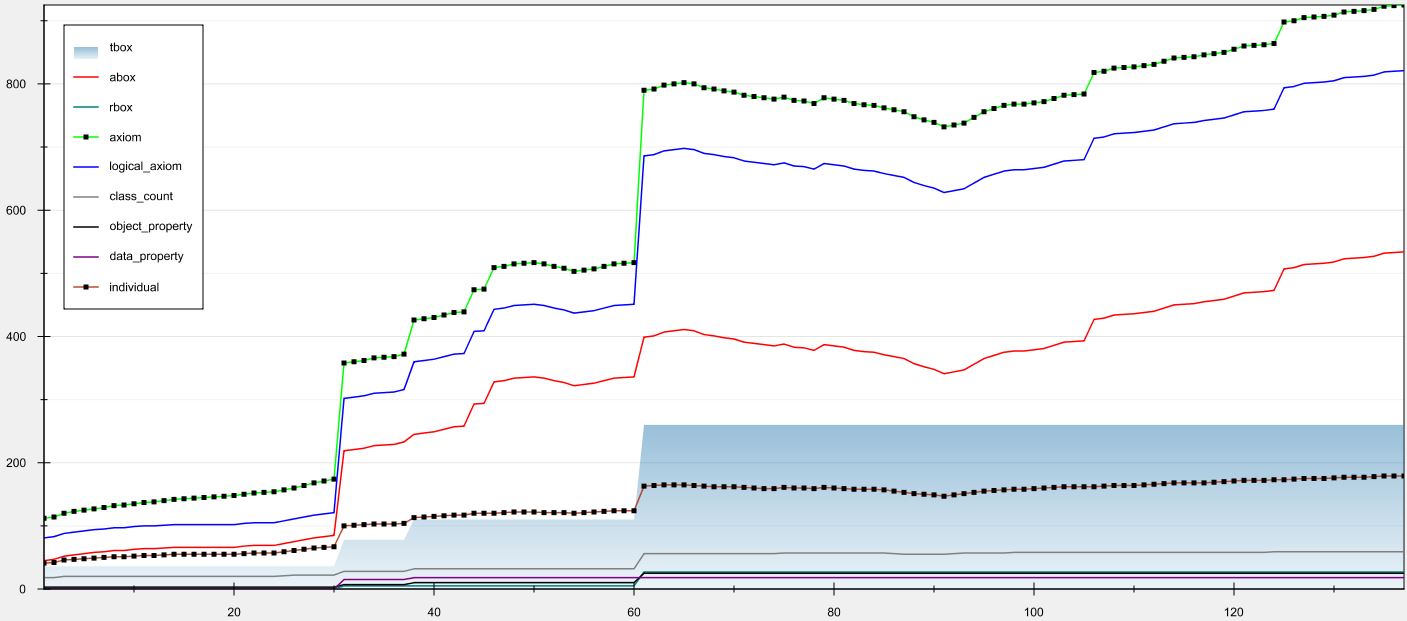
and the upper level ontology.

This approach is well suitable for closed environments (with limited amount of devices). Although scalable, it is unlikely the KB content can indefinitely increase. Limitations may occur in space (system memory limitation) and, most importantly, in time (query processing time). Also, in open environments, if each device brings its own knowledge representation (defined by different peoples or organizations), there is a risk of semantic conflicts (same class name defining two different concepts) and alignments needs.

## Dynamic knowledge management

As previously mentioned, the context knowledge representation is divided into three distinct levels:

1. The formal semantic annotations:
   1. Providing devices and services domain’s formal knowledge description along with the observations on the environment (exteroception) or the objects (proprioception),
   2. Providing contextual properties gathered from proprioceptive and exteroceptive sensors observations and propagated (exteroceptive) through the inheritance graph depicting the devices coupling.
2. The domains terminologies (TBox),
3. The assertions (facts and individuals) on all domains (ABox).

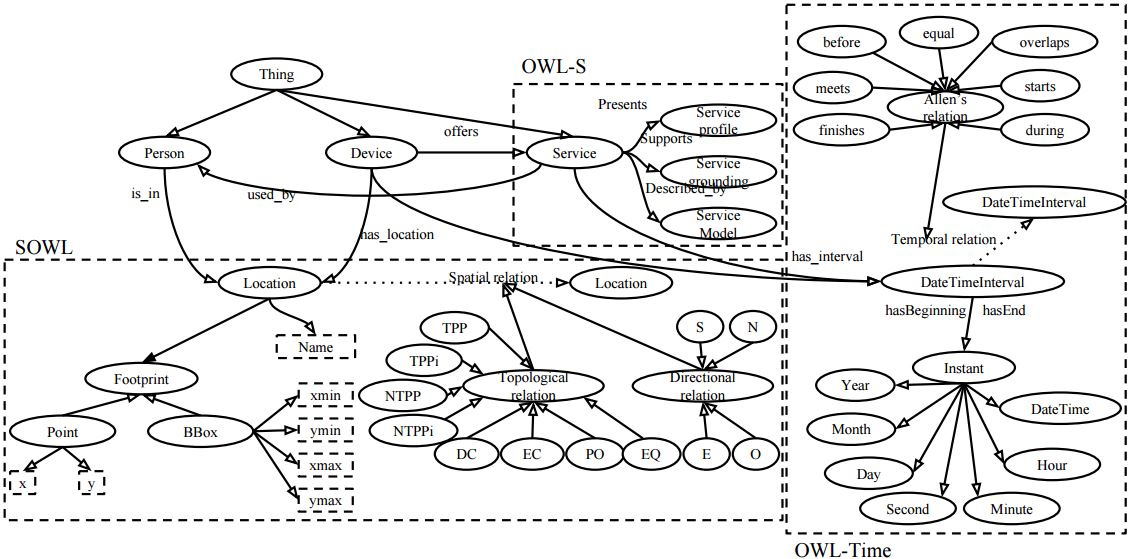


New devices discovery

Devices reappearance

Devices disappearance

**Figure 11** : KB knowledge content evolution



**Figure 12** : Upper level ontology

The context and its elements dynamicity is taken into account and managed at each of the three levels (**Table 2**). At the formal semantic annotation level, each device discovery generates an annotation describing the new device instance (ABox). In addition, when the device monitors a sensor, its annotation (contextual property) is updated real time with the sensor value and propagated (exteroceptive) to the coupled devices. A *time out* feature is put in place allowing to keep coherency between propagated properties and the context structure (if a device monitoring a sensor is defective or has disappeared from the environment, its properties being not propagated any more, all coupled devices are updated consequently by losing the property that was brought by the device).

At the KB level, the disappearance of a device does not remove the concepts he brought. Therefore, the amount of *TBox* elements in the KB will continuously increase while the amount of assertions (*ABox*) will vary in the devices discovery/disappearance cycle (Figure *11*).

|  |  |  |  |
| --- | --- | --- | --- |
| **Event** | **Impacts on G= (**Δ**, predicate)** | **Impacts on formal semantic annotations** | **Impacts on the KB** |
| **New device discovery** | Node addition | New concepts  New instances | + TBox  + ABox |
| “is linked to” link addition | Properties inheritance | ↻ ABox |
| “monitors” link addition | New properties  Properties inheritance | + ABox  ↻ ABox |
| Inferences→ ’is linked to’ link addition | Properties inheritance | ↻ ABox |
| **Device disappearance** | “is linked to” link suppression | Properties suppression  Properties inheritance update | - ABox  ↻ ABox |
| “monitors” link suppression |
| Node suppression | Properties and instances suppression  Properties inheritance update | - ABox  ↻ ABox |
| **Object in the device 's scope** | “monitors” link addition | New properties  Properties inheritance | + ABox  ↻ ABox |
| **Object outside the device’s scope** | “monitors” link suppression | Properties suppression  Properties inheritance update | - ABox  ↻ ABox |
| **Defective connection** | “is linked to” link suppression |
| “monitors” link suppression |
| **Changement de valeur de propriété** |  | Properties value update,  Properties inheritance update | ↻ ABox |
| “is linked to” link addition | New context | + ABox |

**Table 2** : Dynamic knowledge management

+ Add element ↻ Update element

­ Remove element

## WComp and SLCA model

WComp [44], developed by the Rainbow team, is a platform for service composition by assembling light components. This platform implements the SLCA model (Lightweight Service Component Architecture) [45] where the application is formed with an assembly of software components based on the LCA model (Lightweight Component Architecture) and services communicating using events. Containers contain assemblies representing composite services that are then manipulated to manage the application. A structural interface allowing to dynamically modify the assembly and a functional interface giving access to the functional services are exported.

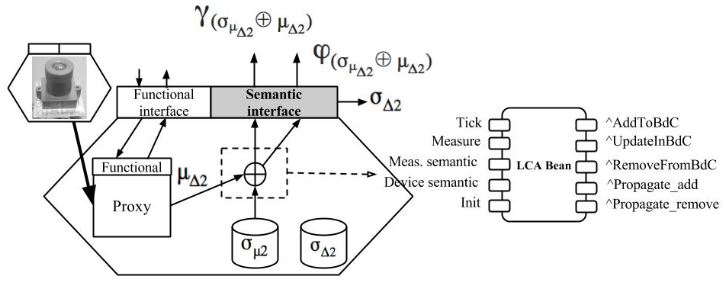


This platform is based on UPnP (Universal Plug and Play). Like DPWS (Device Profile for Web Services), this protocol allows to dynamically manage devices (discovery and disappearance) and registration to the proposed services. Thanks to an event based communication, it brings a good reactivity and a strong interoperability between devices and services. This platform is coupled with *Conquer* [45], a knowledge base encapsulated in an UPnP device.

So far, we have identified two device types: those associated with sensors (proprioceptive or exteroceptive), *producing* information about the objects or the environment, and those associated with objects, potentially *consuming* information from producers. For these two kind of devices, we create a composite web service integrating, in addition to a proxy to the object or the sensor, an LCA component managing the formal semantic annotations. Finally, in addition to structural and functional interfaces, we add a semantic interface.

### Composite web service for sensor

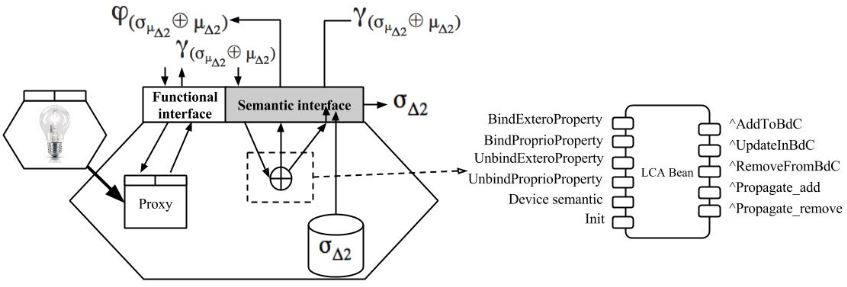
Coupled with a sensor, this device type produces physical measurements about the object or the environment. The LCA component then must be able to acquire the measures, process (fusion) and propagate it to consumers in the form of formal semantic properties γ. Device concepts (*TBox*) are sent to the KB as soon as the device is discovered with, eventually, its state φ (Figure *13*).



**Figure 13** : Composite web service for sensor

### Composite web service for object

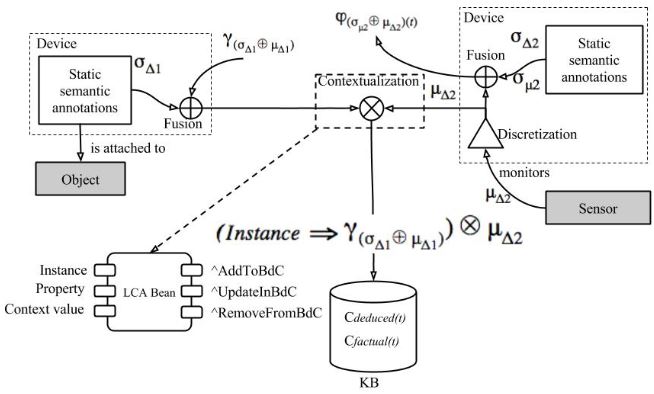
This device type consumes information from composite web service for sensor. The LCA component then must be able to acquire external properties produced by composite web service for sensor, to merge it with its own properties and send its global state φ to the KB (*ABox*) (Figure *14*). In addition, exteroceptive properties must be propagated to other devices coupled to this device. Device concepts (*TBox*) are sent to the KB as soon as the device is discovered.



**Figure 14** : Composite web service for object

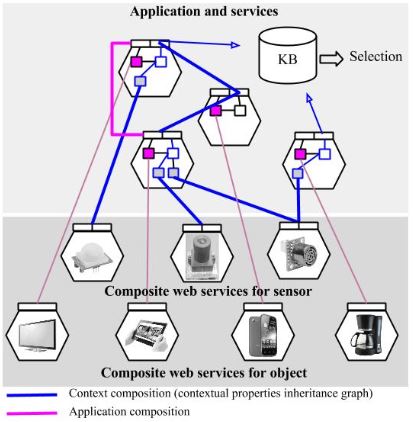
### Contextualization component

This component is used to specify that a device’s properties are valid only under a given context. The LCA contextualization component produces formal semantic annotations based on the context slices pattern (Figure 15). The properties γ of the device instance are only valid under the context given by a context value. For instance, the context value can be computed from a sensor providing a location. The location can be numeric (longitude/latitude) but in that case it may end up with a huge amount of contexts. A discretization element could then be added in order to reduce the amount of possible contexts (1st floor, meeting room, etc…).



**Figure 15** : Properties contextualization

Finally, these devices and associated services are orchestrated in an application:



**Figure 16** : Application composition

# CONCLUSION AND FUTURE WORK

In the field of ambient computing, software services selection mechanism is essential to ensure a continuous application coherency with the context and the desired functionality (service continuity).

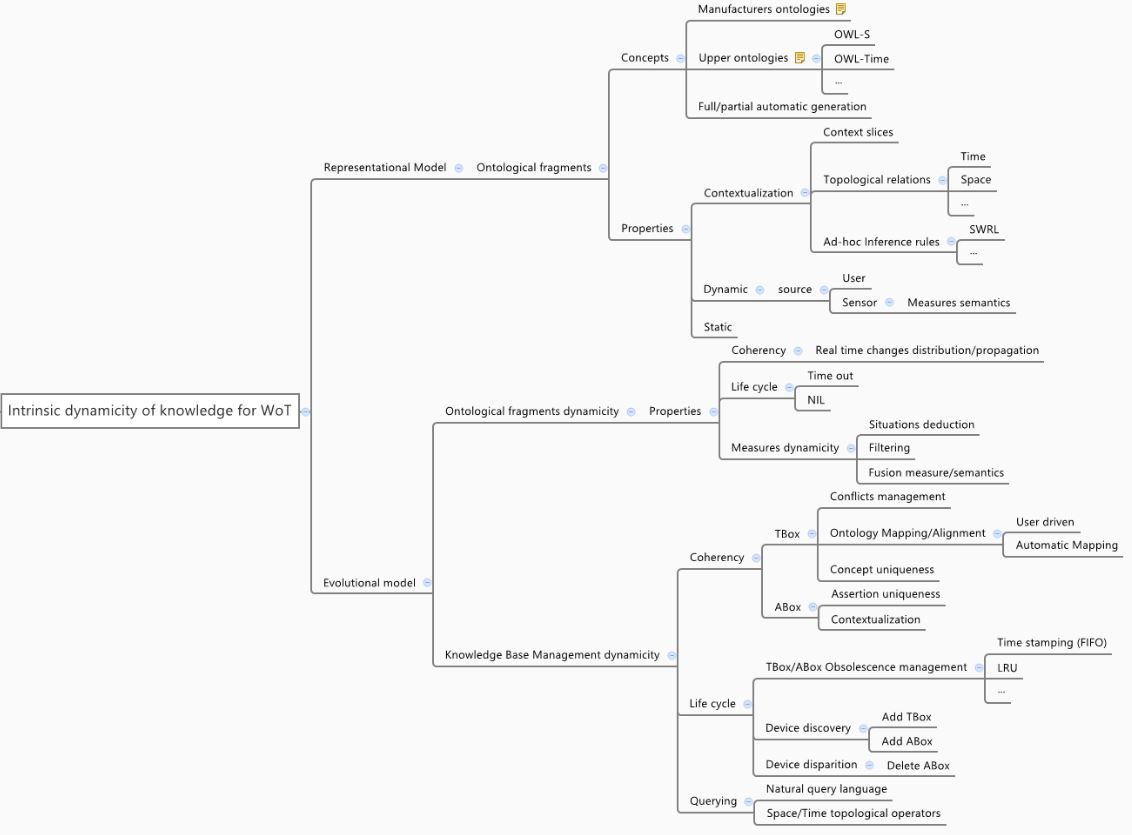
This selection is based on two main properties:

1. The relevance 𝑃ϛ of the functionalities offered by the available services versus a global functionality the application must reach,
2. The devices and services usability depending on their physical state and their availability over the time and space.

In this context, the use of dynamic, distributed and interoperable formal semantic annotations on devices and services, based on semantic web standards, has allowed us to build a dynamic knowledge representation on the devices, the services and the context split in three distinct levels:

1. The formal semantic annotations:
   1. Providing, in the one hand, a formal knowledge description of the devices and services domain’s and in the other hand, a formal knowledge description of the observations on the environment (exteroception) or the objects (proprioception),
   2. Providing contextual properties gathered from sensors observations and propagated (exteroceptive) through the contextual properties inheritance graph depicting the devices coupling.
2. The domains’ terminologies (*TBox*),
3. The assertions (facts and individuals) on all domains (*ABox*).

The context representation is improved through the implementation of a contextualized knowledge representation allowing to reason about the devices and services validity over the time, the space or any other relevant dimension in the context of real physical objects or devices (temperature, quality of service, etc...). The terminological (*TBox*) knowledge in the KB continuously grows while the amount of assertions (*ABox*) varies in the devices discovery/disappearance cycle. Coherency with the context is ensured by a real time sensors exteroceptive properties propagation to all devices annotations the sensor is coupled with.

This work allowed us to define a representational and an evolutional model for the dynamicity in WoT (Figure *17*):

**Figure 17** : WoT dynamicity model

From this model, we can extract several problematics linked to the chosen approach that should be studied:

1. Reasoning on time and space topological relationships requires changes in both the inference the query engines that need to be enhanced with *ad-hoc* inference rules and operators. In this context, the use of SWRL (Semantic Web Rule Language) [47] and SQWRL (Semantic Query-enhanced Web Rule Language) [48] may be considered.
2. Each device embedding the semantic description of its domain, defined by different people, there is a risk :
   1. To have ontological conflicts to manage and alignments to conduct [49]. Enriching formal semantic annotations with synonyms and equivalent concepts definitions may help [50],
   2. To reach system limits in space (due to semantic diversity) and time (query computation time),
   3. To not be able to apply optimal queries. Indeed, query optimality supposes a detailed knowledge about the concepts and the relationships described at each moment in the KB. It is unlikely to happen due to the distributed and heterogeneous formal semantic annotations characteristics which are not known in advance or not known at all. The use of a natural query language may then be a solution [51].
   4. The contextual properties inheritance graph, in the context of open and unbounded environments is more difficult to maintain dynamically and automatically.

Although scalable, **it is unlikely the KB content can indefinitely increase**. Limitations may occur in space (system memory limitation) and time (query processing time). A tradeoff has to be found in between the semantic heterogeneity handling capability (leading to KB terminological content increase improving further selected devices and services relevance), the intrinsic system capabilities (memory) and the user experience (query processing time).

The approach exposed in this work, based on formal semantic annotations describing the devices and services domains is, from the semantic interoperability handling standpoint of view, the most optimal, but, as we have seen, brings serious issues when dealing with ontologies alignment, conflicts management and query elaboration **along with an non-controlled growth of the knowledge volume in the KB**.

Other approaches may have to be considered on the representational model to limit these issues [54] and on the evolutional model by implementing a mechanism allowing to decide about knowledge obsolescence in the KB in order to maintain it below a certain boundary driven by the system memory but also by the selection mechanism responsiveness. The problematic to address is then: how to decide if a given knowledge is obsolete or not? Solutions and algorithms from operating systems memory management can be studied (FIFO, LRU, etc…) in the context of IoT.

# REFERENCES

1. Weiser, M. (1991). The computer for the 21st century. *Scientific american*, *265*(3), 94-104.
2. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.
3. Haller, S. (2010). The things in the internet of things. *Poster at the (IoT 2010). Tokyo, Japan*.
4. Scioscia, F., & Ruta, M. (2009, September). Building a Semantic Web of Things: Issues and Perspectives in Information Compression. In *ICSC* (pp. 589-594).
5. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.
6. Klyne, G., & Carroll, J. J. (2006). Resource description framework (RDF): Concepts and abstract syntax.
7. McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.
8. Prud’Hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. *W3C recommendation*, *15*.
9. Gruber, T. (1993). What is an Ontology.
10. The Description Logic Handbook: Theory, Implementation and Applications, Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider
11. De Giacomo, G., & Lenzerini, M. (1996). TBox and ABox reasoning in expressive description logics. *KR*, *96*, 316-327.
12. Welty, C., Fikes, R., & Makarios, S. (2006, May). A reusable ontology for fluents in OWL. In FOIS (Vol. 150, pp. 226-236)
13. Gutierrez, C., Hurtado, C. A., & Vaisman, A. (2007). Introducing time into RDF. Knowledge and Data Engineering, IEEE Transactions on, 19(2), 207-218.
14. Klein, M. C., & Fensel, D. (2001, July). Ontology versioning on the Semantic Web. In SWWS (pp. 75-91).
15. Carroll, J. J., Bizer, C., Hayes, P., & Stickler, P. (2005,May). Named graphs, provenance and trust. In Proceedings of the 14th international conference on World Wide Web (pp. 613-622). ACM
16. Stoermer, H., Bouquet, P., Palmisano, I., & Redavid, D. (2007). A context-based architecture for rdf knowledge bases: Approach, implementation and preliminary results (pp. 209-218). Springer Berlin Heidelberg.
17. Serafini, L., & Homola, M. (2012). Contextualized knowledge repositories for the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web, 12, 64-87.
18. <http://www.w3.org/TR/rdf11-datasets/>
19. Welty, C. (2010). Context Slices: Representing Contexts in OWL. In Proceedings of the 2nd Workshop on Ontology Patterns (WOP 2010).
20. Welty, C., Fikes, R., & Makarios, S. (2006, May). A reusable ontology for fluents in OWL. In FOIS (Vol. 150, pp. 226-236).
21. Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., & Savio, D. (2010). Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *Services Computing, IEEE Transactions on*, *3*(3), 223-235.
22. Mayer, S., & Basler, G. (2013, September). Semantic metadata to support device interaction in smart environments. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (pp. 1505-1514). ACM.
23. S. Mayer, D. Guinard, and V. Trifa. Searching in a Web-based Infrastructure for Smart Things. In Proceedings of the 3rd International Conference on the Internet of Things (IoT2012), Wuxi, China, Oct. 2012.
24. Gensel, J., Villanova-Oliver, M., et Kirsch-Pinheiro, M. (2008). Modèles de contexte pour l’adaptation à l’utilisateur dans des Systèmes d’Information Web collaboratifs. In 8èmes Journées Francophones d’Extraction et Gestion Des Connaissances (EGC’08),
25. J. Strassner, Y. Liu, M. Jiang, J. Zhang, S. van der Meer, M. Foghlu, C. Fahy, and W. Donnelly, Modelling context for autonomic networking, MUCS 2008, p. 10, april 2008.
26. Dey, A. K., Salber, D., Futakawa, M., & Abowd, G. (1999). *An architecture to support context-aware applications*. GVU Technical Report GIT-GVU-99-23.
27. Tarak Chaari, Frédérique Laforest, and André Flory. Adaptation des applications au contexte en utilisant les services web. In *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pages 111–118. ACM, 2005
28. Anind K Dey. *Providing architectural support for building context-aware applica- tions*. PhD thesis, Georgia Institute of Technology, 2000.
29. James F. Allen. Maintaining knowledge about temporal intervals. Commun. ACM, 1983.
30. Hobbs, J. R., & Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, *3*(1), 66-85.
31. Pan, F., & Hobbs, J. R. (2005, May). Temporal Aggregates in OWL-Time. In *FLAIRS Conference* (Vol. 5, pp. 560-565).
32. Cohn, A. G., Bennett, B., Gooday, J., & Gotts, N. M. (1997). Qualitative spatial representation and reasoning with the region connection calculus. GeoInformatica, 1(3).
33. Max J. Egenhofer and Herring John R. Categorizing binary topological relationships between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono, 1991.
34. Y. Katz and B. Cuenca Grau. Representing Qualitative Spatial Information in OWL-DL. In Proc. of 1st OWL: Experiences and Directions Workshop (OWLED2005), Galway, Ireland, 2005.
35. Stocker, M., & Sirin, E. (2009, October). PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. In *OWLED* (Vol. 529).
36. Batsakis, S., & Petrakis, E. G. (2010, September). SOWL: spatio-temporal representation, reasoning and querying over the semantic web. In *Proceedings of the 6th International Conference on Semantic Systems* (p. 15). ACM
37. O’Connor, M. J., & Das, A. K. (2011). A method for representing and querying temporal information in OWL. In Biomedical engineering systems and technologies (pp. 97-110). Springer Berlin Heidelberg.
38. Katz, Y., & Grau, B. C. (2005). Representing qualitative spatial information in OWL-DL. In In Proceedings of the OWL: Experiences and Directions Workshop.
39. O'Connor, M. J., & Das, A. K. (2009, October). SQWRL: A Query Language for OWL. In OWLED (Vol. 529)
40. Battle, R., & Kolas, D. (2011). Geosparql: enabling a geospatial semantic web. Semantic Web J, 3(4).
41. Rey, G., & Coutaz, J. (2002, November). Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte. In Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine) (pp. 105-112). ACM.
42. Koide, S., Aasman, J., & Haflich, S. (2005, November). OWL vs. object oriented programming. In *International Workshop on Semantic Web Enabled Software Engineering*.
43. Vergoni, C., Rey, G., Lavirotte, S., & Tigli, J. Y. (2011). Construction Bottom-up d’applications ambiantes en environnements partiellement connus à priori. *planning (workflow dynamique)*, *13*, 10.
44. Tigli, Jean-Yves, Riveill, Michel, Rey, Gaëtan, *et al.* A middleware for ubiquitous computing: Wcomp. *Projet RAINBOW, Rapport de recherché, ISRN I3S/RR–2008-01–FR*, 2008.
45. Hourdin, V., Tigli, J. Y., Lavirotte, S., Rey, G., & Riveill, M. (2008, September). SLCA, composite services for ubiquitous computing. In *Proceedings of the International Conference on Mobile Technology, Applications, and Systems* (p. 11). ACM.
46. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., ... & Sycara, K. (2004). OWL-S: Semantic markup for web services. *W3C member submission*, *22*, 2007-04.
47. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, *21*, 79.
48. O'Connor, M. J., & Das, A. K. (2009, October). SQWRL: A Query Language for OWL. In *OWLED* (Vol. 529).
49. Klein, M. (2001, August). Combining and relating ontologies: an analysis of problems and solutions. In *IJCAI-2001 Workshop on ontologies and information sharing* (pp. 53-62).
50. Choi, N., Song, I. Y., & Han, H. (2006). A survey on ontology mapping. *ACM Sigmod Record*, *35*(3), 34-41.
51. Kaufmann, E., & Bernstein, A. (2010). Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web*, *8*(4), 377-393.
52. Ouksel, A. M., & Sheth, A. (1999). Semantic interoperability in global information systems. *ACM Sigmod Record*, *28*(1), 5-12.
53. Swartout, B., Patil, R., Knight, K., & Russ, T. (1996, November). Toward distributed use of large-scale ontologies. In *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems* (pp. 138-148).
54. Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001, August). Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 workshop: ontologies and information sharing* (Vol. 2001, pp. 108-117).