Dynamic SWoT

Three ways to enrich knowledge and gain relevancy for device discrimination

***Abstract*—Ambient intelligence (AmI) term refers to "intelligent" or "smart" environments and systems built upon software services integrated in devices placed in the environment, wore by users or embedded in our everyday life objects (Internet of Things, IoT). The physical, heterogeneous and highly dynamic nature of the environment, the users and the objects, implies a twofold challenge for these systems: (1) adaptability to the dynamicity of the environment and (2) ability at ensuring the service continuity. To address these challenges, there has been a growing interest at using semantic web technologies to semantically annotate devices and give AmI systems the ability, not only to gather the data about their environment, but also to understand and reason about it (Semantic Web of Things, SWoT). As the knowledge is embedded in the devices, it inherits their dynamicity. We have identified three orthogonal levels of knowledge dynamicity: (1) the *property* level handling the devices and environment physical properties, (2) the *instance* level handling devices discovery and disappearance and (3) the *terminological* level handling conceptual knowledge addition. We introduce in this paper how these dynamicity levels could be leveraged in SWoT-based systems in order to address aforementioned challenges. Then, we focus on the terminological dynamicity level because of its capability at increasing AmI systems intelligence by continuously enriching the knowledge and therefore the selected services relevancy as devices are discovered in the environment. We propose a dynamic knowledge management model based on SWoT for AmI able to manage the set of ABox/TBox knowledge brought by the several discovered devices using heterogeneous ontology approach. The model has been validated from dataset and relevance metrics.**

Keywords—Semantic web of things (SWoT); Knowledge representation and management; Devices and services selection.

I. INTRODUCTION

The last decade achievements in computer hardware miniaturization and power consumption reduction has permitted ambient intelligence vision [1] to become a reality by the emergence of *devices* connected to internet and integrated in our everyday life *objects* (chair, table, lamp, etc…) and physical *environments* (house, building, vehicle, etc…). These devices implement *resources* interacting with objects (*actuator*) and/or gathering *data* (*sensor*) about themselves, the objects or the environment (Internet of Things, IoT) [2][4]. Access to these resources is achieved through *services* exposing their interfaces and allowing communication with the digital world. Services are therefore the basic blocks of the ambient computing systems, made available directly to the end users or working in concert from a composition. At the heart of these systems, the devices and services selection mechanism is key at insuring the system *service continuity* (as devices are unpredictably available or unavailable), and *adaptability* to the dynamicity of the environment, theater of physical phenomena (space, time, temperature, quality of service, etc…) subject to evolutionary principles. Services have then been incorporated into a standardized Web service architecture (Web of Things, WoT) allowing well known Web technologies to be leveraged on the data (annotating, searching, etc…) [3]. A step further, there is now a growing interest in the community to evolve from WoT to SWoT (Semantic Web of Things) by using the semantic web standard technologies and tools [5] providing a formal *knowledge* understanding of the data along with querying and reasoning techniques.

Semantic web technologies allow defining the knowledge about devices properties, instances and their associated concepts (terminology). Physical objects induce dynamic evolution of this knowledge. In this paper, we investigate more particularly the incremental evolution of the terminological knowledge as devices are discovered leading to more and more services discrimination effectiveness.

Firstly, in section II, after having described the main semantic web concepts, we identify those that can be leveraged to dynamically bring new knowledge from physical objects connected to the web (WoT). The wide physical objects diversity introduces knowledge heterogeneity. Then, we challenge some ontology architectures according to their heterogeneity handling capability. From this study we finally propose a dynamic knowledge management model for SWoT. In section III**,** we describe some proven metrics in use in the information retrieval (IR) domain and applied to our work in order to measure the selection mechanism effectiveness as the knowledge is increased in a knowledge base. A case-study is detailed and implemented on our experimentation platform to get associated results discussed in section IV. In section V we present some related works and, finally, we conclude in section VI by summarizing the results and introducing the future work.

II. CONCEPTUAL FOUNDATIONS AND CONTRIBUTION

A. *Semantic web concepts*

Before going further, it seems appropriate, at this point, to discuss the several knowledge representation concepts used in the semantic web domain and applied to the SWoT domain.

1) Ontology

The knowledge about the environment, the devices and the services is formally and explicitly described using *ontologies* [5][6], hierarchically structuring the concepts. The main elements composing an ontology are:

a) ***Classes*** (or concepts) and sub-classes hierarchically organized according to a taxonomy (i.e. Device, Service, Display, Speaker, etc…),
b) ***Properties*** allowing to define ***facts*** or ***relations*** between classes. There are mainly two property types:
    i. Object property that defines a relationship between two instances of a class or between classes,
    ii. Data types properties as a relation between a literal value and a class instance.
c) ***Class instances*** (class individual) which may take the characteristics defined by the properties.

2) Vocabulary

The differences between "ontology" and "vocabulary" is subtle[1]: While an ontology formally and strictly describes the concepts and relations of a given domain, a vocabulary enumerates terms without a strict formalism (context-less) allowing them to be shared and used by several domains.

3) Knowledge base

An ontology can be seen as a meta-system for a knowledge base (KB) as a description of the knowledge representation it contains. KB includes facts and individuals of all the defined concepts from which a reasoning engine is used to derive implicit knowledge from explicit knowledge. Knowledge in KB is structured at two description levels, *ABox* and *TBox*, respectively defining *assertions* on the instances and individuals, and the general concepts *terminologies.*

B. *The three SWoT knowledge dynamicity levels*

From the ontology and knowledge base previously described, we denote three main elements: (1) property, (2) instance (*ABox*) and (3) concepts (*TBox*) that can independently and dynamically bring new knowledge in the context of SWoT.

1) The property dynamicity level

Devices placed in the environment, wore by users or embedded in our everyday life objects are semantically annotated. The annotations can bring values gathered from sensors representing the users, the environment or the objects physical states (temperature, location, battery level, etc…). For instance, in Fig. 1, the annotation brings the oven's temperature. The annotation content is updated as the oven temperature value increase or decrease. This dynamicity level is mainly exploited in context aware applications allowing queries such as:

"*What is the current temperature of the oven?*"

This dynamicity is intensively used in context-aware applications [31].

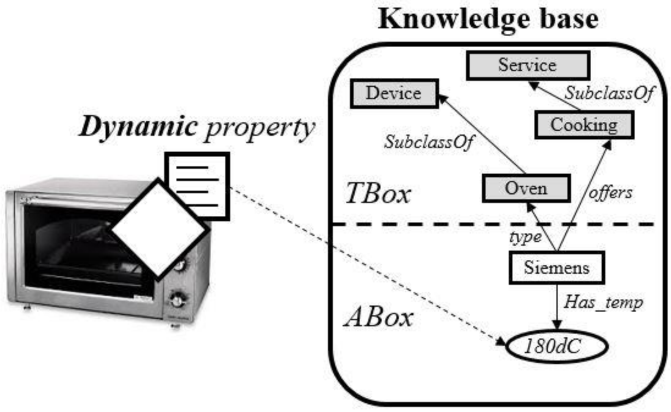

Fig. 1. The property dynamicity level

Back to the AmI identified challenges, context-aware applications are good at adapting to the dynamicity of the environment. Quid of the second challenge regarding the service continuity? Let see how it could be improved with the introduction of the two next knowledge dynamicity levels.

2) The instance dynamicity level

In a closed environment all devices are known. Therefore, all devices and services instances can be populated in the knowledge base (static ABox) at design time. In AmI environments and systems, devices are not known *a priori* and unpredictably appear or disappear in the environment. A devices discovery mechanism is necessary [7][8][9][24], allowing to dynamically keep the knowledge base up to date with the instances of the devices and the services as they appear or disappear in the environment (*knowledge base population*).

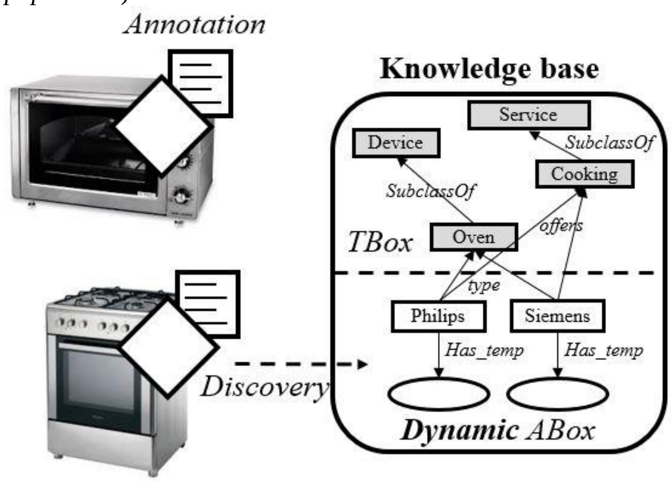

Fig. 2. The instance dynamicity level

[1]http://www.w3.org/standards/semanticweb/ontology

At each instant, the knowledge base content is a snapshot of the environment permitting queries like:

"*What are currently the domestic appliances present in the kitchen?*"

3) The terminological dynamicity level

Properties and instances associated concepts are all defined from classes and relations between classes in the ontologies and the knowledge base (*TBox*). Those concepts and relations define the vocabulary necessary for the machine to understand the meaning of all the instances and the properties in the knowledge base, and possibly infer new implicit knowledge. In general, the vocabulary is bounded to a particular application domain limiting the expressivity of the requests to the classes and relations defined in the vocabulary. When dealing with real world heterogeneous environments and objects like it is the case in AmI systems, it is unlikely that a vocabulary defining all the world concepts and relations can be available. It is therefore necessary to increment the vocabulary (*knowledge base extension*) [10]. This additional knowledge could be either brought by the users [9],

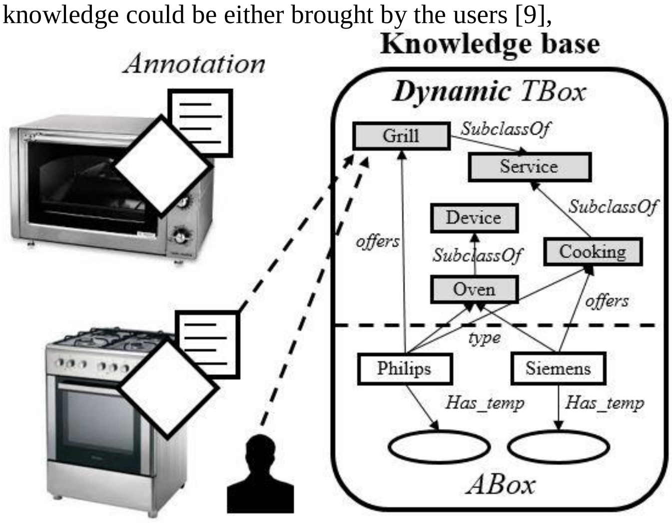

Fig. 3. The terminological dynamicity level

or, for instance, by capitalizing on the devices interactions with other devices: introducing new concepts in the knowledge base from devices and services annotations as they appear in the environment can permit to enhance the initial vocabulary with new concepts and relations between previous concepts (*knowledge merging*). It allows to add more expressivity to the queries allowing to refine the spectrum of the selected devices and services. For instance, an initial query like: "*What are the domestic appliances available allowing to cook*?" corresponding to the Fig. 2 would return two devices (both ovens being linked to the concept "Cooking"). If one of the device adds the new concept "Grill" (Fig. 3), the initial query can be refined with: "*What are the domestic appliances available to grill?*" returning only one result.

Note that along with additional concepts and relations, inference rules can be added as well to refine the knowledge by inferring new relations or adding new properties to the devices and services instances.

C. Ontology architectures analysis for SWoT

Based on the ontology concepts, authors in [11], depict three possible ontology architectures to represent and manage the knowledge. We detail and compare below their abilities in supporting emergent knowledge (*ABox*, *TBox*) and providing scaling capabilities as devices and services appear and disappear in AmI environment.

1) Single ontology approach

With this approach, a single ontology is used to formally and strictly describe *all* the concepts and relations of a given domain. Such an ontology is not supposed to be modified. Although this solution is the best at achieving interoperability, in the context of IoT, an accepted and validated ontology describing the whole world's concepts and relations is unlikely to happen [12].

2) Multiple heterogeneous ontologies approach

Each device defines and embeds its own domain ontology based on its own vocabulary (heterogeneous ontology network). In the context of IoT, although good in supporting knowledge addition, the lack of a common vocabulary leads the necessity of implementing ontologies alignment mechanisms (at the first stage of ontology matching [15] and mapping [16]) in order to smooth the semantic heterogeneity and increase the interoperability. This limits the scaling capability [14] of this approach due to the potential incoherency of the resulting ontology [13] with potential impacts on further devices and services selection effectiveness. The lack of a common vocabulary may also lead to degrade new knowledge inference, the vocabulary being the basic building blocks used by the inference engines[1]. Finally, the alignment process computation time may dramatically increase and degrade the overall service composition mechanism response time and consequently the user experience as the knowledge grow over the time.

3) Hybrid approach

Each device defines and embeds his own domain knowledge built from a common vocabulary. This approach avoids ontologies alignment and inferences issues previously depicted since all of them are based on a common and shared vocabulary. Therefore, this approach is good at aggregating new knowledge. This being said, there is currently no agreed and widely used common and global vocabulary available.

Instead, some efforts are currently made at enumerating existing open vocabularies from several domains, proven to follow the W3C best practices recommendations [LOV[2], LOV4IoT[3]]. In this context, this approach mixes drawbacks depicted by the single ontology approach (inexistence of a global vocabulary is equivalent to the inexistence of a global ontology) and the multiple heterogeneous ontologies approach

[1]http://www.w3.org/standards/semanticweb/ontology
[2]http://lov.okfn.org/dataset/lov/
[3]http://www.sensormeasurement.appspot.com/?p=ontologies

(heterogeneous ontologies alignment problem is equivalent to heterogeneous vocabularies alignment problem).

D. Dynamic knowledge management model for SWoT

Like some authors in the literature [12] we envision each manufacturer to autonomously and independently develop ontologies describing their products from heterogeneous vocabularies. This tends to move toward the heterogeneous ontology approach. Ontologies alignment limitations inherent to this approach are currently well addressed by the semantic web community [13][14][15][16]. Going a step further we also envision that along with a general purpose alignment engine, custom alignments features will also be needed for specific cases. For instance, within its utilization context, an application may consider that "Tv" concept is equivalent to "Display" concept whereas this assumption cannot be applicable to all applications. In addition, syntactical problems in device descriptions may occur (i.e. Television vs television vs telvision vs televisn vs …). Utilizing, in addition of a general purpose alignment engine, an application specific synonyms table might be considered (i.e. the uniform descriptions in [17]). The knowledge management model is presented in Fig. 4:

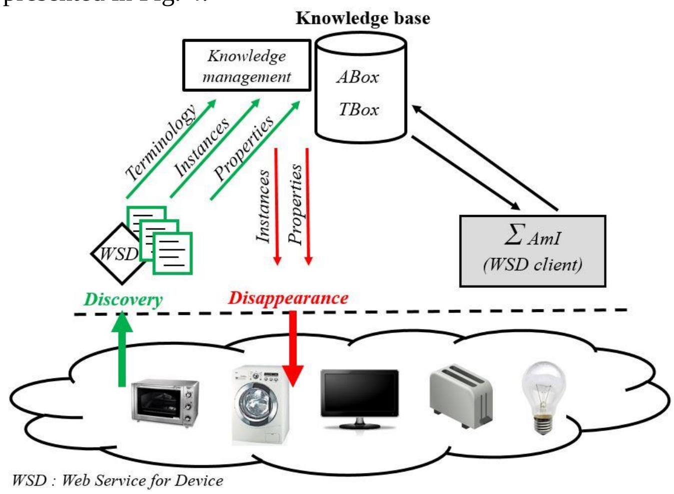

Fig. 4. Dynamic knowledge management model for SWoT

As depicted in the Fig. 4, when a device disappear from the environment, only the associated instance and properties are removed from the knowledge base.

III. Methodology and Metrics

A. Measuring the devices and services selection effectiveness

As we want to increase the devices and services selection effectiveness, we need some associated metrics. Measuring the services selection effectiveness corresponds, in our case, to measure the *relevance* of the selected devices and services instances in the context of user's needs or a composition. The term relevance (or adequacy, pertinence) is defined as follow[1]: "*A state or quality of being to the purpose*". It might be a complex task at a first glance to measure the relevance but doing a parallel with IR domain [18], our problematic can be summarized as considering that the knowledge base contains relevant and nonrelevant devices and services instances for a given query and then apply metrics used in IR domain applications. There are three main metrics commonly used to measure relevance: (1), the *precision*, (2) the *recall* and (3) the *fallout* [19].

1) Precision

The precision P corresponds to the ratio between the number α of relevant retrieved devices and services instances over the total number σ of retrieved devices and services instances in response to a query q.

$$\text{Precision } P = \alpha / \sigma$$

2) Recall

The recall R corresponds to the ratio between the number α of retrieved and relevant devices and services over the total number Δ of relevant devices and services in the knowledge base.

$$\text{Recall } R = \alpha / \Delta$$

3) Fallout

The fallout F is the ratio between the returned devices and services instances out of those nonrelevant.

$$\text{Fallout } F = (\sigma - \alpha) / (O - \Delta)$$

B. Case study

This section motivates the need for leveraging instance and terminological dynamicity levels in addition to the property dynamicity level in order to increase the AmI systems devices and services selection mechanism effectiveness. We consider in this scenario an environment initially comprising several home appliances: a portable TV and a hi-fi system installed in the living room. These appliances embed devices allowing them to be monitored and controlled by an AmI system. Devices provide semantic annotations describing: (1) the appliance power consumption (as a data property), (2) some *incomplete* terminological concepts about their domains. The AmI system has to select the less energy consuming appliances allowing to play a music track. Two problems occur in the context of selecting the relevant devices and services instances for that purpose from the available knowledge:

a) What criteria has to be used to consider that an appliance is too much greedy? Using the raw power consumption property and an arbitrary trigger may lead to inaccurately select the devices…
b) The portable TV terminological annotations describe the TV concept as a "Display" (not a "Speaker" as it is the case for the hi-fi concept).

As a consequence, depending the power consumption trigger used, only the hi-fi system appliance might be selected. This is problematic because the portable TV is also relevant in that case (considering that a TV is able to play a music track). Worst, the portable TV consumption may be lower than the hi-fi system consumption in which case only the portable TV is pertinent, not the hi-fi system.

The user buy a new home TV to be installed in the living room. This appliance embeds a device and semantic

[1]The Cambridge English Dictionary, London, English Edition, 1990
[2]http://en.wikipedia.org/wiki/European_Union_energy_label

annotations bringing new terminological knowledge about the "TV" concept. It describes a "TV" as being a "Display" AND a "Speaker". This new knowledge is brought to the knowledge base upon this new device discovery (instance dynamicity level) and the "TV" concept is updated with the new class relation (Terminological level dynamicity). Executing the previous request allows now the hi-fi system and both TVs to be selected as valid candidates to play a music track.

Let's consider now that the user install a new electric meter in the environment. This electric meter brings new knowledge about the energy classification for home appliances that can be based, for instance, on the European Union energy label[2]. This new knowledge is brought in the form of inference rules defined in the device annotations and enriches the terminological elements (TBox) of the knowledge base (terminological level dynamicity) upon device discovery (instance dynamicity level). The reasoning engine then infers, for each device instance in the knowledge base a new property defining the European Union energy label from the power consumption property. It permits to query and select more accurately the devices instances based on a parameter making sense in the domain of the energy consumption.

IV. Evaluations and results

A. AmI system validation plateform

The previously described scenario has been tested using the CONTINUUM platform[1] enhanced thanks to the contribution presented in this paper.

WComp middleware [20] for service composition by assembling light components is at the heart of this platform. WComp implements the SLCA model (Lightweight Service Component Architecture) [25] where the application is formed with an assembly of software components based on the LCA model (Lightweight Component Architecture) and services communicating using events. 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. This platform is coupled with Conquer knowledge base [21] built on top of Jena API [22]. This knowledge base has been encapsulated in a web service for device (Universal Plug and Play, UPnP) and enhanced with a reasoning engine [23] able to infer on SWRL rules (Semantic Web Rule Language) and some real time ontology metrics monitoring capabilities. Using the aforementioned platform, composite web services have been created for each device, exposing an interface allowing the knowledge base to retrieve the semantic annotations upon device discovery. The annotations are written following the RDF/XML syntax [26].

For the sake of simplicity and because alignment problematics are outside the scope of this work, we have implemented the heterogeneous ontology approach and replaced the alignment engine with a synonyms table aligning the different terms used in our experiments allowing to easily merge the concepts brought by the devices semantic annotations.

It still permit to demonstrate how instance and terminological dynamicity levels can be used to increase the devices and services selection mechanism effectiveness by incrementally enriching a knowledge base providing the knowledge in the form of:

a) Additional concepts,
b) Inference rules which, once processed through the reasoning engine, enrich the knowledge with new inferred data or object properties.

Then, by making persistent the terminological knowledge and inference rules in the knowledge base, the selection mechanism effectiveness enhancement is then achieved in two ways:

a) By refining the query with higher expressiveness leading to reduce the spectrum of possible selection candidates,
b) Given a fixed query, increasing the spectrum of possible selected candidates over the time.

B. Devices discovery and knowledge base enrichment

Following the scenario previously described, two devices are first added in the environment: (1) a portable TV with 8W power consumption, (2) a Hi-fi sound player with 28W power consumption. Those devices are then discovered and their semantic annotations are used to enrich the knowledge base (Fig. 5). We consider that only the portable TV is relevant to play a music track with the lower power consumption. At this point, a query is executed to retrieve "Speaker" devices type with a power consumption lower than 30 watts (value arbitrary chosen):

```
SELECT ?inst ?comment
WHERE
{
?device rdf:type core:Device .
?device core:is_a core:Speaker .
?device core:has_power_consumption ?consumption
?inst rdf:type ?device .
?inst rdfs:comment ?comment
FILTER (?consumption < 30)
}
```

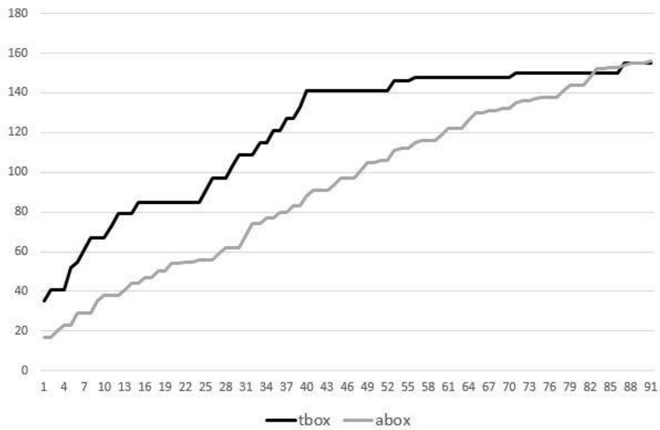

Fig. 5. Devices discovery: ABox/TBox enrichment

[1]Project for service continuity in ubiquitous and mobile computing - French national research agency - ANR-08-VERS-0005.

Only the Hi-Fi sound player device is returned (Step A):

```
?inst =<uuid:85079199-0e2f-4ac3-9e50-dcab2df1294b>
?comment = "Hifi sound player"
```

The only relevant device is actually the portable TV with 8W power consumption but with the current knowledge, this device cannot be selected. A new device is then added (Philips TV) specifying that “TV” concept is a “Display” AND a “Speaker”. As this knowledge is merged with the knowledge base content, the “TV” concept has been enriched with a new relation and all “TV” devices type inherit this new relation. We re-execute the previous query to retrieve “Speaker” devices type:

```
SELECT ?inst ?comment
WHERE
{
?device rdf:type core:Device .
?device core:is_a core:Speaker .
?device core:has_power_consumption ?consumption
?inst rdf:type ?device .
?inst rdfs:comment ?comment
FILTER (?consumption < 30)
}
```

This time, three devices are returned: (1) the newly added device (Philips TV), the previously selected device (Hi-fi sound player), and the first portable TV device that was previously not selected. It is now selected as the “TV” concept has been enriched with the new relation specifying that it is a “Speaker” (Step B).

```
?inst = <uuid:166cd648-952a-4690-8913-3bfd3f7a7f88>
?comment = "Philips 8100 series television"
?inst = <uuid:10f56a9f-f08c-493a-b1cd-afe4a38d2024>
?comment = "Portable television"
?inst = <uuid:85079199-0e2f-4ac3-9e50-dcab2df1294b>
?comment = "Hifi sound player"
```

Finally, an electric counter device is added bringing new knowledge about the energy classification for home appliances that can be based, for instance, on the European Union energy label. This new knowledge is added in the form of SWRL rules. A new query can be executed to show up the inference engine execution results (inferring the property “has_consumption_category”):

```
SELECT  ?c ?p ?j
WHERE
{
?i core:has_power_consumption ?p .
?i rdfs:comment ?c .
?i tst:has_consumption_category ?j
}
```

The new property created allows to classify the devices power consumption under term and values making sense in the power consumption domain:

```
?c = "Hifi sound player"
?p = "28"^^xsd:int
?j = "C"
?c = "Portable television"
?p = "8"^^xsd:int
?j = "A"
?c = "Philips 8100 series television"
?p = "19"^^xsd:int
?j = "B"
```

We are now able to execute a more relevant query and base the selection on this new property:

```
SELECT ?inst ?comment ?category
WHERE
{
?device rdf:type core:Device .
?device core:is_a core:Speaker .
?device tst:has_consumption_category ?category
?inst rdf:type ?device .
?inst rdfs:comment ?comment
FILTER (?category = ”A”^^xsd:string)
}
```

With all the added knowledge, the most relevant device is selected (Step C):

```
?inst = <uuid:10f56a9f-f08c-493a-b1cd-afe4a38d2024>
?comment = "Portable television"
?category = "A"
```

C. Devices disappearance

To take advantage of the terminological dynamicity level, the *TBox* knowledge addition is made persistent in the knowledge base. By doing this, when a device disappear only related assertion elements (ABox) are removed from the knowledge base (Fig. 6). We can therefore capitalize on the devices interactions with other devices (a kind of ontology learning [27]). The new knowledge addition allows maintaining the service continuity. Indeed, in our scenario, the fact to have enhanced the “TV” concept with the new relation allows to still select the portable TV even after the Philips TV disappearance. Also, by keeping persistent the inference rules, the inferences computations categorizing each device based on its consumption are still made even after the electrical meter device disappearance.

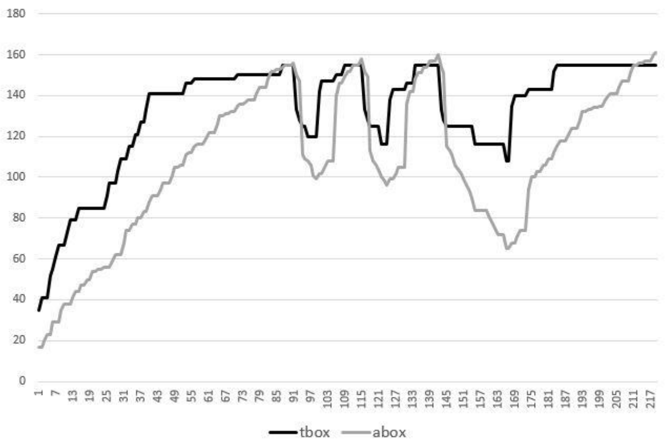

Fig. 6. Devices discovery and disappearance

Note that some TBox elements are removed when a device disappear. This is due to the reasoning engine which adds terminological elements on its own and remove it when instances are removed. Anyway, those TBox elements are not the ones added with the devices semantic annotations.

D. Results summary

The TABLE I. summarizes the evolution of the precision, recall and fallout measures (section III.A) as the knowledge base is enriched with addition terminological elements.

Step	**α**	**σ**	**Δ**	***O***	***P***	***R***	***F***
A	0	1	1	2	0.0	0.0	1.0
B	1	3	1	3	0.33	1.0	1.0
C	1	1	1	4	1.0	1.0	0.0

TABLE I. PRECISION, RECALL AND FALLOUT RESULTS

This new knowledge permits to increase the queries expressivity and clearly improve the relevancy of the selected devices over the time. In our case the precision P, the recall R are improved (from 0.0 to 1.0) as new knowledge is added, ending-up with a perfect fallout F (from 1.0 to 0.0).

V. RELATED WORKS

To the best of our knowledge no paper have identified and exploited the instance and terminological dynamicity levels as a way to incrementally increase the knowledge by capitalizing from devices interactions with other devices over the time as we did in this work. This is maybe due to the fact that the knowledge heterogeneity is addressed as an issue rather than an opportunity to enrich overall knowledge of the system.

Several projects aimed at using semantic annotations to leverage semantic web technologies tools [5] providing a formal knowledge understanding of the devices and the services along with querying and reasoning techniques. We firstly describe works dealing with heterogeneous ontologies in the AmI domain and review the approaches used to manage heterogeneous knowledge sharing and enrichment. The AmI ATRACO project [12] is built around ATRACO agents exchanging data between each other. Each agent owns an ontology aimed at describing its local knowledge. An alignment engine has then been developed (ontology matching) in order to address the syntactic and semantic heterogeneity between agents local ontologies. Authors envision that a global, agreed, and validated ontology, in the AmI domain is unlikely to happen, and that, more realistically, manufacturers will develop their own ontologies, ending-up with countless heterogeneous ontologies. The alignment technic developed aimed at improving knowledge sharing between the agents. In [28] authors expose some challenges relative to SWoT domain. One of the identified challenges, is the ability, for the smart products, to be able to learn new emergent knowledge from user's interactions and feedbacks (user's preference learning) or from wiki pages. In [29] authors address the problem of gathering knowledge in order to improve user's interactions with smart products. They propose to use semantic annotations to enrich smart products workflows aimed at defining tasks and participants in several contexts. Authors highlight the problem of the domain ontologies shipped with smart products that have to be enriched over the time with the knowledge about user's environment and interests. They consider possible changes at the ontology level (ontology extension) and the instance level (ontology population). The instance level described here corresponds to the knowledge base level. While they motivates the need of such knowledge evolution, no automatic mechanism is proposed for the enrichment other than manual.

Other works have been using the hybrid ontology approach.

In [30] authors have defined layered ontologies defining a common vocabulary from which semantic annotations can be defined and deployed on devices. The authors highlight the need for a standardization committee and the need, for the manufacturers to develop their device ontologies based on the defined vocabulary. As it is a good solution from an interoperability standpoint of view it is unlikely that such a standardization could occur.

As a summary, two mains facts are depicted in the studied works:

a) This is unlikely that a global ontology describing all the world concepts is likely to happen. Manufacturers will ship their products with heterogeneous ontologies developed on their own,
b) This knowledge will have to be enriched during the product life, either from users (feedbacks, preferences, etc…) or from their interactions with other devices in the environment.

VI. CONCLUSIONS AND PERSPECTIVES

Semantic web technologies are gaining interest in the domain of ambient intelligence systems. In the SWoT context, devices are semantically annotated, providing those systems not only the ability to gather the data about their environment, but also the knowledge to understand and reason about it.

The dynamic knowledge evolution cannot be ignored anymore as Physical objects are plunged in real physical environments. The first contribution of the present work is to have identified three orthogonal knowledge dynamicity levels: (1) the *property* level handling the devices and environment physical properties, (2) the *instance* level handling devices discovery and disappearance and (3) the *terminological* level handling conceptual knowledge addition. The second contribution concerns a new dynamic knowledge management model for SWoT based on the three aforementioned levels. We focused more particularly on the terminological dynamicity level, its capability at increasing AmI systems intelligence, and therefore the selected services relevancy. The model has been validated using proven metrics and results on a case-study.

Nevertheless, as the knowledge increases, it is unlikely that the KB content can indefinitely increase. As devices are embedded in everyday life objects, and considering their low available computational resources, limitations may occur in space (system memory limitation) and time (query processing time). A tradeoff will have to be found in between handling the semantic heterogeneity, the intrinsic system capabilities

[1]http://www.smartproducts-project.eu/

(CPU, memory) and the user experience (query processing time). Also, care will have to be taken on the data validity over the time (obsolescence management).

ACKNOWLEDGMENT

Our thanks go to Olivier Corby, Isabelle Mirbel, Amélie Gyrard, Gaëtan Rey and the support of EDF Chatou in U-insither project.

REFERENCES

[1] Weiser, M. (1991). The computer for the 21st century. *Scientific american*, *265*(3), 94-104.

[2] Haller, S. (2010). The things in the internet of things. *Poster at the (IoT 2010). Tokyo, Japan*.

[3] Guinard, D., Trifa, V., Mattern, F., & Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In *Architecting the Internet of Things* (pp. 97-129). Springer Berlin Heidelberg.

[4] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.

[5] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.

[6] Gruber, T. (1993). What is an Ontology.

[7] Hourdin, V., Tigli, J. Y., Lavirotte, S., Rey, G., & Riveill, M. (2008). SLCA, Composite Services for Ubiquitous Computing. In *5th International Conference on Mobile Technology, Applications and Systems (Mobility'08)* (p. 8).

[8] Mayer, S., & Guinard, D. (2011, June). An extensible discovery service for smart things. In *Proceedings of the Second International Workshop on Web of Things* (p. 7). ACM.

[9] 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.

[10] Maedche, A. (2002). *Ontology learning for the semantic web*. Springer Science & Business Media.

[11] 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).

[12] Kameas, A., & Seremeti, L. (2011). Ontology-based knowledge management in NGAIEs. In *Next Generation Intelligent Environments* (pp. 85-126). Springer New York.

[13] Euzenat, Jérôme, et al. "Results of the ontology alignment evaluation initiative 2009." *Proc. 4th ISWC workshop on ontology matching (OM)*. No commercial editor., 2009.

[14] Serafini, L., & Tamilin, A. (2005). Drago: Distributed reasoning architecture for the semantic web. In *The Semantic Web: Research and Applications* (pp. 361-376). Springer Berlin Heidelberg.

[15] Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The knowledge engineering review*, *18*(01), 1-31.

[16] Euzenat, J., & Shvaiko, P. (2007). *Ontology matching* (Vol. 18). Heidelberg: Springer.

[17] Gyrard, A., Datta, S. K., Bonnet, C., & Boudaoud, K. Standardizing Generic Cross-Domain Applications in Internet of Things.

[18] Kowalski, G. (1997). Information retrieval systems: theory and implementation. Kluwer Academic Publishers.

[19] Euzenat, J. (2007, January). Semantic Precision and Recall for Ontology Alignment Evaluation. In *IJCAI* (pp. 348-353).

[20] Cheung, D., Tigli, J. Y., Lavirotte, S., & Riveill, M. (2006, June). Wcomp: a multi-design approach for prototyping applications using heterogeneous resources. In *Rapid System Prototyping, 2006. Seventeenth IEEE International Workshop on* (pp. 119-125). IEEE.

[21] Benyelloul, A., Jouanot, F., & Rousset, M. C. (2010). Conquer, an RDFS-based model for context querying. In *6emes Journées Francophones Mobilité et Ubiquité*.

[22] McBride, B. (2002). Jena: A semantic web toolkit. *IEEE Internet computing*, *6*(6), 55-59.

[23] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, *5*(2), 51-53.

[24] Richard, G. G. (2000). Service advertisement and discovery: enabling universal device cooperation. *Internet Computing, IEEE*, *4*(5), 18-26.

[25] 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.

[26] Beckett, D., & McBride, B. (2004). RDF/XML syntax specification (revised). *W3C recommendation*, *10*.

[27] Maedche, A. (2002). *Ontology learning for the semantic web*. Springer Science & Business Media.

[28] Sabou, M., Kantorovitch, J., Nikolov, A., Tokmakoff, A., Zhou, X., & Motta, E. (2009). Position paper on realizing smart products: Challenges for semantic web technologies. In *CEUR Workshop Proceedings* (Vol. 522, pp. 135-147).

[29] Hartmann, M., Uren, V., & Vildjiounaite, E. Gathering knowledge for supporting interaction with smart products.

[30] Dibowski, H., & Kabitzsch, K. (2011). Ontology-based device descriptions and device repository for building automation devices. *EURASIP Journal on Embedded Systems*, *2011*, 3.

[31] Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, *5*(1), 4-7.