

Réunion du Vendredi 24 octobre 2014

Ordre du jour :

1. Revue des livrables attendus et des dates clés du projet,
2. Présentation du cadre général du projet à Mr Corby,
3. Comprendre les possibilités offertes par le langage RDFS autour de la représentation et de la gestion des connaissances dans les ontologies avec la problématique de cohérence de la BdC dans un environnement dynamique (automatisation de la construction de l'ontologie à partir de l'apparition/disparition de triplets RDF, des règles d'inférences,...)
4. Définition des défis et des objectifs du projet.

Personnes présentes : Jean-Yves Tigli, Olivier Corby, Gérald Rocher

1. **Revue des livrables du projet à partir du site des PFE (Projet de Fin d'Etude):**
<http://www.i3s.unice.fr/~mosser/teaching/pfe/start>
2. **Présentation du cadre du projet à partir des minutes de la réunion du 13 Octobre 2014.**
https://www.tigli.fr/lib/exe/fetch.php?media=recherche:masters:minutes_131014.pdf
3. **Discussions RDF/RDFS : représentation et gestion des connaissances dans les ontologies avec les problématiques de gestion dynamique des informations sémantiques dans la BdC et préservation de sa cohérence.**
Généralement une ontologie la plus complète possible est définie en amont et est intégrée dans la BdC dès le début de son existence (Modélisation manuelle d'ontologie de la connaissance par un expert). Dans notre cas, le contenu de l'ontologie (OWL-S) va s'enrichir au fur et à mesure (construction automatique) de la découverte des dispositifs et de leurs annotations sémantiques (statiques (nom, type de service,...) et dynamiques contextuelles (has_location, used_by, ...)).

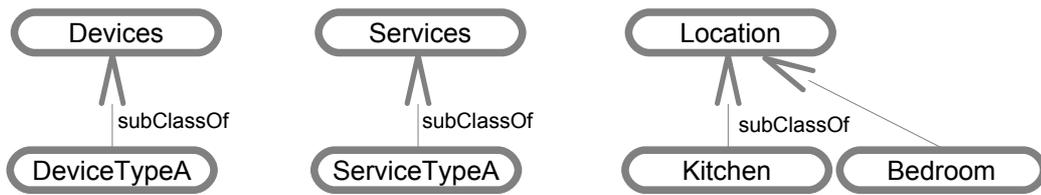
On liste ci-après, dans le cadre de la construction automatique des ontologies, les problématiques liées à la préservation de la cohérence de la BdC.

Problématique#1 : **Isolation des annotations sémantiques statiques et dynamiques**

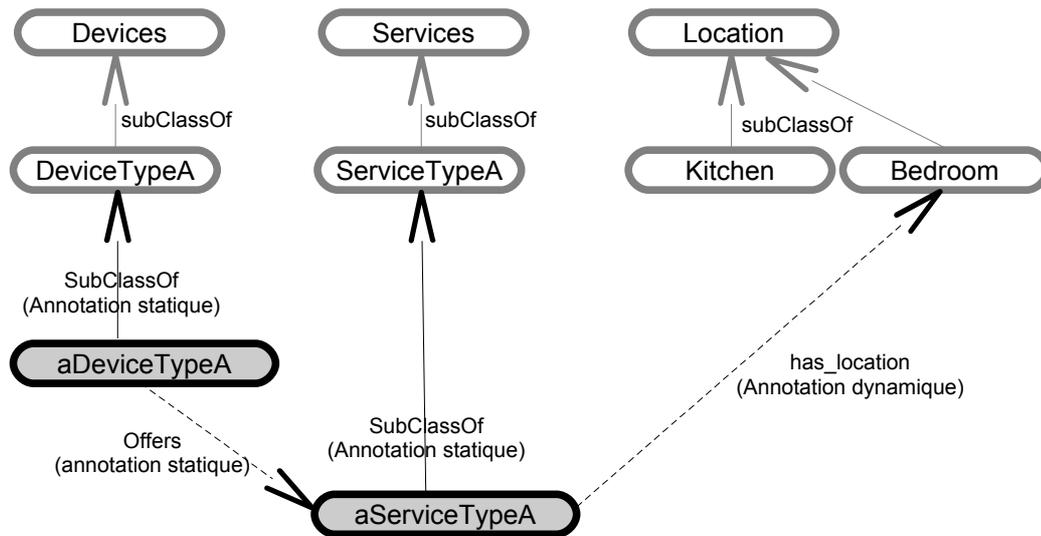
On va vouloir enrichir l'ontologie dans la BdC au fur et à mesure de la découverte de nouveaux dispositifs. Pour cela, les annotations sémantiques statiques et dynamiques attachées au dispositif sont utilisées. Néanmoins, lors de la disparition du dispositif, l'ontologie dans la BdC doit, à minima, conserver les propriétés statiques. Des traitements particuliers pourront être faits sur les annotations sémantiques dynamiques lors de la disparition du dispositif (plutôt que de supprimer purement et simplement ces informations sémantiques on pourrait par exemple procéder à une classification spatio-temporelle de manière à conserver la cohérence de la BdC avec son contexte → problématique#2).

Comment doivent être définies les propriétés statiques et dynamiques ?

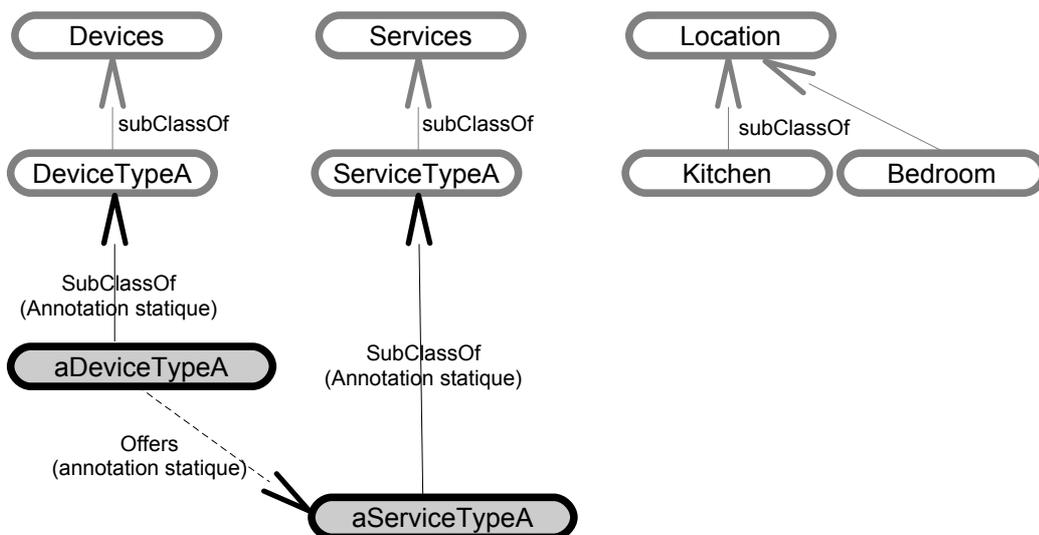
Ontologie minimaliste :



Apparition d'un nouveau dispositif et mise a jour de l'ontologie :



Disparition du dispositif l'ontologie conserve seulement les annotations statiques :



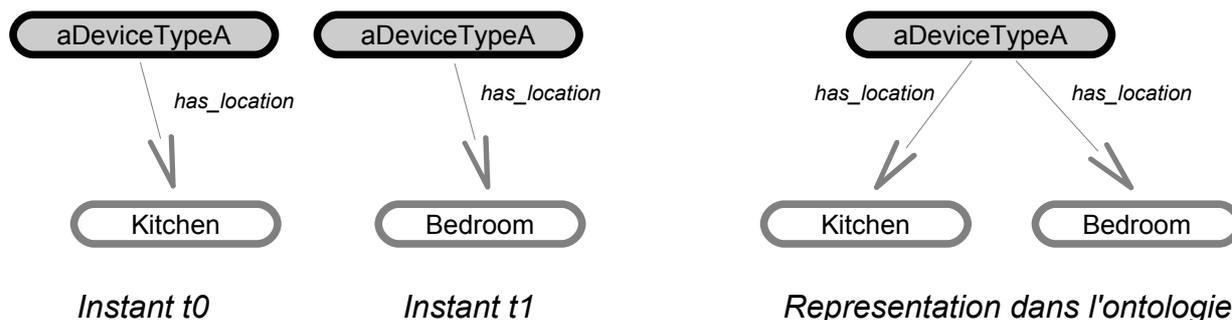
Problématique#2 : Les propriétés dynamiques contextuelles multiples d'un dispositif

Dans un contexte dynamique avec évolution des données sémantiques d'un sujet (triplet RDF < sujet, prédicat, objet > ou < sujet, propriété, valeur > : un sujet donné possède une valeur donnée pour une propriété donnée), les prédicats (propriétés) peuvent évoluer dans la dimension temporelle.

Par exemple :

Instant t0 : triplet RDF #1 <service, has_location, kitchen>

Instant t1 : triplet RDF #2 <service, has_location, bedroom>



Comment gérer la sélection du service des lors qu'il possède deux localisations distinctes? Pour éviter toute incertitude, on peut, avant d'ajouter le second triplet, enlever le premier triplet de la BdC.

La seconde approche serait d'ajouter une notion temporelle (time stamp des informations sémantiques), la sélection du service se faisant alors en tenant compte de cette dimension temporelle. C'est un algorithme possible (most récent location), il y en a sans doute d'autres.

Problématique#3 : Gestion des inférences issues des propriétés fonctionnelles statiques et contextuelles dynamiques des dispositifs

On va pouvoir, avec l'apparition d'un service, enrichir la BdC avec des triplets issus du raisonnement sur les informations fonctionnelles et contextuelles du service pour inférer des nouvelles propriétés. Lors de la disparition du service, comment gérer (detricotter) les propriétés inférées afin que la BdC cohérente avec le contexte ?

Approche possible : Contextualisation des triplets dans la BdC

Le modèle du triplet RDFS ne permet pas d'exprimer des information contextuelles, en particulier les informations temporelles. Une solution serait potentiellement utile dans ce sens :

Graphes nommés

On ne parle plus de triplet mais de quadruplet :

<Sujet><Prédicat><Objet><Graphe>

Ce mécanisme permet :

1. De regrouper les triplets au sein d'un même URI, plus simples à identifier,
2. De gérer l'ensemble des triplets dans un graphe (Ajout, Suppression, update),
3. D'identifier la source des triplets,
4. De versionner → l'URI peut être décrite séparément pour garder la trace des dates de création & modification des triplets dans le graphe,...

API pour Jena (Utilise dans Conquer) :
<http://www4.wiwiss.fu-berlin.de/bizer/ng4j/>

Ce mécanisme permettrait, en contextualisant les triplets au travers de graphes nommés, de pouvoir être en mesure de séparer/classer les annotations sémantiques statiques (nom, type de service,...) et dynamiques contextuelles (localisation,...) pour faciliter leur gestion lors de la disparition des dispositifs (Un graphe distinct par dimension).

4. Discussions annexes en lien avec les problématiques

1. Traitements de flux sémantiques (Langage C-SPARQL)

On considère un flux de triplets RDF représentant des connaissances évoluant continuellement dans le temps et sur lesquels on applique des requêtes (notion de fenêtre temporelle d'acquisition sur les triplets les plus récents). Cette approche est bien adaptée au traitement d'informations envoyées en flux continu par des capteurs par exemple.

2. Automatisation des traitements : BdC 'Event driven'

Des événements dans la BdC déclenchent des traitements. Par exemple, l'apparition de triplets RDF déclenche des calculs de raisonnement, des actions de gestion de cohérence, ou des traitements optionnels.

Cette notion d'événement n'existe pas dans RDFS et se trouve être générique dans le sens où son utilité peut être étendue.

Il s'agirait de définir un modèle de description d'événements 'INSERT', 'DELETE' ou 'UPDATE' qui seraient déclenchés lors :

1. D'occurrence d'un triplet portant sur la localisation,
2. De l'apparition d'un dispositif de type 'x',
3. De la disparition d'un dispositif,
4. ...

Par exemple :

- 1) réception triplet <light><isIn><bedroom>
- 2) réception triplet <bedroom><isA><location>

L'objet 'location' n'étant pas prédéfini dans l'ontologie comment déclencher un événement lors de l'apparition de ce triplet ?

Les propriétés, même si elles ne sont pas définies dans l'ontologie, peuvent être définies dans la modélisation de l'événement, ce n'est pas un problème :

```
Event[Insert/Delete] [x ? rdf:type ex:location] <pattern>
```

Dans l'exemple précédent, bien que l'objet 'location' ne soit pas défini dans l'ontologie, à la réception d'un triplet contenant cet objet, l'événement sera quand même déclenché.

5. Définition des défis et des objectifs du projet.

Défi

Contextualiser les connaissances dans la BdC de manière à faciliter son évolution tout en conservant sa cohérence avec le contexte.

Défi#1.1

Représentation **des connaissances** (triplets RDF) décrivant les services et les propriétés des dispositifs à partir d'annotations sémantiques :

1. **Statiques** (nom, type, domaine d'application, services offerts,...)
2. **Contextuelles dynamiques** (localisation, durée de validité,...)

Défi#1.2

Contextualisation de l'ontologie ainsi définie.

Objectifs

On souhaite permettre la ségrégation de certains types d'annotations sémantiques (les annotations sémantiques statiques et dynamiques contextuelles) de manière à pouvoir faire évoluer l'ontologie associée à la BdC dynamiquement **tout en conservant sa cohérence avec le contexte.**

Bibliographie (préliminaire) associée aux défis

Défi#1.1

[1] *Ontology-Based Representation of UPnP Devices and Services for Dynamic Context-Aware Ubiquitous Computing Applications (2010)*

Ontology and related technologies have been introduced into the Ambient Intelligence domain as a mean to provide declarative formal representations of the domain knowledge. The range of devices available in the scope of an Ambient Intelligence space becomes increasingly heterogeneous and at the same time ubiquitous. Hence there is a need to link the discovery, description and deployment of these ambient devices and their services with context and domain knowledge representations in order to facilitate an Ambient Intelligence space experience. The contribution of this work is an approach for bridging the gap between the non-semantic description mechanisms of XML based devices description protocols, such as UPnP, and the Aml domain knowledge representation. For this we design a prototype ontology-based representation for UPnP devices and services that provide a semantic linking between human-centric abstract description, and the software-centric concrete description that derives from the UPnP descriptors and is necessary to remotely execute method calls on devices. We also demonstrate the benefits of its use with a prototype implementation.

Published in:

Communication Theory, Reliability, and Quality of Service (CTRO), 2010 Third International Conference on

[2] **POWDER (Protocol for Web Description Resources, W3C Recommendation, 2009)**

<http://www.w3.org/TR/powder-dr/>

<http://www.w3.org/TR/powder-primer/>

[3] **Introducing Time into RDF (2007)**

Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman.
IEEE Trans. Knowl. Data Eng., 19(2):207–218, 2007.

[4] **RDF/SPARQL Design Pattern for Contextual Metadata (2007)**

O. Corby, C. Faron-Zucker

The basic principle of the Semantic Web carried by the RDF data model is that a collection of RDF statements coexist all together and are universally true. However some case study imply contextual relevancy and truth. The SPARQL query language is provided with patterns enabling to choose the RDF dataset against which a query is executed. This is a first step to handle contextual metadata. Based on it, we present in this paper a design pattern to handle contextual metadata hierarchically organized. This is done by means of a subStateOf property reifying RDF entailment between contextual graphs. The subStateOf relation is modelled within RDF and therefore context hierarchies can be described and queried by means of SPARQL queries. We propose a slight syntactic extension to SPARQL to facilitate the query of context hierarchies, together with rewriting rules to return to standard SPARQL.

Défi#1.2

[1] **Towards applying dimensions to ontology (Oct. 2012)**

This paper will address the context-sensitive knowledge representation problem of ontology. The author regards the notion of context as a set of dimensions. The approach of applying dimensions to ontology is presented as a framework named Dimensional Ontology (DO). After analyzing different approaches for ontology contextualization, the author explores deeply the concepts and semantics of dimensions. Conforming to some standards, DO is designed and implemented as an integrated and extendable framework with multiple practical supporting components, which is one of the greatest advantages compared with other solutions.

Information Management, Innovation Management and Industrial Engineering (ICIII), 2012 International Conference on (Volume:3)

[2] RDF and Contexts: Use of SPARQL and Named Graphs to Achieve Contextualization (2006)

Heiko Stoermer, Ignazio Palmisano

http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCIQEjAA&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F228697315_rdf_and_Contexts_Use_of_sparql_and_Named_Graphs_to_Achieve_Contextualization%2Flinks%2F0fcfd50be3562b1cee00000&ei=YBBOVJcsjrs9-MeAmAo&usq=AFQjCNEXwsM_LAOuGjgh-i4i-OOdFuL_Aw&sig2=NniphWgTpS0Cr07YLjrqZg&bvm=bv.77880786.d.ZWU

[3] Named Graphs (2005)

Jeremy Carroll, Christian Bizer, Patrick Hayes, Patrick Stickler

Journal of Web Semantics, Vol. 3, Issue 4, p. 247-267, 2005.

[4] An ontology-based context management and reasoning process for UbiComp applications (2005)

Eleni Christopoulou, Christos Goumopoulos, and Achilles Kameas.

UbiComp applications operate within an extremely dynamic and heterogeneous environment and have to dynamically adapt to changes in their environment as a result of users' or other actors' activities. So context definition, representation, management and use become important factors that affect their operation. To ease the development of such applications it is necessary to decouple application composition from context acquisition and representation, and at the same time provide universal models and mechanisms to manage context. In this paper is presented an approach for building a context-aware UbiComp system organised in hierarchical levels. The focus of the paper is on an ontology-based context modelling, management and reasoning process developed for composing context-aware UbiComp applications from Aml artefacts.

In Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies (sOc-EUSA1 '05). ACM, New York, NY, USA, 265-270.

[5] An Ontology-based Context Model in Intelligent Environments (2004)

http://www-public.it-sudparis.eu/~zhang_da/pub/Ontology-2004-2.pdf