Réunion du Lundi 13 octobre 2014

Ordre du jour : Présentation générale du PFE « Sélection sémantique de services pour dispositif dans des systèmes ambiants » et transfert des connaissances (Installation de l'environnement pour que Gérald puisse commencer a comprendre l'architecture générale)

Personnes présentes : Jean-Yves Tigli, Stéphane Lavirotte, Rahma Daikhi, Gérald Rocher

Résumé

Dans le cadre de son projet de fin d'année du Master 2, Rahma Daikhi a travaillé sur la plateforme WComp pour y implémenter la sélection sémantique de services.

Cadre

Historiquement les services étaient sélectionnés grâce a des expressions régulières et les services ainsi sélectionnés étaient alors transmis aux schémas d'assemblage (**Qui correspond, dans son ensemble au schéma d'adaptation**) :

Sélection des services avec expressions régulières	Règles de composition
`a = switch*` `b = light*`	`a^on  b.on()` `a^off  b.off()`

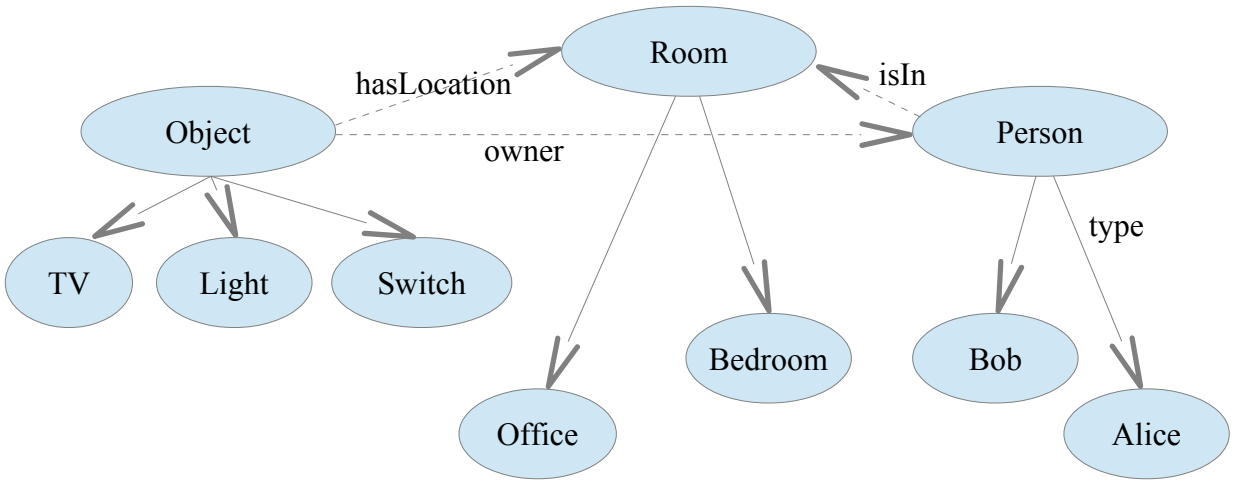
Schéma d'adaptation

L'idée a donc été, pour la sélection des services, de remplacer les expressions régulières par quelque chose de plus flexible et plus « naturel ». Ainsi, être en mesure de sélectionner les services par des requêtes du type :

« La lampe dans le bureau de Bob »

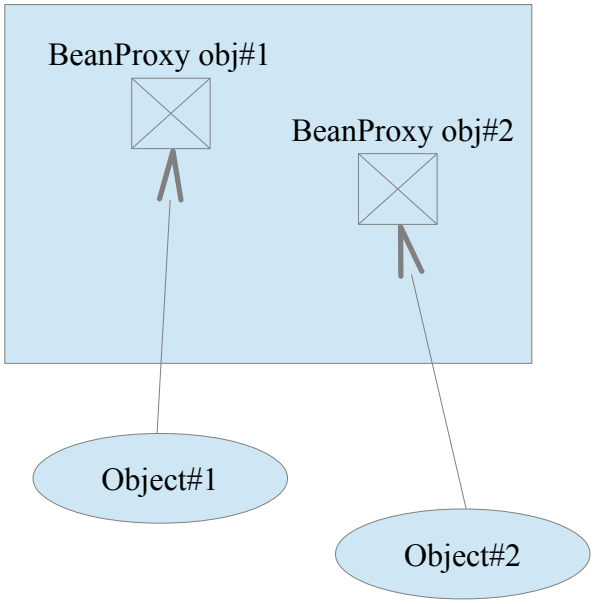
Ontologie

Il a donc été mis en œuvre la possibilité d'exécuter des requêtes SPARQL sur une ontologie utilisée pour décrire les services (Méthodologie issue du WEB sémantique) :



Découverte des dispositifs, annotations sémantiques et metadata

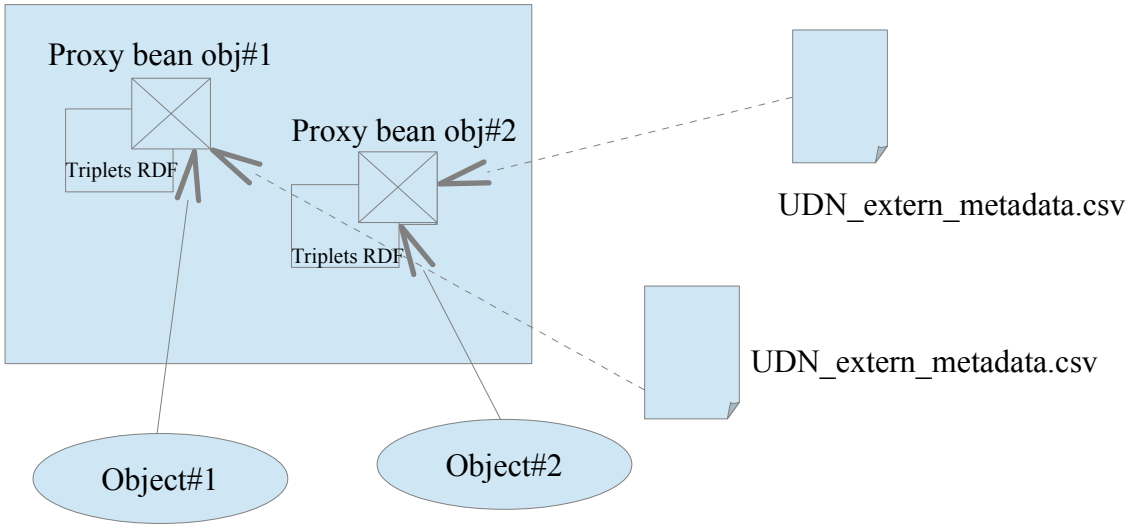
Dans WComp, lorsque un dispositif est détecté, un BeanProxy est automatiquement crée dont l'interface correspond au contrat du dispositif (Web service pour dispositif).



Le `BeanProxy` est donc composé de :

1. L'interface correspondant au contrat du dispositif détecté,
2. Des annotations sémantiques **obtenues a minima par réflexivité :**
   1. Type
   2. Nom
   3. UDN (Unique Device Number) → Pas implémenté mais serait très utile pour pouvoir identifier le bean de manière unique (dans le cas ou il y aurait plusieurs dispositifs de type 'light' par exemple).
3. Enfin, dans le cas des « `BeanProxyWithMetadata` » des metadata sont ajoutées manuellement (Via des fichiers externes). Il s'agit de triplets RDF :

Sujet	Predicat	SPO object
Light	hasLocation	office
Office	hasOwner	Bob



Les `BeanProxy` et `BeanProxyWithMetaData` sont générés automatiquement grâce a un outil tiers : `UpnPWizardDesigner`.

La base de connaissances (BdC)

Une fois les services découverts et annotés, une base de connaissance est alors enrichie/peuplée avec les triplets RDF correspondant aux annotations sémantiques et les metadata.

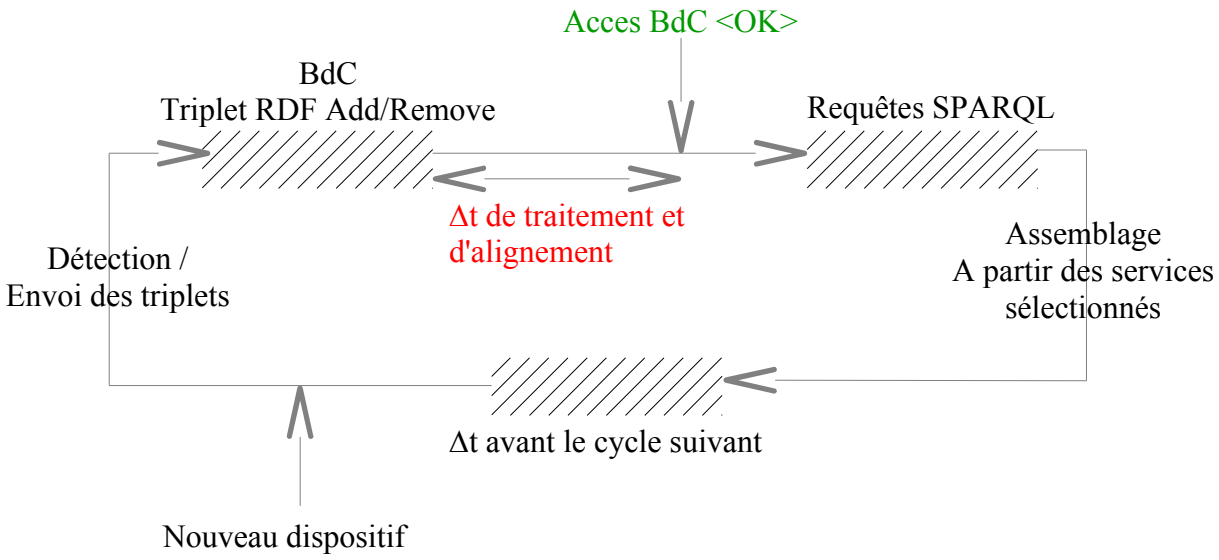
La BdC est basée sur Conquer et est encapsulée dans un service web pour dispositif UPnP et contient la description de l'ontologie vue précédemment.

La BdC est interrogée grâce a des requêtes SPARQL. On permet donc de pouvoir faire de requêtes du type « La lampe dans le bureau de Bob ».

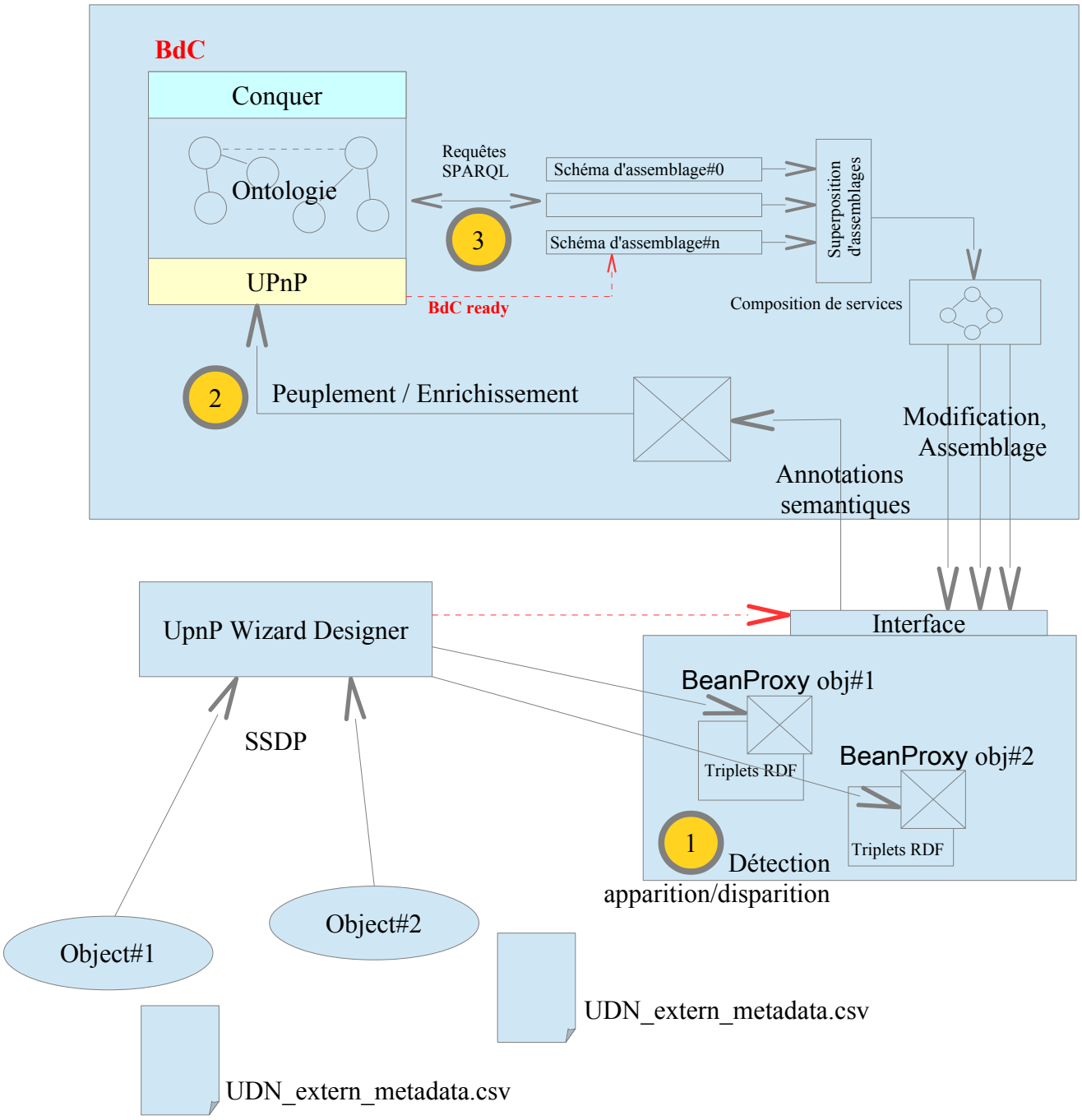
Problèmes :

1. **Les requêtes SPARQL effectuées a partir du schéma d'adaptation ne peuvent être effectuée que lorsque la base de connaissance est prête.**
2. La base de connaissance est prête quand :
    1. L'ajout des nouveaux triplet est terminé,
    2. Les alignements sont terminés (Utilisation de dico wordnet pour compléter l'ontologie avec les synonymes des sujets (Light, Office, etc...),
    3. Les dispositifs disparus sont enlevés de la base (L'instance est enlevée mais il reste des informations (a préciser) dans la base de connaissance. En quelque sorte celle-ci n'arrête jamais de croitre au grès des ajouts des dispositifs ce qui, d'une certaine manière, s'apparente a de l'apprentissage.

Cycle de traitement :



Vue d'ensemble



Périmètre du travail de recherche de Gerald (A detailler apres entrevue avec Mr Corby) :

1. Focus sur la base de connaissance
    1. Etude des ontologies (Voir avec Mr Corby)
    2. Amélioration de la synchronisation avec le schéma d'adaptation (Performance)
    3. Gestion générale (Alignement, cohérence, inférence,....)