

Sélection sémantique de services pour dispositifs dans des systèmes ambiants [Y1415-S051]

Jalon#2 : Etat d'avancement du projet

Gérald Rocher

Master 2 IFI parcours IAM

Université de Nice Sophia-Antipolis

rocher.gerald@gmx.fr

Résumé

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 communicants de notre vie quotidienne. Ces applications logicielles évoluent dans un environnement physique hautement dynamique où la disponibilité des dispositifs, des objets et de leurs services logiciels évolue dans le temps et dans l'espace. Il est donc primordial que le mécanisme de sélection des services logiciels tienne compte de la dynamique spatio-temporelle des dispositifs et des objets. Dans ce cadre, l'utilisation des standards du web sémantique est envisagée pour la représentation et la gestion des connaissances sur les services logiciels et leur évolution dans le temps et dans l'espace. Ce document fait suite à une revue systématique de la littérature [5] et propose, de décrire la solution retenue, de faire un point sur le projet par rapport au plan établi dans le DOW (Description Of Work) [2] et enfin, de présenter les étapes jusqu'au prochain jalon.

Mots clefs

Base de connaissance, Context, Web Sémantique, Informatique ambiante.

1. Rappels sur le contexte du projet

La miniaturisation des composants informatiques permet aujourd'hui de les intégrer à une multitude d'objets et de dispositifs de notre vie quotidienne. Cette évolution technologique est à la base des systèmes informatiques (SI) dits "ambiants" qui visent à s'intégrer à des environnements physiques hautement dynamiques et hétérogènes composés d'une multitude d'utilisateurs, de dispositifs et d'objets communicants. Les applications logicielles, dans ces SI, sont composées à partir des services logiciels mis à leur disposition par les dispositifs et les objets dont la disponibilité peut évoluer dans le temps et dans l'espace et qui forment un contexte opérationnel. Les applications logicielles doivent donc s'adapter dynamiquement aux changements induits par les problématiques de mobilité et de variabilité de l'environnement physique (context aware computing). Cette capacité d'adaptation est primordiale pour assurer la cohérence des applications logicielles avec le contexte opérationnel et se décompose en trois étapes distinctes qui forment un cycle d'adaptation dynamique:

1. Découverte des services logiciels disponibles et enrichissement d'une base de connaissance (BdC) avec leurs propriétés fonctionnelles,
2. Sélection des plus pertinents à partir de règles qui génèrent des requêtes sur la BdC,
3. Composition de l'application à partir de règles sur les services logiciels sélectionnés (composition de services).

L'objectif de ce travail de recherche est d'ajouter la représentation

spatio-temporelle des connaissances sur les services logiciels dans la BdC pour permettre de raisonner sur leur disponibilité dans le temps et dans l'espace. Ainsi, lorsque des requêtes sont émises, la disponibilité des services logiciels et la cohérence de la BdC avec le contexte opérationnel sont considérées comme explicites.

2. Description de la solution retenue

Au delà des dimensions spatio-temporelles, il est souhaitable que la solution retenue puisse permettre la représentation et la gestion des connaissances sur les services pour n'importe quelle dimension. En effet, le domaine de l'informatique ambiante met en œuvre des dispositifs et des objets physiques soumis à de multiples dimensions (consommation électrique, température, etc.). Il s'agit donc de pouvoir contextualiser les connaissances sur les services, c'est à dire, du point de vue du web sémantique, affirmer qu'une relation est vraie uniquement pour un contexte donné (le contexte pouvant représenter un intervalle de temps, une localisation, etc.). Le caractère dynamique des dispositifs et des objets induit l'apparition et la disparition de leurs connaissances dans la BdC. La solution choisie doit donc faciliter la suppression des connaissances sur les services qui disparaissent (individus, propriétés et relations) ainsi que les connaissances sur leurs contextes. Enfin, nous devons permettre la création de relations entre contextes. Par exemple, les relations temporelles d'Allen [1] permettent de dire qu'un contexte temporel (c'est à dire un intervalle de temps) se produit *avant*, *après*, *pendant*, etc. un autre contexte temporel.

A partir des résultats de la revue systématique de la littérature [5], l'approche « context slices » [7], généralisation de l'approche 4D-Fluents [8], a été évaluée. Elle a l'avantage d'être totalement compatible avec le langage OWL (Web Ontology Language) et permet le calcul d'inférences sur l'ensemble des éléments dans l'ontologie (ce qui n'est pas le cas de l'approche « named graphs » qui limite les inférences aux éléments contenus dans les graphes. Il n'est donc pas possible de créer des relations entre contexte [4]).

Plusieurs scénarios ont été mis en œuvre [6] pour l'évaluation de cette approche. Chaque scénario est transcrit dans une ontologie qui est visualisée grâce à l'outil protégé [3]. Cet outil intègre un moteur SPARQL pour effectuer des requêtes sur le contenu de l'ontologie.

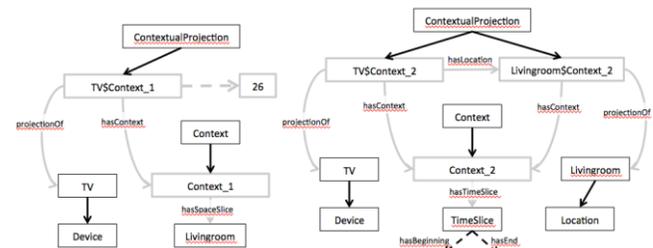


Figure 1 Exemple d'utilisations de l'approche évaluée

La Figure 1 présente deux exemples d'utilisations avec l'approche « context slices ». Le premier exemple met en œuvre la dimension spatiale: l'assertion $\langle TV, hasConsumption, 26 \rangle$ n'est vraie que dans le contexte spatial du salon. Le second exemple met en œuvre la dimension temporelle : l'assertion $\langle TV, hasLocation, Livingroom \rangle$ n'est vraie que dans l'intervalle de temps $[t1, t2]$. On notera que l'inconvénient de cette approche concerne son empreinte mémoire avec la nécessité de créer trois objets (dans le deuxième exemple de la figure 1 nous avons du créer $Context_2$, $TV\$Context_2$ et $LivingRoom\$Context_2$) ainsi que cinq relations ($2 * projectionOf$, $2 * hasContext$ et $1 * hasTimeSlice$). Néanmoins, dans le cadre d'applications pour l'informatique ambiante, le contenu de la BdC n'excède pas quelques centaines d'objets.

Il est par ailleurs important de pouvoir développer des règles d'inférences pour créer des relations entre objets et classes. Par exemple, lors de la suppression d'un objet, il nous faut être capable de retrouver et de supprimer les contextes qui lui sont associés. Ainsi, l'étude du langage SWRL (Semantic Web Rule Language) a permis de mettre en œuvre une règle simple pour ajouter la relation « hasProjection » entre les objets et leurs contextes :

```
?x tst :projectionOf ?y -> ?y tst:hasProjection ?x
```

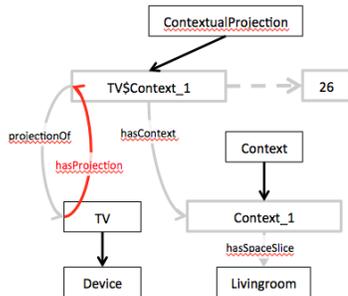


Figure 2 Ajout d'une règle d'inférence simple

SWRL n'est pas encore un standard et certaines fonctions ne sont pas reconnues par tous les raisonneurs. Ainsi, une règle permettant de définir la durée d'un *TimeSlice* n'a pas pu être exploitée avec les raisonneurs disponibles Pellet et Jena :

```
hasBeginning(?x1,?x2),hasEnd(?x1,?x3),
swrlb:subtractTimes(?x4,?x3,?x2)->
hasDuration(?x1,?x4)
```

Enfin, des requêtes SPARQL ont été effectuées démontrant le bon fonctionnement de l'approche sans ajout d'opérateurs *ad-hoc* [6]. Par exemple, pour le premier exemple de la Figure 1, pour sélectionner tous les objets qui ont une consommation électrique inférieure à 30W dans le salon, la requête suivante est exécutée:

```
SELECT ?device
WHERE {
    ?s tst:hasConsumption ?c .
    ?s tst:projectionOf ?device .
    ?s tst:hasContext ?context .
    ?context tst:hasSpaceSlice ?loc .
    FILTER(?c < 30 && ?loc = tst:Livingroom)
}
```

Note: il est possible d'associer plusieurs contextes aux objets. Ainsi, en reprenant les deux exemples de la Figure 1, il est possible de sélectionner les objets dans le salon qui ont une consommation électrique inférieure à 30W à 16h45.

En conclusion, l'approche étudiée permet d'adresser, en grande partie, les problématiques de ce sujet de recherche. Elle a par

ailleurs également été validée sur la BdC Conquer qui est mise en œuvre dans le projet [6]. On remarque néanmoins que la description de règles d'inférences pour le calcul de relations (Allen, etc.) entre contextes, à partir du langage SWRL, risque d'être difficile à mettre en œuvre, toutes les fonctionnalités de ce langage n'étant pas supportées par les raisonneurs.

3. Gestion de projet

La Table 1 montre les tâches planifiées dans le DOW. L'étude de la SLR a demandé un peu plus de temps que prévu (mais cela reste négligeable). La description de la solution est bien avancée mais ne sera pas terminée à la fin de la semaine 51. Il est donc nécessaire de rattraper le retard pendant la semaine 52 prévue à cet effet.

Table 1 Suivi du projet

	Début	Fin	Budget nb. heures	Nb. heures effectives
Planification	w42	w45	48h	48h
Suivi de projet	w46	w9	30h	12h
Etude bibliographique	w44	w50	90h	92h
Description de la solution	w49	w51	22h	17h
Implémentation	w1	w4	42h	
Tests et validations	w3	w6	35h	
Communication des résultats	w6	w9	49h	
TOTAL			316h	164h

4. Plan d'actions

La description de l'implémentation doit inclure la définition des annotations sémantiques relatives aux dimensions spatiale, temporelle, etc. à ajouter aux services logiciels pour permettre la création des contextes dans la BdC.

Une fois complété, ce document sera utilisé comme base pour l'implémentation de la solution dans la BdC. Concernant le calcul d'inférences entre contextes (relations d'Allen, etc.), il sera nécessaire de comprendre dans quelle mesure un autre raisonneur pourrait être intégré à Jena (bibliothèque java utilisée dans la BdC Conquer). Ces tâches correspondent aux objectifs du prochain jalon (semaine#4).

5. REFERENCES

- [1] James F. Allen. Maintaining knowledge about temporal intervals. Commun. ACM, 1983.
- [2] [Description Of Work](#)
- [3] Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Ferguson, R. W., & Musen, M. A. (2001). Creating semantic web contents with protege-2000. *IEEE intelligent systems*, 16(2), 60-71.
- [4] [RDF 1.1: On Semantics of RDF Datasets](#)
- [5] [Systematic Literature Review](#)
- [6] [Vidéo de démonstration](#)
- [7] Welty, C. (2010). Context Slices: Representing Contexts in OWL. In Proceedings of the 2nd Workshop on Ontology Patterns (WOP 2010).
- [8] Welty, C., Fikes, R., & Makarios, S. (2006, May). A reusable ontology for fluents in OWL. In FOIS (Vol. 150, pp. 226-236).