

Sélection sémantique de services pour dispositif dans les systèmes ambiants

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 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 implique 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 clefs

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

1. INTRODUCTION

L'informatique ambiante [1] est un terme qui désigne l'intégration du monde physique réel dans le monde numérique d'internet. Cela concerne tous les *objets* de notre vie quotidienne (chaise, table, lampe, etc...) ou environnements physiques (ville, immeuble, véhicule, espace médicalisé,...), statiques ou dynamiques qui deviennent, à partir d'applications logicielles, observables et contrôlables [2]. Cela est rendu possible grâce à des *dispositifs* communicants embarqués dans les *objets* ou placés dans l'*environnement* [3]. Ces *dispositifs* mettent en œuvre des *ressources* qui permettent d'interagir avec les *objets* (actionneur) et d'obtenir des informations (ID, description, observations de capteurs,...) sur eux-mêmes, sur les *objets* ou l'environnement auxquels ils se rattachent. L'accès à ces *ressources* est réalisé par l'intermédiaire de *services* logiciels qui exposent leur interface et permettent la communication avec le monde numérique. Ces services sont alors en mesure de participer à l'élaboration d'applications logicielles complexes qui doivent, parmi tous les services disponibles, *sélectionner* et *orchestrer* ceux qui vont permettre **d'obtenir une fonctionnalité qui réponde aux besoins des utilisateurs.**

Le caractère dynamique des dispositifs, des objets et des environnements physiques implique que les applications

logicielles, pour conserver leur fonctionnalité et **assurer une continuité de service**, doivent dynamiquement s'adapter aux variations du monde réel (la disponibilité et l'état des services évoluant dans le temps et dans l'espace). Les applications sont alors dites sensibles au contexte. Ainsi, l'élaboration d'applications logicielles sensibles au contexte est le fruit d'une composition de services au sein d'une boucle d'auto-adaptation dynamique (Figure 1).

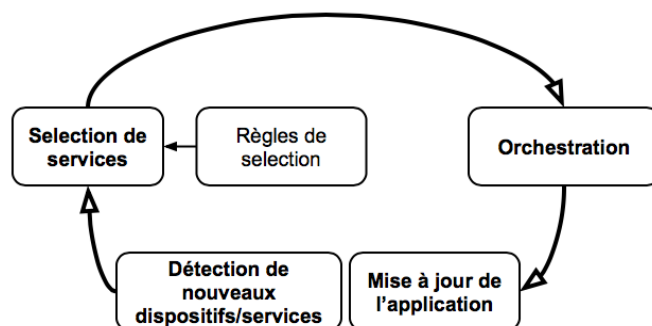


Figure 1 : Boucle d'adaptation dynamique d'une application

Au sein de cette boucle d'auto-adaptation dynamique, la pertinence des services sélectionnés, à partir d'informations sur les dispositifs et les services (annotations), est primordiale. Mais ces informations sont souvent statiques et décollées de l'environnement réel avec lequel les objets, les dispositifs et les services interagissent.

Comment améliorer alors la pertinence des services sélectionnés et augmenter dynamiquement la connaissance en conservant sa cohérence avec l'environnement réel? Celui-ci est en effet le théâtre de nombreux phénomènes physiques souvent modélisés indépendamment les uns des autres (thermique, qualité de service, etc...) mais soumis à des principes d'évolution qui impliquent une temporalité commune.

Pour ce faire, nous proposons une approche qui se base sur les standards du web sémantique (SWoT) [4] et des technologies associées [5]. Chaque dispositif, par le biais d'annotations sémantiques formelles, apporte la connaissance de son domaine, fragments d'ontologies qui, agrégés, vont permettre d'augmenter dynamiquement le contenu d'une base de connaissance (BdC).

Enrichies dynamiquement de valeurs issues de capteurs placés dans l'environnement ou sur les objets, ces annotations vont par ailleurs permettre à la BdC de conserver sa cohérence avec l'environnement réel et son évolution dans le temps.

2. ETAT DE L'ART

2.1 Représentation/Gestion des connaissances à partir des standards du web sémantique

Comme nous l'avons vu précédemment, le mécanisme de sélection des services doit pouvoir interpréter les fonctionnalités offertes par les services et avoir une représentation de la connaissance sur le contexte à partir de laquelle il soit en mesure de raisonner sur leur disponibilité dans le temps et dans l'espace. Dans ce cadre, l'utilisation des technologies éprouvées du web sémantique est tout à fait indiquée. En effet, les langages RDF/RDFS (Resource Description Framework/Schéma) [6], sous-ensemble de la logique du premier ordre avec prédicats binaires, OWL DL (Web Ontology Language) [7], au-dessus de RDF/RDFS, pour la définition d'ontologies, basé sur la logique de description à expressivité plus forte [10], permettent de représenter la connaissance, d'y accéder (SPARQL [8]) et de raisonner sur celle-ci pour inférer de nouvelles connaissances (avec garantie de complétude et de décidabilité des calculs).

2.1.1 Ontologie

Une *ontologie* [9] est un modèle de représentation explicite et formelle de la connaissance qui structure les concepts d'un domaine. L'ensemble des concepts permettent d'exprimer les connaissances du domaine. Une ontologie est composée de trois types d'éléments:

1. Des *classes* (ou concepts) et sous-classes organisées hiérarchiquement selon une taxonomie.
2. Des *propriétés* qui permettent de définir des *faits* ou des *relations* entre classes. Il existe deux types de propriétés:
 - a. Propriété d'objet qui définit une relation entre deux instances d'une classe ou entre plusieurs classes,
 - b. Propriété de type de données c'est à dire une relation entre une valeur ou donnée et l'instance d'une classe.
3. Des *instances* de classe (individu d'une classe) qui peuvent prendre les caractéristiques définies par les propriétés.

2.1.2 Base de connaissance et raisonnement

L'ontologie peut être considérée comme un méta-système pour bases de connaissance (BdC), en tant que description de la représentation des connaissances qu'elles contiennent. Ces BdC, basées sur la logique de description, contiennent les faits et les individus portant sur l'ensemble des concepts et sur lesquels un moteur d'inférence est utilisé pour déduire des connaissances implicites à partir de connaissances explicites. Elles permettent de définir les connaissances suivant deux niveaux de description, *ABox* et *TBox*, définissant respectivement les assertions sur les instances/individus et les concepts généraux [11].

2.1.3 Contextualisation de la connaissance

La représentation contextualisée de la connaissance permet de raisonner sur la disponibilité des dispositifs et des services dans le temps, dans l'espace ou toute autre dimension pertinente dans le cadre d'objets ou de dispositifs physiques réels (température, qualité de service, etc...). Il s'agit donc, dans le cadre du web sémantique, de pouvoir affirmer qu'une relation est vraie

uniquement pour un contexte donné, le contexte pouvant être n'importe quelle mesure permettant de caractériser une grandeur physique.

Dans le domaine du web sémantique, les langages de représentation des connaissances tels que OWL ont une expressivité qui limite les relations à des relations binaires qui ne permettent pas la représentation de relations contextualisées qui nécessiteraient des relations ternaires [12]. Différents mécanismes sont utilisés pour représenter les aspects temporels et/ou spatiaux dans les ontologies. L'approche *temporal RDF* consiste à ajouter un label temporel aux triplets RDF qui indique l'intervalle temporel durant lequel ces derniers sont valides [13]. Le *versioning* d'ontologie [14] suggère que l'ontologie a différentes versions au cours du temps. A chaque changement, une nouvelle version de l'ontologie est créée. L'approche des *graphes nommés* ("named graphs") [15] permet de contextualiser les connaissances [16], [17]. Cependant, cette approche limite les raisonnements aux éléments contenus dans les graphes [18]. Il n'est donc pas possible de créer des relations entre contextes. L'approche *context slices* [19], généralisation de l'approche *4D-Fluents* [20], au-delà de permettre la contextualisation des relations a l'avantage de permettre le calcul d'inférences sur l'ensemble des éléments dans l'ontologie. C'est l'approche qui a été retenue dans le cadre de ce travail de recherche.

2.2 Annotations sémantiques dans WoT

Notre approche est basée sur l'utilisation d'annotations sémantiques formelles distribuées sur chaque dispositif qui permettent une représentation de la connaissance sur les fonctionnalités apportées par les dispositifs et les services ainsi que la représentation de la connaissance multidimensionnelle et dynamique de leur contexte.

Dans [22] les auteurs présentent une approche qui permet de répondre à la problématique de la représentation des connaissances fonctionnelles des dispositifs et des services à partir d'annotations sémantiques. Basée sur une architecture REST (Representational State Transfer), cette approche ne permet pas de gérer la dynamique des dispositifs et des objets (apparition, disparition et abonnement aux événements) [21]. L'ensemble des services disponibles et leur localisation sont codés statiquement dans une base de données [23] qui est accédée par les dispositifs.

2.3 Représentation du contexte

La notion de contexte dans le domaine de l'informatique ambiante est très largement débattue [24], [25] et [26]. Nous retiendrons la définition suivante: [27] : "*Le contexte est l'ensemble des paramètres externes à l'application qui peuvent influencer sur son comportement en définissant de nouvelles vues sur ses données et ses fonctionnalités. Ces paramètres ont un aspect dynamique qui leur permet d'évoluer durant le temps d'exécution*".

Cette définition est intéressante car elle fait apparaître la notion de dynamique inhérente aux environnements physiques réels dans lesquels les applications sont plongées. Cette dynamique implique que les applications doivent s'adapter à leur contexte.

La sensibilité d'une application au contexte se caractérise par trois étapes distinctes [28]:

1. *Perception* des changements de contexte de l'application,

2. *Interprétation* des données contextuelles,
3. *Décision* sur l'adaptation à déclencher.

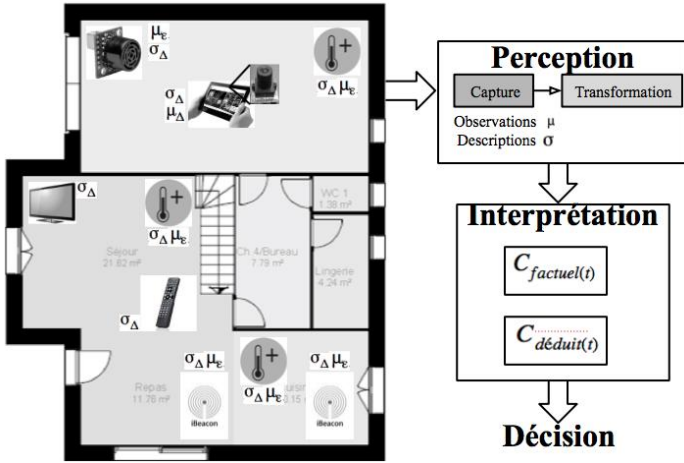


Figure 2 : Extraction du contexte

La notion de dynamicité du contexte, qui peut faire évoluer l'état des entités au cours du temps, nécessite donc une *représentation multidimensionnelle* du contexte permettant de *raisonner* sur la disponibilité des dispositifs et des services dans le temps et dans l'espace.

2.4 Sémantique du temps et de l'espace

Comme indiqué précédemment, le mécanisme de sélection des services va devoir raisonner sur la disponibilité des dispositifs et des services dans le temps et dans l'espace à partir d'informations issues d'annotations sémantiques. Cette approche implique de définir une sémantique pour les dimensions temporelle et spatiale.

2.4.1 Sémantique temporelle

La structure temporelle introduite avec ITL (Interval Temporal Logic) décrite par Allen [29], permet une modélisation linéaire du temps, basée sur des *intervalles* et des *instants*. Ainsi, à partir de deux intervalles temporels sont déduites les 13 relations topologiques d'Allen. Ces relations permettent de raisonner sur la dimension temporelle des objets et de l'environnement. L'ontologie OWL-Time intègre la définition des intervalles de temps et des relations d'Allen [30], [31].

2.4.2 Sémantique spatiale

Apporte une description de l'espace à partir de *points* et des *régions* qui permettent de définir des relations **qualitatives** pour la représentation de topologies (Region Connection Calculus RCC-8 [32], modèle 9-Intersections [33]), **quantitatifs** (A 10km,...), ainsi que des **directions** (N, S, E, O, NO, NE, SE, SO). Cependant, la représentation de relations topologiques telles que celles d'Allen ou RCC-8, bien que faisable, n'est pas triviale à partir de OWL-DL [34] et les auteurs présentent une translation des relations RCC-8 en OWL-DL. Bien que la représentation des concepts et des relations spatiales et temporelles soient réalisables à partir d'OWL-DL, les capacités de raisonnement des moteurs d'inférences et de requêtes sur ces concepts et ces relations doivent être augmentées de règles *ad-hoc*. Ainsi, dans [35], les auteurs, en se basant sur [34], se sont penchés sur le développement d'un raisonneur et d'un moteur de requêtes *ad-hoc* (Pellet Spatial). Le langage SWRL (Semantic Web Rule

Language) est utilisé pour la définition de règles d'inférences à partir de tables de composition définissant les relations d'Allen [36], [37], [38] ou topologiques RCC-8 [36] mises en œuvre dans des moteurs de requêtes spécialisés tels que TOQL [35], SQWRL [39], SOWLQL [36], GeoSPARQL [40],...

3. MÉTHODOLOGIE

3.1 Modélisation du contexte

N'existant pas de modélisation consensuelle du contexte, nous tentons d'en définir une : soient les dispositifs Δ placés dans l'environnement physique réel ε . Leur *état* φ_{Δ} et l'ensemble des services ζ qu'ils proposent, décrits à partir des informations qu'ils fournissent (C 'est à dire la description σ_{Δ} du dispositif et des services ainsi que les mesures μ_{Δ} des objets ou de l'environnement qu'ils observent à partir de capteurs) permettent de décrire, de manière formelle, le contexte $C(t)$ susceptible d'affecter le comportement de l'application, alors :

$$C_{factuel(t)} = \cup\{\varphi_{\Delta(t)} = f(\mu_{\Delta(t)}) \mid \sigma_{\Delta}\} \quad (1)$$

$$C_{déduit(t)} = \cup\{\varphi_{\Delta(déduit)(t)} \mid \sigma_{\Delta(déduit)}\} \quad (2)$$

$$C(t) = \cup\{C_{factuel(t)} \mid C_{déduit(t)}\} \quad (3)$$

La représentation et la connaissance du contexte permettent de définir, à chaque instant t et pour un contexte $C(t)$, l'utilisabilité $U_{\zeta(t)}$ et la pertinence P_{ζ} des services en présence:

$$U_{\zeta(t)} = \varphi_{\Delta(t)} \wedge \varphi_{\Delta(déduit)(t)} \quad (4)$$

$$P_{\zeta} = \sigma_{\Delta} \wedge \sigma_{\Delta(déduit)} \wedge \text{Règle de sélection}_{(1,\dots,n)} \quad (5)$$

$$\text{Services sélectionnés}_{(t)} = U_{\zeta(t)} \wedge P_{\zeta} \quad (6)$$

L'**utilisabilité** $U_{\zeta(t)}$ (4) dénote le fait que l'état du dispositif permet l'utilisation du service pour la composition de l'application. L'utilisabilité dépend donc de l'état physique du dispositif et de son environnement et a donc un caractère dynamique. Ce paramètre nécessite la mise en place de dispositifs (observateurs) associés à des capteurs pour la mesure de l'environnement (capteurs extéroceptifs), et la mesure de l'état physique intrinsèque du dispositif (capteurs proprioceptifs).

La **pertinence** P_{ζ} (5) dénote le fait que le service répond à une fonctionnalité recherchée.

La conjonction des deux paramètres (6) permet la sélection du service comme candidat valide pour la composition de l'application. Voyons à présent l'intérêt de considérer l'utilisabilité $U_{\zeta(t)}$ des dispositifs et des services en plus de leur pertinence P_{ζ} à travers deux scénarios.

3.2 SCENARIOS

Une sélection de services basée uniquement sur la pertinence $P_{\zeta(t)}$ des fonctionnalités qu'ils proposent ne permet pas d'assurer la cohérence de l'application avec son contexte. A chaque instant, la prise en compte de l'utilisabilité $U_{\zeta(t)}$ des services et des dispositifs est nécessaire pour raisonner sur leur état et leur disponibilité dans le temps et dans l'espace et optimiser ainsi la cohérence de l'application avec son contexte.

3.2.1 Scénario 1 : Disponibilité temporelle

Par exemple, un lave-linge, dans un but d'économie d'énergie, ne peut être utilisé que dans certaines tranches horaires (ou

intervalles de temps). Un mécanisme de sélection de services basé uniquement sur la pertinence $P_{\zeta(t)}$ des fonctionnalités proposées par le lave-linge, peut autoriser le déploiement d'une application permettant le contrôle du lave-linge une minute avant la fin de la disponibilité des services logiciels associés. Un mécanisme de sélection de services se basant sur $P_{\zeta(t)}$ et $U_{\zeta(t)}$ va permettre de raisonner sur la disponibilité des services logiciels dans le temps et ne pas autoriser le déploiement de l'application permettant de contrôler le lave-linge une minute avant la fin de la disponibilité des services logiciels associés.

3.2.2 Scénario 2 : Disponibilité spatiale

La localisation des services logiciels a une grande importance dans le processus d'adaptation de l'application à son contexte. Prenons l'exemple d'un agent de contrôle d'un réseau de distribution d'eau. Il est muni d'une télécommande de contrôle et de pilotage d'ouverture et de fermeture de vannes. Durant son parcours il est possible que plusieurs vannes soient détectées en même temps. Un mécanisme de sélection de services basé uniquement sur la pertinence $P_{\zeta(t)}$ des fonctionnalités proposées, va considérer toutes les vannes disponibles autour de l'agent. Un mécanisme de sélection de services se basant sur $P_{\zeta(t)}$ et $U_{\zeta(t)}$ va permettre de raisonner sur la disponibilité des services logiciels en fonction de la position de l'agent par rapport aux vannes disponibles et sélectionner celle qui est la plus proche de lui. L'application déployée alors permettra à l'agent de piloter l'ouverture ou la fermeture de cette vanne à partir de la télécommande.

3.3 Annotations sémantiques formelles

Le calcul des prédicats en logique du premier ordre (équations 4,5 et 6) implique d'associer une valeur booléenne aux formules atomiques. **Celles-ci expriment des propriétés d'objets auxquelles il faut donc donner un sens et donc une sémantique.** Cette sémantique est d'autant plus importante qu'elle va déterminer l'utilisabilité et la pertinence des dispositifs et des services et donc contribuer à la cohérence de l'application avec son contexte.

Dans ce cadre, nous proposons l'utilisation des standards du web sémantique pour fournir une représentation dynamique de la connaissance fonctionnelle et contextuelle des dispositifs, des services et de l'environnement à partir d'annotations sémantiques formelles distribuées et interopérables. Ces annotations vont nous permettre de construire une représentation sémantique dynamique du contexte et d'enrichir une base de connaissance (BdC) centralisée à partir de laquelle pourront être appliqués des raisonnements et des requêtes sur les dispositifs et les services. Les services sélectionnés seront alors en cohérence avec le contexte.

De manière générale, une ontologie, représente la connaissance de l'ensemble d'un domaine. Comme nous l'avons vu précédemment, l'informatique ambiante désigne l'intégration du monde physique réel dans le monde digital d'internet. Exprimer la connaissance complète du monde réel, ouvert, hétérogène et dynamique, à travers l'utilisation d'une ontologie globale n'est donc pas envisageable. De fait, nous proposons une approche où chaque dispositif expose sa propre ontologie, par l'intermédiaire d'annotations sémantiques formelles qui vont permettre la

description du domaine et de ses individus à partir d'éléments ontologiques, à savoir les concepts (classes), les propriétés et les instances du dispositif et des services associés. Une ontologie générique qui exprime les connaissances communes à tous les domaines est par ailleurs définie. Dans ce contexte, le méta-système de la BdC centralisée est constitué de l'union de toutes les ontologies distribuées et définies par les dispositifs au travers de leurs annotations sémantiques formelles.

Le caractère dynamique des dispositifs (et plus précisément des objets ou de l'environnement auxquels ils sont attachés) implique la définition de deux types de portées concernant les éléments ontologiques définis dans les annotations sémantiques formelles:

1. Statiques :

- a. Notés σ_{Δ} , ces éléments ontologiques décrivent :
 - i. Les concepts et les propriétés statiques du dispositif et des services (*TBox*),
 - ii. Les instances du dispositif et des services (*ABox*).
- b. Notés σ_{μ} , ces éléments apportent une description sémantique aux mesures μ_{Δ} de capteurs associés.

2. Dynamiques :

assertions générées par fusion \oplus à partir de σ_{μ} et μ_{Δ} . Notées $\varphi(\sigma_{\mu} \oplus \mu_{\Delta})(t)$ et $\gamma(\sigma_{\mu} \oplus \mu_{\Delta})(t)$, elles permettent de sémantiser :

- a. Des propriétés d'individu ou contextuelles qui varient dans le temps (*ABox*),
- b. Des instances d'individus issues des mesures μ_{Δ} (*ABox*).

$\varphi(\sigma_{\mu} \oplus \mu_{\Delta})(t)$ est une représentation de la connaissance de l'état φ du dispositif telle que décrite dans les équations (1) et (4). A chaque instant t , l'annotation sémantique associée à l'état φ du dispositif est la conjonction de la description sémantique σ_{μ} et des observations μ_{Δ} (Figure 3). Ces annotations sont utilisées pour enrichir la BdC.

$\gamma(\sigma_{\mu} \oplus \mu_{\Delta})(t)$ est une représentation des propriétés contextuelles de telle sorte qu'elles soient héritables par d'autres dispositifs (voir [Interopérabilité des annotations](#))

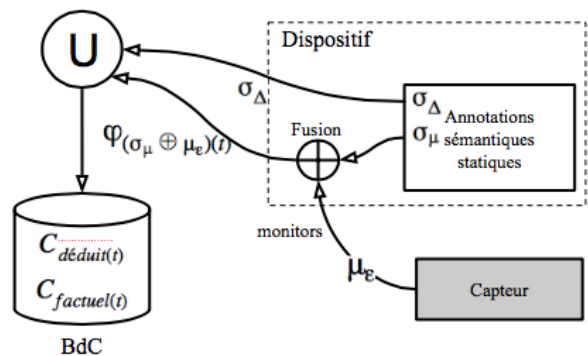


Figure 3 : Annotations statiques et dynamiques

Les concepts et les propriétés statiques ($\sigma_{\Delta} \rightarrow TBox$):

```
<owl:Class rdf:ID="Sensor">
<rdfs:subClassOf rdf:resource="#Device"/>
</owl:Class>
<owl:Class rdf:ID="Localizer">
<rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>
```

```

<owl:ObjectProperty rdf:ID="has_location">
<rdfs:domain rdf:resource="#Location"/>
</owl:ObjectProperty>
La sémantique des mesures ( $\sigma\mu$ )
<Location rdf:ID="#VALUE"/>
<owl:Class rdf:about="#INSTANCE_NAME">
#PROPERTY<has_location rdf:resource="#VALUE"/>
</owl:Class>

```

Annotations sémantiques générées dynamiquement

```

 $\varphi(\sigma\mu \oplus \mu\Delta)(t) \rightarrow ABox$ 
<Location rdf:ID="Kitchen"/>
<owl:Class rdf:about="Localizer_1">
<has_location rdf:resource="#Kitchen"/>
</owl:Class>
 $\gamma(\sigma\mu \oplus \mu\Delta)(t) \rightarrow ABox$ 
<has_location rdf:resource="#Kitchen"/>

```

Jusqu'à présent, à chaque instant t, le contenu de la BdC, à partir des annotations sémantiques formelles, correspond à une vue photographique des dispositifs et de l'environnement. Lorsque des requêtes sont émises, l'utilisabilité des services et la cohérence de la BdC avec le contexte sont considérées comme *implicites* et acquises. Cette représentation des connaissances n'intègre cependant pas les caractéristiques physiques des composantes du monde réel, en particulier les dimensions spatio-temporelles, qui permettraient de raisonner sur leur utilisabilité dans le temps et dans l'espace. Lors de requêtes, l'utilisabilité des dispositifs et des services ainsi que la cohérence de la BdC avec le contexte seraient alors considérées comme *explicites*.

La représentation multidimensionnelle des connaissances sur les dispositifs et les services implique la définition du type de portée « *contextualisé* » concernant les éléments ontologiques définis dans les annotations sémantiques formelles. Notés ($Instance \Rightarrow \gamma_{\Delta 1} \otimes \mu_{\Delta 2}$), ces éléments ontologiques décrivent des propriétés $\gamma_{\Delta 1}$ qui ne sont valides que dans le contexte $\mu_{\Delta 2}$. Comme indiqué dans l'état de l'art, on s'appuie sur les travaux de Chris Welty qui a défini un pattern de représentation de connaissance contextualisée, les 'Context slices' [19], basée sur les standards du web sémantique.

La représentation $\mu_{\Delta 2}$ d'un contexte, peut être issue de mesures discrètes ou continues. Dans le cas de mesures continues, un élément de discrétisation peut s'avérer nécessaire pour réduire le nombre de contextes possibles. Par exemple, dans le cas où $\mu_{\Delta 2}$ soit issue de la mesure d'un capteur de température, la discrétisation va permettre de partitionner les mesures en trois catégories : froid, tiède, chaud. Dans [41], l'auteur introduit la notion de « contexteur » utilisé pour la capture, le calcul et la distribution dynamique d'information contextuelle sous forme d'annotations. Un tel mécanisme pourrait alors être adapté et utilisé pour la discrétisation de valeurs continues issues de mesures sur l'environnement.

3.4 Interopérabilité des annotations

- Jusqu'à présent, nous avons décrit des dispositifs qui,
1. A l'aide d'éléments ontologiques locaux $\sigma\Delta$, décrivent sémantiquement les services qu'ils proposent et les concepts associés (dispositifs associée à des objets),
 2. A l'aide d'éléments ontologiques $\sigma\mu$ et de mesures $\mu\Delta$ décrivent sémantiquement un état φ à chaque instant t (dispositifs associés à des capteurs).

Cependant pour le moment, l'état φ d'un dispositif associé à un capteur ne décrit que lui-même. Comme nous l'avons expliqué précédemment, les capteurs extéroceptifs mesurent des phénomènes physiques de l'environnement et fournissent donc des propriétés contextuelles susceptibles d'être pertinente pour tous les dispositifs placés dans cet environnement (la température d'une pièce, un lieu, etc...). Les capteurs proprioceptifs, 'embarqués' sur des objets, produisent des propriétés contextuelles intrinsèques aux objets. L'état global φ des dispositifs associés aux objets est alors caractérisé par l'ensemble des mesures issues des capteurs proprioceptifs et extéroceptifs. Dans l'exemple de la Figure 4, l'état φ du dispositif associé à l'objet physique réel est caractérisé par des propriétés contextuelles (température, accélération et orientation) issues de capteurs proprioceptifs et par une propriété contextuelle issue d'un capteur extéroceptif (localisation).

Les annotations sémantiques du dispositif associé à l'objet physique réel doivent permettre la représentation et la description de cet état global. Par ailleurs, chacun des dispositifs associés à un capteur extéroceptif, peut être utilisé plusieurs fois pour participer à la description des différents objets placés dans le même environnement. Par ailleurs, la configuration d'utilisation d'un dispositif, quel qu'il soit, ne peut être anticipée. Il faut donc permettre une forte interopérabilité entre les dispositifs et donc entre les annotations sémantiques. L'interopérabilité des annotations sémantiques peut être considérée selon deux niveaux: syntaxique et sémantique [52]. Le niveau syntaxique assure une cohérence dans la représentation des données échangées.

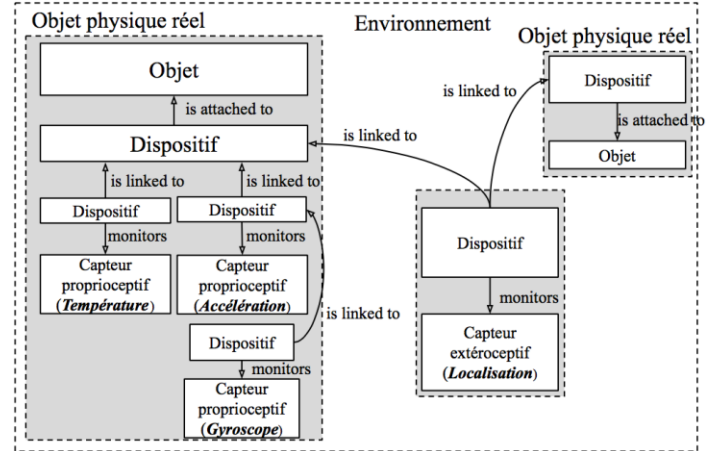


Figure 4 : Exemple de configuration de dispositifs

Nous avons la représentation des annotations sémantiques sur le format XML/RDF. L'utilisation d'éléments ontologiques et du langage OWL offre une solution générique au problème d'interopérabilité sémantique. Ainsi, l'instance d'un dispositif est en mesure d'hériter des propriétés issues de capteurs extéroceptifs des dispositifs qui sont liés à lui par l'intermédiaire de la relation 'is linked to'. Ce faisant, on obtient un graphe orienté $G=(\Delta, \text{prédicat})$ structurel d'héritage des propriétés contextuelles.

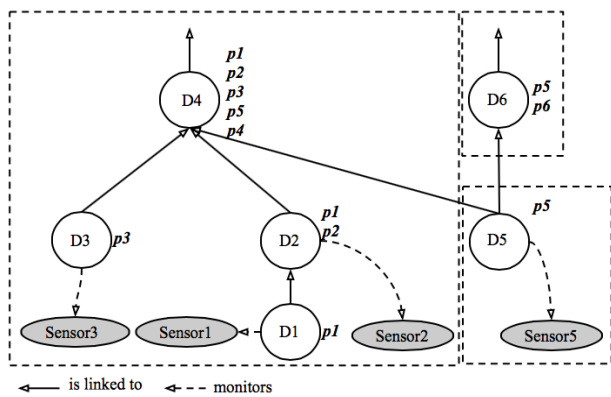


Figure 5 : Graphe structurel d'héritage des propriétés contextuelles

Dans l'exemple de la Figure 5, les propriétés contextuelles du dispositif D4 sont le résultat de l'agrégation des propriétés contextuelles des dispositifs D1, D2, D3 (proprioceptifs) et D5 (extéroceptif). D'un point de vue sémantique cela a un sens : si D1, D2, D3 et D5 mesurent respectivement l'accélération, l'orientation, la température et la localisation, alors D4 possède l'ensemble de ces propriétés contextuelles. On notera par ailleurs que les propriétés contextuelles issues des capteurs proprioceptifs, n'ont pas vocation à être héritées par les dispositifs situés hors de l'objet sur lequel ils sont embarqués (Par exemple, la propriété d'accélération d'un dispositif associé à un objet est intrinsèque à l'objet et ne peut être héritée par un autre objet).

Il n'y a pas de mécanisme d'héritage de propriétés avec les langages RDFS/OWL, qui, bien que permettant la description de classes, de sous-classes et de propriétés ne sont pas des langages orienté objet [42]. De ce fait, chacune des propriétés contextuelles est dupliquée dans les annotations sémantiques des dispositifs qui en héritent grâce au graphe structurel d'héritage des propriétés.

3.5 Graphe structurel d'héritage des propriétés

La représentation de la connaissance de l'état ϕ global d'un dispositif Δ , et sa cohérence avec le contexte physique réel dépend de la construction du graphe orienté structurel d'héritage des propriétés contextuelles. Dans ce graphe, les liaisons structurelles sont conditionnées par le prédicat "is linked to". Ces liens structurels représentent une réalité physique implicite au niveau de la BdC qui n'a pas connaissance de la façon dont est établi l'état ϕ global d'un dispositif Δ :

1. Des dispositifs sont embarqués physiquement sur un objet. La pertinence de l'utilisation d'observateurs dans la sémantique du dispositif est implicite et définie par le concepteur du dispositif,
2. Des dispositifs sont placés dans l'environnement et observent les objets placés dans leur champ d'action. La pertinence de l'utilisation d'observateurs dans la sémantique du dispositif peut être :
 - a. Explicite et définie par l'utilisateur. Mais l'utilisateur ne pense pas à tout et possède une connaissance incomplète de l'environnement qui, par ailleurs, évolue dynamiquement dans le temps.
 - b. Inférée par un moteur de raisonnement à partir de propriétés communes. La représentation du graphe d'héritage des propriétés contextuelles $G=(\Delta, \text{prédicat})$ à l'aide d'une ontologie et d'une BdC permettrait d'ajouter une sémantique structurelle à la représentation du

contexte. Il ne s'agirait pas ici d'inférer sur les assertions des domaines apportées par les annotations sémantiques formelles de chaque dispositif, mais sur l'ensemble des dispositifs et donc du graphe d'héritage des propriétés contextuelles $G=(\Delta, \text{prédicat})$. Par exemple, considérons trois dispositifs D3, D5, D4 (Figure 6) qui mesurent respectivement les propriétés contextuelles environnementales ta , l et i , liés à trois dispositifs D1, D2 et D6. Par transitivité, le dispositif D3 hérite des propriétés contextuelles l et i , le dispositif D5, hérite des propriétés contextuelles ta et i . Enfin, le dispositif D4 hérite des propriétés contextuelles ta et l et le dispositif D6 hérite des propriétés contextuelles l et i de D3. Les nouveaux liens créés impliquent alors de nouvelles inférences. Au final, tous les dispositifs qui apportent des mesures sur l'environnement forment des sous-graphes complets.

Cela nous amène à distinguer deux grands types d'environnements potentiels [43]:

1. **Les environnements à structure fixe dits "bornés"**. Les infrastructures réseau, matérielle et logicielle sont fixes, la position des différents capteurs présents dans l'environnement et connus au moment du développement. La topologie fixe de l'environnement facilite grandement l'élaboration du graphe d'héritage des propriétés.
2. **Les environnements mobiles** ou les dispositifs ne sont plus liés à un environnement physique donné mais sont mobiles. Chaque entité, dispositif ou capteur possède une dynamique qui lui est propre.

Ces environnements sont caractérisés par:

- a. Une infrastructure de topologie fortement variable et imprévisible,
- b. Un environnement physique imprévisible et partiellement accessible dû à des capteurs potentiellement inexistant,
- c. Un caractère non borné.

Dans ces environnements, les applications sont construites en suivant une approche bottom-up et sont décrites à travers un ensemble de règles indépendantes. L'élaboration du graphe d'héritage des propriétés contextuelles peut être aussi décrite par des règles. Cependant, contrairement à la construction d'une application à partir de règles basées sur la sémantique des fonctionnalités apportées par les dispositifs (que nous avons associé à la pertinence $P\zeta$), qui sont alors fonctionnellement liés pour répondre à un besoin, les dispositifs, dans le graphe d'héritage des propriétés contextuelles, ne peuvent être liés sémantiquement entre eux uniquement à partir de la sémantique de leurs mesures propres. **Il est nécessaire qu'ils aient alors un référentiel commun** pour pouvoir être liés sémantiquement de manière automatique ou inférée. Par exemple, si nous considérons deux dispositifs qui embarquent un GPS (référentiel commun), la liaison sémantique peut alors être réalisée de manière automatique si la distance qui sépare les deux dispositifs est inférieure à une certaine valeur.

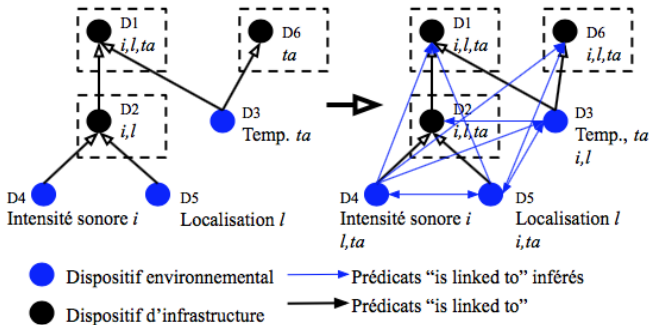


Figure 6 : Inférences sur les dispositifs qui mesurent l'environnement

4. RÉSULTATS

4.1 Ontologie générique

Une ontologie générique exprime les connaissances communes (concepts, relations) à tous les domaines. Cette ontologie correspond à l'agrégation de quatre sous-ontologies définissant les concepts et les relations relatifs aux domaines suivants :

Domaine	Sous-ontologie
Dispositifs et services	OWL-S [46]
Contextualisation des connaissances	Context-slices [19]
Dimension temporelle (Intervalles, instants,	OWL-Time [31]
Dimension spatiale (Points, régions et	SOWL spatial [36]

Table 1 : Sous-domaines de l'ontologie générique

4.2 Gestion dynamique de la connaissance

La représentation de la connaissance du contexte se décompose en trois couches distinctes:

1. Le graphe d'héritage des propriétés qui décrit la **sémantique structurelle** du contexte,
2. Les annotations sémantiques locales aux dispositifs, qui apportent une **sémantique fonctionnelle** aux services et aux observations de l'environnement,
3. La BdC centralisée qui contient **les faits et les individus** portant sur l'ensemble des domaines.

La dynamique du contexte et de ses éléments est prise en compte et gérée au niveau de chacune des trois couches (Table 2).

On notera que la disparition d'un dispositif ne supprime pas de la BdC les concepts qu'il a apporté. De ce fait, le niveau de description TBox dans la BdC va continuellement s'enrichir de la connaissance sur les dispositifs rencontrés (Figure 7).

4.3 WComp et le modèle SLCA

WComp [44], est une plate-forme de composition de services par assemblage de composants légers développée par l'équipe Rainbow. Cette plate-forme implémente le modèle SLCA (Service Lightweight Component Architecture) [45] ou une application se présente sous la forme d'un assemblage de composants logiciels basés sur le modèle LCA (Lightweight Component Architecture) et de services communiquant à l'aide d'événements.

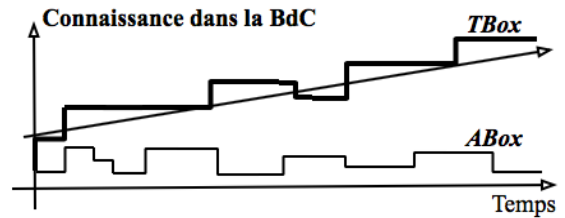


Figure 7 : Evolution TBox/ABox dans la BdC

Évènement	Impacts sur $G = (\Delta, \text{prédicat})$	Impacts sur les annotations sémantiques formelles	Impacts sur la BdC
Apparition d'un dispositif	Ajout d'un noeud	Nouveaux concepts et instances	+ TBox + ABox
	Ajout de liens "is linked to"	Héritage de propriétés	⊙ ABox
	Ajout d'un lien "monitors"	Nouvelles propriétés Héritage de propriétés	+ ABox ⊙ ABox
	Inférences → nouveaux liens 'is linked to'	Héritage de propriétés	⊙ ABox
Disparition d'un dispositif	Suppression de liens "is linked to"	Suppression de propriétés Héritage de propriétés	- ABox
	Suppression du lien "monitors"		⊙ ABox
	Suppression d'un noeud	Suppression de propriétés et d'instances Héritage de propriétés	- ABox ⊙ ABox
Objet dans le champ d'action du dispositif	Ajout d'un lien "monitors"	Ajout de propriétés Héritage de propriétés	+ ABox ⊙ ABox
Objet hors champ d'action du dispositif	Suppression de liens "monitors"	Suppression de propriétés Héritage de propriétés	- ABox ⊙ ABox
	Connectique défectueuse	Suppression de liens "is linked to" Suppression du lien "monitors"	
Changement de valeur de propriété		Mise à jour de la propriété, Héritage de propriétés.	⊙ ABox
	Nouveaux liens 'is linked to'	Nouveau contexte	+ ABox

Table 2 : Gestion dynamique de la connaissance

Des containers renferment des assemblages qui représentent alors des services composites manipulés pour gérer l'application. Une interface structurelle permettant de modifier dynamiquement l'assemblage et une interface fonctionnelle donnant accès aux services fonctionnels sont exportées.

Cette plateforme se base sur le protocole de service web pour dispositif UPnP (Universal Plug and Play). Tout comme DPWS (Device Profile for Web Services), ce protocole permet la gestion dynamique des dispositifs (apparition/disparition) et l'abonnement aux services proposés. Grâce à une communication basée sur des événements, il apporte une bonne réactivité et une interopérabilité forte entre les dispositifs et les services.

Cette plateforme est couplée à Conquer [45], une base de connaissance encapsulée au sein d'un dispositif UPnP.

Comme nous l'avons vu précédemment, nous avons deux types de dispositifs: ceux associés à des capteurs (proprioceptifs ou extéroceptifs) et producteurs d'information sur les objets ou l'environnement, et ceux associés à des objets consommateurs de ces informations. Pour ces deux types de dispositifs, nous créons

un service composite qui intègre, outre un proxy sur l'objet ou le capteur, un composant LCA qui gère les annotations sémantiques formelles. Enfin, en plus des interfaces structurelles et fonctionnelles, nous ajoutons une interface sémantique.

4.3.1 Dispositif associé à un capteur

Ce type de dispositif est producteur d'information sur l'objet ou l'environnement. Le composant LCA doit être en mesure d'acquérir les données, de les traiter (fusion) et de les transmettre aux consommateurs. Les concepts associés sont émis vers la BdC à la découverte du dispositif avec, éventuellement, l'état ϕ du dispositif :

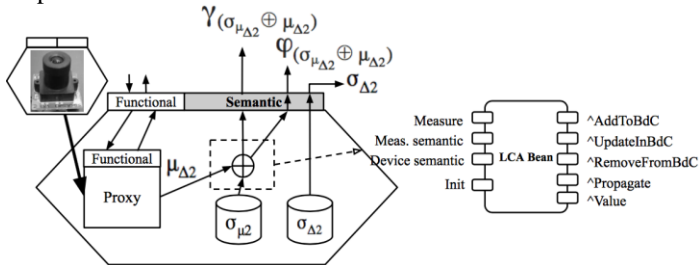


Figure 8 : Web service composite d'un dispositif associé à un capteur

4.3.2 Dispositif associé à un objet

Ce type de dispositif est consommateur d'information provenant de dispositifs associés à des capteurs extéroceptifs ou proprioceptifs.

Le composant LCA doit être en mesure d'acquérir les informations extérieures (produites par les dispositifs associés aux capteurs), de les fusionner et de transmettre, d'une part, son état global, issu de cette fusion, à la BdC, et, d'autre part, de les propager aux autres dispositifs auxquels il serait lié. Les concepts associés sont émis vers la BdC à la découverte du dispositif avec, éventuellement, l'état ϕ du dispositif (Figure 9).

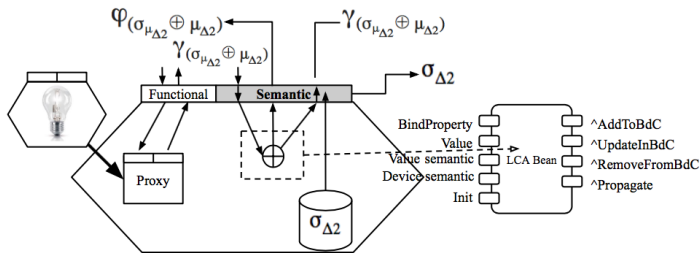


Figure 9 : Web service composite d'un dispositif associé à un objet

4.3.3 Composant de contextualisation

Ce composant (Figure 10) permet de spécifier que les propriétés d'un dispositif d'infrastructure ne sont valides que dans un contexte donné.

Ces services web composites sont alors mis en œuvre dans une application logicielle (Figure 11).

5. CONCLUSION & PERSPECTIVES

Dans le domaine de l'informatique ambiante, le mécanisme de sélection des services logiciels est primordial pour assurer leur cohérence de l'application avec le contexte (continuité de service) et les fonctionnalités recherchées. Cette sélection s'articule autour de deux propriétés:

1. La pertinence P_{ζ} des fonctionnalités offertes par les services en présence par rapport à une fonctionnalité globale que doit atteindre l'application,
2. L'utilisabilité $U_{\zeta(t)}$ des dispositifs et des services qui dépend de leur état physique et de leur disponibilité dans le temps et dans l'espace.

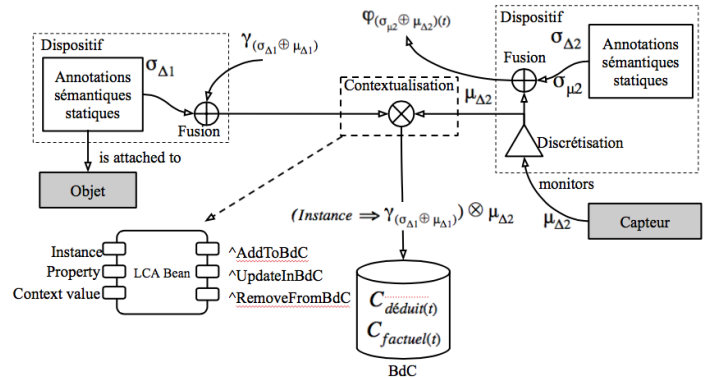


Figure 10 : Contextualisation de propriétés

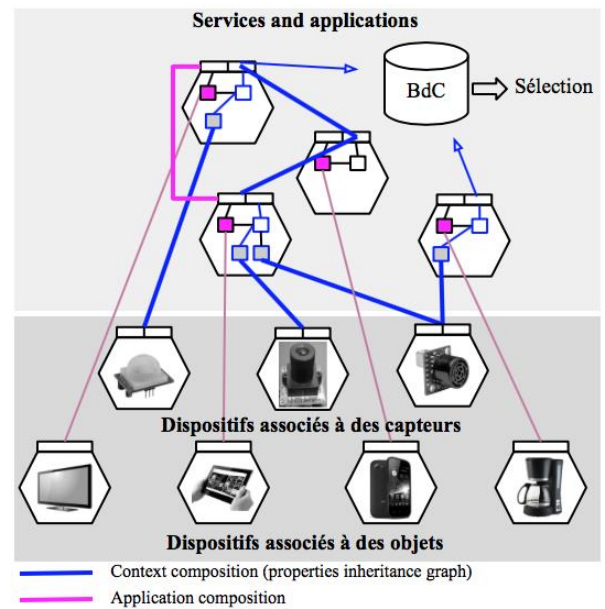


Figure 11 : Composition d'application

Dans ce cadre, l'utilisation d'annotations sémantiques formelles dynamiques, distribuées, et interopérables sur les dispositifs et les services, qui s'appuient sur les standards du web sémantique, nous a permis de construire une représentation dynamique de la connaissance sur les dispositifs, les services et le contexte qui se décompose en trois couches distinctes:

1. Le graphe d'héritage des propriétés contextuelles qui décrit la **sémantique structurelle** du contexte,
2. Les annotations sémantiques locales aux dispositifs qui apportent une **sémantique fonctionnelle** aux services et aux observations et enrichissent une BdC,
3. La BdC centralisée qui contient les faits et les individus portant sur l'ensemble des domaines sur lesquels des raisonnements sont appliqués et dont la **connaissance terminologique TBox évolue incrémentalement**.

La représentation du contexte est améliorée grâce à la mise en œuvre d'une **représentation contextualisée de la connaissance dans la BdC** qui permet de raisonner sur la validité des dispositifs et des services dans le temps, dans l'espace ou toute autre dimension pertinente dans le cadre d'objets ou de dispositifs physiques réels (température, qualité de service, etc...).

Cette approche, se heurte néanmoins à quelques problématiques qu'il conviendra d'étudier :

1. En premier lieu, les raisonnements sur les relations topologiques du temps et de l'espace nécessitent des modifications, tant au niveau du moteur d'inférence que sur le moteur de requêtes qui nécessitent la définition de nouvelles règles d'inférence et d'opérateurs *ad-hoc*. Dans ce cadre, l'utilisation de SWRL (Semantic Web Rule Language) [47] et SQWRL (Semantic Query-enhanced Web Rule Language) [48] peuvent être envisagées.
2. Chaque dispositif embarquant la description sémantique de son domaine, définie par des personnes différentes, il y a un risque :
 - a. De conflits ontologiques à gérer et d'alignements à réaliser [49]. L'ajout, dans les annotations sémantiques statiques, des synonymes sur les concepts définis peut alors aider [50].
 - b. De ne pas pouvoir faire des requêtes optimales, cela supposant une connaissance fine des concepts et des relations décrits à chaque instant dans la BdC, ce qui est peu probable du fait de l'aspect distribué des éléments ontologiques apportés par les annotations sémantiques formelles au grès de la découverte de nouveaux dispositifs, et donc pas connus par avance. L'utilisation d'un moteur de requêtes, au-dessus de SPARQL, en langage naturel peut alors être une solution [51].
 - c. Le graphe d'héritage des propriétés, dans le cadre d'un environnement non borné, est plus difficile à maintenir de manière automatisée et dynamique. Il nécessite un référentiel commun entre tous les dispositifs, ce qui semble improbable à obtenir.

6. REMERCIEMENTS

Je tiens à remercier chaleureusement **Mr Tigli Jean-Yves** pour sa disponibilité, sa patience sans faille et ses conseils avisés qui m'ont beaucoup aidé durant ce travail de recherche. Je remercie également **Mr Corby Olivier** pour les éclaircissements qu'il nous a donné concernant les technologies du web sémantique.

7. 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 applications*. 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., Grosz, 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