

Peuplement d'une ontologie modélisant le comportement d'un environnement intelligent guidé par l'extraction d'instances de relations

Driss Sadoun^{1, 2}

(1) LIMSI/CNRS, B.P 133 91403 Orsay Cedex, France

(2) Université Paris-Sud, 91400 Orsay, France

driss.sadoun@limsi.fr

RÉSUMÉ

Nous présentons une approche de peuplement d'ontologie dont le but est de modéliser le comportement de composants logiciels afin de faciliter le passage de descriptions d'exigences en langue naturelle à des spécifications formelles. L'ontologie que nous cherchons à peupler a été conçue à partir des connaissances du domaine de la domotique et est initialisée à partir d'une description de la configuration physique d'un environnement intelligent. Notre méthode est guidée par l'extraction d'instances de relations permettant par là-même d'extraire les instances de concepts liés par ces relations. Nous construisons des règles d'extraction à partir d'éléments issus de l'analyse syntaxique de descriptions de besoins utilisateurs et de ressources terminologiques associées aux concepts et relations de l'ontologie. Notre approche de peuplement se distingue par sa finalité qui n'est pas d'extraire toutes les instances décrivant un domaine mais d'extraire des instances pouvant participer sans conflit à un des multiples fonctionnements décrit par des utilisateurs.

ABSTRACT

Population of an Ontology Modeling the Behavior of an Intelligent Environment Guided by Instance Relation Extractions

We present an approach for ontology population, which aims at modeling the behavior of software components, for enabling a transition from natural language requirements to formal specifications. The ontology was designed based on the knowledge of the domotic domain and is initialized from a description of a physical configuration of an intelligent environment. Our method focuses on extracting relation instances which allows the extraction of concept instances linked by these relations. We built extraction rules using elements coming from syntactic analysis of user need descriptions, semantic and terminological resources linked to the knowledge contained in the ontology. Our approach for ontology population, distinguishes itself by its purpose, which is not to extract all instances describing a domain but to extract instances that can participate without any conflict to one of the multiple operation described by users.

MOTS-CLÉS : extraction de relations, peuplement d'ontologie, représentation des connaissances.

KEYWORDS: relation extraction, ontology population, knowledge representation.

1 Introduction

Les ontologies permettent une modélisation formelle d'un domaine, en décrivant ses concepts et les relations qu'ils entretiennent ainsi que les individus (ou instances) qui leurs sont associés. La conception d'une ontologie se fait en deux étapes, d'abord sa *conceptualisation* (Bendaoud *et al.*, 2007) (Wang et Turner, 2009) qui a pour but de faire émerger les concepts et relations représentant le domaine visé, ainsi que les axiomes qui permettront d'ordonner concepts et relations dans l'ontologie et de classer leurs instances, ensuite son *peuplement ou instanciation* qui consiste à associer des instances à des concepts et relations existants. C'est cette deuxième tâche qui sera traitée dans cet article.

La méthode de peuplement choisie est en général conditionnée par l'usage auquel est destinée l'ontologie. Dans le cas d'une ontologie modélisant un domaine de façon descriptive (De Boer *et al.*, 2007) (Maynard *et al.*, 2008), le but est de détecter, d'extraire et de lister systématiquement toutes les instances contenues dans les textes pouvant refléter un concept ou une relation du domaine à décrire. L'ontologie que nous avons conçue a pour vocation de modéliser le comportement dynamique d'un système de composants logiciels, en l'occurrence le comportement d'un environnement intelligent, que nous avons modélisé à partir de l'étude des domaines de réseaux de capteurs, des environnements intelligents et de la domotique (Sadoun *et al.*, 2011). Le processus de peuplement aura pour but d'extraire seulement les instances reflétant un comportement possible de ce système spécifique à des exigences d'utilisateurs.

L'ambition du projet (Envie Verte¹) dans lequel s'inscrit ce travail, est de permettre le pilotage d'un environnement intelligent en partant de descriptions en langue naturelle du comportement du système vers des spécifications formelles. Les spécifications sont données par des utilisateurs pour exprimer leurs besoins. L'intérêt est de permettre une vérification formelle de la configuration résultat avant de la déployer. A cette fin, nous avons proposé de passer par l'intermédiaire d'une ontologie du comportement de composants logiciels, pour faciliter le passage de l'informel vers le formel (Sadoun *et al.*, 2012). En effet, du fait de leur formalisme basé sur les logiques de description, les ontologies constituent un outil puissant pour l'aide à la vérification d'exigences et de spécifications formulées en langue naturelle. Nous exploitons ontologie et règles pour représenter et vérifier la consistance et la complétude des spécifications décrivant les comportements de composants logiciels. L'intérêt est de rendre ces comportements facilement paramétrables et adaptables.

L'ontologie ayant été définie, cet article en décrit une méthode de peuplement. La méthode proposée est centrée sur la détection d'instances de relations appartenant à la conceptualisation, point de départ pour l'identification d'instances de concepts.

La section suivante décrit l'état de l'art, la section 3 décrit brièvement l'ontologie, la section 4 décrit les descriptions d'environnement et des exigences. En section 5, nous introduisons la notion d'interprétation des exigences. La section 6 décrit la méthode de peuplement d'ontologie, puis en section 7 nous discutons la gestion de l'implicite avant la conclusion et la définition des perspectives.

1. <http://envieverte.limsi.fr/>, projet financé par DIGITEO 2010

2 État de l'art

Le peuplement automatique d'ontologie à partir de textes est une problématique qui a donné lieu à différentes approches (Alani *et al.*, 2003) , (Alani *et al.*, 2004) , (Amardeilh *et al.*, 2005), (Maynard *et al.*, 2008).

Afin d'extraire des connaissances relatives aux artistes sur le web, (Alani *et al.*, 2004) applique une analyse syntaxique et sémantique pour la reconnaissance d'entités nommées et l'extraction de relations entre instances. La base de connaissance ainsi peuplée servira à la génération automatique de biographies.

(Amardeilh *et al.*, 2005) propose l'enrichissement d'une base de connaissance (Topics Maps) contrainte par une ontologie du domaine. Pour cela il définit des règles d'acquisition en langage XPath pour mettre en correspondance des extractions linguistiques avec l'ontologie du domaine. Ces règles d'acquisition s'appuient sur le parcours d'arbres conceptuels contenant les informations pertinentes pour le domaine. (Maynard *et al.*, 2008) utilise une méthode linguistique et statistique pour la reconnaissance de termes et utilise des informations contextuelles de trois types : syntaxique, terminologique et sémantique. (Witte *et al.*, 2010) utilise des annotations issues d'analyses syntaxiques pour associer des entités nommées à des instances de concepts et relations.

L'identification de relations conceptuelles dans les textes est un problème majeur dans le processus de peuplement d'une ontologie, cette tâche a suscité plusieurs travaux.

(Makki *et al.*, 2008) décrit l'utilisation d'outils de traitement des langues pour le peuplement d'une ontologie de gestion des risques. En partant du principe que les relations sémantiques entre concepts sont le plus souvent représentées par des verbes, il extrait des relations issues d'une analyse des verbes et de leur entourage afin d'extraire des triplets, en utilisant des listes de verbes synonymes des relations conceptuelles construites à l'aide de WordNet. Dans notre méthode nous construisons des classes sémantiques contenant des termes, et pas uniquement des verbes, dont l'association avec des relations sémantiques ou concepts est propre au domaine d'application.

(Hasegawa *et al.*, 2004) propose une approche fondée sur le regroupement de paires d'entités nommées apparaissant dans des contextes similaires, où chaque paire d'une même classe est considérée comme une instance de la même relation. (Bentibebel *et al.*, 2009) généralise cette approche en construisant des classes d'association de termes représentant des relations conceptuelles potentielles. Une fois ces relations conceptuelles validées, ils y associent des règles d'extraction pour extraire d'autres occurrences.

(Lin et Pantel, 2001) avance que des relations binaires peuvent être représentées par des chemins dans les arbres de dépendances syntaxiques, ces chemins auront des significations similaires s'ils ont tendance à se connecter aux mêmes ensembles de mots. (Nakamura-Delloye et Stern, 2011) exploite ces chemins syntaxiques, pour identifier des relations entre couple d'entité nommées préalablement extraits (Nakamura-Delloye, 2011) et leurs règles. Nous généralisons cette approche pour la définition de règles d'extraction d'instances de relations et de concepts, en exploitant des chemins reliant différent types d'unités textuelles.

3 Ontologie de l'environnement intelligent

L'environnement intelligent consiste en un ensemble d'objets communicants (capteurs, actionneurs et processus de contrôle) qui peut être vu comme un réseau de capteurs. Ces objets influencent le comportement des équipements de l'environnement, sous des conditions bien définies. On peut distinguer une partie matérielle : les différents équipements, leur nombre, leur type, leur localisation etc. et une partie logicielle : la configuration du comportement.

Ces deux aspects permettent de décrire le comportement général d'un environnement intelligent :

- Un capteur détecte l'apparition d'un phénomène ou mesure un phénomène quantifiable dans un espace restreint.
- Un phénomène, pour être détecté ou mesuré par un capteur, doit être localisé dans la zone de capture du capteur et être du type perçu par le capteur (température, mouvement, ...)
- Un actionneur est fixé ou connecté à un appareil de l'environnement.
- Quand un phénomène (ou un ensemble de phénomènes) est mesuré ou détecté, un contrôle des informations collectées est effectué et peut conduire à l'activation d'un ou plusieurs actionneurs pour déclencher une ou plusieurs actions (allumer, éteindre, diminuer, augmenter) sur les appareils auxquels ils sont connectés.
- Un actionneur peut être activé par un capteur (ou un ensemble de capteurs), s'il est localisé dans sa zone (leur zone) de contrôle et gère le(s) même type(s) de phénomène.

La figure 1 représente la conceptualisation du comportement de l'environnement intelligent ².

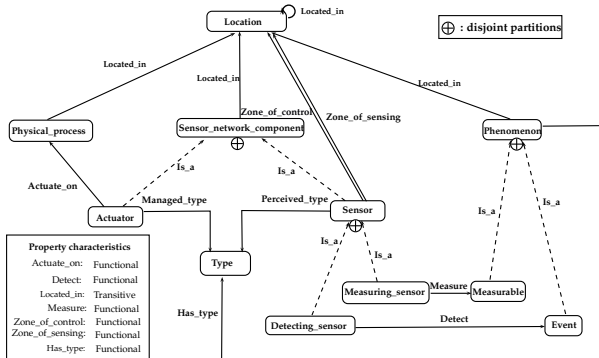


FIGURE 1 – Ontologie d'environnement intelligent

3.1 Rôle de l'ontologie

L'ontologie que nous avons conçue a pour vocation de faciliter le passage de descriptions des besoins à des spécifications formelles, en vue d'une vérification de la consistance et de la

2. L'absence d'un concept utilisateur dans l'ontologie, est dû au fait que les utilisateurs de l'environnement ne sont représentés que par les phénomènes qu'il peuvent engendrer tel qu'une pièce vide ou pleine.

cohérence de la configuration décrite en langage naturel par l'utilisateur de l'environnement intelligent. Elle n'a donc pas vocation à décrire exhaustivement le domaine des environnements intelligents et doit rester à un niveau de description assez élevé pour être la plus générale possible, afin de pouvoir représenter tous les cas de fonctionnement envisageables par un utilisateur. Pour plus de détails sur la conception de l'ontologie voir (Sadoun *et al.*, 2012)

La conceptualisation n'étant pas soumise au changement selon les besoins utilisateurs, la structure de l'ontologie (concepts et relations) a été conçue et figée à partir de l'étude du domaine des environnements intelligents et des réseaux de capteurs. Au contraire, l'instanciation de l'ontologie variera en fonction des descriptions de la configuration physique de l'environnement et des besoins utilisateur, et reflètera pour chaque instanciation un comportement particulier de l'environnement intelligent. La vérification du modèle résultant permettra de détecter d'éventuelles incohérences ou oublis dans les descriptions.

Nous avons décidé d'utiliser comme langage de modélisation, le langage OWL augmenté de SWRL et SQWRL pour sa grande expressivité de représentation des connaissances et ses mécanismes de raisonnement permettant d'effectuer un bon nombre des vérifications nécessaires avant le déploiement du système modélisé.

3.2 Connaissances modélisées

L'ontologie contient deux niveaux : terminologique (*TBox*) et assertionnel (*ABox*). Le premier niveau permet d'exprimer à l'aide d'axiomes des relations entre concepts, ce qui permet par exemple de définir le Domaine et le Range d'une relation ou de marquer la différence entre un capteur qui mesure et un capteur qui détecte.

La relation *Measure* a comme Domaine : *Measuring_Sensor* et comme Range : *Measurable*

Un capteur qui mesure est un capteur qui mesure un mesurable.

$Measuring_Sensor \equiv Sensor \wedge \exists Measure.Measurable$

La relation *Detect* a comme Domaine : *Detecting_Sensor* et comme Range : *Event*

Un capteur qui détecte est un capteur qui détecte un évènement.

$Detecting_Sensor \equiv Sensor \wedge \exists Detect.Event$

Le peuplement portera sur le second niveau de notre ontologie et consistera à créer les assertions identifiées à partir de l'analyse des descriptions. Différents cas de comportement seront modélisés par différentes instanciations, issues des descriptions d'environnement et des besoins utilisateurs. Chaque instanciation est une spécialisation de l'ontologie, qui représente un comportement spécifique d'un environnement donné correspondant aux exigences des utilisateurs.

3.3 Règles de comportement

Des règles SWRL permettent de représenter les règles de comportement dynamique de l'environnement, liant phénomènes et capteurs et capteur et actionneur. Certaines de ces règles sont indépendantes des descriptions, car elles représentent un comportement général du système. D'autres règles sont créées automatiquement lorsqu'elles reflètent un comportement spécifique aux exigences d'utilisateurs.

Les règles SWRL (1),(2),(3) représentent des règles de comportement général de l'environnement, la règle (4) représente une règle spécifique à des spécifications d'exigence. Les deux règles ci-dessous permettent de déduire (ou de créer) les instances de la relation `Shared_type(?s, ?e)`

Le phénomène p et le capteur s partagent le même type
 $Has_type(?p, ?t), Perceived_type(?s, ?t) \rightarrow Shared_type(?p, ?s)$ (1)

L'actionneur a et le capteur s partagent le même type
 $Managed_type(?a, ?t), Perceived_type(?s, ?t) \rightarrow Shared_type(?p, ?s)$ (2)

Ainsi, certaines instances sont issues du processus de peuplement quand d'autres sont déduites lors d'un raisonnement sur les instances présentes dans l'ontologie.

La règle SWRL ci-dessous, stipule que la présence d'un événement e et d'un capteur s qui partagent le même type, quand e est localisé dans la zone de détection de s , entraîne la détection de e par s ($Detect(?s, ?e)$)

$Event(?e), Sensor(?s), Shared_type(s?, ?e), Located_in(?e, l),$
 $Zone_of_sensing(?s, l) \rightarrow Detect(?s, ?e)$ (3)

Dans la règle spécifique ci-dessous, nous distinguons deux parties, une première fixe et indépendante des textes, et une seconde partie (soulignée dans la règle) générée en fonction des exigences décrites dans les textes.

Lorsqu'un capteur mesure une valeur supérieure à 20, l'actionneur du même type augmente l'appareil qu'il contrôle.

$Actuator(?a), Physical_process(?p), Actuate_on(?a, ?p), Located_in(?a, ?l), Measure(?s, ?m),$
 $Shared_type(?s, ?a), Zone_of_control(?s, ?l), \underline{Has_value(?m, ?v), \underline{lessThanOrEqual(?v, "20")}}$
 $\rightarrow \underline{Increase(?a, ?p)}$ (4)

4 Descriptions textuelles

Pour piloter son environnement, un utilisateur décrit, selon la configuration physique de son environnement, les fonctionnalités devant être configurées pour satisfaire ses besoins.

1. *Description de l'environnement intelligent* : décrit les composants de l'environnement (capteurs, actionneurs, processus physique, ...), leur nombre, leur type, leur localisation et leur manière d'interagir. Cette partie définit l'état statique de l'environnement et doit être traitée avant les besoins utilisateur.

Exemple : *L'appartement vert possède un couloir, deux chambres, une salle de bain, et un grand living qui contient une salle à manger et une cuisine. Chaque pièce est équipée de capteurs de mouvement. Chaque ampoule est équipée d'un actionneur.*

2. *Besoins utilisateur* : décrit comment et sous quelles conditions les objets de l'environnement doivent interagir. Cela permet de produire différentes instanciations de l'ontologie selon différents scénarios.

Exemple : *Quand une personne est dans le living room, y allumer la lumière.*

Le but de notre projet étant la vérification automatique d'exigences, la description de l'environnement qui ne fait qu'énumérer des composants physiques et leurs modes d'utilisation. ne nous

intéressera pas dans la phase d'extraction automatique d'informations pour le peuplement de l'ontologie. Cette description sert néanmoins à initialiser l'ontologie (première instanciation) avec les connaissances liées aux caractéristiques physiques de l'environnement qui seront non soumises aux changements. L'instanciation automatique s'appuiera donc plus particulièrement sur les informations contenues dans les descriptions des besoins utilisateur.

Dans le cadre de notre projet nous partons de descriptions en anglais. Pour la collecte de descriptions de besoins utilisateurs (en anglais et français), nous avons mis en place une plateforme³ permettant à des participants de décrire un environnement intelligent, en l'occurrence une maison intelligente, dont la configuration physique est représentée par un plan et une description de différents objets de l'environnement et de leur mode d'interaction. Cette description reste néanmoins assez générale de sorte à permettre aux participants d'exprimer différentes idées de configuration correspondant à des besoins variés.

5 Interprétation des exigences utilisateur

Les descriptions utilisateur décrivent le comportement de l'environnement en fonction de la présence de différents phénomènes. Comme les phénomènes possibles ne peuvent généralement pas tous apparaître en même temps, ces descriptions contiennent en fait différents cas de fonctionnement qu'on appelle *interprétation*.

Une interprétation représente l'état de l'environnement intelligent à un instant t , cet état sera déterminé par toutes les instances qui seront présentes dans l'ontologie à l'instant t . Une interprétation ne contiendra pas toutes les instances reconnues dans les textes mais seulement un sous-ensemble cohérent d'instances modélisant une partie des exigences utilisateur.

Ces instances ont une existence conditionnelle, comme cela est illustré par les deux phrases suivantes "When someone is in the kitchen, turn on the light." et "When nobody is in the kitchen, turn off the light.". Les instances de relations et de concepts pouvant être extraites de ces phrases devront donc logiquement appartenir à deux interprétations différentes, car il ne peut y avoir au même instant quelqu'un et personne dans la cuisine. Dans le cas contraire les mêmes instances déclencheraient deux règles contradictoires engendrant l'assertion des deux relations Turn_on et Turn_off entre les mêmes instances, ce qui créera une incohérence lors du raisonnement.

L'instanciation doit donc prendre garde à ne pas ajouter toutes les instances extraites en même temps ou du moins celles pouvant être en conflit. Pour le moment, l'instanciation et le raisonnement portent sur les instances d'une phrase à la fois pour éviter les incohérences dues à des instances opposées pouvant déclencher des règles contradictoires. Il y a donc autant d'instanciations que de phrases présentes dans les descriptions. L'identification d'instances en conflit s'inscrit dans nos travaux futurs.

Afin de pouvoir naviguer entre descriptions et ontologie, chaque instance extraite se voit allouer un indice composé de deux numéros, celui de la phrase dont elle est issue ainsi que la position de son nœud dans l'arbre syntaxique.

Exemple : Si l'instance i est issue de l'analyse de la phrase numéro 11 et que le numéro du nœud représentant i dans l'arbre syntaxique est 4, on crée les deux instances de relations $Phrase_number(i, 11)$ et $Node_number(i, 4)$.

3. <http://perso.limsi.fr/Individu/sadoun/Application/en/SmartHome.php>

Cette référence à l'arbre syntaxique va permettre d'aller dans les descriptions pointer dans les phrases les termes correspondant aux instances qui sont ambiguës ou qui créent des incohérences, pour permettre aux utilisateurs de corriger et d'améliorer leurs exigences.

6 Peuplement de l'ontologie

La figure 2 illustre le processus de peuplement à partir de spécifications en langue naturelle. L'ontologie est d'abord initialisée avec les instances qui décrivent sa configuration physique. L'utilisation des résultats d'une analyse syntaxique des descriptions utilisateurs faite par le Stanford Parser (de Marneffe *et al.*, 2006), de ressources terminologiques et des connaissances déjà présentes dans l'ontologie, permettent d'appliquer des règles pour l'extraction d'instances. Les instances extraites vont peupler l'ontologie et peuvent à leur tour servir à l'identification de nouvelles instances.

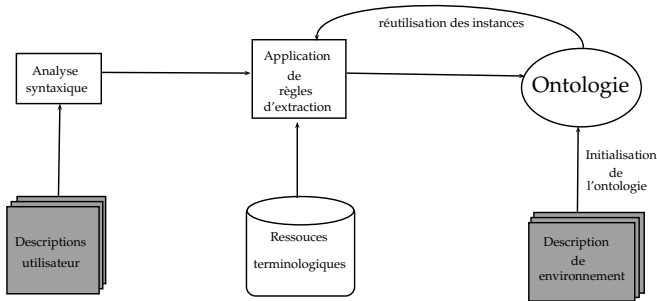


FIGURE 2 – Processus de peuplement de l'ontologie

6.1 Initialisation de l'ontologie

La phase d'initialisation a pour but de peupler l'ontologie avec toutes les instances qui représentent la configuration physique de l'environnement intelligent (instances de capteurs, d'actionneurs, appareils, localisations et relations entre instances). Ces instances sont peu sujettes aux changements, et donc l'initialisation ne se fait qu'une fois via une interface de saisie.

Ci-dessous un exemple d'instanciation du concept *Detecting_sensor* :

Detecting_sensor(s_1) : s_1 est une instance de *Detecting_sensor*. (I_1)

Perceived_type($s_1, movement$) : s_1 détecte des phénomènes de type *movement*. (I_2)

Zone_of_sensing($s_1, kitchen$) : s_1 a comme zone de détection *kitchen* (I_3)

Zone_of_control($s_1, kitchen$) : s_1 a comme zone de contrôle *kitchen* (I_4)

Lors de l'initialisation de l'ontologie, une instance est créée pour chaque capteur avec comme

propriétés son type, sa zone de détection et sa zone de contrôle.

6.2 Extraction des instances

Nous avons choisi de prendre comme point de départ la détection d'instances de relations plutôt que de concepts pour aborder le processus de peuplement, choix qui se justifie, par la liste plus limitée des formulations pouvant représenter des instances de relations dans les corpus. De plus, seule l'extraction des instances de concepts participant à une relation sémantique modélisée dans l'ontologie nous intéresse.

Pour extraire une instance de relation, il faut reconnaître d'abord des concepts pouvant la refléter, puis vérifier que les concepts auxquels elle est liée dans le texte correspondent aux concepts du Domaine ou du Range de la relation.

Nous distinguons deux types de classes sémantiques, certaines calquées sur les concepts et relations modélisés dans l'ontologie, telle que la classe "Turn on" contenant tous les termes pouvant refléter l'action d'allumer ou de mettre en marche (light on, switch on, open, ...), à l'opposé de la classe "Turn off" (light off, switch off, close, ...) qui reflète l'action d'éteindre ou de mettre à l'arrêt, ou la classe "Location" contenant tous termes correspondant à une instance de localisation issue de la description de l'environnement (hall, kitchen, parent's bedroom, living room, ...) et d'autres classes sémantiques portant sur des connaissances moins spécifiques, telle que la classe Sémantique *Conditional Introducer* (when, if, each time, ...) qui contient les termes introduisant une condition.

L'utilisation de ressources telles que WordNet, VerbNet ou FrameNet s'avère utile pour la construction de ressources terminologiques. Néanmoins, étant basées sur des connaissances générales, les classes sémantiques qu'elles définissent se révèlent être moins pertinentes en domaine spécialisé et ne correspondent pas aux concepts que nous avons définis.

L'ontologie permet de définir un vocabulaire conceptuel sans toutefois contenir le volet terminologique. L'essentiel des connaissances terminologiques que nous utilisons pour l'extraction d'instances de relations et de concepts sont issues de ressources terminologiques extérieures à l'ontologie mais liées aux connaissances qu'elle modélise.

6.3 Extraction des instances

Nous avons choisi de prendre comme point de départ la détection d'instances de relations plutôt que de concepts pour aborder le processus de peuplement, choix qui se justifie par la liste plus limitée des formulations pouvant représenter des instances de relations dans les corpus. De plus, seule l'extraction des instances de concepts participant à une relation sémantique modélisée dans l'ontologie nous intéresse.

Pour extraire une instance de relation, il faut reconnaître d'abord des termes et structures syntaxiques pouvant la refléter, puis vérifier que les concepts auxquels elle est liée dans le texte correspondent aux concepts du Domaine ou du Range de la relation.

Nous avons donc construit des lexiques permettant de lier termes et concepts. Nous distinguons deux types de classes sémantiques. Certaines sont calquées sur les concepts et relations modélisés

dans l'ontologie, telle que la classe "Turn on" contenant tous les termes pouvant refléter l'action d'allumer ou de mettre en marche (light on, switch on, open, ...), à l'opposé de la classe "Turn off" (light off, switch off, close, ...) qui reflète l'action d'éteindre ou de mettre à l'arrêt, ou la classe "Location" contenant tous termes correspondant à une instance de localisation issue de la description de l'environnement (hall, kitchen, parent's bedroom, living room, ...). D'autres classes sémantiques portent sur des connaissances moins spécifiques, telle que la classe Sémantique *Conditional Introducer* (when, if, each time, ...) qui contient les termes introduisant une condition.

L'utilisation de ressources telles que WordNet, VerbNet ou FrameNet s'avère utile pour la construction de ressources terminologiques. Néanmoins, étant basées sur des connaissances générales, les classes sémantiques qu'elles définissent se révèlent être moins pertinentes en domaine spécialisé et ne correspondent pas directement aux concepts que nous avons définis.

L'ontologie permet de définir le vocabulaire conceptuel sans toutefois contenir le volet terminologique. L'essentiel des connaissances terminologiques que nous utilisons pour l'extraction d'instances de relations et de concepts sont issues de ressources terminologiques extérieures à l'ontologie mais liées aux connaissances qu'elle modélise.

6.4 Construction de règles d'extraction

6.4.1 Chemin syntaxique

La construction de règles d'extraction s'appuie sur une analyse des dépendances syntaxiques. Dans un arbre de dépendances (cf figure 3), les dépendances syntaxiques lient deux nœuds, le premier que nous appellerons nœud *directeur* et le second nœud *dépendant*. Un chemin syntaxique entre deux nœuds est représenté par les dépendances qui les lient.

Par exemple, soit la dépendance syntaxique *subj(is, someone)*

- nsubj : nom de la relation (dépendance sujet nominal)
- is : mot contenu dans le nœud directeur.
- someone : mot contenu dans le nœud dépendant.

Exemple : *When someone is in the kitchen, turn on the light.* La figure 3 représente l'arbre des dépendances syntaxiques de cette phrase⁴.

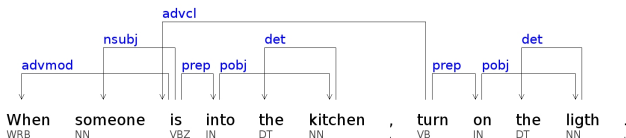


FIGURE 3 – Arbre des dépendances syntaxiques

4. Image produite à l'aide de DependenceSee.jar <http://chaoticcity.com/dependensee-a-dependency-parse-visualisation-tool/>

6.4.2 Règles d'extraction

Les règles d'extraction que nous construisons doivent être assez générales pour s'appliquer sur différentes formulations de phrases et assez spécifiques pour n'extraire que les instances de relations et de concepts les plus pertinentes.

L'utilisation de chemins syntaxiques pour reconnaître la présence d'instances candidates à l'extraction permet de traiter différents types de phrases car les dépendances syntaxiques formant ces chemins peuvent apparaître dans différentes formulations. La vérification de la pertinence des instances candidates se fait à l'aide d'éléments issus de l'analyse syntaxique des descriptions, des ressources terminologiques, ainsi que des connaissances modélisées dans l'ontologie.

Dans les exemples qui suivent, les termes commençant par des minuscules et contenant des majuscules représentent des fonctions codées en java permettant d'accéder aux ressources terminologiques ou aux connaissances contenues dans l'ontologie.

Règle pour l'extraction d'une instance de la relation `Located_in()` :

Cette règle identifie tous les chemins syntaxiques existant entre les deux dépendances `prep`⁵ et `pobj`⁶, en vérifiant qu'il existe un nœud commun p entre elles, que ce nœud est une préposition de lieu (`isPrepLocation(p)`), que le nœud directeur l de la dépendance `pobj` corresponde à une instance du concept `Location` (`isLocation(l)`) existante dans l'ontologie, puis que le nœud dépendant i de la dépendance `prep` a comme catégorie syntaxique un nom ou un verbe.

$$pobj(l, p) \wedge prep(p, i) \wedge isPrepLocation(p) \wedge isLocation(l) \wedge (isNoun(i) \vee isVerb(i)) \\ \rightarrow Located_in(i, l)$$

L'application de cette règle à la phrase précédente a pour résultat :

$$pobj(kitchen, \underline{in}) \wedge prep(\underline{in}, is) \wedge isPrepLocation(in) \wedge isLocation(kitchen) \\ \wedge (isNoun(is) \vee isVerb(is)) \rightarrow Located_in(is, kitchen)$$

Dans le cas où le directeur de la relation `Located_in` i est issu du verbe auxiliaire `to be` et qu'il existe une relation `nsubj`⁷ entre cet auxiliaire et un nom, l'auxiliaire est remplacé par ce nom :

$$Located_in(i, l) \wedge isToBe(i) \wedge nsubj(i, n) \rightarrow Located_in(n, l)$$

Ce qui a pour résultat :

$$Located_in(is, kitchen) \wedge isToBe(is) \wedge nsubj(is, someone) \rightarrow Located_in(someone, kitchen)$$

L'instance de relation `Located_in(someone, kitchen)` extraite ne permet pas de typer ou de déduire lors d'un raisonnement sur les instances, le type de `someone` car le domaine de la relation `Located_in` est égal à tous les concepts (`Thing`). Aussi, nous décrivons des règles d'extraction d'instances de concepts permettant de désambigüiser des instances qui ne pourraient l'être au sein de l'ontologie.

Règle pour l'extraction d'une instance du concept `Event` :

Cette règle identifie tous les chemins syntaxiques existant entre les deux dépendances `advmod`⁸

5. `pobj` : objet de la préposition

6. `prep` : modificateur prépositionnel

7. `nsubj` : sujet nominal

8. `advmod` : modificateur adverbial

et *nsubj*, en vérifiant qu'il existe un nœud commun *v* entre elles, que ce nœud est issu du verbe auxiliaire *to be*, que le nœud dépendant *c* de la dépendance *advmod* appartient à la classe Sémantique *Conditional Introducer* et que le nœud dépendant *n* de la dépendance *nsubj* participe déjà à une instance de la relation *Located_in* dans l'ontologie, car un événement n'a aucun intérêt si l'on ne connaît pas sa localisation.

$$advmod(v, c) \wedge nsubj(v, n) \wedge isToBe(v) \wedge IsCondIntroducer(c) \wedge Located_in(n, l) \\ \rightarrow Event(n)$$

Ce qui a pour résultat :

$$advmod(is, When) \wedge nsubj(is, someone) \wedge isToBe(is) \wedge IsCondIntroducer(When) \wedge \\ Located_in(someone, kitchen) \rightarrow Event(someone)$$

Règle pour l'extraction d'une instance de la relation *Has_type* :

La règle ci-dessous identifie la présence des deux instances *Event* et *Located_in* issues de l'ontologie, contenant l'instance commune *n* dénotant le type *t* (information présente dans nos ressources terminologiques et sémantiques)

$$Event(n) \wedge Located_in(n, l) \wedge denoteType(n, t) \rightarrow Has_type(n, t)$$

Le terme *someone* en tant qu'évènement dénote le type *movement*, on donc a pour résultat :

$$Event(someone) \wedge Located_in(someone, kitchen) \wedge denoteType(someone, movement) \\ \rightarrow Has_type(someone, movement)$$

A l'aide de ces règles, nous obtenons des instances de relations et de concepts qui sont ajoutées à l'ontologie et qui peuvent être réutilisées dans d'autres règles.

L'ordre d'application des règles d'extraction n'a pas de conséquences sur le résultat du peuplement. Néanmoins comme l'application de certaines règles nécessite des résultats (instances) issues de l'application d'autres règles, il est préférable d'ordonner leur application pour optimiser le processus.

Les règles varient en fonction des instances à extraire, mais aussi pour une même instance en fonction de toutes les formulations. Dans un premier temps, nous compléterons celles-ci pour définir les plus courantes, afin de valider notre méthodologie d'instanciation de l'ontologie. Nous étudierons ensuite comment disposer d'un plus large corpus si le nombre de descriptions collectées ne nous permet pas d'avoir recours à des méthodes d'acquisition automatique.

7 Discussion

Dans les descriptions utilisateur, les références aux capteurs ne sont pratiquement jamais présentes, ces références sont souvent implicites, réduisant le capteur à son fonctionnement, ce que l'on voit bien dans la phrase : *When someone is in the kitchen, turn on the light.*

Or, une instance de relation que nous souhaitons extraire est la relation *Detect*, qui représente la détection d'un événement par un capteur.

C'est là que réside tout l'intérêt de notre modèle, qui ne s'appuie pas seulement sur les informations présentes dans les descriptions, mais tient aussi compte de connaissances issues du

domaine, qui, le cas échéant, permettent de lever les ambiguïtés. L'ontologie modélise ainsi le fait qu'un capteur est identifiable par sa zone de détection et le type d'événement qu'il peut détecter. Aussi la reconnaissance d'instances de ces concepts permettra de déduire le capteur concerné.

Nous allons détailler ce processus pour l'exemple :

Les connaissances issues des initialisations I_1, I_2, I_3 (Section 6.1) font que le déclenchement de la règle SWRL (3) (Section 3) n'est plus soumis qu'à l'extraction des instances suivantes : `Located_in(someone,kitchen)`, `Event(someone)`, `Has_type(someone,movement)`, car le capteur devant intervenir dans la règle, sera déduit automatiquement.

Instanciation de la règle SWRL (3)

Event(someone), Sensor(s1), Shared_type(s1,someone), Located_in(someone,kitchen), Zone_of_sensing(s1,kitchen) → Detect(s1,someone)

8 Conclusion et travaux futurs

Nous avons décrit une méthode pour le peuplement automatique d'une ontologie modélisant le comportement d'un environnement intelligent, laquelle a pour but de faciliter le passage de descriptions de besoins utilisateurs en langue naturelle vers des spécifications formelles de ces besoins.

Cette problématique est distincte des approches habituelles pour le peuplement d'ontologie à partir de textes qui ont pour but de peupler une ontologie avec toutes les instances extraites des textes permettant de décrire un domaine. L'approche adoptée dans cet article vise à extraire uniquement les instances participant à la modélisation d'un des multiples comportements d'un système de composants logiciels.

Notre méthode est guidée par l'extraction d'instances de relations permettant par là-même d'extraire les instances de concepts liées par ces relations. L'extraction d'instances se fait à l'aide de règles construites manuellement à partir de résultats d'analyse syntaxique, de ressources terminologiques et des connaissances modélisées dans l'ontologie. Ainsi seules les instances les plus pertinentes, car participant à au moins une relation conceptuelle, sont extraites. Le recours aux connaissances modélisées par l'ontologie offre un cadre formel pour la résolution des ambiguïtés.

La modélisation résultante permet la navigation entre descriptions des exigences en langue naturelle et spécifications formelles, pour la vérification de ces exigences et leur amélioration.

Nos travaux futurs porteront sur l'approfondissement de l'acquisition de règles, la création automatique de règles SWRL spécifiques au comportement décrit dans les descriptions utilisateurs et l'identification d'instances pouvant être en conflit et dont la présence simultanée dans une même instanciation serait incohérente.

Références

ALANI, H., KIM, S., MILLARD, D. E., WEAL, M. J., HALL, W., LEWIS, P. H. et SHADBOLT, N. (2004). Using protege for automatic ontology instantiation. *In 7th International Protégé Conference.*

- ALANI, H., KIM, S., MILLARD, D. E., WEAL, M. J., HALL, W., LEWIS, P. H. et SHADBOLT, N. R. (2003). Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*.
- AMARDEILH, F., LAUBLET, P. et MINEL, J.-L. (2005). Document annotation and ontology population from linguistic extractions. In *Proceedings of the 3rd international conference on Knowledge capture*.
- BENDAOU, R., ROUANE HACENE, M., TOUSSAINT, Y., DELECROIX, B. et NAPOLI, A. (2007). Construction d'une ontologie à partir d'un corpus de textes avec l'ACF. In *IC 2007*.
- BENTIBEBEL, R., NAZARENKO, A. et SZULMAN, S. (2009). Mise en lumière de relations sémantiques pour la construction d'ontologies à partir de textes. In *TIA 2009*.
- DE BOER, V., VAN SOMEREN, M. et WIELINGA, B. J. (2007). Relation instantiation for ontology population using the web. In *Proceedings of the 29th annual German conference on Artificial intelligence*.
- de MARNEFFE, M.-C., MACCARTNEY, B. et MANNING, C. D. (2006). Generating typed dependency parses from phrase structure trees. In *LREC*.
- HASEGAWA, T., SEKINE, S. et GRISHMAN, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.
- LIN, D. et PANTEL, P. (2001). Discovery of inference rules for question answering. *Natural Language Engineering*.
- MAKKI, J., ALQUIER, A.-M. et PRINCE, Vp, V. (2008). Ontology Population via NLP Techniques in Risk Management. In *ICSWE : Fifth International Conference on Semantic Web Engineering*.
- MAYNARD, D., LI, Y. et PETERS, W. (2008). Nlp techniques for term extraction and ontology population. In *Proceeding of the 2008 conference on Ontology Learning and Population : Bridging the Gap between Text and Knowledge*.
- NAKAMURA-DELLOYE, Y. (2011). Named entity extraction for ontology enrichment. In *IPSI Special Interest Group - Information Fundamentals and Access Technologies (IFAT)*.
- NAKAMURA-DELLOYE, Y. et STERN, R. (2011). Extraction de relations et de patrons de relations entre entités nommées en vue de l'enrichissement d'une ontologie. In *TOTh 2011 : Terminologie & Ontologie : Théories et Applications*.
- SADOUN, D., DUBOIS, C., GHAMRI-DOUDANE, Y. et GRAU, B. (2011). An ontology for the conceptualization of an intelligent environment and its operation. *Mexican International Conference on Artificial Intelligence*.
- SADOUN, D., DUBOIS, C., GHAMRI-DOUDANE, Y. et GRAU, B. (2012). Formalisation en OWL pour vérifier les spécifications d'un environnement intelligent. In *Actes de la conférence RFIA 2012*.
- WANG, F. et TURNER, K. J. (2009). An ontology-based actuator discovery and invocation framework in home care systems. In *Proceedings of the 7th International Conference on Smart Homes and Health Telematics : Ambient Assistive Health and Wellness Management in the Heart of the City*.
- WITTE, R., KHAMIS, N. et RILLING, J. (2010). Flexible ontology population from text : The owl exporter. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.