Classes d'annotation pour l'annotation sémantique

Sylvain Tenier, Yannick Toussaint

Atelier OntoTexte-2007, Sophia-Antipolis, 10 octobre 2007

Classes d'annotation pour l'annotation sémantique

Sylvain Tenier et Yannick Toussaint

Équipe Orpailleur - LORIA - Campus Scientifique
BP 239 - 54506 Vandoeuvre-lès-Nancy Cedex

Résumé

Les classes d'annotation constituent une méthode d'annotation sémantique de pages web fondée sur les logiques de descriptions. Elles désignent l'annotation à la fois comme processus et comme résultat de ce processus. Cette approche est motivée par un parallèle entre la structure d'une page web et la sématique qui lui est associée. Ces deux dimensions de structure et de sémantique sont formalisées en OWL-DL un langage fondé sur les logiques de descriptions. L'annotation est ensuite traitée comme un problème d'instanciation : une page web est interprétée comme instance d'une classe d'annotation en fonction de sa structure et de sa sémantique.

Mots clés

Représentation des connaissances, annotation sémantique, web sémantique.

1. Intégrer les connaissances au processus d'annotation sémantique

Les moteurs de recherche devraient être capable de répondre à des requêtes complexes tenant compte des connaissances de l'utilisateur. La proposition du Web Sémantique (Berners-Lee, 1999) vise ainsi à permettre à des agents logiciels d'accéder à des pages web, de manipuler et de raisonner sur leur contenu. Pour cela, ce contenu doit être représenté formellement dans un langage de représentations des connaissances, tel que OWL-DL.

L'annotation représente formellement le contenu d'une page web existante. Une annotation désigne à la fois une métadonnée associée à une partie du document et le processus de génération de cette métadonnée. Une annotation est sémantique lorsqu'elle se réfère à une ontologie, qui décrit, dans une logique de descriptions, les concepts et les relations entre les concepts d'un domaine. L'ontologie est au centre de la méthodologie proposée dans cet article : d'une part, elle guide le processus d'annotation ; d'autre part, les annotations générées sont des instances de cette ontologie.

Comment distinguer les pages web pertinentes pour un domaine ? Les pages portant sur un domaine donné partagent généralement le même type d'information. Par exemple, la page d'une équipe de recherche présente les thèmes et projets de ses chercheurs. Il s'agit de conditions nécessaires et suffisantes : il faut qu'une personne soit reliée à un thème et à un projet pour être considérée comme un chercheur. La présence de chercheurs permet alors d'identifier

la page comme une page d'équipe. Ces conditions sont formalisées dans l'ontologie sous la forme de concepts définis. En outre, ces pages présentent généralement des régularités. Ces connaissances sur la structure des pages sont intégrées dans l'ontologie pour prendre en compte l'interaction entre la structure et la sémantique de la page.

Le problème posé est l'identification dans des pages web d'instances de ces concepts définis. Comme le montre Uren et al. (2006), les travaux actuels en annotation sémantique exploitent des techniques d'extraction d'information pour associer les chaînes de caractères de la page à des concepts de l'ontologie. La structure de la page est en outre exploitée dans certaines méthodes, comme Kushmerick (1997) et Carme (2007) pour identifier les relations entre les éléments. Cependant, pour pouvoir identifier des instances de concepts définis, il est nécessaire d'intégrer les connaissances au processus d'annotation.

Nous proposons la notion de classe d'annotation comme méthode d'annotation guidée par les connaissances. Les classes d'annotations prennent en compte l'interaction entre la structure et la sémantique des pages web en les formalisant dans un langage commun fondé sur les logiques de descriptions. Dans un premier temps, des structures représentatives des pages d'équipes sont formalisées en LD. Des classes d'annotation sont alors définies comme des sous-classes de ces concepts de structure : chaque classe d'annotation représente un concept du domaine associé à une structure. Enfin, étant donnée une page à annoter, le processus d'annotation est défini comme une opération d'instanciation au sens des logiques de descriptions et les annotations résultantes comme des instances de concepts.

La prochaine section présente l'importance de la structure dans l'interprétation des pages web. La section 3 introduit les concepts de structure. La section 4 décrit l'ontologie de domaine. La section 5 présente les techniques syntaxiques d'identification des concepts primitifs utilisées dans la littérature. La section 6 présente l'application des classes d'annotation à l'annotation de pages web. Enfin, les résultats sont discutés dans la section 7.

2. Le rôle de la structure dans l'interprétation

La figure 1 présente une page web annotée. Tous les éléments constitutifs sont présents : personnes, thèmes et projets. Le problème est d'établir les relations entre ces éléments. Ceux-ci sont contenus dans des structures tabulaires, pour lesquelles les méthodes linguistiques telles que (Amardeilh *et al.*, 2005) ne fonctionnent pas.

La régularité de la structure permet cependant de définir un modèle et de lui associer une sémantique. Par exemple, un extrait du code HTML présenté figure 1 est :

```
<div><h2>Jack</h2>
     <p><em>Semantic Web</em>
        <a>Knowledge Web</a></p></div>
```

Un modèle d'annotation pour cette page doit définir que les parties de la page entre et décrivent un chercheur. Récursivement, des modèles sont définis pour l'identification de sous-structures identifiant les personnes, projets et thèmes.

Cette modélisation peut être effectuée grâce à la structure arborescente des pages web définie par le Document Object Model (DOM). Le DOM définit chaque balise HTML comme un noeud de l'arbre et les relations entre balises comme des arêtes. Dans la suite, nous distinguons deux

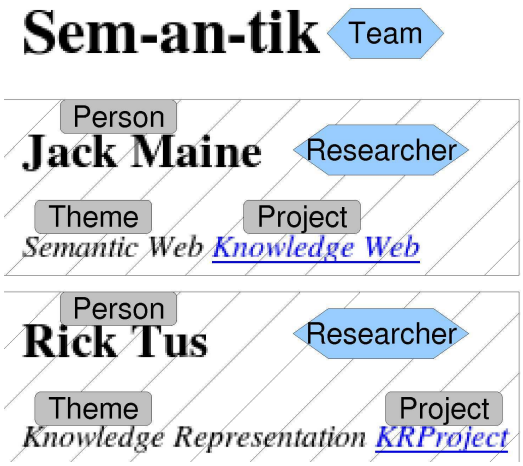

Figure 1: Page web annotée d'une équipe de recherche

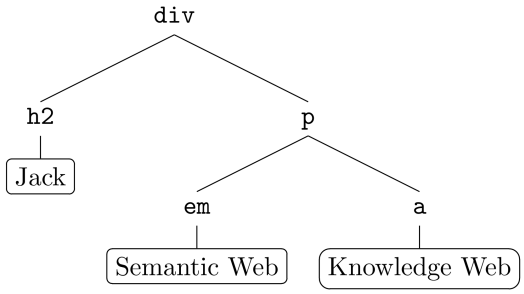

Figure 2: Représentation DOM du code HTML de la page web figure 1

types de noeuds. : les chaînes de caractères sont représentées par des *noeuds textes*, les balises HTML par des *noeuds éléments*. Ces noeuds éléments sont typés par le nom de la balise.

La figure 2 présente l'arbre DOM d'une partie de l'exemple. Les balises `div`, `h2`, `p` et `em` sont transformées en noeuds éléments. Les chaînes *Jack*, *Semantic Web* et *Knowledge Web* en noeuds textes. L'arbre est ordonné, chaque arête représentant une relation parent/fils entre deux noeuds. L'arbre DOM est alors transformé en arbre binaire en introduisant les relations suivantes :

- `firstChild`, écrit `fC`, identifie le premier fils d'un noeud.
- `nextSibling`, écrit `nS`, identifie le prochain frère d'un noeud.
- `noChild`, écrit `noC`, identifie un noeud sans fils.
- `noSibling`, écrit `noS`, identifie un noeud sans frère.

Ainsi, le sous-arbre de l'exemple figure 2 se décrit comme suit :
$\texttt{fC}(\texttt{div}, \texttt{h2})$, $\texttt{nS}(\texttt{h2}, \texttt{p})$, $\texttt{fC}(\texttt{p}, \texttt{em})$, $\texttt{nS}(\texttt{em}, \texttt{a})$, $\texttt{noS}(\texttt{a})$, $\texttt{noC}(\texttt{a})$.

Un tel arbre binaire peut être encodé directement sous la forme de concepts et de rôles en logique de descriptions. La prochaine section détaille l'instanciation de pages web dans ce formalisme.

3. Représentation formelle de la structure des pages

La structure des pages est formalisée par les concepts et rôles suivants :

- chaque balise HTML est représentée par un concept primitif nommé par le nom de la balise : `div`, `h2`, `p`, `em` and `a` sont de tels concepts,
- la relation parent/fils entre deux balises `A` et `B` est formalisée par le rôle `fC` tel que $\texttt{A} \sqsubseteq \exists \texttt{fC}.\texttt{B}$,

- la relation " le prochain frère de A est B " est formalisée par le rôle nS tel que: $\mathtt{A} \sqsubseteq \exists \mathtt{nS}.\mathtt{B}$
- les relations noC et noS représentent respectivement l'absence de fils et de frère.

Cette formalisation permet de créer des concepts définis de structure et de les organiser dans un ordre partiel. Prenons les concepts suivants comme exemple :

1. $\mathtt{S1} \equiv \mathtt{body} \sqcap \exists \mathtt{fC}.\mathtt{h1}$
2. $\mathtt{S2} \equiv \mathtt{body} \sqcap \exists \mathtt{fC}.(\mathtt{h1} \sqcap \exists \mathtt{noC} \sqcap \exists \mathtt{noS})$
3. $\mathtt{S3} \equiv \mathtt{body} \sqcap \exists \mathtt{fC}.(\mathtt{h1} \sqcap \mathtt{nS}.\mathtt{div})$

S1 est un concept défini représentant toute page web dont le corps de texte débute par un titre de niveau 1. S2 représente les pages web ne contenant qu'un titre. S3 représente les pages dont le titre est suivi d'une structure en div. S1 est plus général que S2 et S3. S2 et S2 sont disjoints : une page web ne peut pas être à la fois instance de S2 et de S3. Par contre, toute page instance de S2 ou S3 est aussi instance de S1. En termes de classification, l'ordre partiel est le suivant : $\mathtt{S2} \sqsubseteq \mathtt{S1}$, $\mathtt{S3} \sqsubseteq \mathtt{S1}$ and $\mathtt{S2} \sqcap \mathtt{S3} \sqsubseteq \bot$.

Afin de classifier une nouvelle page à annoter par rapport aux concepts définis, la page est formalisée par des individus de la LD. Dans un premier temps, la page est codée sous la forme d'un arbre binaire à partir de sa représentation DOM. Puis, pour chaque noeud élément x, un individu $\mathtt{s_x}$ est généré. Si $\mathtt{s_x}$ a un frère (resp. un fils), l'instance de rôle $\mathtt{nS}(\mathtt{s_x}, \mathtt{s_y})$ (resp. $\mathtt{fC}(\mathtt{s_x}, \mathtt{s_y})$) est générée. S'il n'a pas de frère (resp. de fils) une instance de noS (resp.noC)) lui est associée. La formalisation de la page d'exemple figure 1 est la suivante :

$\mathtt{body}(\mathtt{s_1})$	$\mathtt{h1}(\mathtt{s_2})$	$\mathtt{div}(\mathtt{s_3})$	$\mathtt{h2}(\mathtt{s_4})$	$\mathtt{p}(\mathtt{s_5})$	$\mathtt{em}(\mathtt{s_6})$
$\mathtt{a}(\mathtt{s_7})$	$\mathtt{div}(\mathtt{s_8})$	$\mathtt{h2}(\mathtt{s_9})$	$\mathtt{p}(\mathtt{s_{10}})$	$\mathtt{em}(\mathtt{s_{11}}$	$\mathtt{a}(\mathtt{s_{12}})$
$\mathtt{fC}(\mathtt{s_1}, \mathtt{s_2})$	$\mathtt{noS}(\mathtt{s_1})$	$\mathtt{nS}(\mathtt{s_2}, \mathtt{s_3})$	$\mathtt{fC}(\mathtt{s_3}, \mathtt{s_4})$	$\mathtt{nS}(\mathtt{s_3}, \mathtt{s_8})$	$\mathtt{nS}(\mathtt{s_4}, \mathtt{s_5})$
$\mathtt{noC}(\mathtt{s_4})$	$\mathtt{fC}(\mathtt{s_5}, \mathtt{s_6})$	$\mathtt{noS}(\mathtt{s_5})$	$\mathtt{nS}(\mathtt{s_6}, \mathtt{s_7})$	$\mathtt{noC}(\mathtt{s_6})$	$\mathtt{noC}(\mathtt{s_7})$
$\mathtt{noS}(\mathtt{s_7})$	$\mathtt{fC}(\mathtt{s_8}, \mathtt{s_9})$	$\mathtt{noS}(\mathtt{s_8})$	$\mathtt{nS}(\mathtt{s_9}, \mathtt{s_{10}})$	$\mathtt{noC}(\mathtt{s_9})$	$\mathtt{fC}(\mathtt{s_{10}}, \mathtt{s_{11}})$
$\mathtt{noS}(\mathtt{s_{10}})$	$\mathtt{nS}(\mathtt{s_{11}}, \mathtt{s_{12}})$	$\mathtt{noC}(\mathtt{s_{11}})$	$\mathtt{noS}(\mathtt{s_{12}})$	$\mathtt{noC}(\mathtt{s_{12}})$	

La classification des instances est réalisée par un raisonneur sur la LD. L'individu est instance du concept de structure le plus spécifique qu'il instancie dans la hiérarchie des concepts de structure. Par exemple, $\mathtt{s_1}$ est instance de S1 et S3. Le concept de structure retenu pour $\mathtt{s_1}$ sera donc S3, car $\mathtt{S3} \sqsubseteq \mathtt{S1}$. Ainsi, le concept de structure le plus spécifique que l'on peut définir pour les individus issus de la balise div ($\mathtt{s_3}$ et $\mathtt{s_8}$) de l'exemple figure 2 est :

$$\begin{aligned}\mathtt{S5} \equiv \mathtt{div} \sqcap \exists \mathtt{fC}.(&\mathtt{h2} \sqcap \\ &\exists \mathtt{nS}.(\ \mathtt{p} \sqcap \\ &\quad \exists \mathtt{fC}.(\ \mathtt{em} \sqcap \\ &\quad\quad \exists \mathtt{nS}.(\ \mathtt{a} \sqcap \exists \mathtt{noC} \sqcap \exists \mathtt{noS}))))\end{aligned}$$

S5 instancie à la fois $\mathtt{s_3}$ et $\mathtt{s_8}$ car la définition de S5 ne contraint pas le fait que div ait un fils ou non. Cette possibilité de ne pas spécifier certaines contraintes rend cette formalisation adaptée à la formalisation des structures régulières. Ainsi, les LD permettent de définir des concepts de structure et de classifier les pages par instanciation. La section suivante introduit la formalisation du domaine de la recherche avec lequel les pages doivent être annotées.

4. Une ontologie de la recherche

Afin d'annoter les pages en fonction de connaissances du domaine, nous définissons une ontologie au sens de Gruber (1995) : une spécification explicite et partagée d'une conceptualisation. L'ontologie doit guider la génération d'individus de domaine à partir des chaînes de caractères identifiées dans une page web et permettre la génération d'instances de rôles entre les individus reliés. Cette ontologie, que nous appelons $\mathcal{O}$, est définie comme suit :

- Les concepts sont hiérarchisés par la relation de subsomption $\sqsubseteq$. Les sous-classes directes de $\top$ sont des concepts primitifs. Pour chaque concept primitif, un ensemble de termes permet d'identifier une instance dans une page web. Par exemple, "jack" appartient à l'ensemble des termes du concept `Personne`,
- les rôles définissent les relations binaires entre concepts. Soient `C`, `D` $\in \mathcal{O}$, l'existence d'une relation entre `C` et `D` est formalisée comme suit : `C` $\sqsubseteq \exists r.D$. Dans $\mathcal{O}$, r est unique pour deux concepts donnés,
- les concepts définis sont définis par un ensemble de rôles. Pour chaque concept défini `D`, les rôles forment l'ensemble des conditions nécessaires et suffisantes pour qu'un individu soit instance de `D`. Par exemple, un chercheur est un sous-concept de personne qui a au moins un thème et un projet. Formellement, `Chercheur` $\equiv$ `Personne` $\sqcap \exists r_1$.`Theme` $\sqcap$ $\exists r_2$.`Projet`.

Dans le cadre de notre application, les concepts primitifs de $\mathcal{O}$ sont issus de l'ontologie SWRC (Semantic Web for Research Communities), une ontologie standard utilisée dans diverses applications (Sure *et al.*, 2005). Elle modélise des entités de la recherche comme des personnes, des publications, des projets, des thèmes de recherche et leurs relations. Par souci de clarté, seul un sous-ensemble de SWRC est présenté dans les exemples. Il s'agit des concepts primitifs `Personne`, `Theme`, and `Projet` ainsi que des concepts définis `Chercheur` et `Equipe`. L'identification de ces concepts primitifs dans les pages web est présentée dans la prochaine section.

5. Traitement des concepts primitifs : apport des travaux actuels

Les systèmes actuels d'annotation traitent exclusivement les concepts primitifs. Pour instancier les concepts d'une ontologie de domaine, ils identifient des chaînes de caractères qui correspondent à ces concepts. Ils appliquent pour cela des techniques de Recherche et Extraction d'Information (Uren *et al.*, 2006). Ces applications peuvent être classifiées en quatre catégories :

Annotation interactive. Ces systèmes proposent une interface affichant l'ontologie et la page à annoter. Les utilisateurs l'utilisent pour annoter des instances de concepts. Par exemple, Amaya (Quint & Vatton, 1997) et Mangrove (McDowell *et al.*, 2003) sont des outils spécifiques aux pages web qui permettent d'annoter depuis un navigateur web.

Système par apprentissage. Dans un premier temps, le système apprend des règles depuis un corpus annoté. Ensuite, de nouvelles pages sont annotées par application des règles. De tels systèmes sont évalués dans des concours, tels que le MUC (Machine Understanding Conference) (Chinchor, 1997)

Annotation semi-automatique. Ces systèmes sont centrés sur l'utilisateur mais automatisent certaines tâches. Ainsi, S-CREAM (Handschuh *et al.*, 2002) intègre un module d'extraction d'information. Lixto (Gottlob *et al.*, 2004) et SHOE (Heflin *et al.*, 2003) assistent l'utilisateur pour qu'il crée des règles qui associent des chaînes de caractères à des concepts de l'ontologie.

Annotation non supervisée. La redondance de l'information sur le web permet de valider les annotations sans intervention humaine. Armadillo (Norton *et al.*, 2005) associe des techniques d'extraction d'information à une validation statistique des connaissances extraites. KnowItAll (Etzioni *et al.*, 2004) definit une mesur à partir de résultats fournis par différents moteurs de recherche. Enfin, OntoSyphon (McDowell & Cafarella, 2006) effectue des requêtes sur un moteur de recherche à partir d'une ontologie.

L'apport de notre approche porte sur l'annotation par des instances de concepts définis. Pour l'identification initiale des instances de concepts primitifs, nous reposons sur les travaux précédents. Dans notre système, la structure de la page permet d'identifier des chaînes de caractères candidates pour l'annotation. Pour chaque structure à laquelle un concept est potentiellement associé, la chaîne candidate est comparée à l'ensemble des termes associés à ce concept dans l'ontologie de domaine. En cas de non concordance, l'utilisateur valide lui-même l'annotation et le terme manquant est ajouté à la liste de termes le cas échéant. Le processus complet est détaillé dans la section suivante.

6. Définition de classes d'annotation pour l'annotation

Soit $\mathtt{S}$ le sommet de la hiérarchie des concepts de structure et $\mathtt{C}$ un concept de l'ontologie $\mathcal{O}$. La classe d'annotation $\mathtt{CA}$ la plus générale annotant tout élément d'une page par une instance du concept $\mathtt{C}$ est définie comme : $\mathtt{CA} \equiv \mathtt{S} \sqcap \mathtt{annotePar.C}$. Une classe d'annotation pour un concept défini de structure $\mathtt{S_x} \sqsubseteq \mathtt{S}$ est alors défini comme $\mathtt{CA_x} \equiv \mathtt{S_x} \sqcap \mathtt{annotePar.C}$. Cette classe est alors instanciée par tous les individus de structure instances de $\mathtt{S_x}$ qui sont annotés par $\mathtt{C}$. Par exemple, la figure 3 présente la classe d'annotation CA5 qui définit l'annotation de la structure S5, introduite en section 3 par le concept de Chercheur. Une visualisation DOM est présentée figure 4. La figure 3 présente la classe d'annotation CA5 qui définit l'annotation de la

$$
\begin{aligned}
\mathtt{CA5} \equiv\ & \mathtt{div} \sqcap \exists \mathtt{annotePar.Chercheur} \sqcap \\
& \exists \mathtt{fC.}(\mathtt{h2} \sqcap \exists \mathtt{annotePar.Personne} \sqcap \\
& \quad \exists \mathtt{nS.}(\mathtt{p} \sqcap \\
& \quad\quad \exists \mathtt{fC.}(\mathtt{em} \sqcap \exists \mathtt{annotePar.Theme} \\
& \quad\quad\quad \sqcap \exists \mathtt{nS.}(\mathtt{a} \sqcap \exists \mathtt{annotePar.Projet} \sqcap \\
& \quad\quad\quad\quad \exists \mathtt{noC} \sqcap \exists \mathtt{noS}))))
\end{aligned}
$$

Figure 3: Classe d'annotation CA5, sous-classe du concept de structure S5

structure S5, introduite en section 3 par le concept de Chercheur. Une visualisation DOM de cette annotation est présentée figure 4.

Annoter une page web signifie créer des individus qui instancient une classe d'annotation et sont reliés par des instances de rôles. Étant donnée une page à annoter, le processus consiste en deux grandes étapes : identifier la structure de la page web, puis associer une sémantique à cette structure, parmi les sémantiques possibles.

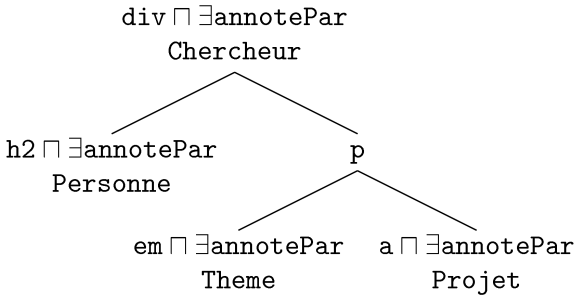

Figure 4: Visualisation DOM de CA5

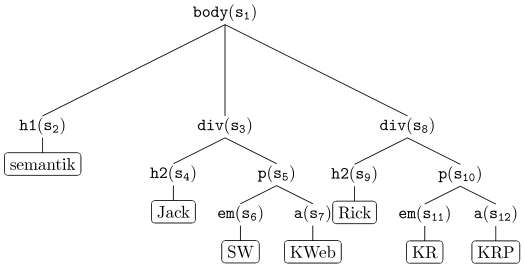

Figure 5: DOM (et individus de structure) de la page à annoter

À titre d'exemple, annotons la page web de la figure 1, dont le DOM est présenté figure 5. La première étape se déroule comme décrit en section 3 : tout d'abord, les individus de structure sont générés à partir du DOM de la page. Puis un raisonneur classifie ces individus comme instances de concepts de structure.

La seconde étape consiste à associer une sémantique aux individus $\mathtt{s_i}$ instances d'une structure $\mathtt{S_x}$. Les annotations potentielles de $\mathtt{s_i}$ sont définies par les classes d'annotations subsumées par $\mathtt{S_x}$, telles que $\mathtt{CA_x} \equiv \mathtt{S_x} \sqcap \exists \mathtt{annotePar.C}$.
Parmi les cas possibles, seuls certains sont satisfiables en fonction des concepts primitifs qui participent à la définition de C et qui sont identifiés ou non dans la page. Le procesus consiste ainsi à vérifier que les conditions nécessaires et suffisantes définies dans $\mathcal{O}$ sont satisfaites. Par exemple, s_3 peut être annoté par une instance de Chercheur uniquement si des instances de Personne, Theme et Projet annotent des individus de structure instances de concepts faisant partie de la définition de S5. Le processus fondé sur une résolution de contraintes est détaillé dans les sous-sections suivantes.

6.1. Identification des candidats à l'annotation

Cette étape identifie pour chaque instance de structure créée à partir de la page web, la liste des classes d'annotation définies dans l'ontologie $\mathcal{O}$. La sortie consiste en des couples candidats, qui sont des couples (individus de structure, concept de domaine) dans lesquels chaque individu de structure est instance d'une structure $\mathtt{S_x}$ qui subsume au moins une classe d'annotation. Le concept de domaine est le codomaine du rôle annotePar d'une CA subsumée. Dans l'exemple figure 3 les candidats sont :

1. $(\mathtt{s_3}, \mathtt{Chercheur})$ and $(\mathtt{s_8}, \mathtt{Chercheur})$,
2. $(\mathtt{s_4}, \mathtt{Personne})$ and $(\mathtt{s_9}, \mathtt{Personne})$,
3. $(\mathtt{s_6}, \mathtt{Theme})$, $(\mathtt{s_{11}}, \mathtt{Theme})$,
4. $(\mathtt{s_7}, \mathtt{Projet})$ and $(\mathtt{s_{12}}, \mathtt{Projet})$.

6.2. Calcul des contraintes structurelles, terminologiques et sémantiques

La validation de chaque couple candidat dépend de la résolution des contraintes suivantes :

Contraintes structurelles. Les concepts de structure sont définis par les rôles (fC et nS) qui sont des conditions nécessaires et suffisantes. Les *contraintes structurelles* reflètent ces conditions au niveau des individus. Par exemple, la classification de $\mathtt{s}_3$ comme instance de S5 dépend de la classification de $\mathtt{s}_4$, $\mathtt{s}_5$, $\mathtt{s}_6$ et $\mathtt{s}_7$. Les contraintes de $\mathtt{s}_3$ sont donc $\mathtt{s}_4$, $\mathtt{s}_5$, $\mathtt{s}_6$ et $\mathtt{s}_7$.

Contraintes terminologiques. Certains individus de structure sont généré à partir de noeuds DOM parents d'un noeud texte. Les *contraintes terminologiques* sont les mots contenus dans ces noeuds textes. Dans l'exemple, les contraintes terminologiques sont "Jack" pour $\mathtt{s}_4$, "SW" pour $\mathtt{s}_6$, "KWeb" pour $\mathtt{s}_7$, "Rick" pour $\mathtt{s}_9$, "KR" pour $\mathtt{s}_{11}$ et "KRP" pour "$\mathtt{s}_{12}$".

Contraintes sémantiques. Elles sont calculées pour chaque concept défini C faisant partie d'au moins un candidat. Il s'agit de l'ensemble des concepts de l'ontologie $\mathcal{O}$ appartenant à la définition de C. Par exemple, Chercheur est un concept défini de $\mathcal{O}$ appartenant au candidat ($\mathtt{s}_3$, Chercheur). Ses contraintes sémantiques sont Personne, Theme et Projet.

6.3. Génération des annotations

Pour chaque couple ($\mathtt{s}_x$, C), la résolution des contraintes dépend de C :

Si le concept de domaine C est primitif Les contraintes terminologiques de $\mathtt{s}_x$ sont comparées aux termes associés à C dans $\mathcal{O}$. Si un terme correspond, un individu de domaine instance de C est généré et $\mathtt{s}_x$ est annoté avec. Dans l'exemple,

- la contrainte terminologique pour $\mathtt{s}_4$ ("Jack") et pour $\mathtt{s}_9$ ("Rick") concorde avec un terme de Personne. Deux individus $\mathtt{ind}_4$ et $\mathtt{ind}_9$, instances de Personne, sont générés. Les annotations $\mathtt{annotePar}(\mathtt{s}_4, \mathtt{ind}_4)$ et $\mathtt{annotePar}(\mathtt{s}_9, \mathtt{ind}_9)$ sont générées,
- la contrainte terminologique pour $\mathtt{s}_6$ ("SW") et pour $\mathtt{s}_{11}$ ("KR") concorde avec un terme de Theme. Deux individus $\mathtt{ind}_6$ et $\mathtt{ind}_{11}$, instances de Theme sont créés. Les annotations $\mathtt{annotePar}(\mathtt{s}_6, \mathtt{ind}_6)$ et $\mathtt{annotePar}(\mathtt{s}_{11}, \mathtt{ind}_{11})$ sont générées,
- la contrainte terminologique pour $\mathtt{s}_7$ ("KWeb") et $\mathtt{s}_{12}$ ("KRP") concorde avec un terme de Projet. $\mathtt{ind}_7$ et $\mathtt{ind}_{12}$ sont créées comme instances de Projet. Les annotations $\mathtt{annotePar}(\mathtt{s}_7, \mathtt{ind}_7)$ et $\mathtt{annotePar}(\mathtt{s}_{12}, \mathtt{ind}_{12})$ sont générées.

Si C est un concept défini L'individu de structure $\mathtt{s}_i$ est annoté par une instance de C si une instance de chaque contrainte sémantique de C annote un individu appartenant aux contraintes structurelles de $\mathtt{s}_x$.

Dans l'exemple, $\mathtt{s}_3$ est annoté par une instance de Chercheur si
$\mathtt{s}_4$, $\mathtt{s}_5$, $\mathtt{s}_6$ ou $\mathtt{s}_7$ est annoté par une instance de Personne
$\mathtt{s}_4$, $\mathtt{s}_5$, $\mathtt{s}_6$ ou $\mathtt{s}_7$ est annoté par une instance de Theme
$\mathtt{s}_4$, $\mathtt{s}_5$, $\mathtt{s}_6$ ou $\mathtt{s}_7$ est annoté par une instance deProjet

Comme $\mathtt{s}_4$ est annoté par $\mathtt{ind}_4$, $\mathtt{s}_6$ est annoté par $\mathtt{ind}_6$ et $\mathtt{s}_7$ par $\mathtt{ind}_7$, toutes les contraintes sont résolues. En conséquence, les instances de rôles et les annotations suivantes sont générées :

1. Les rôles du concept de Chercheur sont instanciés comme suit : $\mathtt{r}_1(\mathtt{ind}_4, \mathtt{ind}_6)$, $\mathtt{r}_2(\mathtt{ind}_4, \mathtt{ind}_7)$. Ainsi, Chercheur est instancié par ind_4.

2. $\mathtt{s}_3$ est annoté par $\mathtt{ind}_4$. L'annotation est formalisée par $\mathtt{annotePar}(\mathtt{s}_3, \mathtt{ind}_4)$.

De la même manière, l'annotation de $\mathtt{s}_8$ en fonction de $\mathtt{CA}_5$ génère les rôles $\mathtt{r}_1(\mathtt{ind}_9, \mathtt{ind}_{11})$ et $\mathtt{r}_2(\mathtt{ind}_9, \mathtt{ind}_{12})$, ainsi que l'annotation $\mathtt{annotePar}(\mathtt{s}_8, \mathtt{ind}_9)$.

7. Évaluation et perspectives

Nous avons programmé un prototype pour évaluer l'approche sur des données réelles. Un script génère un arbre DOM depuis le code HTML de la page à annoter puis crée les individus de structure dans le langage OWL-DL. Les instanciations sont calculées par le raisonneur Pellet (Sirin & Parsia, 2004).

- Il est possible d'écrire une classe d'annotation pour toutes les pages du corpus. Ceci devient toutefois couteux si la structure n'est pas régulière. Pour les pages générées depuis des bases de données avec des langages dynamiques de type PHP, une classe d'annotation est décrite par 15 à 20 concepts
- La majeure partie du temps de calcul est utilisé par le raisonneur. Le calcul des contraintes est négligeable par rapport à ce temps

De plus, les logiques de descriptions fournissent l'explication de l'annotation des pages. En outre, la maintenance des annotations devient un problème de satisfaction : si la structure change, l'instanciation échoue.

Les perspectives consistent à automatiser la construction des classes d'annotation. L'objectif est de permettre à un expert du domaine de peupler son ontologie sans avoir à construire des classes à la main. Au final, le prototype devrait permettre la maintenance de l'ontologie de domaine, la recherched de pages web pertinentes, la génération automatique des classes d'annotation, le peuplement de l'ontologie, la maintenance des annotations et la générations de requêtes.

8. Conclusions

Les classes d'annotation traitent l'annotation comme un problème d'instanciation. Les éléments de pages web formalisés comme individus de structure sont annotés par des instances de concepts du domaine. Les relations entre ces éléments sont formalisées par des instances de rôles. Il s'agit, à notre connaissance, du premier système dans lequel l'annotation est formellement définie à la fois comme processus et comme résultat de ce processus. Bien qu'il reste du travail pour obtenir un framework complet, les premiers tests ont montré la faisabilité d'un processus d'annotation guidé par les connaissances.

Références

AMARDEILH F., LAUBLET P. & MINEL J.-L. (2005). Document annotation and ontology population from linguistic extractions. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, p. 161–168, New York, NY, USA: ACM Press.

BERNERS-LEE T. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco.

CARME J., GILLERON R., LEMAY A. & NIEHREN J. (2007). Interactive learning of node selecting tree transducers, machine learning. *Machine Learning*, **66**(1), 33–67.

CHINCHOR N. (1997). Overview of MUC-7. In *Proceedings of the Seventh Message Understanding Contest*. Fairfax, VA, USA.

ETZIONI O., CAFARELLA M., DOWNEY D., KOK S., POPESCU A. M., SHAKED T., SODERLAND S., WELD D. S. & YATES A. (2004). Web-scale information extraction in knowitall: (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, p. 100–110, New York, NY, USA: ACM Press.

GOTTLOB G., KOCH C., BAUMGARTNER R., HERZOG M. & FLESCA S. (2004). The lixto data extraction project: back and forth between theory and practice. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, p. 1–12, New York, NY, USA: ACM Press.

GRUBER T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, **43**(5-6), 907–928.

HANDSCHUH S., STAAB S. & CIRAVEGNA F. (2002). S-cream-semi-automatic creation of metadata. *Proc. of the European Conference on Knowledge Acquisition and Management*.

HEFLIN J., HENDLER J. A. & LUKE S. (2003). Shoe: A blueprint for the semantic web. In *Spinning the Semantic Web*, p. 29–63.

KUSHMERICK N. (1997). *Wrapper induction for information extraction*. PhD thesis. Chairperson-Daniel S. Weld.

MCDOWELL L. & CAFARELLA M. (2006). Ontology-driven information extraction with ontosyphon. In I. CRUZ, S. DECKER, D. ALLEMANG, C. PREIST, D. SCHWABE, P. MIKA, M. USCHOLD & L. AROYO, Eds., *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, p. 428–444: Springer.

MCDOWELL L., ETZIONI O., GRIBBLE S. D., HALEVY A. Y., LEVY H. M., PENTNEY W., VERMA D. & VLASSEVA S. (2003). Mangrove: Enticing ordinary people onto the semantic web via instant gratification. In *International Semantic Web Conference*, p. 754–770.

NORTON B., CHAPMAN S. & CIRAVEGNA F. (2005). Orchestration of semantic web services for large-scale document annotation. In *ESWC European Semantic Web Conference*, p. 649–663.

QUINT V. & VATTON I. (1997). An introduction to amaya. *World Wide Web J.*, **2**(2), 39–46.

SIRIN E. & PARSIA B. (2004). Pellet: An owl dl reasoner. In V. HAARSLEV & R. MÖLLER, Eds., *Description Logics*, volume 104 of *CEUR Workshop Proceedings*: CEUR-WS.org.

SURE Y., BLOEHDORN S., HAASE P., HARTMANN J. & OBERLE D. (2005). The swrc ontology - semantic web for research communities. In C. BENTO, A. CARDOSO & G. DIAS, Eds., *Proceedings of the 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence (EPIA 2005)*, volume 3803 of *LNCS*, p. 218–231, Covilha, Portugal: Springer.

UREN V., CIMIANO P., IRIA J., HANDSCHUH S., VARGAS-VERA M., MOTTA E. & CIRAVEGNA F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, (4), 14–28.