Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle

Florence Amardeilh

UNIVERSITE PARIS X – NANTERRE

ECOLE DOCTORALE CONNAISSANCE, LANGAGE ET MODELISATION
LABORATOIRE MODYCO (MODELES, DYNAMIQUES, CORPUS) – UMR CNRS 7114
CONVENTION CIFRE N° 422/2003

THESE DE DOCTORAT

Discipline : Informatique

Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle

Présentée par Florence Amardeilh

Sous la direction de Messieurs Jean-Luc Minel et Philippe Laublet

Mai 2007

Membres du jury :

Rapporteurs :	**Mme Nathalie Aussenac-Gilles**, Chargée de recherche (HDR), CNRS
	Mr Gilles Kassel, Professeur d'université, Université de Picardie
Examinateurs :	**Mr Benoît Habert**, Professeur d'université, Université Paris X-Nanterre
	Mme Maria-Teresa Pazienza, Professeur d'université, Università di Roma Tor Vergata
Co-Directeur :	**Mr Jean-Luc Minel**, Ingénieur de recherche (HDR), CNRS
Co-Directeur :	**Mr Philippe Laublet**, Maître de conférences, Université Paris IV-Sorbonne
Invité :	**Mr Jean Delahousse**, PDG de Mondeca

modyco Modèles, Dynamiques, Corpus UMR 7114 Sciences du langage. CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE UPX UNIVERSITE PARIS X NANTERRE

MONDECA

A mon grand-père Elie Amardeilh,

et à mon futur petit bout…

Remerciements

J'aimerais remercier toutes les personnes qui m'ont aidée durant mon processus de recherche et d'écriture de cette thèse :

Jean-Luc Minel, mon directeur de thèse, pour sa disponibilité, ses critiques et ses conseils toujours opportuns ainsi que pour sa grande gentillesse.

Philippe Laublet, mon co-directeur de thèse, pour ses idées constructives, ses remarques toujours pertinentes, sa relecture rigoureuse de ce mémoire et nos discussions animées autour des concepts manipulés par cette thèse.

Nathalie Aussenac-Gilles, pour qui j'ai énormément d'admiration, et qui me fait l'honneur d'être rapporteur de cette thèse après m'avoir déjà épaulée lors de mon DEA.

Gilles Kassel qui a également eu l'amabilité d'accepter d'être rapporteur de ma thèse.

Benoit Habert et Maria Teresa Pazienza pour l'intérêt qu'ils ont porté à cette thèse en acceptant de faire partie de mon jury.

Jean Delahousse, CEO de Mondeca, qui m'a offert l'opportunité de rejoindre son équipe, me procurant ainsi un cadre de travail et de réflexion propice à la réalisation de cette thèse, grâce notamment aux divers projets et conférences auxquels j'ai pu participer activement.

Toute l'equipe de Mondeca, et notamment Benoît, Thomas, Anh, Louis, Olivier, Laurence, Bernard et Gilles, pour les nombreuses heures passées ensemble sur les projets ou au Merle Moqueur, pour leur aide précieuse et pour leur bonne humeur revigorante.

Les linguistes de Témis, et en particulier Vincent, Amandine, Sylvie, Jean-Pierre, Sophie, Christian, Françoise et Stéphanie avec qui j'ai eu plaisir à travailler et à échanger à propos des analyses linguistiques réalisées dans les projets que nous avons menés ensemble.

L'ensemble de mes amis (désolée encore pour ces derniers longs mois d'absence) ainsi que toutes les personnes qui ont croisé un jour mon chemin et qui ont su me supporter durant ces années de thèse, parfois éreintantes, mais toujours enrichissantes.

Ma famille, et plus particulièrement mes parents, qui ont su m'encourager jour après jour et qui ont toujours cru en moi. Sans leur soutien constant, tant affectif que matériel, je n'aurai jamais pu accomplir mes études et envisager cette thèse.

Et enfin, mon « coach perso », mon mari, mon épaule, mon cerveau aussi parfois, Stéphane, pour son amour, sa tendresse, sa présence et sa grande patience, sans qui je ne serai peut-être pas arrivée au bout de cette aventure…

Résumé

Cette thèse aborde les problématiques liées à l'annotation sémantique et au peuplement d'ontologies dans le cadre défini par le Web Sémantique (WS). La vision du WS a pour objectif de structurer les informations disponibles sur le Web. Pour cela, les ressources, textuelles ou multimédias, doivent être sémantiquement étiquetées par des métadonnées afin que les agents logiciels puissent les exploiter. Dans le processus d'annotation sémantique, les ontologies jouent un rôle primordial puisqu'elles modélisent les concepts, leurs attributs et les relations utilisées pour annoter le contenu des documents. Mais il est aussi important que la base de connaissance, associée à cette ontologie, contienne les instances à utiliser pour l'annotation sémantique. C'est pourquoi la tâche de peuplement d'ontologie a pour but d'enrichir (semi-)automatiquement la base de connaissance avec de nouvelles instances de concepts, d'attributs et de relations.

L'idée proposée ici est de combiner les outils d'extraction d'information (EI) avec les outils de représentation des connaissances du WS pour la réalisation de ces deux tâches. Malgré tout, il existe actuellement un fossé entre les formats de représentation des outils linguistiques et ceux des outils WS pour la représentation des connaissances. Cette thèse propose de combler ce fossé en concevant un médiateur capable de transformer les étiquettes générées par les outils d'EI en une représentation plus formelle, que ce soit sous la forme des annotations ou des instances d'une ontologie. Autrement dit, nous tentons de répondre à la problématique suivante : comment pouvons-nous passer d'une certaine représentation du texte à une représentation sémantique de la connaissance ? L'enjeu consiste aussi bien à proposer une réflexion méthodologique sur l'interopérabilité des différentes technologies qu'une conception de solutions opérationnelles dans le monde des entreprises, et à plus large échelle du Web.

Dans le cadre de cette thèse, nous avons donc conçu une démarche que nous avons nommée OntoPop, pour « Ontology Population ». Cette démarche propose une passerelle sous la forme de règles, dites « d'Acquisition de Connaissance ». Le langage OPAL (Ontology Population and Annotation Language) définit la grammaire pour l'implémentation de ces règles. Enfin, nous soumettons des propositions pour l'opérationnalisation de la démarche OntoPop à travers une méthodologie en cinq étapes et une plateforme logicielle basée sur l'outil de représentation des connaissances ITM de la société Mondeca.

Mots-clefs : Annotation Sémantique, Peuplement d'Ontologie, Web Sémantique, Acquisition de Connaissance, Extraction d'Information.

Abstract

This thesis deals with the issues related to semantic annotation and ontology population within the framework defined by the Semantic Web (SW). The vision of the Semantic Web aims to structure information available on the Web. To achieve that goal, the resources, textual or multimedias, must be semantically tagged by metadata so that the software agents can exploit them. In the process of semantic annotation, ontologies play a major part since they model the concepts, their attributes and the relations used to annotate the contents of the documents. But it is also important that the knowledge base, associated with this ontology, contains the instances to be used for semantic annotation. This is why the purpose of the ontology population task aims to enrich (semi-)automatically the knowledge base with new instances of concepts, attributes and relations as defined by the ontology model.

The idea suggested in this thesis is to combine the information extraction (IE) tools with the knowledge representation tools of the WS for the achievement of these two tasks. Despite all integration efforts, there is currently a gap between the representation formats of the linguistic tools and those of the knowledge representation tools in the field of the Semantic Web. This thesis proposes to fill this gap by designing a mediator able to transform the tags generated by the IE tools into a more formal representation. In other words, we try to answer the following issue: how can we map a certain textual representation into a semantic knowledge representation? The stake consists in proposing a methodological reflexion about the interoperability of various technologies as well as a design of operational solutions in the world of the companies, and on broader scale of the Web.

Within this thesis, we thus conceived a framework named OntoPop for "Ontology Population". This framework proposes a bridge in the form of rules, known as "Knowledge Acquisition Rules". The OPAL language (Ontology Population and Annotation Language) defines a grammar for the implementation of these rules. Lastly, we submit proposals for the implementation of the OntoPop through a methodology in five stages and a software platform based on the knowledge repository ITM designed by Mondeca.

Mots-clefs : Semantic Annotation, Ontology Population, Semantic Web, Knowledge Acquisition, Information Extraction.

Table des Matières

Liste des Figures

Liste des Tableaux

Introduction

I. Du besoin particulier de Mondeca…

Cette thèse a été réalisée au sein de la société Mondeca[1], dans le cadre d'un contrat CIFRE. Mondeca est un éditeur de logiciels qui s'adresse particulièrement aux entreprises gérant de vastes ensembles de ressources documentaires professionnelles dans le cadre de leurs activités : documentation réglementaire, financière, médicale, juridique, économique, technique, presse, etc. Ces ressources fournissent aux employés, aux équipes de R&D, aux partenaires comme aux clients des informations pertinentes organisées autour de taxonomies, de thésaurus, d'ontologies et bien évidemment de bases de connaissance orientées métier. Mondeca a conçu un logiciel, l'Intelligent Topic Manager™ (ITM), pour répondre aux trois catégories de besoins exprimés par ces entreprises : 1) la fédération et l'organisation de contenus hétérogènes ; 2) la gestion des référentiels métiers ; 3) la gestion des bases de connaissances dédiées. ITM permet aux entreprises d'organiser leurs contenus et connaissances autour d'une vision métier de leurs activités et de leur offrir des solutions efficaces de recherche, de navigation, de raisonnement et de réutilisation de ces contenus et connaissances.

ITM est construit sur les normes et standards développés notamment dans le cadre du Web Sémantique : il permet de gérer des ontologies modélisées en OWL pour y être importées, des bases de connaissance et autres référentiels métiers formant un réseau sémantique écrit avec le langage des Topic Maps, et des publications ou exports générés selon les besoins en RDF, OWL ou Topic Maps. Nous verrons plus en détail chacun de ces langages dans la section 1.3 de ce mémoire. Le fait qu'ITM repose sur ces langages permet une meilleure interopérabilité des référentiels. Ceux-ci deviennent facilement intégrables via les APIs et Web Services aux applications externes de gestion de contenus et de moteurs de recherche.

Comme représenté dans la Figure 1, ITM propose l'ensemble des fonctionnalités suivantes :

- La gestion d'ontologies, de thésaurus, de taxonomies et de bases de connaissances
- La navigation dans une représentation métier sous forme d'un vaste réseau sémantique
- La recherche multi-axes dans les bases de connaissances et les contenus documentaires
- La réutilisation des contenus : composition, publication et diffusion des contenus
- Le travail collaboratif de capitalisation des connaissances
- Le raisonnement par l'intégration d'un moteur d'inférence (en cours de réalisation)

[1] http://www.mondeca.com

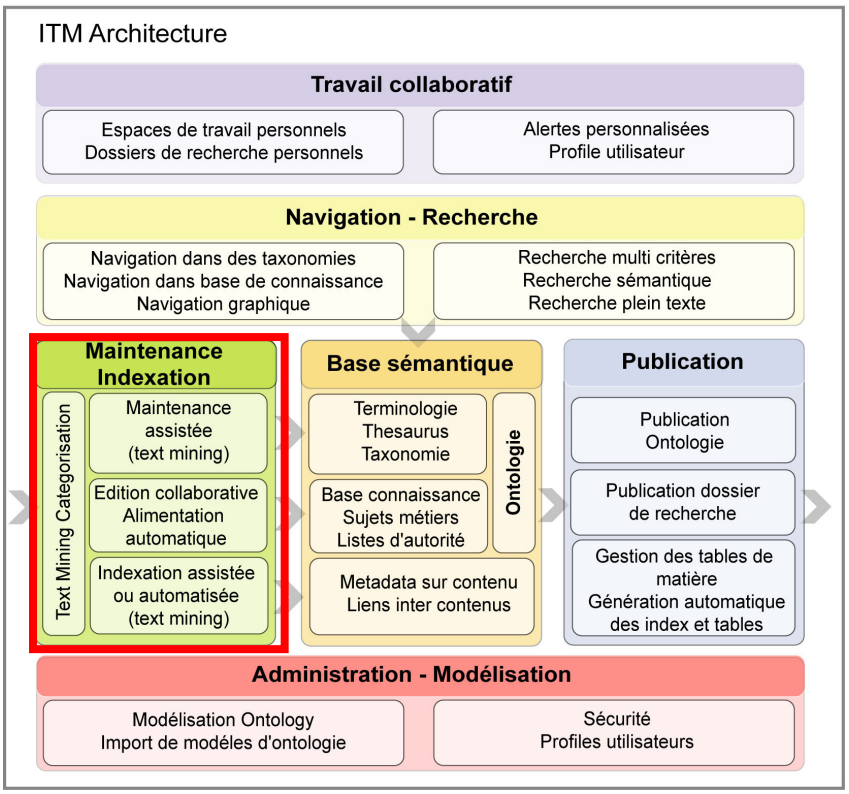


Figure 1. Fonctionnalités proposées par l'outil Intelligent Topic Manager™

Néanmoins, à mon arrivée chez Mondeca, une fonctionnalité pourtant essentielle à ITM manquait à cette architecture : l'annotation sémantique et l'enrichissement automatique des contenus, des bases de connaissances et autres référentiels métiers (cf. la case « Maintenance Indexation » de la Figure 1). En effet, ITM fournit un ensemble d'interfaces à l'utilisateur lui permettant de créer, d'éditer et de supprimer manuellement du contenu dans les différents référentiels. Pour autant, ce travail manuel n'est pas viable dans le cas d'applications devant traiter quotidiennement des centaines, voire des milliers, de ressources documentaires. Il était donc devenu crucial pour Mondeca d'intégrer une solution de Traitement Automatique du Langage Naturel (TALN) afin d'automatiser ce processus d'acquisition de la connaissance et de permettre également l'utilisation de la connaissance acquise pour l'annotation sémantique des ressources documentaires analysées. C'est dans cette optique que Mondeca a signé un partenariat OEM avec la société Temis, éditeur de l'outil d'extraction d'information IDE™, décrit au chapitre 2.

Une partie de mon travail de thèse a donc consisté à concevoir une passerelle entre IDE de Temis et ITM de Mondeca. Mais plus qu'une intégration spécifique, Mondeca souhaitait une solution générique, indépendante du moteur d'extraction d'information utilisé. Autrement dit, la solution doit être capable d'intégrer n'importe quel outil d'analyse linguistique. L'objet de notre recherche porte donc sur **le couplage d'outils d'extraction d'information et de représentation des connaissances à des fins de peuplement d'ontologie et d'annotation sémantique**. C'est dans ce cadre de travail que j'ai conçu la démarche OntoPop présentée tout au long de ce mémoire de thèse.

II. …vers une problématique plus générale dans le cadre du Web Sémantique

La vision du Web Sémantique initiée en 1998 par Sir Tim Berners-Lee [BER 98] a pour objectif de structurer les informations disponibles sur le Web. Pour cela, les ressources, textuelles ou multimédias, doivent être sémantiquement étiquetées par des métadonnées afin que les agents logiciels puissent les exploiter. La représentation explicite des contenus des ressources documentaires du Web est rendue possible grâce notamment aux ontologies. Ces ontologies représentent une technologie clef pour la mise en œuvre de ce Web Sémantique (cf. section 1.2.1). Elles ont été développées en Intelligence Artificielle pour faciliter le partage de la connaissance et leur réutilisation. Depuis les années 90, les ontologies sont devenues un sujet au cœur des recherches de différentes communautés, incluant l'ingénierie de la connaissance, le traitement automatique du langage naturel, la recherche d'information, les systèmes collaboratifs, etc. La raison de cette popularité est en partie due au fait que les ontologies proposent une compréhension commune et partagée d'un domaine, tant au niveau des utilisateurs humains qu'au niveau des applications logicielles [FEN 03]. Parmi les autres propositions du Web Sémantique figurent par exemple la traçabilité de l'information, à savoir l'identification de sa provenance et de ses émetteurs afin de lui accorder un niveau de confiance adéquat, l'interconnexion des informations entre elles, la découverte de nouvelles connaissances à partir de raisonnements logiques, etc.

La mise en place d'un Web Sémantique permet trois améliorations d'envergure du Web actuel et qui se répercutent au niveau des systèmes d'information des entreprises [URE 06]. La première est l'amélioration des moteurs de recherche d'information par la capacité à effectuer des requêtes qui exploitent la structure de l'ontologie et qui infèrent de la connaissance à partir des informations existantes dans cette ontologie. La deuxième consiste à donner les moyens d'accès à l'information permettant aux utilisateurs d'exploiter la connaissance représentée dans leur application. De tels moyens incluent des fonctionnalités pour trouver, partager, résumer, visualiser, naviguer et organiser la connaissance [FEN 03]. La troisième est l'interopérabilité entre les différents systèmes d'information, particulièrement importante pour les entreprises souhaitant communiquer avec leur propre réseau ou avec ceux de ses prestataires par exemple. Dans ces cas-là, l'annotation basée sur l'utilisation d'une ontologie commune peut fournir un cadre de travail commun pour l'intégration d'informations provenant de sources hétérogènes. Pour ce faire, le Web Sémantique fournit un ensemble de langages et de technologies pour la modélisation des ontologies et l'annotation sémantique des contenus documentaires en fonction de ces ontologies [URE 06].

La thèse aborde donc les problématiques liées à l'annotation sémantique et au peuplement d'ontologies dans le cadre défini par le Web Sémantique. L'annotation sémantique consiste à ajouter (semi-)automatiquement des métadonnées structurées aux ressources documentaires du Web mais aussi des intranets des entreprises (cf. section 1.1). Les annotations décrivent aussi bien le document dans son ensemble, comme son titre, son auteur, etc., que son contenu par des descripteurs

provenant de vocabulaires contrôlés comme les thésaurus ou par des instances d'une base de connaissance. Ces annotations sont alors exploitables par les utilisateurs finaux d'une application donnée pour rechercher, partager, accéder, publier des documents, des métadonnées ou même de la connaissance [LAU 02].

Dans le processus d'annotation sémantique, les ontologies jouent un rôle primordial puisqu'elles modélisent les concepts, leurs attributs et les relations utilisées pour annoter le contenu des documents. L'ontologie contraint l'application sur les vocabulaires et les instances autorisés comme métadonnées. Mais s'il est essentiel pour une application Web Sémantique de reposer sur une ontologie pour la réalisation de cette tâche d'annotation sémantique, il est aussi important que la base de connaissance, associée à cette ontologie, contienne les instances à utiliser pour l'annotation sémantique. C'est pourquoi la tâche de peuplement d'ontologie a pour but d'enrichir (semi-)automatiquement la base de connaissance avec de nouvelles instances de concepts, d'attributs et de relations comme défini par la modélisation de l'ontologie [HAN 05].

En fait, ces deux tâches peuvent être perçues comme très proches. Premièrement elles reposent toutes deux sur la modélisation de ressources terminologiques et ontologiques [BOU 04] (comme les taxonomies, thésaurus, ontologies) pour normaliser la sémantique des annotations documentaires comme celle des concepts du domaine concerné. Deuxièmement, elles utilisent les méthodes et outils du Traitement Automatique du Langage Naturel (TALN), comme l'Extraction d'Information (EI), pour extraire une information structurée des ressources documentaires ou encore la Catégorisation pour classifier un document dans des catégories prédéfinies ou calculées. Troisièmement, elles s'appuient de plus en plus sur les standards et langages du Web Sémantique comme RDF pour l'annotation et OWL pour le peuplement.

L'idée proposée dans cette thèse, de combiner des outils d'extraction d'information avec des outils de représentation des connaissances pour la réalisation de ces deux tâches, n'est certes pas nouvelle, même dans le cadre du Web Sémantique. Dans [BEN 02], Benjamins et al. présentent les six challenges qui vont conditionner le succès du Web Sémantique. Parmi ces challenges figurent la mise à disposition de contenus annotés et le développement et l'évolution des ontologies. La résolution de ces challenges passe notamment par l'utilisation de technologies issues de l'informatique linguistique, et plus particulièrement du domaine de l'Extraction d'Information (cf. Chapitre 2), car le principal mode de transfert de la connaissance se fait par l'utilisation du langage naturel dans les ressources documentaires [BUI 03].

Plusieurs autres articles [STE 03] [KAT 02] [BON 03] et un atelier de travail consacré à cette question[2] ont récemment présenté les avantages à utiliser les méthodes et outils du domaine du TALN pour l'étiquetage des ressources documentaires textuelles du Web afin notamment d'assister les annotateurs humains dans cette tâche fastidieuse et coûteuse. Les solutions d'Extraction d'Information parcourent une ressource textuelle, créent des étiquettes linguistiques pour marquer le

[2] Workshop on Human Language Technology for the Semantic Web: http://gate.ac.uk/conferences/iswc2003

contenu pertinent par rapport au domaine concerné. Ces étiquettes sont alors utilisées pour différents objectifs :

- annoter le contenu avec des métadonnées [KAH 02] [HAN 02] ;
- acquérir de la nouvelle connaissance, i.e. pour semi-automatiquement construire et maintenir les terminologies du domaine [VAR 02] ;
- semi-automatiquement enrichir les bases de connaissance avec les entités et les relations sémantiques extraites [BOU 04] [POP 04] [ALA 03] [CEL 04].

En fait, il existe une véritable synergie entre les domaines du Web Sémantique et de l'Informatique Linguistique qu'il est important de dégager et d'exploiter dans les futures applications de gestion de la connaissance. En effet, les technologies et standards mis en place dans le cadre du Web Sémantique peuvent aussi profiter aux outils d'extraction d'information. Ces derniers produisent des corpus annotés ou des lexiques par exemple. Or il est intéressant que ces derniers soient normalisés grâce à un langage standard permettant ainsi l'interopérabilité entre les différents outils d'analyse linguistique. Par ailleurs, ces outils ont parfois besoin d'intégrer des ressources linguistiques ou sur le domaine traité pour améliorer les différents traitements du langage naturel. Or ceci peut être réalisé par le fait de disposer d'ontologies et de ressources terminologiques, comme les thésaurus, qui soient formellement explicites. Ainsi, apparaît un cercle vertueux dans lequel les outils linguistiques permettent de peupler les ontologies du Web Sémantique qui, à leur tour, enrichissent les ressources nécessaires aux différentes analyses linguistiques (cf. Chapitre 4).

Malgré tout, il existe actuellement un fossé entre les formats de représentation des outils linguistiques et ceux des outils pour la représentation des connaissances dans le domaine du Web Sémantique. Cette thèse se propose de combler ce fossé en concevant un médiateur capable de transformer les étiquettes générées par les outils d'EI en une représentation plus formelle, que ce soit sous la forme des annotations ou des instances d'une ontologie (cf. Chapitre 3). Autrement dit, nous tenterons dans ce mémoire de répondre à la problématique suivante : comment pouvons-nous passer d'une certaine représentation du texte à une représentation sémantique de la connaissance ? L'enjeu consiste aussi bien à proposer une réflexion méthodologique sur l'interopérabilité des différentes technologies qu'une conception de solutions opérationnelles dans le monde des entreprises, et à plus large échelle du Web.

III. Déroulement de la thèse et guide de lecture

Tout au long de la thèse, nous avons eu à travailler sur des applications concrètes pour divers clients de Mondeca. Il nous a fallu prendre en compte les différents besoins suscités par ces applications : simplement peupler l'ontologie du domaine, peupler mais aussi annoter les ressources documentaires, automatiser tout ou partie du traitement, proposer une interface de validation générique, sauvegarder en interne ou en externe les annotations comme les nouvelles instances, etc.

La prise en charge de ces besoins nous a conduit à concevoir une application modulaire, qui puisse être flexible vis-à-vis des divers besoins exprimés et facilement adaptable à de nouveaux besoins.

Par ailleurs, si nous avons commencé à travailler avec l'outil d'extraction d'information IDE de la société Temis, nous avons progressivement élargi notre champ de vision en testant notre solution sur d'autres outils d'ingénierie linguistique comme l'outil GATE de l'Université de Sheffield.

Nous avons également suivi au fur et à mesure l'avancée des champs de recherche liés aux méthodes et outils d'annotation sémantique ou de peuplement d'ontologie, à la standardisation des langages et technologies du Web Sémantique, aux propositions d'architectures pour des applications liées à la gestion de la connaissance, entre autres.

Ces différents apports, tant au niveau du travail de recherche que de l'expérimentation, nous ont permis de concevoir la démarche proposée dans le cadre de cette thèse que nous avons nommée OntoPop, pour « Ontology Population ». Le plan de ce mémoire s'organise autour d'une progression de la réflexion, partant de la problématique telle qu'elle est traitée dans l'état de l'art (cf. partie 1) vers les solutions opérationnelles qui ont été conçues (cf. partie 3), en passant par la description de la démarche d'un point de vue plus global (cf. partie 2). Chaque partie est illustrée par des exemples concrets issus des projets auxquels nous avons participé.

La première partie présente une vue d'ensemble des différents champs de recherche concernés par notre problématique guidée par la perspective de l'annotation sémantique, à savoir qu'est-ce qu'une annotation sémantique dans le cadre du web sémantique, quels sont les langages et ressources disponibles pour l'annotation de ressources documentaires, quels sont les outils existants et leurs limites actuelles (cf. Chapitre 1). Cette première étude nous permet d'exposer les notions essentielles pour la compréhension du domaine de recherche et de mettre l'accent sur les points clefs liées à notre problématique. Puis, nous abordons la question de l'extraction d'information avec pour objectif d'expliciter le format de représentation des textes généré par les moteurs d'extraction, à savoir les arbres conceptuels (cf. Chapitre 2). En effet, ces arbres conceptuels constituent la matière première des tâches d'annotation sémantique et de peuplement d'ontologie et il est particulièrement important d'en connaître les caractéristiques.

La deuxième partie débute par la présentation des Règles d'Acquisition de Connaissance qui constituent le cœur de la démarche proposée dans OntoPop (cf. Chapitre 3). Ce sont elles qui vont permettre le passage des arbres conceptuels d'une extraction à une représentation sémantique et formelle de la connaissance sous forme d'annotations ou d'instances. Nous verrons que cette transformation dépend fortement de la notion de contexte dans les arbres. Le langage OPAL (Ontology Population and Annotation Language) d'écriture des règles est également présenté dans ce chapitre ainsi que l'outil logiciel pour la création et la compilation des règles. Le chapitre suivant concerne l'utilisation de ces règles dans un cycle complet d'extraction d'information, d'enrichissement des ressources terminologiques et ontologiques, d'annotation de ressources documentaires et de

mise à jour des lexiques utilisés par l'outil d'extraction linguistique (cf. Chapitre 4). L'accent est notamment porté sur la résolution des problèmes imposés par un tel cycle de vie, à savoir la transformation des règles, la consolidation des nouvelles annotations et instances vis-à-vis du modèle de l'ontologie et la maintenance des lexiques.

Dans la troisième partie, nous soumettons des propositions pour l'opérationnalisation de la démarche OntoPop à travers une méthodologie en cinq étapes (cf. Chapitre 5) et une plateforme logicielle basée sur l'outil de représentation des connaissances ITM de la société Mondeca (cf. Chapitre 6). La méthodologie a pour objectif de fournir un mode d'emploi simple et efficace pour la réalisation d'une application concrète d'annotation sémantique ou de peuplement d'ontologie au sein d'une entreprise. La plateforme logicielle offre des exemples de composants logiciels pouvant être développés en concordance avec la démarche proposée par OntoPop.

Enfin la quatrième et dernière partie présente le cadre des expérimentations réalisées sur deux projets choisis pour leurs différences et leur complexité (cf. Chapitre 7). Nous y exposons les différentes mesures utilisées pour l'évaluation ainsi que les résultats obtenus pour chacun des projets et discutons de la validité des résultats et de la nécessité à mettre rapidement en place un protocole d'évaluation standard aux applications d'annotations sémantique ou de peuplement d'ontologie. Enfin, nous proposons un bilan du travail accompli dans cette thèse en relevant les limites possibles de notre démarche et en dégageant des perspectives de recherche qui permettraient peut-être de les dépasser (cf. Chapitre 8).

Première partie.

Etat des lieux autour de l'annotation sémantique

Chapitre 1. L'annotation et le Web Sémantique

L'objectif de ce chapitre est de dresser le portrait de l'annotation sémantique en s'appuyant sur la vision glanée tout au long de notre recherche. Nous soulignerons tout d'abord les différentes facettes et définitions de l'annotation. Puis, nous étudierons le lien existant entre l'annotation sémantique et les ressources terminologiques ou ontologiques existantes. Ensuite, nous donnerons quelques exemples de langages actuels pour la représentation de ces ressources et leur impact quand à l'écriture des annotations sémantiques. Enfin, nous présenterons une synthèse des différents outils d'annotation sémantique issus de la recherche de ces cinq dernières années.

1.1 L'annotation sémantique

1.1.1 Quelques définitions

Le Petit Robert définit le terme **annotation** comme une « note critique ou explicative qui accompagne un texte – une note de lecture qu'on inscrit sur un livre ». De son côté, le Dictionnaire de l'Académie Française (9ème édition)[3], précise que le terme **annotation** est un dérivé du terme latin *annotare,* signifiant « noter ; annoter ». L'annotation correspond donc à « l'action d'annoter, au résultat de cette action » comme dans la phrase « *Cet exemplaire contient des annotations manuscrites de l'auteur* ». Le terme **annoter** est quant à lui défini comme « accompagner un texte de notes, de remarques, de commentaires », par exemple « *Montaigne annotait sans cesse les premières éditions des "Essais"* ».

Ainsi, le terme **annotation** réfère à une note, une critique, une explication ou encore à un commentaire. Or, nous rédigeons une note sur un sujet ou bien nous critiquons, expliquons, commentons un sujet. Une annotation seule ne fait pas sens, elle est toujours associée à l'objet qui a été annoté. C'est pourquoi les annotations sont considérées comme des **métadonnées**. Comme le souligne Handschuh [HAN 05], si une métadonnée est une donnée sur une donnée, une annotation constitue un cas particulier d'une métadonnée puisqu'elle représente une nouvelle donnée attachée à une ressource documentaire. Prié & Garlatti distinguent une métadonnée comme une description normalisée attachée à une ressource identifiée (sur le Web notamment) et une annotation comme un commentaire libre situé à l'intérieur de la ressource documentaire [PRI 04]. Il est important ici de préciser la notion de **ressource documentaire** : elle peut correspondre à l'ensemble d'un document ou bien seulement à un fragment de celui-ci et contenir du texte, de l'image, du son, de la vidéo

[3] http://atilf.atilf.fr/academie9.htm

ou une combinaison de ces contenus. Dans le cadre du Web, elle sera identifiée et accessible via une adresse URL [LAU 07].

L'annotation de ressources documentaires est une vieille tradition dans le monde de la documentation et des bibliothèques. La Digital Library Federation[4] (DLF), une association constituée des quinze bibliothèques américaines les plus importantes aux Etats-Unis, a défini trois sortes d'annotations qui peuvent s'appliquer aux ressources documentaires d'une bibliothèque numérique [HAN 05] :

- *L'annotation administrative*, ou *annotation documentaire* [LAU 07], indique les informations associées à la création et à la maintenance de la ressource documentaire telles « qui, quoi, où et comment ». Depuis l'avènement du Web, le langage DublinCore fait office de standard pour l'annotation avec des descripteurs tels que l'auteur, le titre, la source, l'éditeur, la date de publication, la langue, etc.
- *L'annotation structurelle* relie des parties de ressources documentaires entre elles afin de constituer une représentation logique d'un document [RIN 03].
- *L'annotation descriptive* décrit une ressource documentaire vis-à-vis de son contenu, c'est-à-dire qu'elle va dégager les concepts mentionnés dans la ressource documentaire, les relations entre ces concepts ainsi que leurs instances [LAU 07] [RIN 03].

Dans la suite de ce mémoire, nous nous intéressons principalement à l'annotation descriptive. Le contenu d'un document peut être analysé selon différents angles et chacun d'entre eux peut être utile pour un but bien précis de l'application qui utilise les annotations descriptives issues de ces analyses. Euzenat [EUZ 05] distingue notamment trois structures pouvant constituer des angles d'approches différents pour l'annotation descriptive : « *la structure grammaticale pour analyser les relations entre syntagmes, la structure rhétorique pour dégager l'argumentation d'un texte ou encore la structure logique pour interroger le sens d'un document* ».

Dans le cadre du Web Sémantique, une annotation descriptive, notamment lorsqu'elle s'intéresse à la structure logique du contenu d'un document, est le plus souvent appelée **annotation sémantique** [PRI 04]. Comme souligné dans [LAU 07], le terme « sémantique » est ambigu mais il indique une volonté de faire émerger le sens d'un contenu et ce, de manière plus ou moins formelle selon les préceptes de la logique. Les annotations sémantiques ont donc pour objectif d'exprimer la « sémantique » du contenu d'une ressource afin d'en améliorer sa compréhension, sa recherche et donc sa réutilisation par les utilisateurs finaux [COR 06]. Par conséquent, nous définissons l'annotation sémantique comme **une représentation formelle d'un contenu, exprimée à l'aide de concepts, relations et instances décrits dans une ontologie (cf. §1.2.1), et reliée à la ressource documentaire source**.

[4] http://www.diglib.org/dlfhomepage.htm

Dans la prochaine section, nous présentons les différentes dimensions qui caractérisent une annotation, et plus particulièrement dans la perspective de l'annotation sémantique.

1.1.2 Les dimensions de l'annotation sémantique

Dans les années 90, Marshall s'est intéressée aux annotations créées par des étudiants dans leurs livres de cours universitaires [MAR 98a] [MAR 98b]. Dans cette étude, il apparaît que les étudiants ont de nombreuses ressources à leur disposition pour annoter leurs livres de cours, comme des surligneurs, des stylos, des crayons, etc. Leurs annotations consistaient à ajouter des notations symboliques ou de longs commentaires en langage naturel. Ils écrivaient aussi bien dans les marges, entre ou sur les lignes des pages, dans les couvertures. Ils pouvaient aussi bien encercler ou surligner les différents passages qu'ils jugeaient importants comme des chapitres entiers, des sections ou sous-sections, des paragraphes, des phrases ou tout simplement un mot. Ainsi, les annotations varient en fonction des utilisateurs, de l'objet annoté et du but de l'annotation.

De cette étude, Marshall a distingué sept dimensions permettant de caractériser les annotations de ressources documentaires [MAR 98a]. Le Tableau 1 présente les résultats de cette étude avec en perspective l'application de ces dimensions à l'annotation sémantique de ressources numériques, que ces dernières soient disponibles sur le Web ou sur un réseau d'entreprise. L'analyse de Marshall concluait que les annotations liées aux ressources documentaires traditionnelles, c'est-à-dire sur support « papier », sont généralement personnelles, informelles, intensives, transitoires et privées [MAR 98 a]. A contrario, les annotations de ressources numériques sont créées dans un monde documentaire où ces ressources ont pour vocation d'être partagées et exploitées par des utilisateurs divers et variés. Ceci se répercute sur leurs annotations qui sont alors les plus formelles possibles, hyper-extensives, permanentes, globales et publiques.

Dimension	Analyse effectuée par Marshall	Analyse des annotations sémantiques
Formelle versus informelle	Les annotations informelles sont celles écrites en langage naturel dans la marge du document alors que les annotations formelles prennent la forme de métadonnées structurées par l'utilisation d'un langage standard définissant un ensemble de conventions de nommage et de valeurs par défaut. Ces annotations formelles permettent d'assurer l'interopérabilité entre les différentes annotations qui suivent ce standard et leur interprétation par des outils qui implémentent ce même standard	Différents langages, plus ou moins formels selon les langages de représentation de la connaissance utilisés, permettent de représenter des annotations sémantiques.

Tacite versus explicite	Les annotations personnelles sont très souvent tacites (un passage souligné ou une marque allusive comme un point d'exclamation par exemple). Elles posent des problèmes d'interprétation pour les autres utilisateurs que l'auteur de l'annotation. Plus les annotations ont pour but d'être partagées avec d'autres utilisateurs, plus elles doivent être explicites.	Les ressources numériques, tout comme leurs annotations, ont souvent pour but d'être partagées entre divers utilisateurs. Par conséquent, les annotations doivent être les plus explicites possibles et pour ce faire, s'appuyer sur des langages formels afin de pouvoir désambiguïser le contenu des documents.
Ecriture versus lecture	Les annotations oscillent entre représenter une aide ou une explication à la lecture du document ou bien constituer une nouvelle forme d'écriture en tant que telle, ajoutant du sens au texte écrit.	Les annotations de ressources numériques représentent non seulement une aide à la lecture de la ressource annotée, mais plus encore à la recherche de ces ressources. Elles permettent également de générer de nouvelles instances de connaissances pouvant être stockées et réutilisées par diverses applications informatiques.
Hyper-extensive versus extensive versus intensive	Une annotation hyper-extensive est une annotation de surface (structurée, un peu à la manière d'un lien hypertexte) alors qu'une annotation dite « intensive » est une annotation de fond (un commentaire descriptif par exemple). Les annotations dites « extensives » représentent un intermédiaire entre ces deux distinctions.	Les annotations sémantiques exploitent surtout l'hyper-extensivité, notamment par l'utilisation des liens hypertextes pour le référencement et l'adressage.
Permanente versus transitoire	Certaines annotations ne sont utiles qu'à son auteur à un moment donné alors que d'autres peuvent perdurer tout en gardant leur valeur ajoutée aussi bien pour l'auteur que pour d'autres utilisateurs	Comme l'objectif premier des annotations sémantiques est le partage et la réutilisation, elles sont donc préférablement permanentes plutôt que transitoires. Mais d'un autre côté les documents numériques sont plus sujets à modification que les documents papiers, surtout les pages Web. Les annotations sémantiques doivent alors évoluer en fonction du contenu modifié
Publique versus privée	Les annotations peuvent être destinées à rester dans l'intimité d'un auteur, qui y consigne ses impressions de lecture par exemple, ou au contraire à être divulguées à de multiples utilisateurs. Ces utilisateurs pourront à leur tour compléter les annotations produites par l'auteur initial	Les annotations sémantiques ont principalement une visée publique, notamment lorsque les ressources documentaires associées sont mises en ligne sur le Web. Néanmoins, un utilisateur peut désirer créer des annotations sémantiques pour son usage personnel
Globale versus institutionnelle versus personnelle	Les bénéfices attendus des annotations créées varient en fonction des groupes d'utilisateurs qui exploiteront ces annotations.	Dans le cadre du Web Sémantique, les utilisateurs ne sont plus seulement des humains, mais aussi des machines, des agents logiciels. La portée de ces annotations est donc plus institutionnelle, voire globale, que personnelle.

Tableau 1. Analyse de l'annotation sémantique de documents numériques en perspective avec les sept dimensions déterminées par Marshall dans [MAR 98a]

Aux dimensions définies ci-dessus, Prié & Garlatti [PRI 04] ajoutent d'autres dimensions permettant de caractériser plus finement la spécificité des annotations de ressources documentaires numériques. Ces dimensions comprennent la nature des ressources documentaires, la structuration des modèles formels utilisés pour l'annotation, l'automatisation de la création des annotations, leur stockage vis-à-vis de la ressource annotée et l'utilisation de ces annotations par les agents logiciels. Nous allons revenir sur chacune de ces dimensions afin d'en préciser le sens et leur impact sur les annotations sémantiques.

1.1.2.1 La nature des ressources documentaires

Comme nous l'avons dit plus haut, une ressource documentaire peut correspondre à l'ensemble d'un document ou à un fragment de celui-ci. Mais cette ressource peut contenir des informations de natures différentes : du texte, des images, du son, de la vidéo, etc. D'autre part, même un texte peut être plus ou moins structuré. Cette nature de l'information peut jouer un rôle prépondérant dans la constitution et la création des annotations sémantiques. Par exemple, dans le cadre d'un texte non structuré, il peut être fait appel aux méthodes et techniques initiées par le domaine du traitement automatique du langage naturel (TALN). Dans la suite de ce mémoire, nous nous intéressons uniquement aux ressources documentaires de nature textuelle, structurées ou non.

1.1.2.2 La structuration des modèles formels utilisés pour l'annotation sémantique

Les modèles formels utilisés pour l'annotation sémantique peuvent être plus ou moins structurés. Le standard DublinCore, évoqué précédemment, peut être utilisé, même si les valeurs de ces descripteurs renvoient généralement à des chaînes de caractères non normalisées, cf. Figure 2.

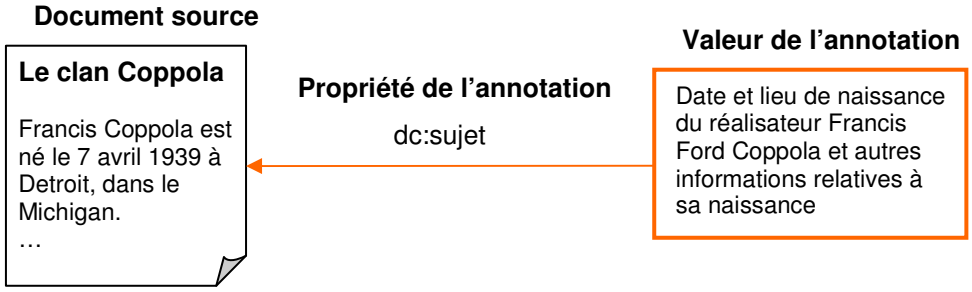


Figure 2. Exemple d'une annotation utilisant le descripteur « dc:sujet » du DublinCore pour annoter le contenu du document source

Les valeurs de ces annotations sont en langage naturel. Elles ont donc l'inconvénient d'être surtout exploitables par des utilisateurs humains et moins par des machines. Analyser une expression en langage naturel est une tâche très complexe pour une machine. Traditionnellement [NAZ 05] [NAZ 06], l'analyse linguistique d'un texte est découpée et organisée en différents niveaux relevant de la structure interne des mots (morphologie), de leur organisation en groupes de mots et en phrases (syntaxe), de l'analyse du sens des mots et des phrases (sémantique) [SOW 00]. Un dernier niveau peut être ajouté, celui de la pragmatique qui est l'étude de l'interprétation en contexte en fonction des

connaissances générales du monde et de la situation de communication, mais que nous occulterons ici car trop difficile à traiter informatiquement [WEH 97]. Ainsi, la complexité de l'analyse linguistique est décomposée en un ensemble de problèmes réputés plus simples selon le précepte cartésien [FUC 93]. Chaque niveau de l'analyse linguistique génère un ensemble d'annotations (au sens général du terme, i.e. où de l'information est ajoutée au document textuel afin d'en caractériser son contenu ou sa forme). Ici, chaque annotation représente le résultat d'un niveau donné de l'analyse linguistique et peut être exploitée par le niveau suivant de l'analyse. Ces annotations sont aussi appelées étiquettes dans la mesure où chaque niveau d'analyse attache directement un ensemble d'étiquettes prédéterminées à chaque unité textuelle traitée [HAB 05]. Un exemple d'annotations générées[5] à chaque niveau d'analyse linguistique est donné dans la Figure 3.

L'analyse sémantique est au cœur de tous les mécanismes de compréhension de la langue, permettant d'analyser, de traduire et d'interpréter les phrases et plus globalement les textes. Le lexique joue un rôle central dans la résolution des ambiguïtés et des exceptions liées à cette analyse sémantique – deux problèmes majeurs pour le TAL [NAZ 06]. Parmi les ressources lexicales disponibles aujourd'hui, citons WordNet [FEL 98] ou Memodata [DUT 03].

Deux grandes familles de formalismes sont utilisées pour construire les représentations sémantiques : d'une part les structures comme les frames, les réseaux sémantiques ou encore les graphes conceptuels [SOW 00] et d'autre part les formalismes logiques, avec notamment l'utilisation de la logique des prédicats, dite aussi logique du premier ordre ou logique classique [BOU 98]. Toutefois de nombreux phénomènes échappent à la logique classique, comme le rapport au temps, l'action, les modalités (nécessaires, possibles, contingentes), les croyances, les commandements et interrogations, etc. Pour tenter de caractériser plus finement les liens sémantiques qui unissent le prédicat à ses divers arguments, certains chercheurs et en particulier Fillmore, préconisent l'utilisation de grammaires de cas [FIL 68]. Mais en pratique, il s'est avéré extrêmement difficile de mettre en œuvre de façon opératoire une théorie des cas sémantiques, dès lors que l'on s'éloigne de petits schémas de phrases simples « sujet – verbe – complément » [FUC 93].

La très grande majorité des systèmes de TAL ont adopté une approche séquentielle bien qu'elle soit quelque peu théorique en raison des multiples ambiguïtés de la langue [CHA 05]. Par ailleurs, comparativement à la morphologie et la syntaxe, les travaux en sémantique n'ont pas atteint le même niveau de développement. Le niveau sémantique est beaucoup plus complexe à décrire et à formaliser que les précédents. Aussi les réalisations opérationnelles sont-elles plus difficiles à réaliser, et concernent-elles des applications très limitées où l'analyse sémantique se réduit de fait à l'analyse d'un domaine parfaitement circonscrit [FUC 93]. Par ailleurs, comme le souligne Habert [HAB 05], « *Les résultats de la plupart des annotations fournies jusqu'à présent relèvent de formats propriétaires. Ils ont été développés pour les sorties d'un logiciel déterminé et ne sont pas prévus a*

[5] L'analyse morphologique a été réalisée avec l'outil PILAF, dont un démonstrateur est en ligne sur le site http://www-clips.imag.fr/geta/User/damien.genthial/Pilaf/analyse.html. L'analyseur syntaxique est disponible sur la page personnelle, http://www.lirmm.fr/~chauche/ExempleAnl.html, de Jacques Chauché, chercheur au LIRMM. L'analyse sémantique a été réalisée manuellement.

priori pour faciliter les échanges et le travail en aval ». Il est donc absolument crucial de disposer d'un format de représentation standard, consensuel et formel afin que les annotations puissent être créées, exploitées et maintenues par différents utilisateurs, qu'ils soient humains ou logiciels.

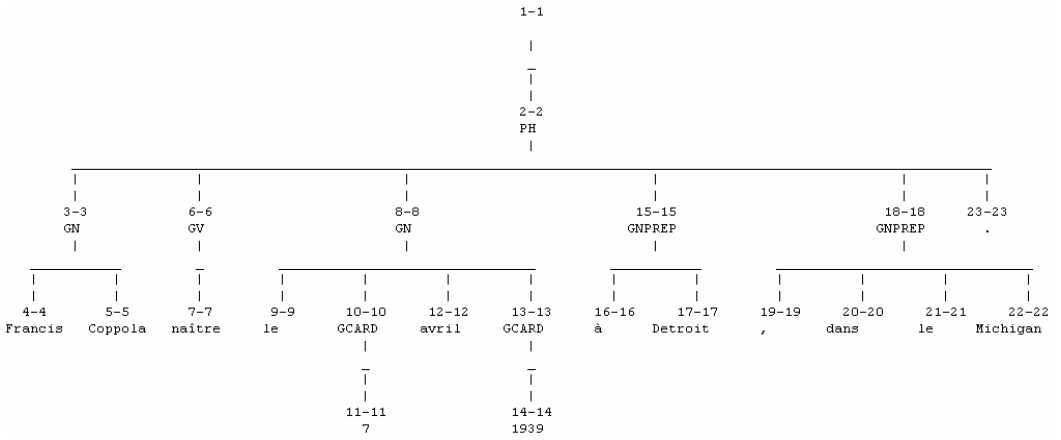
Document source

Le clan Coppola
Francis Coppola est né le 7 avril 1939 à Detroit, dans le Michigan. …

Analyse morphologique

Lemme	Forme	**CAT.** Variables
francis	francis	**cls**
coppola	coppola	**cls**
est	est	**vet** sin tre pre ind
né	né	**ppas** sin mas
le	le	**detp** sin mas tre cod
7	7	**chf**
avril	avril	**subc** sin mas
1939	1939	**chf**
à	à	**à**
detroit	detroit	**cls**
,	,	**virg**
dans	dans	**prep**
le	le	**detp** sin mas tre cod
michigan	michigan	**cls**
.	.	**pnt**

Analyse syntaxique



Analyse sémantique

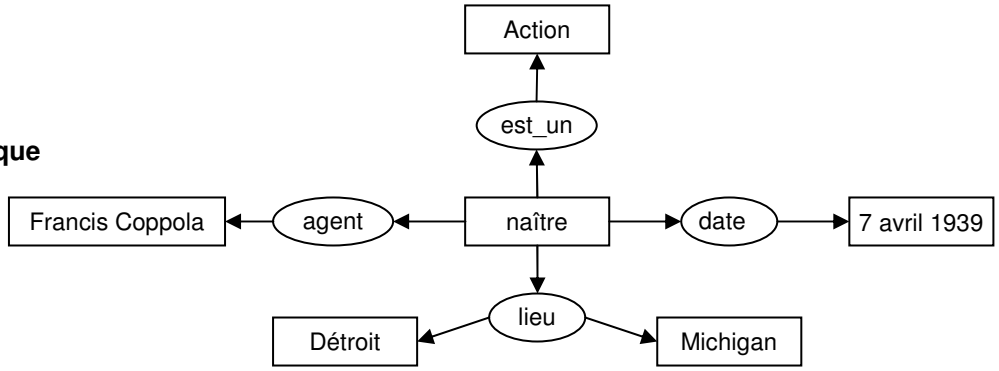


Figure 3. Exemple d'annotations générées aux différents niveaux morphologique, syntaxique et sémantique d'une analyse linguistique

Une ontologie, telle que définie dans le cadre de l'Ingénierie des Connaissances, représente à la fois cet objet de consensus pour les humains et un objet formel permettant son exploitation par un agent

logiciel [LAU 07]. Sa représentation s'inspire en effet à la fois des réseaux sémantiques et de la théorie des logiques de description. Elle se compose de concepts (ou classes) et de propriétés (relations ou attributs) et de contraintes qui définissent et précisent l'utilisation de ces concepts et propriétés (cf. §1.2.1). Elle est décrite dans un langage formel de représentation des connaissances, tel que RDF(S) ou mieux encore OWL (cf. §1.3.2). Une annotation dans ce contexte permet de relier le contenu du texte à des instances de concepts ou de propriétés décrits dans l'ontologie de référence, cf. Figure 4.

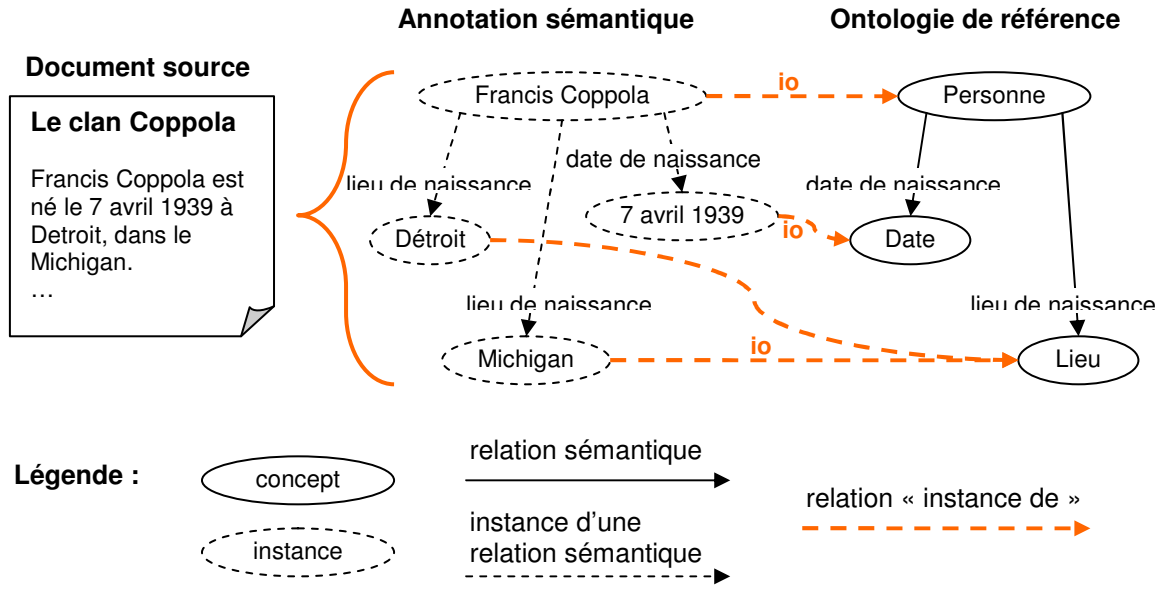


Figure 4. Exemple d'une annotation sémantique orchestrée par une ontologie de référence

Le fait d'utiliser la sémantique de ces concepts et relations telle que modélisée dans l'ontologie pour créer l'annotation, permet alors d'assigner cette sémantique au contenu de la ressource et donc de le rendre exploitable par des agents logiciels.

1.1.2.3 L'automatisation de la création des annotations

Fuchs définit un traitement automatique comme opérant par des « moyens mécaniques » alors qu'un traitement manuel, ou « instrumental » est opéré par un être humain [FUC 93]. Les moyens mécaniques correspondent aux ordinateurs, et plus précisément aux agents logiciels. Il existe des traitements entièrement automatisés, i.e. sans intervention d'un être humain dans l'exécution du traitement, et d'autres partiellement automatisés, i.e. que l'intervention d'un humain soit requise en amont pour la préparation du traitement ou en aval pour la vérification et la validation des résultats produits.

En ce qui concerne la création des annotations sémantiques, le traitement manuel consiste simplement à mettre en place une interface utilisateur dans laquelle l'utilisateur humain peut sélectionner la ressource à annoter, choisir le modèle formel servant à la création des annotations sémantiques et, tout en respectant les contraintes imposées par le modèle formel, créer les annotations voulues sur la ressource sélectionnée [KAH 01] [HAN 01].

Le traitement automatisé pour la création des annotations sémantiques est en fait plutôt semi-automatique. En effet, les traitements entièrement automatisés utilisent des algorithmes basés sur des modèles statistiques ou sur l'exploitation de la redondance dans un corpus de ressources [CIM 04] [DIL 03]. Mais leur perspicacité est limitée et leurs annotations souvent non exploitables dans une application concrète, notamment en entreprise. Les outils d'annotation qui reposent sur cette méthode ne sont viables que pour des projets en laboratoires, à titre expérimental. Par contre, les traitements semi-automatiques ont prouvé au cours des dernières années qu'ils pouvaient apporter une aide non négligeable à l'annotateur humain. Ces traitements semi-automatiques s'appuient généralement sur un moteur d'extraction d'information. La construction d'un moteur s'effectue selon deux méthodes : soit la création manuelle de ses patrons d'extraction [VAR 02b] [POP 03], soit leur apprentissage supervisé [HAN 02] [DIN 03b].

- **Création manuelle des patrons :**

Cette première méthode permet de faire fonctionner un moteur d'extraction d'information à partir d'un ensemble de règles définies par un utilisateur expérimenté, généralement un linguiste. Ces règles sont soit des expressions régulières (par exemple l'expression régulière d'une date sous la forme « jj/mm/aaaa ») soit par des résultats d'analyses linguistiques plus poussées (par exemple si l'analyse syntaxique trouve un sujet représentant une entité nommée de type personne et un verbe signifiant la naissance et un complément de lieu représentant une entité nommée de type lieu, alors on peut annoter cette phrase comme étant le lieu de naissance de la personne du sujet) [GRI 97]. L'ensemble de ces règles est ensuite compilée dans un automate à états finis. L'automate parcourt l'ensemble du corpus documentaire et l'annote en fonction des règles qui sont déclenchées. Les annotations sont ensuite proposées à l'annotateur humain via une interface dédiée. Celui-ci doit alors les passer en revue pour les valider, les modifier ou les supprimer. L'analyse linguistique peut être plus ou moins fine, mais elle convient bien à l'annotation de textes semi ou non structurés. Par contre, la définition des règles s'appuyant sur des expressions linguistiques particulières à un domaine donné, cette méthode est fortement dépendante de ce domaine et souvent difficile à adapter à de nouveaux domaines sans avoir besoin de redéfinir les règles d'extraction.

- **Apprentissage supervisé des patrons :**

Cette méthode permet de mettre en place un système capable d'apprendre à annoter un corpus donné. L'annotateur humain initie cet apprentissage en annotant manuellement un sous-ensemble du corpus. Ces annotations servent alors d'exemples au moteur d'extraction qui en déduit un ensemble de règles : de simples expressions régulières du langage ou des expressions régulières s'appuyant sur la structure même des documents, par exemple l'utilisation des balises HTML ou XML contenues dans le document analysé. L'annotateur humain peut aussi initier l'apprentissage en fournissant un ensemble de règles de base. Le moteur d'extraction est ensuite capable d'apprendre à annoter des ressources en redéfinissant itérativement ces expressions régulières, soit par généralisation soit par spécification, à partir des corrections fournies au fur et à mesure par l'annotateur humain et ce, jusqu'à l'obtention d'un seuil de réussite acceptable pour ce dernier. Une fois la phase d'apprentissage achevée, le moteur peut alors automatiquement annoter le reste du corpus

documentaire. Cette méthode est plutôt adaptée aux documents très structurés (issus des bases de données) ou semi-structurés (comme les pages Web) dans lesquels les constructions sont suffisamment simples et fixes pour que des expressions régulières linguistiques ou structurelles puissent être apprises facilement sans générer trop d'annotations non pertinentes au domaine étudié. Le problème avec cette méthode, c'est que non seulement un nombre suffisant d'exemples d'annotations doit être fourni au moteur d'apprentissage mais en plus, celles-ci doivent être pertinentes vis-à-vis du domaine concerné. Or, il n'est pas toujours facile de constituer ou de récupérer un tel corpus d'annotations [HAB 05].

1.1.2.4 Le stockage des annotations et de leurs ressources

Les annotations peuvent être soit « embarquées » soit « débarquées » vis-à-vis de la ressource documentaire source [HAB 05]. Une annotation est dite « embarquée » lorsqu'elle est ajoutée directement au contenu du document d'origine. Ce dernier pourra être mis à jour sur le site Web d'origine, dans le système de gestion de contenu documentaire de l'entreprise, ou simplement enregistré localement sur un ordinateur pour réutilisation ultérieure. A contrario, l'annotation est dite « débarquée » lorsqu'elle est stockée à l'extérieur du document source. Non seulement l'annotation elle-même doit être stockée, mais le lien avec la ressource annotée doit aussi être préservé. C'est ce qui permettra ensuite de retrouver toutes les annotations correspondantes à une ressource. Généralement, ces annotations sont stockées sur des serveurs d'annotations qui peuvent être interrogés afin de retrouver les annotations d'une ressource donnée. L'avantage de l'annotation débarquée est qu'il devient possible d'annoter toute ressource documentaire, y compris celles dont l'application n'est pas propriétaire. Ceci est particulièrement vrai dans le cadre de l'annotation de pages web. L'inconvénient réside dans le fait que si le document source est modifié ou supprimé, les annotations deviennent obsolètes voire orphelines.

Rapellons ici que nous nous focalisons sur les annotations sémantiques et que nous supposons que celles-ci sont structurées et décrites formellement pour représenter de la connaissance sous la forme d'instances de concepts et de relations de l'ontologie de référence. Ces instances peuvent être stockées dans une base de connaissance, indépendamment de la ressource annotée, afin d'être réutilisées pour de nouvelles annotations d'autres ressources ou tout simplement pour être interrogées par les utilisateurs de cette base de connaissance. Dans ce cas précis, nous parlons de **peuplement d'ontologie** ou d'**enrichissement de base de connaissance**. Ces notions décrivent l'action d'ajouter de nouvelles instances à une base de connaissance contrainte par une ontologie, autrement dit à peupler une ontologie avec des instances de concepts et de relations [PRI 04]. Les outils d'annotation sémantique peuvent donc être capables de peupler une ontologie existante à partir des annotations sémantiques créées pour une ressource donnée, à moins que l'ontologie ne soit déjà pré-peuplée avec toutes les instances possibles du domaine concerné. Il est aussi intéressant de noter que certains outils discutés à la section 1.4 n'ont pas pour objectif premier la tâche d'annotation. Ils considèrent plutôt l'annotation comme un moyen de capturer la connaissance d'un domaine pour peupler une ontologie.

Remarque : Attention toutefois à bien distinguer la notion de peuplement d'ontologie et celle d'**enrichissement d'ontologie**. Dans ce dernier cas, il s'agit non pas d'ajouter de nouvelles instances à des concepts existants, mais plutôt d'ajouter de nouveaux concepts ou relations au modèle formel de l'ontologie. Il faut néanmoins prévoir la manière dont l'ontologie est stockée afin que les outils d'annotation puissent interroger et exploiter son modèle formel, voire raisonner à partir de celui-ci. Nous reviendrons sur la notion de peuplement d'ontologie dans la suite de ce mémoire.

1.1.2.5 L'utilisation des annotations par les agents logiciels

L'annotation sémantique permet de nombreuses applications comme la recherche d'information sémantique, la catégorisation, la composition de documents, etc. [PRI 04]. L'annotation sémantique est applicable à n'importe quel type de contenu : pages web, documents textuels non structurés, champs d'une base de données, documents audio ou vidéo, etc. Enfin, plus le modèle de l'annotation est formalisé, plus les services proposés à partir de cette annotation peuvent devenir « intelligents ». En effet, les agents logiciels pourront inférer de la nouvelle connaissance, raisonner sur cette connaissance et ainsi améliorer les résultats de la recherche d'information ou bien dégager un sens implicite contenu dans le document d'origine [LAU 07].

Nous venons de définir ce qu'est l'annotation, et plus particulièrement l'annotation sémantique de ressources textuelles, à partir des dimensions proposées par Marshall puis par Prié et Garlatti. Comme nous l'avons expliqué, ces annotations sémantiques ont besoin d'être exprimées de manière formelle. Nous allons donc à présent étudier quels sont les ressources et les langages disponibles qui permettent de créer des annotations sémantiques, notamment dans le cadre du projet du Web Sémantique.

1.2 L'annotation et le Web Sémantique

Dans le cadre du Web Sémantique, l'objectif est de décrire le contenu des ressources en les annotant avec des informations non ambigües afin de favoriser l'exploitation de ces ressources par des agents logiciels [PRI 04]. Or, les données actuelles du Web sont encore trop souvent écrites en langage naturel, car destinées aux humains. Le langage naturel étant par essence trop ambigu, des alternatives formelles et sémantiquement explicites doivent être mises en place pour lever les ambiguïtés du langage naturel, aussi bien dans le contenu des ressources que dans leurs annotations. La tâche d'annotation pour le Web Sémantique consiste donc à prendre en entrée une ressource documentaire et fournir en sortie le même contenu enrichi par des annotations sémantiques basées sur des représentations de la connaissance plus ou moins formelles.

Nous allons tout d'abord nous intéresser aux ressources terminologiques ou ontologiques (RTO) [BOU 04] qui permettent de représenter la connaissance d'un domaine. Nous illustrerons ensuite

comment les annotations sémantiques relient le document source aux RTO [CON 03]. Enfin, nous décrirons quelques-uns des langages permettant d'exprimer plus ou moins formellement des annotations sémantiques selon les modèles formels utilisés.

1.2.1 Les Ressources Terminologiques ou Ontologiques (RTO)

Bourigault et al. ont défini la notion de Ressources Terminologiques ou Ontologiques (RTO) à la croisée des domaines de la Terminologie et de l'Intelligence Artificielle, et plus particulièrement de l'ingénierie des connaissances [BOU 04]. Cette notion regroupe plusieurs sortes de ressources, allant des index et glossaires jusqu'aux ontologies en passant par les bases de données lexicales et les thesaurus. Nous allons présenter les trois principales RTO permettant de représenter et de modéliser la connaissance d'un domaine : les taxonomies, les thesaurus et les ontologies[6].

1.2.1.1 Les Taxonomies

Les systèmes de représentation des connaissances et de raisonnement en Intelligence Artificielle ont besoin de représenter l'existant de manière formelle. Il peut très bien s'agir d'objets concrets tels une Personne ou une Voiture ou bien de notions abstraites telles un Style de peinture ou une Idée. Or, la science a toujours eu pour premier but de repérer et classifier les objets du monde pour les étudier et les comprendre [CHA 02]. Ceci est particulièrement vrai pour les sciences naturelles qui ont construit des classifications [BOU 04], appelées **taxinomies** ou **taxonomies**, au sujet de la botanique ou de la faune, comme celle présentée dans la Figure 5. L'objectif d'une taxonomie est de conceptualiser les objets du monde et de les organiser hiérarchiquement les uns par rapport aux autres.

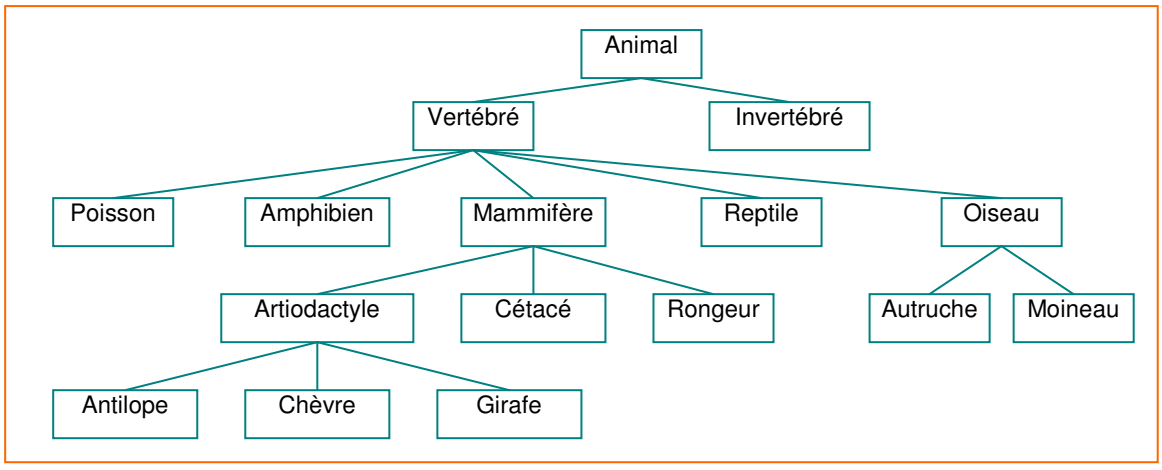


Figure 5. Extrait d'une taxonomie sur la représentation simplifiée de la faune[7]

La hiérarchie d'une taxonomie, modélisée via une structure arborescente, représente le plus souvent une relation de **subsomption** ou d'hyperonymie, i.e. « est un ». Cette relation est très ancienne puisqu'elle remonte à Aristote [CHA 02]. Plus les concepts sont proches de la racine, plus leur

[6] Se référer aussi à la thèse d'Audrey Baneyx [BAN 07] au sujet des différentes RTO existantes
[7] http://www.educ.csmv.qc.ca/mgrparent/vieanimale/taxonomie2.html

signification est générale et à l'inverse, plus ils sont proches des feuilles, plus leur signification est spécifique au domaine concerné. On appelle aussi cette relation « concept/sous-concept » ou « classe/sous-classe ». Cette relation compose généralement la structure de base des ontologies comme nous le verrons par la suite car elle est à la base des mécanismes de raisonnements en Intelligence Artificielle [LAU 07]. Dans l'exemple de la Figure 5, le concept « Cétacé » est considéré comme un sous-concept de « Mammifère », héritant donc de toutes les spécificités de ce concept. Cela n'est pas sans poser des problèmes de modélisation [ENJ 05c] : si un concept « Oiseau » est défini comme un « animal vertébré qui vole » et si le concept « Autruche » est représenté comme un sous-concept de « Oiseau », il ne peut pas pour autant hériter de la propriété « voler » de ce concept « Oiseau ». Pour une discussion approfondie des problèmes de représentation de la connaissance liée aux taxonomies, se référer à [CHA 02].

Il se peut que la hiérarchie représente une autre relation, celle d'**agrégation** ou de méronymie, i.e. « partie/tout ». Cette relation permet de modéliser le fait qu'un concept se situant à un niveau inférieur de la taxonomie représente une sous-partie du concept du niveau supérieur. La classification des lieux géographiques est un exemple d'utilisation d'une taxonomie d'agrégation : par exemple, la classe « France » peut être modélisée comme une sous-partie de la classe « Europe ». Cette relation est également très fréquente en médecine pour décrire l'organisation anatomique [CHA 02].

1.2.1.2 Les Thesaurus

Les domaines de l'Informatique documentaire et de la terminologie ont depuis longtemps mis au point des modélisations de ressources terminologiques comme les thesaurus [CHA 04]. Bourigault & al définissent un thesaurus comme « *un langage documentaire fondé sur une structuration hiérarchisée* », sachant qu'un langage documentaire est un « *ensemble organisé de termes normalisés, utilisé pour représenter le contenu des documents à des fins de mémorisation pour une recherche ultérieure* » [BOU 04]. Un thesaurus est donc considéré comme un vocabulaire contrôlé et structuré dans lequel les relations entre les termes du domaine considéré sont clairement spécifiées formant ainsi un réseau terminologique. La structuration hiérarchisée correspond à la relation d'hyperonymie déjà vue pour les taxonomies sauf qu'elle ne structure plus des concepts mais les termes du vocabulaire. On dit alors qu'un terme X a un sens plus général (TG) ou plus spécifique (TS) qu'un terme Y, par exemple « Véhicule » a un sens plus général que « Automobile » dans la Figure 6. D'autres relations constituent le réseau terminologique comme les relations suivantes (cf. Figure 6) :

- Synonymie → un terme X est le synonyme d'un terme Y, par exemple « Voiture » et « Automobile ».
- Homonymie → un terme X a la même forme orale ou écrite qu'un terme Y alors qu'ils ont des sens différents, par exemple le terme « Velot » signifiant « la peau d'un veau mort-né » est un homonyme du terme « Vélo ».
- Associative → un terme X est associé à un terme Y s'il y a une sorte de relation non sémantiquement spécifiée entre les deux, par exemple le terme « Conduite » est souvent associé au terme « Véhicule ».

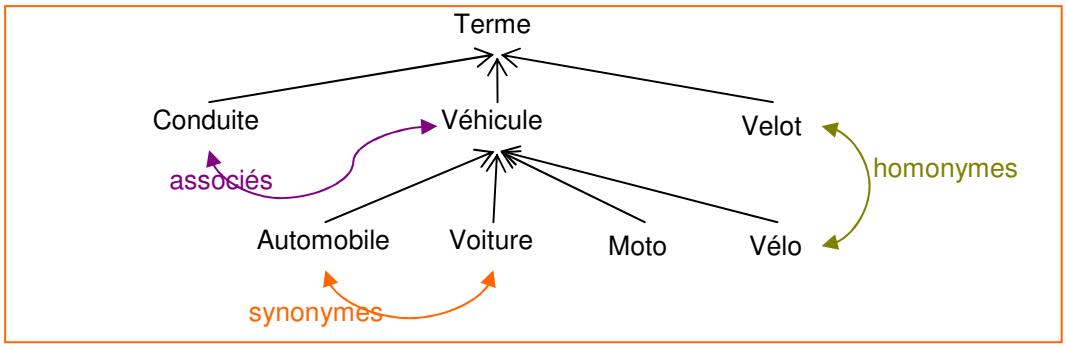


Figure 6. Les différentes relations qui composent un thesaurus

Quelques propriétés comme une définition ou une abréviation peuvent également être ajoutées aux termes d'un thesaurus. Comme mentionné dans sa définition, l'objectif premier d'un thesaurus est de faciliter la recherche de document et de rendre l'indexation de documents consistante[8] à l'aide des termes, aussi appelés descripteurs dans ce contexte. Un exemple de thesaurus disponible sur le Web est le thesaurus médical MeSH[9] (Medical Subject Heading) utilisé pour indexer la base bibliographique MEDLINE [CHA 04]. Nous voulons ici souligner le fait que les thesaurus ne sont pas des ontologies : ils permettent de modéliser le vocabulaire d'un domaine ou d'une application mais ne fournissent pas de représentation de la connaissance de ce domaine ou de cette application. Par contre, ils peuvent être complémentaires à l'utilisation d'une ontologie, notamment pour les applications d'indexation et d'annotation [HER 05], comme nous le verrons ultérieurement. Ils peuvent également être utilisés comme ressources pour l'aide à la création des ontologies [CHA 04].

1.2.1.3 Les Ontologies

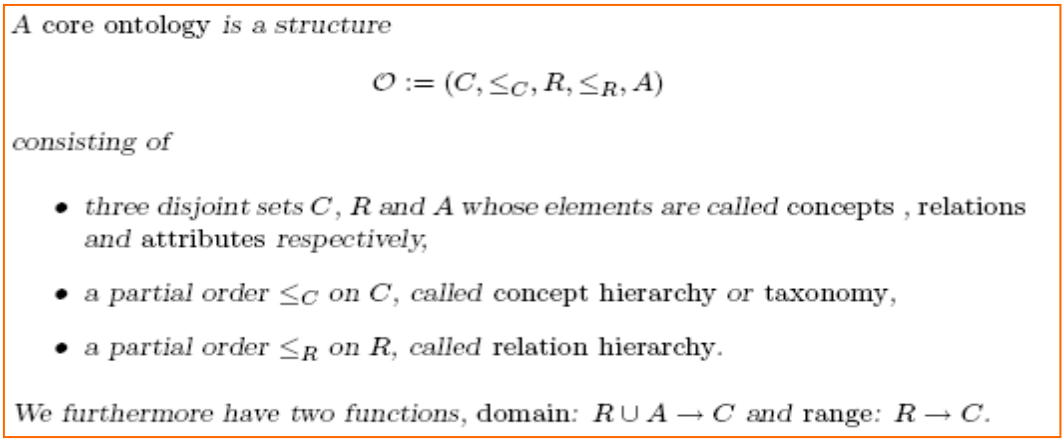
Le terme **Ontologie** (avec un O majuscule) a tout d'abord été défini en Philosophie comme une branche de la Métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant [CHA 02]. En effet, ce terme est construit à partir des racines grecques **ontos**, i.e. ce qui existe, l'Etre, l'existant, et **logos**, i.e. l'étude, le discours, d'où sa traduction par « l'étude de l'Etre » et par extension « de l'existence »[10]. Au début des années 90, des chercheurs en Intelligence Artificielle se sont intéressés à la cette notion pour la formalisation des connaissances. Dans cette discipline, ce qui « existe » peut être « représenté ». Dans ce contexte, ils ont défini une **ontologie** (avec un o minuscule) comme un artefact permettant de représenter l'existant par l'utilisation d'un vocabulaire formel et consensuel. Une des premières définitions de l'ontologie communément admise en Intelligence Artificielle a été énoncée par Gruber [GRU 93] comme la « *spécification explicite d'une conceptualisation* ». Cette définition de l'ontologie a ensuite été affinée par R. Studer et al. [STU 98] comme « *spécification formelle et explicite d'une conceptualisation partagée* » :

- *Formelle* → l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.

[8] Cf. Norme ANSI/NISO Z39.19-1993 (R1998), p. 1.
[9] http://www.nlm.nih.gov/mesh/
[10] Cf. Fabien Gandon dans Interstices : http://interstices.info/display.jsp?id=c_17672&part=0

- *Explicite* → la définition explicite des concepts utilisés et des contraintes de leur utilisation.
- *Conceptualisation* → le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- *Partagée* → l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.

En clair, une **ontologie** fournit les moyens d'exprimer les concepts d'un domaine en les organisant hiérarchiquement et en définissant leurs propriétés sémantiques dans un langage de représentation des connaissances formel favorisant le partage d'une vue consensuelle sur ce domaine entre les applications informatiques qui en font usage [BOU 04]. Définir des concepts et les relier entre eux par des relations sémantiques correspond au premier niveau d'une ontologie, le modèle conceptuel, inspiré des réseaux sémantiques, et plus encore des graphes conceptuels de Sowa [SOW 00]. Handschuh [HAN 05] présente une définition formelle d'une ontologie comme suit :



A core ontology *is a structure*

$$\mathcal{O} := (C, \leq_C, R, \leq_R, A)$$

consisting of

- *three disjoint sets* C, R *and* A *whose elements are called* concepts , relations *and* attributes *respectively,*
- *a partial order* $\leq_C$ *on* C, *called* concept hierarchy *or* taxonomy,
- *a partial order* $\leq_R$ *on* R, *called* relation hierarchy.

We furthermore have two functions, domain: $R \cup A \rightarrow C$ *and* range: $R \rightarrow C$.

Figure 7. Définition formelle d'une ontologie donnée par Handschuh [HAN 05]

Les **concepts** (aussi appelées « classes ») représentent les objets, abstraits ou concrets, réels ou fictifs, élémentaires ou composites, du monde réel. Ces concepts sont organisés en taxonomie, par l'utilisation de la relation de subsumption, dans laquelle ils peuvent appartenir à plusieurs sur-concepts différents. Dans la Figure 8, le concept « Personnalité » est une sous-classe de « Personne ».

Les **relations** représentent des interactions entre concepts permettant de construire des représentations complexes de la connaissance du domaine [CHA 04]. Elles établissent des liens sémantiques binaires, organisables hiérarchiquement. Dans le domaine modélisé ci-dessous, les concepts « Personnalité » et « Film » sont reliés entre eux par la relation sémantique « réalise(Personnalité, Film) » dans laquelle « Personnalité » est le domaine et « Film » la portée (ou « range » en anglais).

Les **attributs** correspondent à des caractéristiques, des spécificités particulières, attachées à un concept et qui permettent de le définir de manière unique dans le domaine [CHA 04]. Leurs valeurs sont littérales, i.e. de type primitif, comme une chaîne de caractère ou un nombre entier. Par exemple, un concept « Personne » peut avoir les attributs suivants : un « numéro de sécurité sociale », une « date de naissance », un « alias », etc.

Remarque : Nous laissons volontairement de côté le terme propriété qui est trop ambigu : parfois utilisé pour représenter un attribut [CHA 04], parfois une relation, parfois les deux. Par exemple, le langage RDF (cf. §1.3.2) ne distingue pas les attributs des relations, utilisant uniquement le terme propriété. Par contre, le langage de référence des ontologies pour le Web Sémantique, OWL, distingue les relations, « object properties », des attributs, « datatype properties ». Pour plus de clarté dans la suite de ce mémoire, nous emploierons le terme propriété de la même manière que dans OWL englobant ainsi les attributs et les relations, c'est-à-dire toutes les caractéristiques propres à un concept. Par ailleurs, nous jugeons important de souligner, comme le fait Charlet [CHA 04], qu'il n'est pas toujours aisé de savoir si une propriété doit se modéliser comme une relation ou comme un attribut dans la future ontologie lorsque le langage de représentation offre ces deux possibilités. Ce choix est laissé au concepteur de l'ontologie et peut être entièrement arbitraire ou, mieux, en fonction de l'application utilisant cette ontologie. Par exemple, une propriété « date de naissance » peut à la fois être modélisée comme attribut avec un type de donnée primitif tel que « xml:date » ou bien comme une relation avec un concept « Date ».

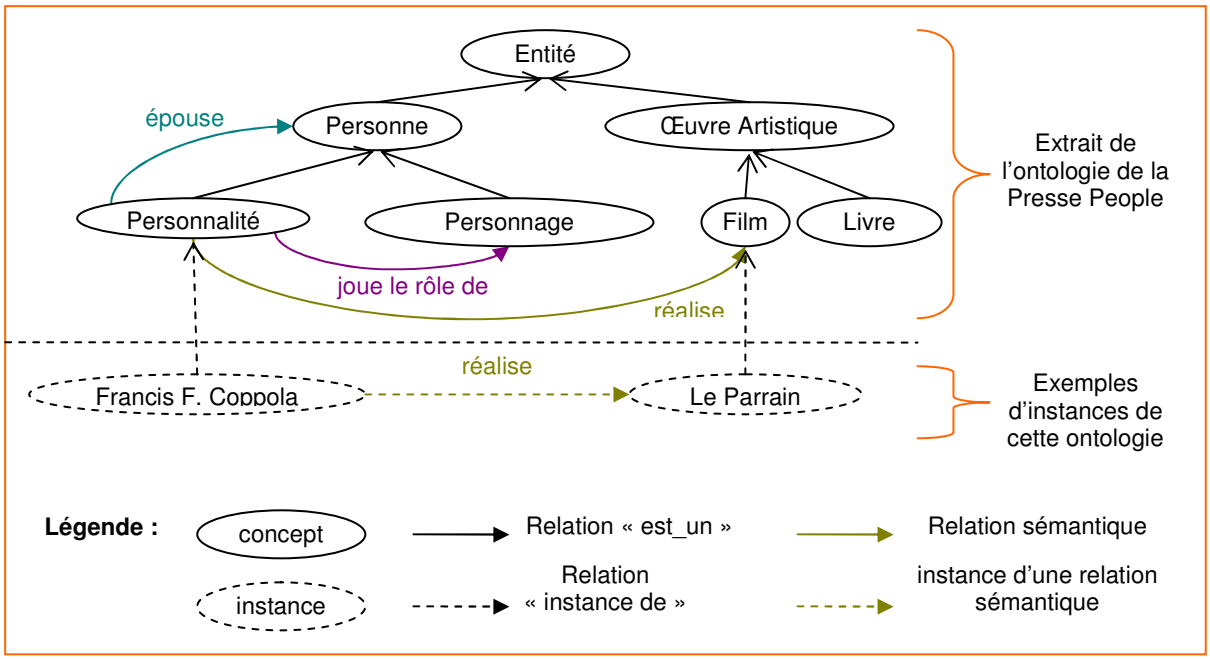


Figure 8. Exemple d'une ontologie dans le domaine de la presse « People »

Les **instances** de concepts (aussi nommés individus) ne font pas à proprement parler partie de l'ontologie, mais plutôt de la base de connaissance [HAN 05]. En effet, ces dernières permettent de stocker les instances des concepts, mais aussi les instances de relations et les valeurs des propriétés en fonction des contraintes imposées par l'ontologie. Dans le monde de l'ingénierie des connaissances, par référence aux logiques de description, on parle aussi de la **Terminological-Box** (ou T-Box) pour l'ontologie et de la **Assertion-Box** (ou A-Box) pour la base de connaissance. Dans l'exemple de la Figure 8, « Francis F. Coppola » est une instance du concept « Personnalité » et une relation sémantique « réalise » est instanciée entre cette instance et celle du concept « Film », i.e. « Le Parrain». La base de connaissance contiendra donc les informations *Personnalité(Francis F.*

Coppola), *Film(Le Parrain)* et *réalise(Francis F. Coppola, Le Parrain)*. L'action de définir et d'instancier une base de connaissance a été récemment appelée « peuplement d'ontologie »[11]. Comme nous l'avons mentionné précédemment, les annotations sémantiques permettent non seulement d'enrichir le contenu d'un document avec des métadonnées sémantiques, mais également de peupler la base de connaissance avec les nouvelles instances repérées dans ce contenu [HAN 05].

La définition d'ontologies plus complexes fait référence aux théories de la Logique, et notamment celle des logiques de description [LAU 07]. Ces ontologies comportent en plus un ensemble d'axiomes et de règles d'inférence. Ensemble, ils sont utilisés pour contraindre l'ontologie, vérifier sa validité et raisonner sur le domaine représenté par l'ontologie. Ils permettent aussi de « définir » [CHA 04] de nouveaux concepts par conjonction de plusieurs concepts (par exemple « Personnage de Film » est une conjonction des concepts « Personnage » et « Film ») ou par restriction d'un concept à l'aide de contraintes (« Personnage féminin français ») [LAU 07]. La plupart des recherches en ingénierie des connaissances aspirent à construire des ontologies de ce niveau afin d'exprimer le plus formellement possible la sémantique du domaine.

Jusqu'à présent, nous avons beaucoup parlé des ontologies de domaine. Mais il faut savoir que Studer et al. [STU 98] distinguent quatre autres types d'ontologie, à savoir :

- Les ontologies de représentation n'appartiennent à aucun domaine, mais définissent et « organisent les primitives de la théorie logique » [CHA 02] spécifiant ce qui doit être représenté. Par exemple, la Frame Ontology[12] définit des concepts pour exprimer de la connaissance dans un environnement implémentant les langages de Frame.
- Les ontologies génériques sont suffisamment abstraites pour être valides quel que soit le domaine étudié. Elles permettent par exemple de formaliser les aspects temporels ou spatiaux des objets du monde réel, cf. l'ontologie Upper Cyc[13].
- Les ontologies de tâche (ou de méthode de résolution de problème) fournissent les concepts modélisant une activité générique [HAN 05] ou explicitant le « rôle joué par chaque concept dans le raisonnement » [CHA 02] pour la résolution de problème.
- Les ontologies d'application sont une double spécialisation [CHA 02] d'une ontologie de domaine par rapport à une ontologie de tâche permettant ainsi de modéliser une activité spécifique dans un domaine donné.

Dans la suite de ce mémoire, nous continuerons de nous intéresser uniquement aux ontologies de domaine, car ce sont celles les plus couramment utilisées, notamment dans le cadre du Web Sémantique mais plus particulièrement dans notre périmètre lié à l'annotation sémantique.

[11] Cf. le workshop « Ontology Learning & Population » à la conference ECAI 2004: http://olp.dfki.de/ecai04/cfp.htm
[12] http://www-ksl.stanford.edu/people/brauch/demo/frame-ontology/
[13] http://www.cyc.com/cycdoc/upperont-diagram.html

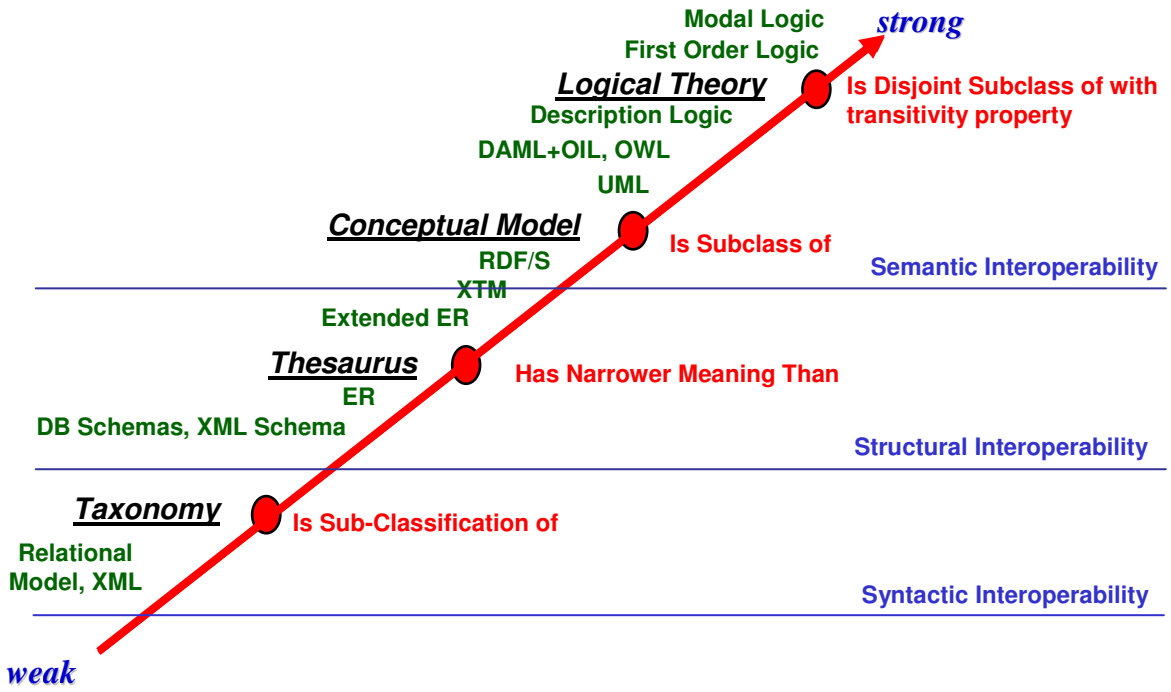


Figure 9. **Le continuum RTO, issu d'un tutoriel de D. Riañyo**[14]

Nous venons de présenter différentes ressources terminologiques ou ontologiques : d'abord les taxonomies (conceptualisation organisée hiérarchiquement), puis les thesaurus (les termes d'un vocabulaire contrôlé et leurs relations lexicales) et enfin les ontologies (conceptualisation plus riche, plus complexe et permettant la mise en place de mécanismes d'inférence pour la découverte et l'exploitation de la connaissance d'un domaine. La Figure 9 présente une vue d'ensemble de ces différentes ressources les unes par rapport aux autres, ainsi que les langages correspondants (dont certains seront présentés à la section 1.3), sur un axe symbolisant une représentation formelle de plus en plus forte.

1.2.2 Les RTO et l'annotation sémantique

A présent, nous allons étudier comment les ressources terminologiques ou ontologiques peuvent être utilisées pour annoter sémantiquement un document. Nous avons vu que les RTO n'offrent pas toutes le même niveau de représentation de la connaissance : les taxonomies étant moins expressives que les thesaurus, eux-mêmes moins formels et précis qu'une ontologie. Le choix d'une RTO va donc influencer la modélisation des annotations sémantiques possibles.

Bien que nous ayons vu que les thesaurus ne constituaient pas une ontologie en soi, leur capacité à exprimer un vocabulaire contrôlé ainsi qu'un ensemble de relations lexicales entre les termes de ce vocabulaire permet de constituer un premier niveau d'annotation sémantique. Dans ce cas, les annotations consistent en de simples pointeurs vers les termes du thesaurus, aussi appelés **descripteurs**. Elles fonctionnent alors comme des entrées d'index améliorant les résultats des

[14] www.etse.urv.es/~drianyo/teaching/KM.pdf

moteurs de recherche qui utilisent ce genre d'annotation. En exemple, nous avons déjà cité le Medical Subject Headings[15] (MeSH), on pourrait aussi ajouter le Thesaurus of Geographical Names[16] (TGN), etc.

Les ontologies de domaine sont de plus en plus utilisées pour annoter des ressources documentaires. En fait, une même ressource peut être annotée par différentes ontologies de domaine en même temps, offrant ainsi différents points de vue sur un même contenu. Non seulement, les annotations sémantiques créées à partir d'une ontologie améliorent la recherche d'information, car elles peuvent contribuer à faire jouer des mécanismes d'inférence et de raisonnement, mais elles peuvent être combinées avec l'enrichissement d'une base de connaissance du domaine. En effet, la connaissance du domaine contenue dans le document peut être extraite à partir des annotations sémantiques afin d'être stockée et exploitée dans une base de connaissance contrainte par l'ontologie du domaine.

En fait, les thesaurus et les ontologies offrent des accès aux contenus d'après des angles différents : celui du vocabulaire pour les thesaurus et celui de la conceptualisation d'un domaine pour l'ontologie. Ils ont donc des rôles complémentaires à jouer dans l'annotation sémantique comme nous allons le voir dans l'exemple suivant. Nous utiliserons d'ailleurs cet exemple tout au long de notre mémoire comme un fil directeur. Il s'agit d'un extrait de l'article « Le Clan Coppola » paru le 30/02/2003 dans le magasine ELLE, au sujet de la famille de Francis Ford Coppola, cf. Figure 10.

Annotons tout d'abord l'extrait avec un thesaurus géographique comme le TGN. Dans la Figure 10, ces annotations correspondent aux mots surlignés en vert (foncé) :

- **Detroit** est une référence au descripteur ayant l'ID 7013547 dans le TGN et qui fait référence à la ville de Détroit dans l'état du Michigan.
- **Michigan** est une référence au descripteur ayant l'ID 7007520 et qui fait référence à l'état Michigan aux Etats-Unis
- **New York** est une référence au descripteur ayant l'ID 7007567 dans le TGN et qui fait référence à la ville de New York aux Etats-Unis.

LE CLAN COPPOLA

Francis Coppola naît le 7 avril 1939 à Detroit, dans le Michigan. Il est le deuxième des trois enfants de Carmine et Italia Coppola. Son père, originaire de New York, est chef d'orchestre. Francis fera appel à lui pour composer la musique du « Parrain », en 1972. Sa mère, elle, est la fille du célèbre compositeur napolitain Francesco Pennino, auteur de l'opéra « Senza Mamma », dont un extrait figure dans « Le Parrain II ». Comédienne, elle joue dans plusieurs films de Vittorio De Sica, avant d'embrasser la carrière de « mamma ». Francis a de qui tenir ! […]

Figure 10. Extrait de l'article « Le Clan Coppola » paru dans le magasine ELLE, le 30/02/2003.

Puis, supposons qu'il existe une ontologie dans le domaine de la « Presse People » que nous pourrions utiliser pour annoter l'extrait ci-dessus. Cette ontologie contiendrait les concepts « Personnalité » et « Œuvre Artistique », des attributs comme « lieu de naissance », « date de

[15] http://www.nlm.nih.gov/mesh/meshhome.html
[16] http://www.getty.edu/research/tools/vocabulary/tgn/index.html

naissance », « date de création », « profession », etc. et des relations telles que « est créé par(Œuvre Artistique, Personnalité) » ou « a lien parenté avec(Personnalité, Personnalité) ». Les documents seraient alors annotés en fonction de ces éléments, à savoir :

- Par les instances de concepts. Par exemple, une annotation sera créée avec « Francis Ford Coppola » qui représente une instance du concept « Personnalité » et avec « Le Parrain », instance du concept « Œuvre Artistique ».
- Par les valeurs d'attributs. Par exemple, « 7 avril 1939 » peut être la valeur de l'attribut « date de naissance » qui sera attaché à l'instance « Francis Ford Coppola » de notre annotation.
- Par les instances de relations. Par exemple, l'instance « Le Parrain » du concept Œuvre Artistique et l'instance « Francis Ford Coppola » du concept Personnalité peuvent être connectées par la relation « est-créée-par ». Le document sera annoté avec cette instance de relation « est créée par (Le Parrain, Francis Ford Coppola) ».

Par cet exemple, nous voyons bien comment les thesaurus et autres vocabulaires contrôlés peuvent être utilisés pour fournir un ensemble de termes agréés dans des domaines particuliers comme les lieux géographiques. Ce premier niveau d'annotation peut ensuite être combiné avec les annotations sémantiques basées sur l'ontologie du domaine qui décrivent le contenu du document en fonction des concepts, attributs et relations modélisés dans cette ontologie. Plus une ontologie est précise dans sa modélisation du domaine, plus les annotations seront contraintes en fonction des restrictions imposées sur leurs valeurs autorisées, sur leurs relations, etc. Par conséquent, bien que de nature différente, ces approches sont complémentaires les unes par rapport aux autres. Il est donc généralement intéressant d'annoter une ressource documentaire par ces différentes approches afin d'offrir des angles différents aux services et autres agents logiciels exploitant les annotations créées. Cela permet notamment de fournir des accès différents vers les mêmes ressources documentaires en recherche d'information (recherche par mots-clefs, par extension sémantique, recherche sémantique, etc.).

Nous venons de présenter la notion de RTO, depuis les taxonomies jusqu'aux ontologies en passant par les thesaurus, et nous avons montré comment elles pouvaient être employées pour l'annotation sémantique. A présent, nous allons montrer quels langages informatiques peuvent être utilisés pour représenter la connaissance d'un domaine, l'instrumenter dans un agent logiciel et ainsi créer les annotations sémantiques qui en découlent.

1.3 Les Langages de l'annotation sémantique

Plusieurs langages de représentation de la connaissance, et d'ontologies, ont été définis par les chercheurs du domaine de l'Intelligence Artificielle. Citons notamment OntoLingua [6], LOOM [16], OCML [17] et F-Logic [12]. Ces langages ont été conçus à partir de différents modèles issus de théories de la Logique comme les logiques de description, les prédicats du premier et second ordre, les frames, etc. Mais ils ne peuvent être utilisés tels quels dans le Web Sémantique sans une certaine

adaptation, notamment syntaxique [BAG 04]. Outre HTML, de nouveaux langages ont été créés pour le Web comme XML et RDF. Puis, avec la perspective naissante du Web Sémantique, ces deux ensembles ont été fusionnés pour donner naissance à de nouveaux langages pour la spécification d'ontologies orientées Web, comme RDF Schéma, DAML+OIL et OWL. Mais il est évident que le Web Sémantique ne peut exister sans une standardisation de ses langages [BAG 04], tel que ce fut le cas pour HTML dans le Web actuel. C'est pourquoi le World Wide Web Consortium (W3C) a participé activement à l'élaboration de ces nouveaux métalangages pour le Web par la publication des recommandations autour d'XML, RDF(S) et OWL. D'autres organismes, comme le Defense Advanced Research Projects Agency (DARPA) ou l'International Standard Organisation (ISO) ont également sponsorisé des projets d'envergure autour de ces métalangages, respectivement DAML et XML Topic Maps. Nous allons étudier quelques uns de ces langages dans la perspective de l'annotation sémantique, à commencer par les précurseurs.

1.3.1 Les précurseurs

- **HTML-A**

L'initiative (KA)² proposait d'utiliser une extension d'HTML, appelée HTML-A, pour insérer des annotations sémantiques dans les pages Web. Cette approche ne spécifiait pas le langage d'implémentation de l'ontologie de référence. Par contre, comme décrit dans [BEN 99], les agents logiciels du Web devaient ensuite savoir interpréter cette extension pour l'exploiter correctement, tel l'outil OntoAnnotate [STA 01a].

```
<html>
<head><Title>Le Clan coppola</Title>
<A ONTO="Personnalité:FFCoppola"/>
</head>
<body>
Francis Coppola naît le <A ONTO="Personnalité[dateNaissance=body]">7 avril 1939</A> à <A
ONTO="Personnalité[lieuNaissance=body]">Detroit</A>, dans le <A
ONTO="Personnalité[lieuNaissance=body]">Michigan</A>.
</body>
</html>
```

Figure 11. Exemple d'une annotation sémantique en HTML-A

La Figure 11 présente un exemple d'annotation sémantique proposée par l'initiative (KA)². L'entête de cette page Web déclare le concept et l'instance de l'ontologie de référence, « Personnalité:FFCoppola ». Les annotations sémantiques qui vont être insérées dans le corps du document feront toutes références à l'instance « FFCoppola » du concept « Personnalité ». Parmi ces annotations sémantiques, deux attributs, « dateNaissance » et « lieuNaissance », encadrent trois éléments textuels correspondant à leur valeur.

- **SHOE : Simple HTML Ontology Extension**

L'approche de SHOE [HEF 01] était similaire à celle de l'initiative (KA)² : créer une extension d'HTML ayant pour objectif l'insertion d'annotations sémantiques pour la description des ressources du Web.

Au lieu d'utiliser l'attribut **ONTO** dans l'élément **A**, SHOE proposait d'utiliser un ensemble d'éléments prédéfinis tels **INSTANCE**, **CATEGORY**, **RELATION**, etc. Ces éléments étaient directement insérés dans le code HTML de la page Web annotée.

Les annotations devaient ensuite être interprétables par des agents Web. L'objectif de ce langage était de permettre aux agents de glaner de la connaissance au sujet de pages Web afin d'améliorer les mécanismes de recherche d'information et de fouille de données. Contrairement à l'initiative (KA)², une ontologie correspondant à une simple hiérarchie « classe/sous-classe » devait d'abord être implémentée en SHOE. Outre la définition de cette classification, les ontologies SHOE décrivaient aussi les relations entre les classes, appelées catégories en SHOE, et un ensemble de règles d'inférence simplifiées.

La Figure 12 montre un exemple d'annotation d'une version HTML de SHOE. Contrairement à HTML-A, les concepts de l'ontologie de référence sont directement déclarés dans le corps du document et les annotations sont débarquées vis-à-vis du texte originel.

```
<html>
<head><Title>Le Clan coppola</Title></head>
<body>
Francis Coppola naît le 7 avril 1939 à Detroit, dans le Michigan.

<INSTANCE KEY="FFCoppola">
<USE-ONTOLOGY ID="People-Ontology" URL="http://www.elle.com/SHOE/people.html"
VERSION="1.0" PREFIX="people">
<CATEGORY NAME="people.FFCoppola">
<RELATION NAME="people.dateNaissance">
        <ARG POS=1 VALUE="FFCoppola">
        <ARG POS=2 VALUE="7 avril 1939">
</RELATION>
<RELATION NAME="people.lieuNaissance">
        <ARG POS=1 VALUE="FFCoppola">
        <ARG POS=2 VALUE="Detroit">
</RELATION>
</INSTANCE>

</body>
</html>
```

Figure 12. Exemple d'une annotation sémantique en SHOE

Aujourd'hui, très peu d'outils savent créer et traiter ces langages d'annotations, cf. §1.4.2. Ces deux approches ont été abandonnées au profit de langages plus formels et recommandés par le W3C, présentés dans la suite de ce chapitre.

1.3.2 La pyramide des langages du Web Sémantique

L'architecture du Web Sémantique se compose d'un ensemble de langages, généralement représentés sous la forme d'une pyramide. Chaque niveau repose sur les résultats définis au niveau inférieur, c'est-à-dire que chaque niveau est progressivement plus spécialisé et plus complexe que son niveau précédent. D'autre part, tout niveau est indépendant des niveaux supérieurs afin qu'il

puisse être développé et rendu opérationnel de manière autonome par rapport aux développements des niveaux supérieurs. Cette pyramide des langages, cf. Figure 13, a initialement été présentée par Sir Tim Berners-Lee [LEE 2000] lors de la conférence XML en 2000[17].

A la base, les ressources sont encodées au format Unicode et utilisent le système d'adressage par Universal Resource Indicator (URI) servant aussi d'identifiant sur le Web. Ces ressources doivent être conformes à un format XML qui utilisera un espace de nom particulier et qui sera valide par rapport à une grammaire définie dans une DTD ou un XML Schéma. Puis vient RDF qui permet de décrire les métadonnées des documents sous la forme de triplets comme spécifié par RDF Schéma. Il permet d'implémenter un premier niveau d'ontologies, relativement simples. Afin de représenter des ontologies plus complexes, le langage OWL qui définit des classes, attributs, relations et axiomes peut être combiné à un langage de règles permettant de raisonner sur les ressources et d'inférer de la nouvelle connaissance. Le niveau Logique (Logic) est utilisé pour établir la cohérence des annotations et pouvoir inférer des conclusions non explicitement énoncées. Le niveau Preuve (Proof) pourrait fournir les moyens pour tracer et expliciter les différentes étapes du raisonnement logique afin de pouvoir lui accorder un niveau de confiance. Le dernier niveau (Trust) a pour objectif d'authentifier l'identité et la véracité des données et services disponibles sur le Web Sémantique.

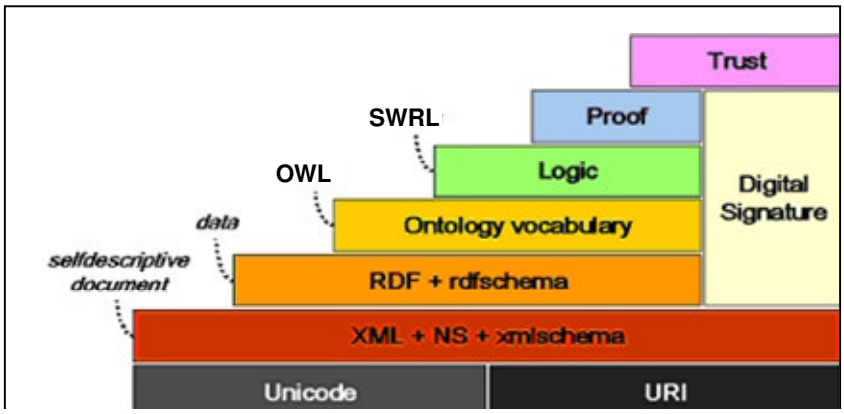


Figure 13. Pyramide des langages du Web Sémantique en 2005[18]

Aujourd'hui, XML[19] et XML Schéma[20] sont largement répandus dans les applications orientées Web mais ils restent limités car ils ne disposent pas d'une sémantique formelle [BAG 04]. RDF, RDF Schéma et OWL ont été récemment publiés comme des recommandations du W3C. Ils s'inspirent notamment des réseaux sémantiques, des graphes conceptuels [SOW 01] et des logiques de description qui permettent d'exprimer la connaissance d'un domaine. Nous allons les présenter plus en détail dans la suite de cette section. Les autres niveaux supérieurs sont progressivement étudiés et développés. Dernièrement, un langage pour le niveau Logique, le Semantic Web Rule Language

[17] XML 2000 presentation: http://www.w3.org/2000/talks/1206-xml2k-tbl/slide1-0.html
[18] http://www.sebastiankruk.com/storage/presentation/elearning_on_sw20/img17.jpg
[19] http://www.w3.org/XML/
[20] http://www.w3.org/XML/Schema

(SWRL)[21], a été soumis au W3C pour devenir une recommandation au même titre que les autres langages ci-dessus.

1.3.2.1 RDF : Resource Description Framework

Le W3C a publié la première recommandation de RDF[22] en 1999 pour répondre aux besoins du Sémantique Web. Sa dernière version en date est celle de février 2004[23]. L'objectif premier de RDF est la description de « ressources » [BAG 04]. Un document RDF peut contenir plusieurs descriptions. Une **description** correspond à un ensemble d'**énoncés** (ou « statements ») au sujet d'une **ressource**. Un énoncé RDF est aussi appelé **triplet** car il est composé de trois éléments, sujet-prédicat-objet, où :

1) *le sujet* représente la ressource décrite, i.e. tout document accessible sur le Web comme les pages HTML, les documents textuels (PDF, Ms Word) ou multimédias (images, vidéo), etc., mais aussi tout objet, abstrait ou non, du monde réel. Les ressources sont nommées en utilisant une URI.

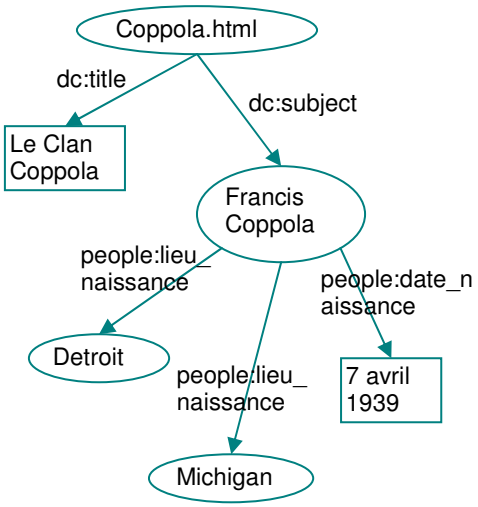
2) *le prédicat* représente la propriété descriptive, i.e. une caractéristique spécifique, un attribut ou une relation, utilisée pour décrire une ressource.

3) *l'objet* représente la valeur de cette propriété, soit une valeur littérale, comme un nombre entier ou une chaîne de caractère, soit une autre ressource accessible par son URI. Par contre, une valeur littérale ne peut en aucun cas être le sujet d'un énoncé.

Un triplet peut s'écrire « prédicat(sujet, objet) » ou encore « propriété(sujet, valeur) ». Par exemple, la phrase « Francis Coppola est né à Détroit » sera traduite par le triplet « né(Francis Coppola, Détroit) ». Ici, l'objet « Détroit » correspond soit à une simple chaîne de caractères, soit une ressource identifiée par le site Web de la ville. D'autres notations peuvent être utilisées pour afficher des données RDF. En particulier, les graphes étiquetés orientés sont parfaitement adaptés à la représentation des énoncés RDF puisque le modèle RDF a été influencé par les réseaux sémantiques [BAG 04]. Dans ces graphes, un nœud représente un sujet ou un objet et l'arc un prédicat dont l'origine est le sujet et la destination l'objet de l'énoncé (cf. Figure 14). RDF peut aussi être exprimé en XML afin notamment d'échanger ses données avec les agents logiciels du Web ou autres.

A propos du langage, un document RDF est délimité par l'élément `rdf:RDF` qui comporte un ou plusieurs éléments `rdf:Description` pour chacune des descriptions de ressources comprises dans le document. Chaque description comprend un attribut `rdf:about` qui pointe vers l'URI de la ressource à décrire et un à plusieurs éléments représentant chacun un prédicat. Lorsqu'un prédicat a pour valeur une autre ressource, l'attribut `rdf:resource` pointera vers son URI, comme c'est le cas pour le prédicat « `dc:subject` » dans la figure ci-dessous. Par contre, si c'est un littéral, la valeur est insérée dans l'élément, cf. « `dc:title` » dans la Figure 14 .

[21] http://www.w3.org/Submission/SWRL/
[22] http://www.w3.org/RDF/
[23] http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/




```
<xml "version="1.0" encoding="UTF-8">
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:people="http://www.mondeca.com/onto#people">

<rdf:Description
   rdf:about="http://www.elle.fr/Coppola.html">
   <dc:title>Le Clan Coppola</dc:title>
   <dc:subject rdf:resource="#Francis Coppola"/>
</rdf:Description

<rdf:Description rdf:about="Francis Coppola">
   <people:date_naissance>7 avril 1939
     </people:date_naissance>
   <people:lieu_naissance rdf:resource="#Detroit/>
   <people:lieu_naissance rdf:resource="#Michigan/>
</rdf:Description>

</rdf:RDF>
</xml>
```

Figure 14. Exemple d'annotation sémantique en RDF (notation graphique à gauche et XML à droite)

Les autres caractéristiques de RDF permettent la composition des énoncés en structures plus complexes comme le groupement de sujets et/ou d'objets en listes énumérées ou bien la réification d'énoncés, i.e. la création de nouveaux énoncés à partir d'énoncés existants.

RDF répond aux besoins de la plupart des outils d'annotation. En effet, les documents RDF sont des documents XML valides, leur modélisation sous forme de réseau sémantique apporte une flexibilité nécessaire et il est possible de réutiliser des énoncés existants pour composer des documents RDF plus complexes. Par contre, RDF ne fournit pas de mécanisme de contrainte de classes ou de types pour les différentes parties du triplet. Il n'est donc pas assez puissant pour représenter de vraies ontologies avec un système de raisonnement approprié.

1.3.2.2 RDFS : Resource Description Framework Schema

Comme son nom l'indique, RDFS[24] est un langage utilisé pour la définition de schémas RDF. Son modèle de données est basé sur celui des Frames. Alors qu'RDF exprime les relations sémantiques au niveau des instances sous la forme de triplets, RDFS exprime donc les relations au niveau des classes et des propriétés (les prédicats RDF), contraignant ainsi les instances possibles dans les triplets RDF [BAG 04]. Par conséquent, RDFS marque une étape de plus vers la conception d'un formalisme de représentation plus riche et introduit les primitives de bases de la modélisation ontologique pour le Web Sémantique.

[24] http://www.w3.org/TR/rdf-schema/

```
<xml ”version=”1.0” encoding=”UTF-8” xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”>

<rdf:Description rdf:about="http://www.elle.fr/Coppola.html">
   <rdf:type rdf:about="#Article"/>
   <rdfs:label>Le Clan Coppola</rdfs:label>
   <dc:subject rdf:resourceRef="#FFCoppola"/>
</rdf:Description>

<rdfs:Class rdf:ID="#Lieu">
<rdfs:Class rdf:ID="#Personne">
<rdfs:Class rdf:ID="#Personnalité">
   <rdfs:subClassOf rdf:resource ="#Personne"/>
</rdfs:Class>

<rdf:Property rdf:about="lieu_naissance">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range rdf:resource="#Lieu"/>
</rdf:Property>
<rdf:Property rdf:about="date_naissance">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</rdf:Property>

<Personnalité rdf:about="FFCoppola">
   <rdfs:label>Francis Coppola</rdfs:label>
   <lieu_naissance rdf:resource="#Detroit"/>
   <lieu_naissance rdf:resource="#Michigan"/>
   <date_naissance>7 avril 1939</date_naissance>
</Personnalité>
</xml>
```

Figure 15. Exemple d'annotation sémantique basée sur un schéma RDFS

RDFS est construit au dessus de RDF, c'est-à-dire que ce langage réutilise la syntaxe des triplets RDF et l'étend avec de nouveaux éléments pour définir les classes, `rdfs:class`, et les propriétés, `rdfs:property`. L'élément `rdfs:subClassOf` permet de spécifier la taxonomie des classes. Dans la Figure 15, « Personne » et « Personnalité » sont définies comme étant des classes de l'ontologie, « Personnalité » étant une sous-classe de « Personne ». Enfin la ressource « Francis Coppola » est déclarée comme étant une instance de la classe « Personnalité ». Les hiérarchies de propriétés sont définies par l'élément `rdfs:subPropertyOf`. Les définitions de propriétés sont restreintes par deux contraintes :

1) le domaine de la propriété, représenté par `rdfs:domain` et indiquant la classe à laquelle cette propriété s'applique et,

2) la portée de la propriété, représentée par `rdfs:range` et indiquant la classe dont les instances seront les valeurs autorisées pour cette propriété.

Ainsi, dans l'exemple de la Figure 15, nous définissons la propriété « Lieu_Naissance » avec la classe « Personne » comme domaine et « Lieu » comme sa portée. Ceci signifie donc que cette propriété s'applique à toute instance de la classe « Personne » et que sa valeur doit être une instance de la classe « Lieu ».

Comme RDF et RDFS sont des recommandations du W3C, ils ont été largement acceptés pour l'annotation des ressources du Web. Les Schémas RDF apportent des caractéristiques importantes à l'annotation sémantique en RDF par la définition d'une ontologie relativement simple à travers une taxonomie de classes, de leurs propriétés ainsi que la restriction du domaine et de la portée des propriétés. Ces aspects sont particulièrement importants pour l'annotation sémantique car ils offrent assez d'expressivité pour mettre en place un premier niveau sémantique. Par contre, il n'est pas suffisant de définir un ensemble de classes et de propriétés RDFS pour constituer une véritable ontologie formelle. Parmi les caractéristiques manquantes, citons la restriction des cardinalités par des quantifieurs, la combinaison de classes par des opérateurs ensemblistes, l'équivalence entre classes, etc.

1.3.2.3 OWL : Web Ontology Language

OWL[25], recommandé par le W3C en février 2004, est le plus expressif des langages ontologiques pour le Web [BAG 04]. La conception d'OWL a bénéficié de plusieurs générations de langages de représentation des connaissances, d'une base théorique solide en logique et d'une volonté de la part de ses concepteurs pour créer un langage approprié à une utilisation dans le cadre du Web Sémantique. En fait, OWL est issu des travaux autour du langage DAML+OIL[26], lui-même fusion de deux projets l'un européen, OIL, et l'autre américain, DAML. La plupart des chercheurs ayant participé à l'élaboration du langage DAML+OIL ont ensuite travaillé à OWL.

Le langage OWL fournit des mécanismes pour créer tous les composants d'une ontologie : classes, instances, propriétés et axiomes. OWL repose également sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS. Comme en RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés. Par contre, en OWL, on distingue :

1) les propriétés objet (object property), i.e. les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.

2) les propriétés type de données (datatype property), i.e. les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,…). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale.

Les axiomes fournissent de l'information au sujet des classes et des propriétés, spécifiant par exemple l'équivalence entre deux classes.

OWL se compose de trois sous-langages : OWL Lite, OWL DL et OWL Full [BAG 04]. OWL Lite est le sous-langage offrant le maximum de restrictions. L'objectif principal est d'aider les utilisateurs à prendre en main la syntaxe d'OWL, de leur permettre d'implémenter des agents logiciels capables de raisonner facilement puis d'évoluer vers des utilisations plus complexes avec les autres sous-langages. OWL Lite permet de définir les classes et propriétés d'une ontologie. Une sous-classe dans

[25] http://www.w3.org/2004/OWL/
[26] http://www.daml.org/2001/03/daml+oil-index.html

OWL Lite peut avoir plusieurs classes parentes, elle héritera alors des propriétés de chacune de ses classes parentes. Les propriétés sont définies par `owl:objectProperty` pour les relations entre classes ou par `owl:datatypeProperty` pour les attributs (cf. Figure 16). L'élément `owl:subProperty` permet de spécifier une hiérarchie de propriétés et `owl:domain`/`owl:range` précisent respectivement le domaine et la portée d'une propriété donnée, de la même manière qu'avec RDFS. Par contre, OWL Lite ajoute la possibilité de contraindre les propriétés en utilisant des quantifieurs, `owl:allValuesFrom` et `owl:someValuesFrom`, et des cardinalités (uniquement de valeur 0 ou 1), `owl:minCardinality` et `owl:maxCardinality`. Enfin, OWL Lite permet de caractériser une propriété grâce aux éléments `owl:inverseOf`, `owl:TransitiveProperty` et `owl:SymmetricProperty`. Comme on peut le constater, OWL Lite fournit donc les moyens de modéliser un premier niveau d'ontologie permettant un raisonnement simple.

```
<xml "version="1.0" encoding="UTF-8" xmlns:owl="http://www.w3.org/2002/07/owl#">
<rdf:RDF>
<owl:Ontology rdf:about="http://www.mondeca.com/onto#people">
   <dc:title>People</dc:title>
</owl:Ontology>

<owl:Class rdf:ID="Lieu">
<owl:Class rdf:ID="Personne">
<ow:Class rdf:ID="Personnalité">
   <rdfs:subClassOf rdf:resource ="#Personne"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="lieu_naissance">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range rdf:resource="#Lieu"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="date_naissance">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</owl:DatatypeProperty>

<owlThing rdf:ID="FFCoppola">
   <rdf:type rdf :resource="#Personnalité"/>
   <rdfs:label>Francis Coppola</rdfs:label>
   <lieu_naissance rdf:resource="#Detroit"/>
   <lieu_naissance rdf:resource="#Michigan"/>
   <date_naissance>7 avril 1939</date_naissance>
</Personnalité>

<rdf:Description rdf:about="http://www.elle.fr/Coppola.html">
   <rdf:type rdf:about="#Article"/>
   <rdfs:label>Le Clan Coppola</rdfs:label>
   <dc:subject rdf:resourceRef="#FFCoppola"/>
</rdf:Description>

</rdf:RDF>
</xml>
```

Figure 16. Exemple d'annotation sémantique en OW Lite

OWL DL étend le langage OWL Lite et s'appuie plus fortement encore sur la théorie des Logiques de Description comme son nom l'indique. Son objectif est de modéliser des ontologies plus complexes que celles autorisées par OWL Lite tout en assurant la complétude des calculs de raisonnement par

les systèmes logiques. Pour citer la spécification d'OWL DL[27], ce langage permet d'obtenir un « *maximum d'expressivité tout en gardant la complétude de calcul (toutes les conclusions sont garanties calculables) et de décision (tous les calculs finiront dans un temps fini)* ». Les logiques de description permettent de redéfinir certaines restrictions, notamment les contraintes de cardinalité dont les valeurs ne sont plus restreintes à 0 ou 1. D'autre part, les classes peuvent être définies par des valeurs de propriétés spécifiques comme la propriété `owl:hasValue` ou par l'utilisation d'opérateurs booléens comme `owl:unionOf`, `owl:intersectionOf` et `owl:complementOf`. Les classes peuvent aussi être construites par des énumérations par l'utilisation de l'élément `owl:oneOf` ou par opposition à des classes existantes par l'utilisation de l'élément `owl:disjointWith`.

OWL Full étend à son tour OWL DL en permettant aux classes d'être traitées à la fois comme des collections et des instances. En effet, dans OWL Lite et OWL DL, une restriction particulièrement importante stipule qu'une classe ne peut être une instance d'une autre classe. OWL Full est assez complexe pour donner du fil à retordre à un raisonneur et il se peut que les calculs n'aboutissent jamais à une conclusion. C'est pourquoi OWL Lite et OWL DL ont donc été conçus pour être moins puissant et permettre aux raisonneurs d'aboutir à une conclusion quelles que soient les assertions.

1.3.3 Une alternative, les Topics Maps

Outre les développements menés par le W3C, d'autres initiatives comme DAML+OIL ou les Topics Maps sont nées pour tenter de mettre au point des langages pour la modélisation d'ontologies ou la création de bases de connaissances. Parmi ces initiatives, nous retiendrons celle des Topic Maps, concurrent du langage RDF, car ce langage est utilisé par l'outil de représentation des connaissances qui compose notre solution présentée au chapitre 6 de ce mémoire.

Originellement développés pour permettre la construction de listes, de glossaires, de thesaurus et de tables de matières dans les systèmes d'indexation automatique, les Topic Maps ont su évoluer pour s'adapter au monde du Web. Les Topic Maps permettent de caractériser sémantiquement le contenu d'un document en fonction des sujets, ou **topics**, dont parlent ces documents. En plus d'un index, ces sujets peuvent être reliés entre eux, facilitant ainsi la navigation dans le contenu des documents. Le standard international pour les Topic Maps, ISO/IEC 13250[28], a été créé pour définir l'organisation des sujets dans une **topic map**, i.e. une « carte de sujets » ou encore « carte topique » [BAG 04]. Ce standard spécifie également le XML Topic Map[29] (XTM), i.e. la sérialisation des Topics Maps en XML, servant à l'échange et à l'utilisation de topic maps par des agents logiciels.

Dans les Topic Maps, un **concept** appartient au monde du discours, i.e. une idée ou un objet physique pouvant être pensé et évoqué, alors qu'un **topic** correspond à une représentation du concept sous la forme d'une donnée stockée dans la mémoire d'un ordinateur. Un attribut

[27] http://www.w3.org/TR/owl-ref/#OWLDL
[28] http://www.isotopicmaps.org/rm4tm/
[29] http://www.topicmaps.org/xtm/index.html

`instanceOf` spécifie la classe d'appartenance du `topic` et, comme dans OWL Full, une classe peut être à la fois sous-classe et instance d'autres classes. Un topic possède toujours au moins un nom, son `baseName`, un libellé plutôt qu'un identifiant. Il peut avoir d'autres noms comme c'est le cas dans le monde réel où les objets peuvent être dénommés de différentes manières (le `displayName` pour l'affichage et le `sortName` pour la recherche). Une propriété spécifique d'un topic, comme un numéro de page, est appelé une `occurrence`. Elle fonctionne un peu comme un triplet RDF : elle est attachée à un topic (la ressource) et a pour valeur, soit un lien URI vers une ressource, soit une valeur littérale sous la forme d'une chaîne de caractères. Par ailleurs, une `association` permet de représenter des relations plus complexes entre topics, notamment n-aires. C'est une des particularités essentielles du standard qui s'oppose à la notion de prédicat dans RDF. En effet, un prédicat RDF ne peut relier que deux ressources au plus dans un même énoncé (relations binaires uniquement). Lorsqu'un topic est `membre` d'une association, c'est qu'il joue un `rôle` bien spécifique dans cette association. Autant de topics peuvent jouer autant de rôles que nécessaire dans une même association et afin de les identifier, ces rôles sont typés.

Tout comme RDF, les Topics Maps sont représentées par un réseau sémantique. Les nœuds correspondent à des topics (cercle bleu dans la Figure 17) ou à des associations (rectangle orange dans la Figure 17). Les arcs représentent les rôles joués par les topics dans les associations. Contrairement à RDF, il s'agit d'un graphe non orienté : les rôles sont explicites et par conséquent les arcs sont étiquetés en fonction du type du rôle. Dans l'association « A pour sujet » de la Figure 17, ces rôles sont « article_annoté » joué par le topic « Clan_Coppola » et « sujet_article » joué par le topic « Francis Ford Coppola ». Les occurrences peuvent aussi être représentées dans ce réseau sémantique : l'arc représente alors le type de propriété, comme « date naissance », et relie un topic à la valeur de cette propriété, par exemple « Francis ford Coppola » à la valeur « 7 avril 1939 ».

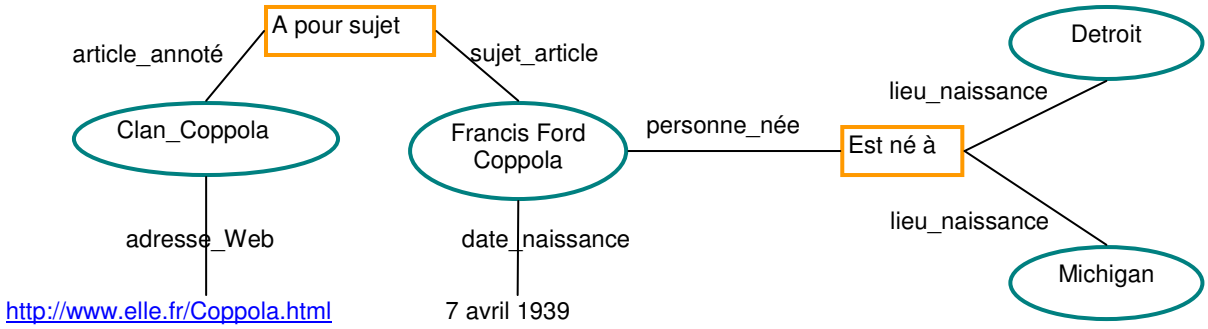


Figure 17. Exemple d'une annotation sémantique en Topic Maps (notation graphique)

Lars Marius Garshol [GAR 03] et Bernard Vatant [VAT 03], [VAT 04] se sont penchés sur les correspondances entre RDF et Topics Maps. Ces langages ont été développés par des communautés différentes, pour des tâches différentes : la création de métadonnées pour RDF et l'indexation de contenu pour les Topic Maps. RDF possède un niveau supérieur d'abstraction grâce à l'utilisation de RDF Schéma qui lui permet de décrire un premier niveau d'ontologie alors que les Topics Maps ne possèdent pas la sémantique nécessaire à la représentation d'ontologies. Enfin, le modèle des Topic Maps semble plus complexe que RDF. Rappelons que la structure de base en RDF est le triplet qui

sert à relier exactement deux ressources au plus (relations binaires) alors que les Topic Maps peuvent relier de multiples sujets entre eux (relations n-aires), ce qui les rend attrayantes quant à la navigation dans des bases de connaissances. Pourtant, tout comme RDF, les Topics Maps tentent de décrire l'information en termes de ressources. Les deux standards permettent de créer des annotations sémantiques au sujet de documents du Web et de rendre documents et contenus plus facilement accessibles aux différents utilisateurs, humains ou machines. Garshold et Vatant ont également comparé la syntaxe des Topic Maps à celle d'OWL. Nous renvoyons donc à leurs travaux pour plus de précisions à ce sujet.

Bien que nous ayons présenté les langages les plus largement adoptés pour l'annotation sémantique, il existe par ailleurs d'autres langages pouvant être utilisés pour les exprimer. Par exemple, OCML [MOT 99] a été utilisé dans le cadre du projet Planet-Onto et de l'outil MnM présenté dans la prochaine section. Toutes les annotations sémantiques générées dans ces langages ne sont pas faites pour être embarquées dans le corps des documents annotés. Certaines sont plutôt stockées dans un serveur d'annotation ou même une base de connaissances. Elles seront alors exploitées ultérieurement selon les applications pour lesquelles elles auront été créées : annotation de nouvelles ressources, création de fiches de connaissances, recherche de ressources annotées, génération de nouveaux documents, etc. [PRI 04]. Nous allons à présent nous intéresser aux outils d'annotation sémantique existants.

1.4 Les outils d'annotation sémantique

Ce chapitre aborde l'état de l'art des outils d'annotation sémantique existants. Après en avoir donné une définition, nous présenterons la synthèse à laquelle nous avons aboutie après une étude attentive de la plupart des outils d'annotation sémantique créés durant ces cinq dernières années. L'Annexe 1 de ce mémoire comprend la grille de lecture qui a servi à notre étude, le détail de l'analyse de chacun de ces outils suivant cette grille de lecture ainsi qu'un tableau récapitulatif des outils selon les différents critères de la grille de lecture.

1.4.1 Qu'est-ce qu'un outil d'annotation sémantique ?

Un outil d'annotation sémantique est un outil logiciel qui permet d'insérer et de gérer des annotations sémantiques liées à au moins une ressource documentaire donnée. Dans le cadre du Web Sémantique, les outils d'annotation sémantique utilisent une ontologie, ou tout au moins un modèle formel, qui formalise et structure les annotations produites en fonction des concepts et des contraintes définis dans cette ontologie.

Ces outils d'annotation sémantique ont pour objectif d'alléger le fardeau de l'annotation manuelle dans les pages Web. La plupart d'entre eux ont évolué vers des environnements de plus en plus automatisés grâce aux méthodes issues des domaines de l'Extraction d'Information et des Systèmes

d'Apprentissage [COR 06]. En plus de l'interface traditionnelle d'annotation manuelle, ces méthodes sont capables de suggérer à l'utilisateur un ensemble d'annotations sémantiques relatives à la ressource documentaire analysée.

Un outil d'annotation sémantique peut aussi être utilisé pour peupler une ontologie, i.e. pour instancier la base de connaissance contenant les instances de l'ontologie de référence. Si l'approche du Web Sémantique se consacre principalement à la production automatisée de pages annotées sémantiquement, le domaine de l'acquisition des connaissances considère l'annotation sémantique des documents comme un moyen d'enrichir une base de connaissance existante grâce aux documents annotés [CON 03]. Ces deux approches convergent donc vers les mêmes outils avec pourtant une différence majeure : c'est l'objectif de l'application et ses utilisateurs qui détermineront le choix de tel ou tel outil par rapport à la tâche devant être accomplie.

1.4.2 Synthèse des outils existants

S'il est impossible de réaliser une étude exhaustive de tous les outils d'annotation sémantique, les quelques uns décrits ci-dessus nous montre que ce domaine est particulièrement actif et en constant progrès. La preuve en est que de nouveaux outils ou de nouvelles versions des outils existants continuent à voir le jour régulièrement. De plus, plusieurs conférences ou ateliers de travail ont chaque année l'annotation sémantique pour thème ainsi que ses dérivés, i.e. l'acquisition des connaissances, le peuplement et la maintenance des ontologies, etc. Nous n'avons considéré dans notre étude que les outils d'annotations de ressources textuelles, mais il existe également des outils ou projets s'intéressant également à l'annotation sémantique de ressources multimédia.

Nous allons à présent synthétiser notre étude en fonction des différents points critères que nous avons établis dans notre grille de lecture, suite à l'analyse des différents outils (cf. Annexe 1).

1.4.2.1 Les Ressources Documentaires

La standardisation des formats de l'annotation est essentielle pour construire des applications compatibles et interopérables avec un vaste choix de systèmes. Parmi les outils décrits ci-dessus, tous ont été développés entre 2000 et 2006 et nous pouvons très clairement constater une évolution des langages utilisés pour l'annotation en fonction des recommandations formulées par le W3C. En effet, les premiers outils comme SHOE KA, OntoAnnotate ou MnM utilisent des langages non standardisés, respectivement SHOE, HTML-A et OCML. Puis avec l'arrivée de RDF et de DAML+OIL, ces précédents formats sont plus ou moins tombés en désuétude : les anciens outils ont migré leurs applications vers la prise en compte de RDF et/ou DAML+OIL alors que les nouveaux outils ont bien sûr directement implémenté l'utilisation de ces langages pour leurs annotations. Enfin, on constate récemment l'arrivée d'annotations générées en OWL. Mais RDF reste pourtant un standard en matière d'annotations documentaire, OWL étant plutôt utilisé pour l'enrichissement des bases de connaissance.

Concernant le contenu des documents, on constate que les outils d'annotation sémantique, et en particulier ceux offrant un processus semi-automatisé voire entièrement automatisé, ne s'intéressent qu'aux entités nommées dans les textes. Il est rare qu'ils sachent extraire les attributs et encore moins les relations. Ou bien, ces relations sont de très haut niveau comme le fait qu'une Personne soit membre d'une Organisation, ou qu'une Organisation se situe dans un Lieu donné. Or dans le cadre de certaines applications Web Sémantique, notamment destinées aux entreprises, il est nécessaire de pouvoir capter toute la sémantique d'un domaine, de manière plus précise que ces relations basiques. Ceci est notamment possible par l'utilisation de règles d'extraction basées sur une analyse linguistique fine. Le corollaire de ceci est aussi dû au fait que les applications s'appuyant sur des l'apprentissage supervisé ont pour but d'annoter des pages web ayant une forte propension à être structurées par des listes ou des tableaux. Ces pages web structurées ou semi-structurées contiennent pour la plupart des ensembles d'entités nommées, mais les relations entre elles ne sont pas explicitement mentionnées.

1.4.2.2 Les Traitements

Nous l'avons dit, l'annotation sémantique peut être réalisée de plusieurs manières, depuis une procédure manuelle jusqu'à un traitement entièrement automatisé. L'automatisation du processus d'annotation est vitale pour faciliter l'acquisition de la connaissance. Si dans les premières années la plupart des outils ont surtout développé une interface utilisateur dédiée à une annotation manuelle, leurs concepteurs ont également ressenti le besoin d'automatiser tout ou partie du processus d'annotation. Les outils ont ainsi évolué pour procurer à l'utilisateur final un support automatique ou semi-automatique basé sur l'utilisation de techniques issues du domaine de l'extraction d'information. Ces techniques implémentent généralement des règles d'extraction construites manuellement ou produites à partir d'un processus d'apprentissage.

La forme la plus commune de systèmes d'extraction intégrés aux outils d'annotation est le « wrapper », originellement développé par Kushmerick [KUS 97]. Il exploite la structure des pages Web, comme les listes ou les tableaux par exemple, pour identifier des fragments d'information à étiqueter. Les wrappers sont donc très efficaces lorsque les documents comportent ces structures régulières. En fait, plus le contenu des pages est structuré, plus les wrappers réussissent à extraire de l'information pertinente, i.e. en conservant un niveau élevé de la qualité des annotations produites, plus le processus d'annotation peut être automatisé. Les outils d'annotation supervisés, comme MnM, Melita ou OntoMat utilisent tous le moteur d'extraction Amilcare : celui-ci apprend à reconnaître les entités qui seront annotées par apprentissage, grâce à l'utilisation d'un corpus d'entraînement constitué de documents précédemment annotés. Ceci requiert l'étiquetage d'un nombre important de documents afin que l'apprentissage puisse atteindre un niveau de pertinence satisfaisant. Par ailleurs, non seulement le corpus d'entraînement doit être conséquent, mais ses annotations doivent être de « bons » exemples, ceci afin d'optimiser la tâche d'apprentissage et obtenir des résultats qui soient exploitables durant la phase d'annotation.

Les outils d'annotation comme KIM, ArtEquAkt ou Onto-H sont basés sur un ensemble d'analyses linguistiques issues du Traitement du Langage Naturel. Celles-ci implémentent des patrons d'extraction le plus souvent compilés dans des automates à états finis. Les patrons s'attachent à

l'étude de la syntaxe et parfois même de la sémantique du contenu des documents. Cette technique permet généralement d'extraire des informations à partir de contenus semi à non structurés. L'écriture de ces patrons peut devenir fastidieuse, surtout si l'objectif est d'extraire des informations sémantiquement très riches. Et si ces patrons doivent être adaptés à un nouveau domaine, leur concepteur aura alors besoin de fortes compétences linguistiques dans l'écriture des expressions régulières implémentées dans les patrons.

Enfin, les outils d'annotation comme Armadillo ou SemTag se sont intéressés aux problèmes du besoin en corpus annotés pour les systèmes d'extraction basés sur l'apprentissage supervisé. Ils implémentent différentes techniques basées sur les statistiques du corpus documentaire analysé au sein de systèmes d'extraction par apprentissage. Par exemple, PANKOW démontre comment la distribution de certains patrons sur le Web peut être utilisée pour en déduire l'annotation d'entités. Néanmoins, les utilisateurs de ces systèmes d'annotation, qui peuvent devenir entièrement automatisés, doivent être prévenus de leurs limites : ces systèmes semblent ne fonctionner que sur des pages fortement structurées. Cependant, certaines applications, ayant pour objectif le traitement de grandes collections de documents comme cela est le cas pour l'annotation du Web, préfèrent une annotation imparfaite plutôt que pas d'annotation.

Un autre problème lié aux outils d'extraction d'information est l'extraction des relations. Cette extraction est cruciale pour l'étiquetage de l'information ontologique et la création de documents intelligents. La plupart des systèmes d'extraction savent reconnaître les concepts et leurs instances mais plus rares sont ceux repérant des relations sémantiques explicites entre ces instances. Pour cette raison, la tâche d'enrichissement de la base de connaissance ne peut être complète sans une intervention manuelle de l'utilisateur final. Pour certaines applications, ceci peut constituer un frein majeur à la mise en place de tels outils par manque de richesse, de couverture et de précision. Si les utilisateurs connaissent assez bien les entités (nommées) de leur domaine, notamment dans le cadre d'applications déterminées comme la Veille ou la Publication en entreprise, ce qui les intéresse d'autant plus ce sont les relations qu'elles entretiennent entre elles et qui évoluent dans le temps.

1.4.2.3 Les Ontologies

De rares outils, comme KIM ou AeroDAML, imposent à l'utilisateur final l'utilisation d'une ontologie générique de référence. Mais généralement, le choix de l'ontologie est laissé à l'utilisateur final afin que ce dernier puisse orienter la tâche d'annotation en fonction de ses besoins. Ces ontologies sont donc modélisées en fonction du domaine concerné pour l'application utilisatrice. Elles peuvent être chargées localement ou bien à partir d'un site Web suivant leur disponibilité.

La plupart des outils supportent désormais le format OWL. Ils ont su s'adapter à l'évolution des langages et standards définis pour le Web Sémantique. Pourtant rares sont les outils fournissant d'autres fonctionnalités que la recherche et la navigation dans une ou plusieurs ontologies. Mis à part les outils qui intègrent un éditeur d'ontologie tel SMORE ou OntoMat, la maintenance des ontologies n'est quasiment pas supportée par la génération actuelle des outils alors que cette tâche affecte directement la maintenance des annotations qui en dépendent [URE 06]. Les outils d'annotation que

nous avons étudiés ne permettent pas jusqu'à présent d'automatiser cette maintenance, ni même d'apporter un quelconque support à l'utilisateur chargé de cette tâche. Or, nous pensons que ce problème est un challenge important et que la maintenance des ontologies pourrait être favorisée par l'ajout de fonctionnalités supplémentaires dans les interfaces actuelles.

Enfin, un autre genre de maintenance doit être pris en compte. En effet, les ontologies sont peuplées par de nouvelles instances issues des annotations. Il est intéressant, dans le cas où les outils d'extraction d'information reposent sur l'utilisation de lexiques, de faire usage de ces nouvelles connaissances afin d'enrichir ces lexiques. La tâche concernant le peuplement d'ontologies peut en effet servir à réaliser une maintenance des entrées de ces lexiques. Par exemple, une règle d'extraction basée sur une expression régulière du langage a permis d'extraire le nom d'une société du domaine. Cette société est ajoutée à la base de connaissance soit automatiquement soit à la suite d'une validation par l'utilisateur final. En retour, cette société vient enrichir le lexique de l'outil d'extraction. Ainsi, au prochain document, même si l'expression régulière premièrement utilisée pour reconnaître cette société n'a pas été déclenchée, le fait que cette société appartienne au lexique permettra automatiquement sa reconnaissance et son extraction. Là encore, peu d'outils d'annotation se sont penchés sur l'implémentation d'une solution pour cette maintenance des outils d'extraction. Et nous pensons que ce retour sur investissement permettrait pourtant d'améliorer les résultats des outils d'extraction d'information au fur et à mesure de leurs utilisations sur un domaine particulier.

1.4.2.4 Le Stockage des annotations

Une variété d'approches pour stocker les annotations ont vu le jour, depuis l'utilisation de serveurs d'annotation comme Annotea ou le Label Bureau de SemTag ou bien de bases de connaissance comme KIM. La nuance est parfois subtile entre un serveur d'annotation et une base de connaissance : un serveur d'annotation pouvant stocker de la connaissance et une base de connaissance pouvant stocker des annotations. Toutefois, précisons que, dans le cas du serveur d'annotation, l'annotation est explicitement attachée à une ressource documentaire et sa valeur peut être une référence vers une instance de connaissance, que cette dernière soit stockée dans le même serveur d'annotation ou en externe. La base de connaissance est directement construite à partir de la modélisation donnée par l'ontologie de référence. Elle sert à y stocker ses instances de concepts, d'attributs et de relations en concordance avec les restrictions, contraintes et axiomes définis dans l'ontologie. Il se peut que ces instances soient enregistrées avec une métadonnée indiquant le nom ou l'adresse du document ayant servi à leur création dans la base de connaissance. Mais les instances existent en tant que telles, indépendamment des annotations ou des documents les ayant produites. Ces instances sont en relation les unes avec les autres fournissant un réseau de connaissance du domaine concerné.

Outre le fait de pouvoir stocker ces annotations dans un des deux modèles présentés ci-dessus, il est également possible de les enregistrer directement en local sur l'ordinateur de l'utilisateur final ou bien sur un réseau distribué tel que le Web. Dans ce cas, les annotations peuvent être stockées soit dans le même fichier que le document source, soit dans un fichier séparé. Dans le cadre du Web Sémantique, les ressources documentaires et les annotations sont généralement stockées

séparément, comme avec Annotea ou AktiveDoc. Ceci est inévitable puisque les documents, i.e. les pages Web analysées, appartiennent au site Web qui en est le seul propriétaire. Par contre, dans une organisation comme l'entreprise, un modèle alternatif consiste à stocker les annotations directement dans le document lorsque l'entreprise est bien sûr propriétaire du document en question. Ceci est le cas pour SemanticWord [TAL 03] ou même MnM. L'impact de l'approche choisie se ressentira sur la maintenance des annotations vis-à-vis du document annoté. Lorsque les annotations ne sont pas stockées avec le document source, il est nécessaire d'avoir un processus permettant de conserver la conformité des annotations avec le document annoté. Cette tâche est extrêmement fastidieuse et complexe. Cependant, le stockage séparé comporte aussi ses avantages. Le découplage entre la sémantique et le contenu facilite la réutilisation du document mais aussi de ses annotations. Cela permet de lancer des requêtes sur des ressources hétérogènes comme si elles appartenaient à une même base de connaissances. Cela rend aussi plus facile la production de différentes vues sur un même document pour les utilisateurs ayant différents rôles et différents droits d'accès, facilitant ainsi le partage, la collaboration et la réutilisation [URE 06]. C'est pourquoi beaucoup d'outils d'annotation n'ont finalement pas opté pour l'une ou l'autre de ses deux approches, autorisant l'utilisateur final à choisir s'il préfère stocker ces annotations dans ou hors le document source. Parmi ces outils, on retrouve SHOE KA, COHSE ou encore les différentes versions d'Ont-O-Mat.

1.4.2.5 L'Interfaçage

Un des environnements le plus commun de tous ces outils est un navigateur Web, un résultat naturel du fait que la plupart d'entre eux sont conçus pour une utilisation dans le cadre du Web Sémantique. Même pour une utilisation dans le cadre plus particulier des entreprises, ceci a l'avantage d'être une technologie très familière. Un autre environnement fréquent est l'utilisation d'applications indépendantes Java possédant un navigateur d'ontologie et un éditeur de document. Par contre, l'inconvénient de ces environnements est qu'ils se focalisent principalement sur les formats du Web tels que HTML et XML en oubliant d'autres formats utilisés notamment en entreprise comme MS Word ou PDF [URE 06].

Par ailleurs, concernant les outils implémentant un navigateur d'ontologies ou bien permettant l'enrichissement d'une base de connaissance, bien peu d'entre eux possède des fonctionnalités avancées pour aider l'utilisateur à sélectionner les annotations valides vis-à-vis des contraintes et restrictions modélisées dans l'ontologie de référence. Seules les différentes versions d'Ont-O-Mat fournissent un support utilisateur dans ce domaine. Mais les outils qui apparaissent depuis peu, comme SMORE, ont compris que l'annotation documentaire et le peuplement d'ontologies devaient être contrôlés afin de garantir l'intégrité et la cohérence des bases de connaissances ainsi générées.

Enfin, une grande proportion de ces outils s'intègrent dans un environnement de travail généralement développé par l'équipe ou le laboratoire dans lesquels ils ont été créés. Certains permettent également de s'interfacer avec des outils devenus des standards, comme le serveur d'annotation Annotea. Les outils issus de la recherche universitaire sont pour la plupart accessibles en open-source afin d'être testés par la communauté et améliorés.

1.5 Discussion

1.5.1 Synthèse au sujet de l'annotation sémantique

Nous venons de présenter dans ce chapitre la notion d'annotation sémantique et nous avons vu que, dans le cadre du Web Sémantique, elle est intrinsèquement liée à la modélisation d'une ontologie. En effet, cette ontologie va représenter les concepts, attributs et relations d'un domaine à l'aide d'un langage de représentation des connaissances orienté Web comme OWL. Les annotations sémantiques sont structurées à l'aide de cette ontologie et leurs valeurs pointent vers les instances de l'ontologie de référence ou, dans certains cas, directement vers les concepts eux-mêmes. Les autres Ressources Terminologiques ou Ontologiques, tels que les thesaurus, peuvent aussi être utilisés comme valeur d'annotation sémantique afin de fournir une perspective différente à l'utilisateur final au sujet d'une même ressource documentaire.

Nous avons vu qu'il existe une gamme assez importante d'outils d'annotation sémantique ayant chacun des caractéristiques propres mais dont le but de plus en plus orienté vers l'assistance des annotateurs humains à la création des annotations. Le niveau d'automatisation dépend du moteur d'extraction d'information intégré dans l'outil. Les annotations peuvent être stockées dans un serveur d'annotation, en mode embarqué ou débarqué vis-à-vis du document d'origine, ou bien dans une base de connaissance. Elles peuvent, entre autres, servir à améliorer les systèmes de recherche d'information, à peupler une ontologie existante, voire à aider à la construction des ontologies. Nous retenons de notre étude que les outils présentés sont tous plus ou moins issus de la recherche. Malgré une évolution rapide des langages et standards, grâce notamment à l'essor du Web Sémantique, ces outils ne sont toujours pas adaptés aux utilisations concrètes du monde réel. Et pour que les entreprises, ayant envie de bénéficier du nouveau modèle spécifié par le cadre fondateur du Web Sémantique, puisse les intégrer dans leurs applications, certaines limites encore existantes ont besoin d'être repoussées.

1) Les outils d'annotation sont intrinsèquement liés au moteur d'extraction utilisé pour extraire l'information des textes et ainsi les annoter. Je pense qu'il faut à tout prix dissocier les deux pour que l'outil d'annotation puisse utiliser tel ou tel moteur d'extraction en fonction des besoins de l'application dans laquelle il s'inscrit.

2) Les moteurs d'extraction reposent pour la plupart sur des processus d'apprentissage supervisés. Ces systèmes sont peut-être performants pour du contenu structuré, voire semi-structuré. Mais sur le Web et dans les entreprises, le contenu non structuré est celui contenant potentiellement de nouvelles informations stratégiques, notamment les relations sémantiques entre entités, et dont la sémantique est la plus difficile à extraire [AUS 00a]. J'estime que dans ce cas, il est nécessaire de donner la priorité aux systèmes d'extraction basés sur différentes analyses linguistiques fines permettant d'identifier cette sémantique dans le contenu des documents traités.

3) L'annotation sémantique doit continuer à privilégier les approches basées sur des ontologies de domaine, et non génériques. Cela est surtout le cas dans les applications destinées au monde de

l'entreprise. Ces dernières sont intéressées par des faits inhérents à leur propre domaine d'application et non par des généralités.

4) Il faut pouvoir fournir une aide maximale à l'utilisateur et notamment lorsque le but consiste aussi à peupler l'ontologie. Les processus et les interfaces doivent non seulement lui présenter des suggestions, des propositions mais ils doivent également être en mesure de le guider en prenant notamment en compte les contraintes et restrictions modélisées dans l'ontologie de référence. Je suis convaincue que la question de l'intégrité des annotations, et plus particulièrement de la base de connaissance qui accueille les nouvelles instances issues de ces annotations, est primordiale pour l'exploitation des résultats fournis par une telle application.

1.5.2 Vers une méthodologie d'annotation sémantique

Je me suis inspirée de ces desirata pour construire une méthodologie pour l'implémentation d'une solution d'annotation sémantique qui puisse aussi être utilisée pour peupler une ontologie. Cette méthodologie représente la base pour implémenter un outil s'attachant à résoudre des cas concrets d'applications dans le monde de l'entreprise. Néanmoins, plutôt que de « réinventer la roue », nous nous basons sur des outils existants du Traitement Automatique du Langage (TAL) et de Représentation des Connaissances. Le but de la méthodologie est de définir une passerelle entre ces deux catégories d'outils.

La mise en place de la passerelle doit également résoudre les problèmes inhérents à l'interfaçage entre deux types d'outils différents, ayant des entrées/sorties différentes. Grâce à l'expérience acquise lors de nos divers projets, nous avons constaté les problèmes récurrents suivants :

- Problème de format : comment passer d'un format de représentation du document textuel à un autre format de représentation dépendant de l'ontologie et de l'implémentation de la base de connaissance ? A l'issue de l'analyse linguistique, les outils de TAL produisent un ensemble d'étiquettes sémantiques le plus souvent sous la forme d'un document XML, aussi appelé arbre conceptuel. L'ontologie est modélisée dans le but de stocker et d'exploiter de la connaissance de manière rigoureuse et contraignante grâce aux formalismes de représentation des connaissances (tels OWL, RDF, XTM). Comme nous avons vu dans ce chapitre, ces formalismes ont des exigences plus ou moins fortes sur la qualité et la formalisation des informations extraites.

- Problème de la couverture du domaine lors du passage du langage naturel au modèle : comment gérer le décalage entre le domaine couvert par l'ontologie et le vocabulaire contenu dans les ressources linguistiques pour les outils de TAL ? Les spécifications et le développement des ontologies et des ressources linguistiques utilisées par les outils de TAL sont indépendants les uns des autres. La couverture du domaine concerné n'est donc pas forcément alignée entre les étiquettes linguistiques et les concepts de l'ontologie. En effet, l'ontologie peut servir à d'autres tâches impliquant une couverture du domaine différente de celle couverte par les outils du TAL. Ces derniers peuvent réutiliser des ressources linguistiques existantes ou bien définir de nouveaux patrons linguistiques spécifiques au domaine étudié. Bien que ceci soit tout à fait

justifié, le non-alignement de la couverture peut néanmoins poser des difficultés d'intégration entre les deux systèmes, outre les contraintes techniques posées par l'implémentation même de la passerelle entre les deux types d'outils.

- Problème de la conceptualisation lors du passage du langage naturel au modèle : comment savoir si la signification d'une étiquette sémantique produite par les outils du TAL correspond à la signification d'un concept de l'ontologie ? En effet, une étiquette sémantique a priori « identique » (ayant au moins le même libellé) à un concept de l'ontologie peut servir à annoter ou à instancier un tout autre concept de l'ontologie. Ceci dépend des besoins de l'application et de la modélisation de l'ontologie. A l'inverse, deux concepts de l'ontologie différents peuvent être instanciés à partir de la même étiquette sémantique et deux étiquettes sémantiques différentes peuvent servir à annoter ou à instancier le même concept de l'ontologie. Qu'est-ce qui peut servir à assimiler ou à distinguer une étiquette sémantique d'un concept de l'ontologie ? Gruber dans [GRU 91] se demandait aussi : « *what information about terms is most critical for supporting sharability? The names? Textual definitions? Type, arity and argument restrictions? Arbitrary axioms?* ».

Dans le prochain chapitre de ce mémoire, nous allons étudier les caractéristiques des étiquettes fournies par les outils de TAL, et plus particulièrement des moteurs d'extraction d'information. Puis, dans la deuxième partie de ce mémoire, nous présenterons notre solution à ces constats : les Règles d'Acquisition de Connaissance (cf. Chapitre 3). Nous verrons comment elles s'intègrent dans la démarche globale d'OntoPop (cf. Chapitre 4). Nous définirons une méthodologie permettant d'implémenter cette démarche dans des applications concrètes (cf. Chapitre 5) et nous décrirons la plateforme logicielle associée (cf. Chapitre 6). Cette plateforme est d'ors et déjà utilisée dans des diverses applications en entreprise comme nous le verrons dans les expérimentations (cf. Chapitre 7).

Chapitre 2. L'Extraction d'Information, une application du TAL pour l'annotation sémantique

Nous venons de voir que les moteurs d'extraction d'information permettent d'automatiser tout ou partie du processus d'annotation sémantique. Le choix d'un moteur d'extraction est donc particulièrement important pour la mise en pratique de ces tâches, surtout dans un contexte d'applications en entreprise ou sur le Web. Or, nous sommes arrivés à la conclusion dans le chapitre précédent qu'il existait aujourd'hui un fossé non encore comblé entre le résultat de ces moteurs d'extraction et la représentation sémantique des annotations.

Nous allons donc tenter de comprendre dans ce chapitre pourquoi un tel fossé persiste en étudiant le domaine de l'extraction d'information, ses tâches et ses méthodes. Puis nous donnerons l'exemple de deux outils figurant parmi les plus aboutis, que ce soit du côté de la recherche ou de l'industrie. Enfin, nous chercherons à apporter des réponses aux constats que nous avons posés précédemment en nous penchant sur la représentation des résultats des outils présentés.

2.1 Présentation de l'Extraction d'Information

L'Extraction d'Information (EI) est une des applications du Traitement Automatique du Langage Naturel (TALN), aussi appelé Traitement Automatique des Langues (TAL) [FUC 93]. Le TALN, discipline à la frontière entre la linguistique et l'informatique, a été créée pour tenter d'apporter des réponses à une société qui manipule un volume croissant de documents exprimés en langage naturel, qu'ils soient écrits ou parlés. Son objectif est donc de concevoir des modèles d'analyse et de génération du langage naturel à partir desquels il devient possible de réaliser des logiciels capables de traiter automatiquement des données linguistiques, i.e. de comprendre ou de produire des énoncés exprimés en langue naturelle. Le langage naturel se réfère au langage humain : complexe, irrégulier, divers, avec tous les problèmes liés au sens et au contexte [WEH 97]. Les données linguistiques peuvent être de différentes tailles : des textes jusqu'aux mots, en passant par l'étude des phrases, des énoncés, des groupes de mots, etc. [FUC 93]. Elles ne sont pas forcément monolingues, le TAL s'étant également intéressé aux problématiques liées au multilinguisme. Outre l'extraction d'information, le TALN a donné lieu à d'autres sortes d'applications telles que la recherche d'information, les vérificateurs et correcteurs orthographiques, la dictée vocale, la synthèse de la parole, la traduction automatique, le résumé automatique, les systèmes de question/réponse, etc. [PIE 00].

Si le TALN est né dans les années 1950, l'extraction d'information (EI) a été clairement définie seulement à partir de 1987 grâce aux conférences des Message Understanding Conference (MUC)

organisées par le DARPA. Dans ce cadre, l'EI « consiste, dans un domaine restreint, à extraire des éléments d'information précis à partir d'un ensemble de textes homogènes et à remplir des formulaires prédéfinis avec ces éléments d'information » [POI 99]. Les campagnes d'évaluation MUC ont été organisées afin de confronter les systèmes d'extraction d'information réalisés par différentes équipes en comparant leurs performances avec des mesures précises et objectives. Ces mesures, inspirées de celles définies pour le domaine de la Recherche d'Information, sont devenues un standard pour toute évaluation des résultats de l'EI. Ainsi, la *précision* mesure le bruit produit par le système, c'est-à-dire le nombre d'informations extraites correctement par rapport au nombre d'informations extraites. Le *rappel* lui mesure le silence du système, c'est-à-dire le nombre d'informations correctement extraites par rapport au nombre d'informations correctes présentes dans le corpus. Enfin, la F-mesure permet de disposer d'une évaluation globale du système en combinant précision et rappel [GRI 96]. Nous reparlerons plus en détail de ces mesures dans le chapitre consacré à l'évaluation (cf. Chapitre 7).

L'apport des conférences MUC a été considérable : aussi bien en termes d'identification des problèmes à prendre en compte (linguistique, représentation des connaissances, acquisition de ressources, travail sur corpus…) qu'en termes de méthodes et de techniques pour les résoudre. Divers systèmes d'extraction d'information ont été testés sur différents types de textes : récits d'attentats (MUC-3 et MUC-4), annonces de produits (MUC-5), annonces financières concernant les prises de participation des entreprises (MUC-6), etc. Les systèmes en compétition devaient remplir un ou plusieurs formulaires (« template » en anglais) fixés à l'avance en fonction du domaine. Par exemple, pour les annonces financières, ils devaient extraire les différentes sociétés (acheteurs, vendeurs, achetés), la date, le lieu et le montant de la transaction financière, etc.

2.1.1 Les tâches de l'extraction d'information

Lors de la conférence MUC-7, cinq épreuves ont été identifiées : la reconnaissance des entités nommées, la coréférence, la reconnaissance des attributs, la reconnaissance des relations et enfin la reconnaissance des scénarios [ENJ 05b]. Nous allons brièvement présenter en quoi consiste chacune de ces tâches. Les chiffres présentés pour chacune des tâches représentent le score moyen atteint par les systèmes évalués lors du MUC-7 [CUN 99].

2.1.1.1 Les Entités Nommées (EN)

Le terme « entité nommée » [GRI 96] désigne toutes les formes linguistiques bien identifiées, à l'instar des noms propres (de personnes, d'organisations, de lieux) mais également les expressions temporelles (dates, durées, horaires), les quantités (monétaires, unités de mesure, pourcentages), etc. La tâche de reconnaissance des entités nommées consiste donc à les repérer dans le texte concerné et à leur affecter une étiquette sémantique choisie dans une liste prédéfinie. Certaines entités peuvent être ambiguës (« Peugeot » peut représenter une voiture particulière, des personnes ou une société) mais elles sont généralement faciles à repérer et moins polysémiques que d'autres unités textuelles.

Les systèmes de reconnaissance d'entités nommées exploitent généralement des dictionnaires, ou lexiques, (de noms propres, de noms de villes, etc.) couplés avec des règles d'extraction permettant de repérer de nouvelles entités nommées sur la base de leur contexte. Par exemple, la règle « <titre><prénom><Mot inconnu avec majuscule> » détecte un nom propre de personne à la place du « <Mot inconnu avec majuscule> », comme « Hugo » dans « Mr Victor Hugo ». Des méthodes d'apprentissage [POI 03] ont aussi été développées pour induire des règles d'extraction à partir de documents à la fois suffisamment fiables et productives.

Les performances des systèmes de reconnaissance d'entités nommées sont évidemment variables en fonction du type des entités nommées recherchées, de la couverture des dictionnaires et des règles, du style rédactionnel et de la structuration des textes analysés. Mais en général, ils fournissent une bonne précision à défaut d'avoir un bon rappel [NAZ 05]. En effet, d'après [CUN 99], la précision des systèmes atteint environ 95%, ce qui correspond à un taux similaire à celui atteint par les annotateurs humains (taux mesuré en comparant les annotations produites par des annotateurs humains sur un même corpus de documents).

2.1.1.2 La Coréférence (CO)

Cette épreuve consiste à reconnaître toutes les formes linguistiques qui se réfèrent à une entité nommée. Par exemple, dans « Sofia Coppola est la fille du réalisateur américain. Elle s'est mariée à Paris avec Spike Jonze », la résolution des coréférences devrait relier « Elle » à « Sofia Coppola ». Cette tâche est importante pour les tâches suivantes de résolution des attributs et des relations.

Elle se subdivise en deux sous-tâches : la résolution des anaphores (l'exemple précédent) et l'identification des variantes de forme des noms propres. Cette deuxième tâche revient à trouver toutes les occurrences des mêmes entités orthographiées différemment ou leurs alias, comme pour « FT », « France Telecom », « France Télécommunication », etc. Mais la coréférence est un processus imprécis, particulièrement lorsqu'elle est appliquée à la résolution des références anaphoriques. Selon [CUN 99], les résultats varient donc grandement d'un domaine à un autre, atteignant une précision entre 50% et 60%.

2.1.1.3 Les Attributs (Element Template)

La reconnaissance des éléments du formulaire associe en fait de l'information descriptive, généralement sous la forme de groupes nominaux, aux entités précédemment identifiées. Cette information descriptive correspond à un attribut de l'entité concernée. Dans l'exemple précédent, cette tâche devrait identifier « réalisateur américain » par rapport à l'entité nommée « Francis Ford Coppola » si suffisamment d'information est présente dans le texte environnant.

De bons scores sont obtenus par les systèmes avec une précision moyenne de 80% [CUN 99], malgré le fait qu'ils dépendent fortement d'un domaine en particulier et que l'évolution de ces systèmes vers un autre domaine entraînerait une redéfinition importante des règles ou des dictionnaires sur lesquels ils s'appuient.

2.1.1.4 Les Relations (Relation Template)

La reconnaissance des relations s'attache à identifier un certain nombre de relations, le plus souvent binaires, entre les entités extraites précédemment. Ainsi, dans l'exemple précédent, cette tâche permet de repérer une relation de mariage entre les entités personnes « Sofia Coppola » et « Spike Jonze » et une relation de parenté entre les entités « Sofia Coppola » et « Francis Ford Coppola ».

L'extraction des relations entre les entités est une tâche centrale pour les applications d'extraction d'information, surtout pour des applications en entreprise [APP 99]. En général, les bons systèmes de reconnaissance des relations ont des scores environnant 75% et tout comme la tâche précédente, ils dépendent fortement du domaine pour lequel ils ont été créés [CUN 99].

2.1.1.5 Les Scénarios (Scenario Template)

L'épreuve de reconnaissance des scénarios relient entre eux les entités et les relations précédemment repérées dans des descriptions d'événement relatif au domaine étudié (un attentat, une transaction financière, une hospitalisation, etc.). Pour chaque événement, sont également associés les différents traits complémentaires tels que la localisation spatiale et temporelle s'il y a lieu. Par exemple, la reconnaissance d'entités a repéré « Sofia Coppola » et « Spike Jonze » comme des entités personnes et « Paris » comme une entité de lieu. La reconnaissance de relations a identifié une relation de mariage entre ces personnes. La reconnaissance d'un scénario identifie l'événement mariage dans son ensemble, i.e. que ces deux personnes se sont mariées dans le lieu cité.

La reconnaissance des scénarios est une tâche particulièrement difficile. Elle dépend des résultats des étapes précédentes et possède donc un score plus faible, dépendant de la composition de leurs résultats. Les meilleurs systèmes d'EI ont un score de 60% environ. Le score des annotateurs humains est autour de 80% de consensus, ce qui illustre la complexité de la tâche [CUN 99]. En plus d'être liée au domaine concerné, cette tâche est également fortement dépendante de l'application devant être réalisée pour les utilisateurs finaux.

2.1.2 Les règles d'extraction d'information

Les moteurs d'extraction d'information reposent sur un ensemble de règles d'extraction [NAZ 05]. Ces règles (cf. Figure 4) comportent une première partie qui stipule quelles sont les conditions que la portion de texte analysée doit vérifier pour qu'on puisse extraire certains éléments textuels. Il s'agit du patron d'extraction. L'ensemble des patrons d'extractions sont ensuite compilés dans un automate (ou transducteurs) à états finis [WEH 97]. La seconde partie indique comment interpréter ces éléments pour remplir un ou plusieurs champ(s) du formulaire. Elle correspond à l'action qui sera déclenchée dans le cas où un patron est reconnu dans le texte analysé : remplir le formulaire prédéfini pour la tâche d'extraction, étiqueter le texte avec les résultats obtenus, alimenter automatiquement des bases de données ou mieux encore, des bases de connaissances, qui dès lors pourront être consultées en lieu et place des textes eux-mêmes [FUC 93].

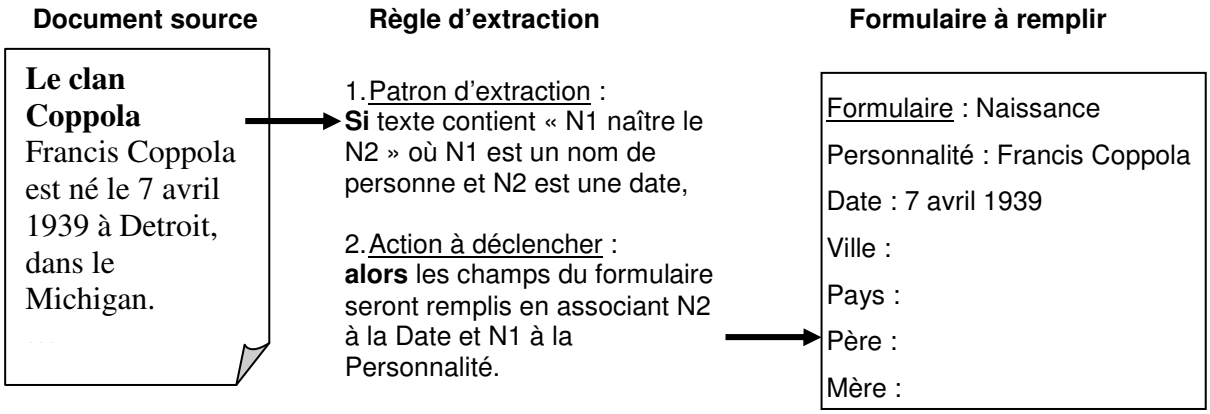


Figure 18. Exemple d'application d'une règle d'extraction pour remplir un formulaire « Naissance »

Les patrons d'extraction reposent généralement sur des expressions régulières. Celles-ci s'écrivent soit à partir de la structure des ressources documentaires lorsque celles-ci sont explicites (les balises HTML d'une page Web par exemple), soit à partir d'une analyse linguistique plus ou moins complète (impliquant partie ou tout des différents niveaux d'analyse évoqués à la section 1.1.2, i.e. morphologique, syntaxique et sémantique), soit à partir d'une combinaison des deux. L'écriture de ces règles d'extraction peut être entièrement manuelle ou bien guidée par des systèmes d'apprentissage, supervisés ou non [APP 99].

Dans la plupart des applications, les informations à extraire sont étroitement ciblées. Elles répondent à un format prédéfini, le formulaire à remplir ou les étiquettes à apposer aux unités textuelles du document source, dans un domaine de connaissance déterminé et restreint. Le corpus documentaire à traiter est lui-même fortement spécialisé et réputé contenir cette information. Autrement dit, en extraction d'information, on connaît ce que l'on cherche (scénarios), on sait où le trouver (corpus) et à peu près dans quelle forme l'information sera exprimée (expression) [PIE 00]. Pourtant, il n'existe aucune méthodologie formelle pour l'écriture des règles d'extraction même si nous pouvons dégager deux types d'approches, qui peuvent également être combinées [ENJ 05b] :

- Ascendante : l'étude du corpus textuel déclenche et guide la définition des scénarios et formulaires qui feront l'objet des règles d'extraction d'information.
- Descendante : en fonction des formulaires et scénarios prédéfinis, le corpus est analysé pour dégager les expressions régulières pertinentes qui permettront de les instancier.

Il est également possible d'utiliser des méthodes quantitatives, d'ordre statistique, pour aider le linguiste à rechercher des régularités langagières, et notamment lexicales, dans les corpus documentaires à analyser [CON 05].

Les règles d'extraction seront ensuite intégrées au moteur d'extraction d'information. Ce dernier procèdera au traitement du corpus documentaire, lequel se découpe généralement en trois phases [ENJ 05b] :

1) Les prétraitements qui suivant les moteurs consistent à filtrer les textes à analyser, à découper le texte en unités textuelles (phrases et/ou mots le plus souvent), à corriger les fautes de surface et enfin à extraire les entités nommées connues à partir de listes (« gazetteers » en anglais) ou de dictionnaires.

2) L'application des règles d'extraction, et plus particulièrement des patrons d'extraction au fur et à mesure des analyses morphologique, syntaxique et sémantique. Ces analyses ne sont pas toujours toutes réalisées, cela dépend de l'approche choisie par le moteur d'extraction. Il peut également opter pour des analyses partielles, comme l'analyse syntaxique partielle (« shallow parsing » en anglais) [HOB 97], puisque une analyse linguistique complète est souvent difficile à réaliser. Les premières entités nommées repérées à l'aide de dictionnaires sont ici complétées avec les entités nommées extraites à l'aide des patrons d'extraction. Ces derniers peuvent aussi extraire, dans certains cas, les relations reconnues entre ces entités en fonction des syntagmes nominaux et verbaux contenus dans le texte. Le résultat correspond à une représentation sémantique du texte constituée des fragments pertinents étiquetés [CHA 05].

3) L'instanciation des formulaires constitue la dernière étape et comprend les calculs de coréférence, les inférences qui peuvent être déduites de la représentation sémantique ou des identifications des entités nommées et relations repérées à l'étape précédente et enfin le remplissage des scénarios et des formulaires ou des bases de données s'il y a lieu.

Cette méthode d'extraction d'information soulève des questions majeures liées à la place importante occupée par l'analyse linguistique. Par exemple, « quelle doit être la complexité d'analyse à chacun des niveaux ? » [ENJ 05b]. Lors des conférences MUC, le système FASTUS a montré que les analyses syntaxiques partielles permettent d'implémenter une solution à moindre coût et tout aussi efficace [HOB 97]. Par ailleurs, le découpage traditionnellement admis en analyse morphologique, syntaxique et sémantique n'est pas aussi net qu'il y paraît et ces analyses ont plutôt besoin les unes des autres pour pouvoir résoudre les ambiguïtés soulevées à leur niveau [WEH 97]. Enfin, il est souvent reproché aux méthodes classiques d'extraction d'information l'écriture souvent fastidieuse des patrons d'extraction et leur forte dépendance vis à vis du domaine. De ce fait, les outils ne sont ni génériques, ni réutilisables pour de nouvelles applications que celle pour laquelle ils ont été créés [POI 03]. C'est pourquoi beaucoup d'outils d'extraction d'information, tel Amilcare [CIR 03b], se sont tournés vers l'apprentissage supervisé des règles d'extraction. Mais cette solution demande la constitution de corpus annotés à la main [HAB 05], souvent de taille importante pour que le mécanisme d'apprentissage puisse atteindre des niveaux satisfaisants pour l'extraction. De plus, les moteurs implémentant cette solution ne savent généralement pas traiter les textes non structurés et sont moins performants dans l'extraction des relations sémantiques entre les entités nommées [AUS 00a].

C'est pourquoi nous avons volontairement occulté dans cette partie les méthodes basées sur l'apprentissage supervisé ainsi que les méthodes probabilistes pour l'extraction d'information. En effet, dans notre problématique d'annotation sémantique et de peuplement d'ontologies dans le cadre du Web Sémantique, et notamment dans le contexte de leur application à l'entreprise, la précision des informations extraites doit être privilégiée ainsi que la possibilité de constituer des réseaux sémantiques riches comprenant aussi bien les entités nommées du domaine concerné que leurs propriétés (attributs et relations).

2.2 Deux exemples d'outils d'extraction d'information

Plutôt que de brosser un tableau complet de tous les outils d'extraction d'information, nous avons choisi de mettre l'accent sur deux outils basés sur l'écriture manuelle des règles d'extraction, parmi les plus aboutis aussi bien dans le domaine de la recherche que dans celui de l'industrie. Le premier, GATE™, est une plateforme d'ingénierie linguistique créée par le laboratoire de recherche « Natural Language Processing Group » de l'Université de Sheffield et disponible gratuitement sur Internet. Il est utilisé dans divers projets de recherche internationaux, à plus ou moins grande échelle, et intégré à de nombreux outils d'annotation sémantique comme nous avons pu voir dans la section 1.4. Le second, IDE™, est un logiciel commercialisé par la société Temis. Il est utilisé dans des projets industriels, notamment de veille économique et scientifique, pour de grands comptes français et internationaux.

2.2.1 GATE

GATE (pour « General Architecture for Text Engineering ») [CUN 02b] n'est pas à proprement parler un moteur d'extraction d'information mais plutôt une plateforme d'ingénierie linguistique développée depuis 1995 à l'Université de Sheffield (UK) par le groupe de recherche « Natural Language Processing ». En effet, cet outil permet de construire ses propres composants logiciels pour traiter du langage naturel. D'après ses concepteurs, GATE est à la fois [CUN 02b] :

- une architecture logicielle car il définit l'organisation d'un système d'ingénierie linguistique et l'assignement des responsabilités des différents composants, et s'assure de la satisfaction des besoins du système à partir des interactions de ces composants ;
- un cadre de travail car il fournit une conception réutilisable (librairie de classes, APIs) par tout logiciel d'ingénierie linguistique et un ensemble de composants logiciels préfabriqués que les ingénieurs linguistes peuvent utiliser, étendre et personnaliser en fonction de leurs besoins spécifiques ;
- un environnement de développement car il a pour but d'aider les ingénieurs linguistes à minimiser le temps passé à développer de nouveaux systèmes d'ingénierie linguistique ou à modifier les systèmes existants, grâce aux fonctionnalités de son environnement graphique et au mécanisme de débogage.

L'architecture de GATE suppose que les éléments des systèmes logiciels qui traitent du langage naturel peuvent être divisés en divers composants, réutilisables indépendamment les uns des autres,

connus dans GATE sous le nom de « ressources ». Ainsi les ingénieurs linguistes n'ont plus besoin de recréer continuellement les mêmes ressources. Il y a trois sortes de ressources [CUN 02b] :

- les Ressources Linguistiques, ou « Linguistic Resource », qui représentent les entités telles que les lexiques et les corpus ;
- les Ressources Algorithmiques, ou « Processing Resource », qui représentent les entités algorithmiques comme les parsers, les générateurs, les analyseurs, etc.
- les Ressources Visuelles, ou « Visualisation Resource », qui représentent les composants d'édition et de visualisation utilisés par les interfaces utilisateurs.

Toutes les ressources font parties de CREOLE, i.e. « a Collection of REusable Objects for Language Engineering ». Elles peuvent être locales ou bien distribuées et disponibles sur le Web. Toutes peuvent être étendues par les utilisateurs sans modification de GATE lui-même. La séparation entre les ressources algorithmiques et les ressources linguistiques permet l'indépendance de leur développement respectif par des utilisateurs ayant différentes expertises (informaticiens ou linguistes). Similairement, séparer les ressources linguistiques de leur visualisation leur permet de développer des ressources visuelles alternatives [CUN 02b].

Parmi les ressources algorithmiques fournies par GATE, le plugin ANNIE pour « A Nearly-New IE system » regroupe un ensemble de ressources pour les principales étapes d'extraction d'information que nous avons évoquées à la précédente section. Ces ressources peuvent aussi être utilisées individuellement ou couplées avec d'autres ressources pour la création de nouvelles applications. Par exemple, certaines applications peuvent avoir besoin d'un découpeur de phrases ou d'un étiqueteur morpho-syntaxique sans pour autant nécessiter de ressources plus spécifiques comme la reconnaissance des entités nommées. D'après [CUN 02a], ANNIE se compose :

- d'un découpeur de « tokens », i.e. les mots, les nombres, la ponctuation, etc. (tokenizer) ;
- d'un découpeur de textes en phrases (sentence splitter) ;
- d'un correcteur d'orthographique (orthomatcher) ;
- de divers lexiques sur les prénoms, les villes, les organisations, les jours de la semaine, etc. (gazetteers) ;
- d'un étiqueteur morpho-syntaxique qui est une version modifiée de l'étiqueteur de Brill [] ;
- d'un automate à états finis basé sur le langage d'expressions régulières JAPE, pour « Java Annotation Patterns Engine » [CUN 00], qui permet d'utiliser une chaîne de caractères particulière ou bien les étiquettes (ou annotations) créées par les précédentes ressources (cf. Figure 19) ;
- et enfin de deux solutionneurs de coréférence, l'un basé sur les pronoms et l'autre sur les syntagmes nominaux.

```
Rule: PersonJobTitle
Priority: 20
 (
 {Lookup.majorType == jobtitle}
):jobtitle
(
 {TempPerson}
):person
-->
    :jobtitle.JobTitle = {rule = "PersonJobTitle"},
    :person.Person = {kind = "personName", rule = "PersonJobTitle"}
```

Figure 19. Exemple d'une expression régulière exprimée en JAPE

Lorsque GATE est utilisé pour développer une fonctionnalité de traitement du langage pour une application donnée, le développeur dispose de l'environnement de développement et du cadre de travail pour construire ses propres ressources. L'environnement de développement est utilisé pour la visualisation des structures de données produites et consommées pendant le traitement et pour le débogage, l'évaluation de la performance, etc. A chaque nouvelle application (cf. Figure 20), l'utilisateur sélectionne les ressources de traitement dont il a besoin (tokeniser, étiquetteur morpho-syntaxique, etc.), les place selon l'ordre dans lequel elles doivent être exécutées et choisit les données sur lesquelles elles vont s'appliquer (corpus ou document). Il peut aussi configurer chaque paramètre d'exécution des ressources sélectionnées.

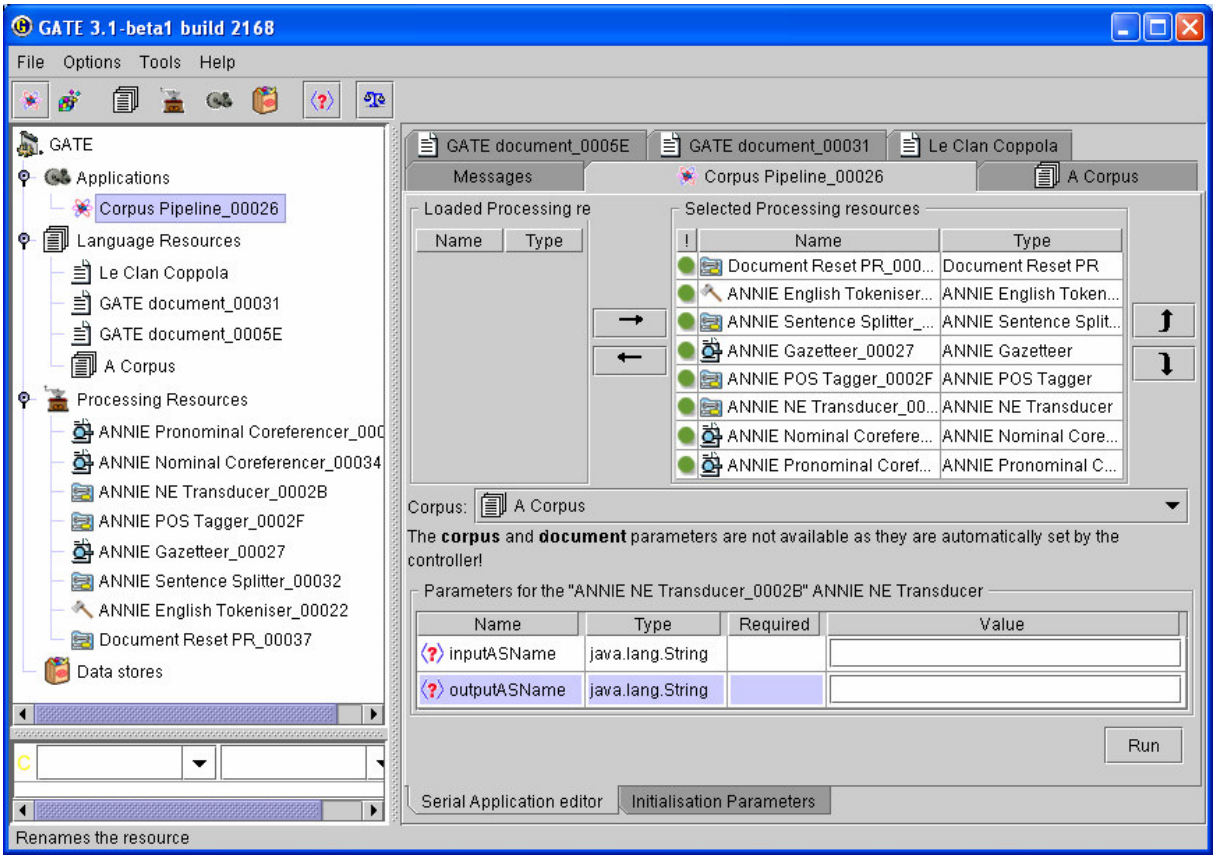


Figure 20. L'environnement de développement d'une application de GATE

Lorsque l'application est lancée, les modules vont être exécutés dans l'ordre spécifié sur le document donné. Les résultats sont alors présentés dans l'interface de visualisation documentaire (Cf. Figure 21). Cette interface est également utilisée pour l'annotation manuelle notamment dans le cadre d'applications basées sur des algorithmes d'apprentissage supervisé. Elle permet aux utilisateurs de constituer le corpus d'entraînement nécessaire à l'apprentissage des règles d'extraction.

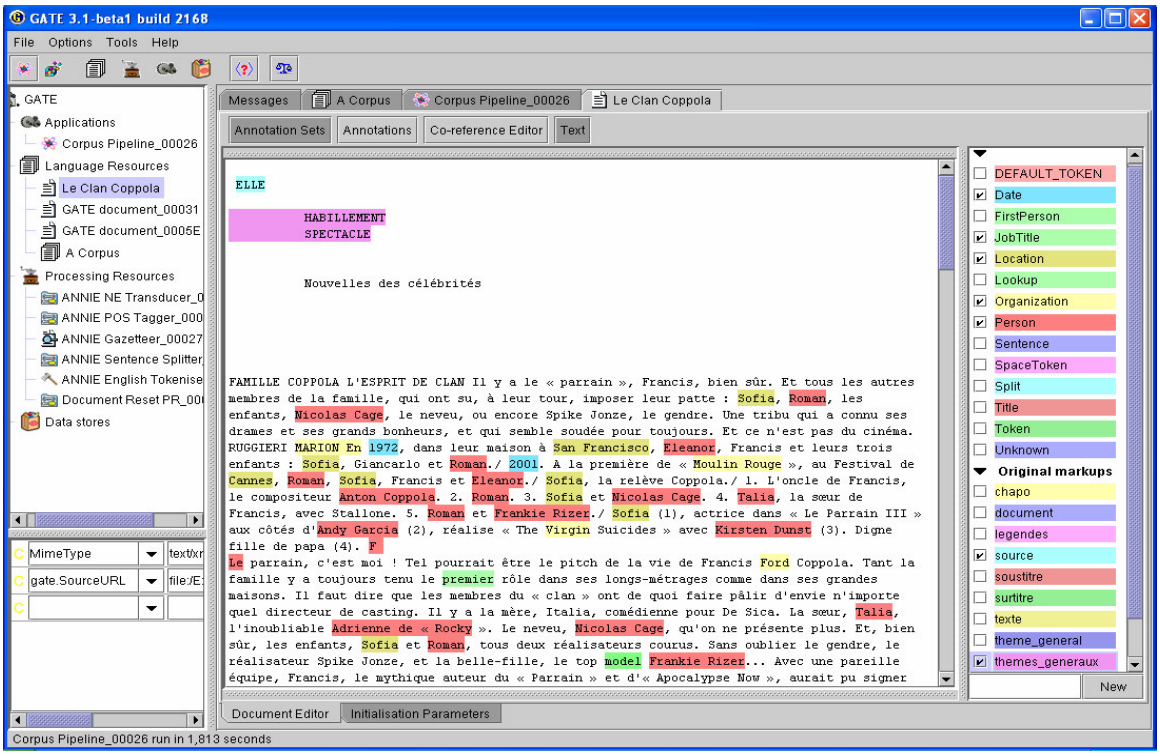


Figure 21. La visualisation des informations extraites et annotées dans GATE

GATE supporte une variété de formats documentaires incluant XML, RTF, HTML, SGML, email et du texte simple. Dans tous les cas, lorsqu'un document est créé/ouvert dans GATE, le format est analysé et converti dans un même modèle unifié d' « annotation » [CUN 02b]. Ce modèle est une structure centrale à GATE, issue du format TIPSTER [GRI 97], qui encode les ressources linguistiques lues et traitées par chacune des ressources de traitement. GATE fournit plusieurs moyens de stocker les informations extraites et annotées par ce modèle : soit par l'utilisation d'une base de données relationnelle comme Oracle, soit leur sérialisation en objet Java ou enfin une sérialisation en fichier XML dans lequel les annotations peuvent être embarquées dans le corps du texte (cf. Figure 22) ou débarquées à la fin de ce dernier. Nous verrons dans la suite de ce chapitre les conséquences de cette représentation pour l'annotation sémantique et le peuplement d'ontologie.

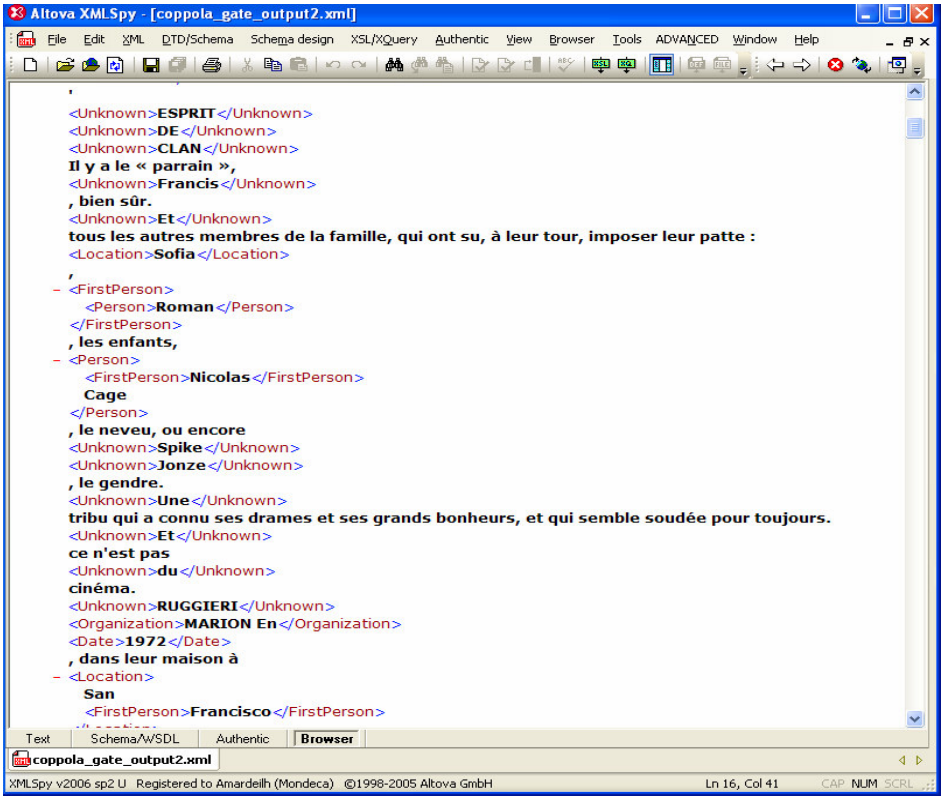


Figure 22. Exemple d'une sérialisation en XML des annotations embarquées générées par GATE

En conclusion, GATE est donc une plate-forme d'ingénierie linguistique qui fournit les ressources nécessaires à la réalisation d'un moteur d'extraction d'information générique grâce au plugin ANNIE. La facilité de sa mise en œuvre explique l'intérêt suscité dans la communauté des chercheurs, tant en informatique linguistique qu'en ingénierie des connaissances. Par contre, pour des moteurs d'extraction d'informatique plus poussés, notamment avec extraction des relations sémantiques et de scénarios complets, la mise en œuvre implique le développement de patrons d'extraction en JAPE. Cette tâche est loin d'être à la portée du tout venant puisqu'elle implique un minimum de connaissances en écriture d'expressions régulières et en programmation Java. C'est pourquoi, pour des applications très concrètes, les entreprises ayant besoin de moteurs d'extraction robustes et spécifiques à leur domaine d'activité se tournent généralement vers des solutions élaborées par des ingénieurs linguistes professionnels, comme celle proposée par l'outil IDE.

2.2.2 Insight Discoverer™ Extractor

L'outil Insight Discoverer™ Extractor (IDE) est commercialisé par la société Temis depuis juin 2002. C'est un serveur d'extraction d'information dédié à l'analyse de ressources textuelles capable de traiter 16 langues, principalement européennes. Il est spécialisé dans les domaines de la veille économique ou scientifique ainsi que dans le domaine pharmaceutique [GRI 01a]. Son approche est fondée sur l'acquisition de connaissances à partir d'un corpus selon un processus itératif. Elle exploite les complémentarités des différentes étapes de l'analyse linguistique (morphologique, syntaxique et

sémantique) à partir des étiquettes générées à chacune de ces étapes. L'IDE peut être employé pour une simple extraction d'éléments syntaxiques comme les noms, les verbes, etc. ou bien pour une extraction d'éléments sémantiques tels que des noms de sociétés, des noms de lieux, des dates, des prix,… et des relations sémantiques (fusion de X avec Y, achat de W par Z).

L'architecture de l'IDE repose sur le moteur XeLDA™ pour l'analyse morpho-syntaxique, quelle que soit la langue utilisée. Ce moteur XeLDA™ est un étiqueteur grammatical qui découpe chaque texte en unités lexicales puis lemmatise ces unités lexicales pour qu'elles soient reconnues indépendamment de leurs formes fléchies. Enfin, il assigne à ces unités lexicales une catégorie grammaticale (nom, adjectif, verbe…) assortie de traits morphosyntaxiques (genre, nombre). Pour réaliser cette analyse, le moteur XeLDA™ dispose de diverses ressources comme des dictionnaires, des règles morphologiques ainsi que des modèles statistiques (utilisation des chaînes de Markov) pour résoudre les ambiguïtés concernant l'affectation des catégories grammaticales. Ces sources ne peuvent être modifiées par l'utilisateur, elles sont identiques quel que soit le domaine concerné. La sortie de l'analyseur morphosyntaxique se compose donc de la forme fléchie de chaque unité textuelle, de son lemme, de sa catégorie grammaticale ainsi que de sa position dans le texte (point d'entrée plus longueur de la chaine de l'unité textuelle considérée).

Puis vient l'analyse sémantique pour laquelle les ingénieurs linguistes de la société définissent des cartouches linguistiques, les Skill Cartridges™, en fonction du domaine concerné et des attentes de l'application finale. D'après [GRI 01a], une Skill Cartridge™ est une hiérarchie de « *composants de connaissance* », appelés « Skill Units », décrivant l'information à extraire pour un métier, une activité, un domaine spécifique ou une thématique donnée. Un composant de connaissance peut avoir la forme d'un dictionnaire ou d'un ensemble de règles d'extraction qui décrivent l'information à extraire. Les dictionnaires peuvent exploiter des ressources existantes comme WordNet[30] [FEL 98] pour l'anglais ou MemoData[31] [DUT 03] pour le français. Les règles d'extraction se composent d'un ensemble de patrons d'extraction contextuels, combinant lemmes, étiquettes syntaxiques et étiquettes sémantiques (cf. Figure 23). Chaque règle associe ensuite une nouvelle étiquette, par exemple « actor_in_agribusiness » dans la figure ci-dessous, au fragment de texte repéré. Cette étiquette peut ensuite être utilisée dans de nouvelles règles, et ainsi de suite.

```
<actor_in_agribusiness>

        ;; a seed and flour miller
(company_Adj|Loc_Adj|#OD)* / (NOUN)* / (food_|brand_product) / (actor|company)

       ;; maker of private label pasta
(company_Adj|Loc_Adj|#OD)* / (actor|company) / of / (food_|brand_product)
```

Figure 23. Exemple d'une expression régulière dans IDE, combinant étiquettes syntaxiques (« NOUN ») et étiquettes sémantiques (« brand_product »)

[30] http://wordnet.princeton.edu/
[31] http://www.memodata.com/2004/fr/index.shtml

Les linguistes de Témis ont pour consigne de suivre les étapes de construction suivantes pour tout développement d'une nouvelle cartouche linguistique [GRI 01b] :

1) l'analyse morpho-syntaxique des textes qui constituent le corpus d'entraînement et la définition du vocabulaire pertinent relatif au secteur d'activité ou au domaine économique étudié (agroalimentaire, automobile, etc.) ;
2) le regroupement des termes ainsi définis sous des étiquettes sémantiques elles-mêmes organisées, selon les besoins, en une hiérarchie simple de 3 à 4 niveaux généralement ;
3) la définition des règles d'extraction d'information niveau par niveau hiérarchique afin d'aboutir aux concepts visés : les acteurs du domaine d'activité étudié, leurs relations et toute information relative aux lieux, au temps ou à une donnée particulière, ou attribut, liée au domaine comme un montant financier par exemple ;
4) l'exécution interactive des règles d'extraction sur le corpus d'entraînement afin d'évaluer le résultat des extractions. Il est alors possible d'enrichir le vocabulaire, de modifier des règles d'extraction et d'en vérifier l'impact sur le corpus de travail.

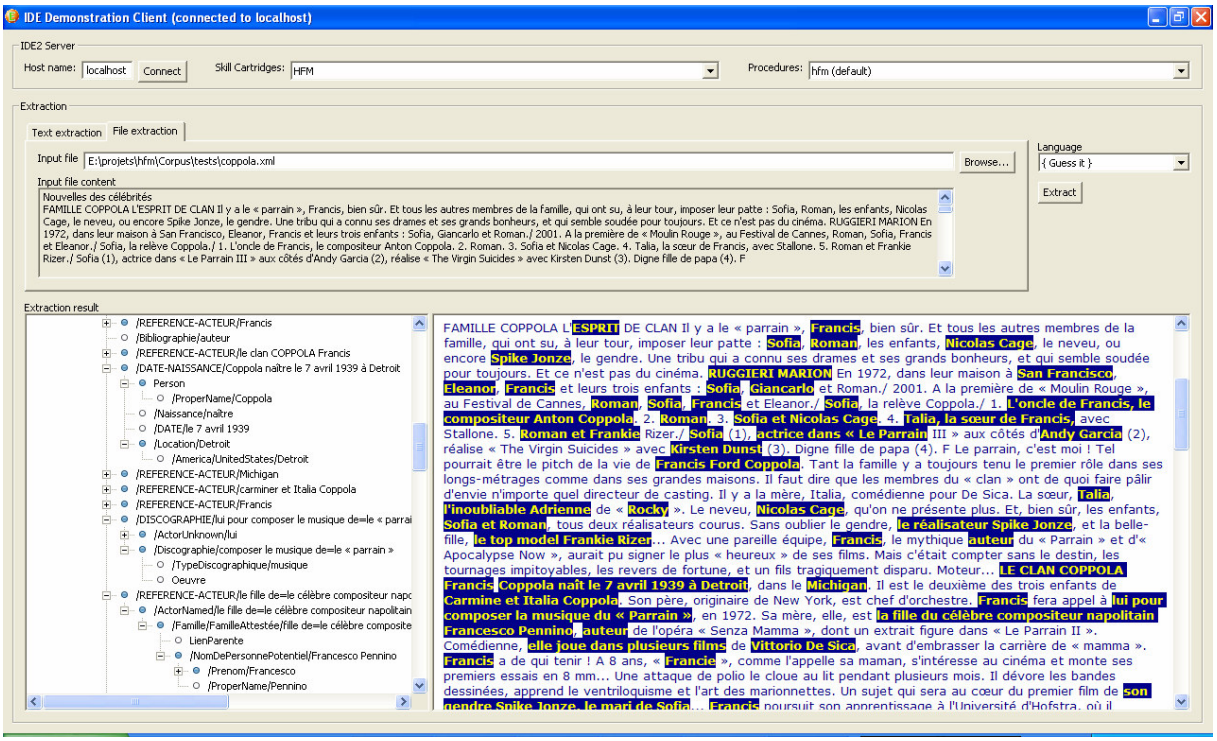


Figure 24. La visualisation des informations extractions et annotées dans IDE

Cette méthodologie de construction des règles d'extraction basée sur la notion d'organisation des règles d'extraction par niveaux hiérarchiques permet de mieux contrôler leur exécution sur les corpus [GRI 01b]. IDE utilise la règle classique « le plus à gauche, le plus long » pour savoir quelle séquence de mots associer aux patrons candidats. Ainsi, pour chaque segment textuel, l'algorithme cherche parmi les patrons celui qui correspond le mieux. Mais les patrons n'étant pas tous de même niveau (du plus général au plus spécifique), ils ne s'appliquent pas tous au même moment. Une information extraite à un niveau hiérarchique donné encapsule les unités textuelles qui la composent, les rendant inaccessibles aux niveaux supérieurs. Par contre, un mot isolé qui ne participe pas à la construction

d'un patron reste disponible pour l'application d'un autre patron en changeant de niveau hiérarchique. L'algorithme utilisé par l'IDE est décrit plus en détail dans [GRI 01a]. L'ensemble des patrons d'extractrion sont compilés dans un transducteur [HOB 97], la cartouche linguistique, qui applique d'abord les patrons de reconnaissance des entités nommées, puis les patrons indépendants du domaine et qui s'appuient sur l'analyse morphosyntaxique et, enfin, les patrons spécifiques au domaine de l'application pour laquelle ils ont été conçus avec, notamment, le repérage des relations sémantiques entre les entités précédemment extraites.

Le serveur d'extraction d'IDE prend en charge les ressources documentaires aux formats tels que le texte simple (txt ou rtf), MS Word, pdf, html, xml, etc. Il applique automatiquement la cartouche linguistique définie pour l'application cible. A l'issue du traitement, il génère un objet Java, appelé « arbre conceptuel », représentant l'organisation hiérarchique des étiquettes sémantiques apposées en fonction de la structure de l'information extraite du texte analysé (cf. Figure 25). Si cet arbre conceptuel Java correspond à un format propriétaire à l'IDE, sa sérialisation dans un fichier XML en fait un objet tout à fait exploitable par d'autres applications.

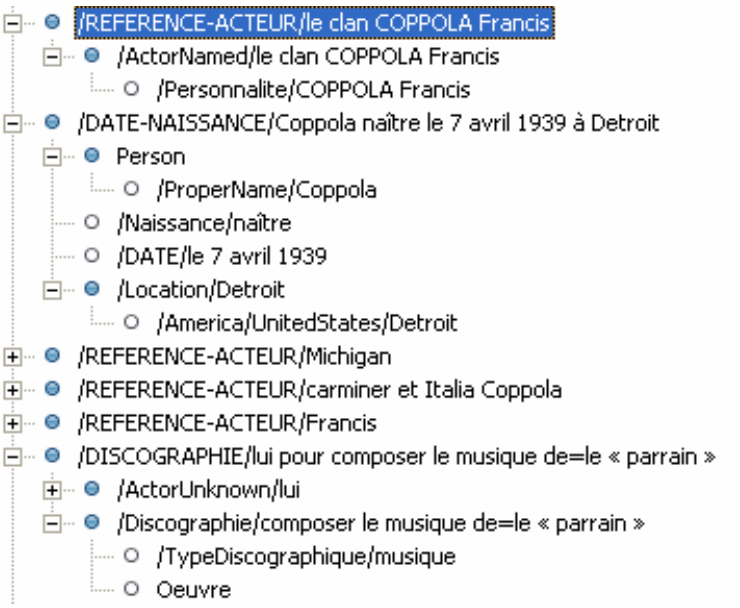


Figure 25. Extrait d'un arbre conceptuel généré par IDE

En conclusion, GATE et IDE sont deux outils de conception différente. Le premier est plus orienté vers la communauté scientifique puisqu'il a pour objectif de fournir les ressources nécessaires à la construction d'une application d'extraction d'information pour un ingénieur linguiste. A contrario, IDE est conçu pour l'industrie puisqu'il est livré à des entreprises avec des applications clef en main en fonction de leur domaine d'activité et de leur métier. Ces applications sont développées par la propre équipe d'ingénieurs linguistes de Témis. Néanmoins, ils réalisent tout deux une analyse linguistique plus ou moins fine sur laquelle repose les patrons d'extraction construits manuellement. Au final, les informations sont extraites similairement et annotées par des étiquettes sémantiques constituant un

arbre conceptuel, même si GATE ne l'appelle pas ainsi. En effet, la sérialisation des annotations produites par GATE en fichier XML est équivalente à la représentation du texte sous la forme d'un arbre conceptuel. Par ailleurs, nous avons vu au §1.3.2 que le langage XML se trouve à la base de la pyramide des langages recommandés par le W3C. Or, en ce qui concerne l'annotation sémantique et le peuplement d'ontologie, nous cherchons à atteindre les niveaux supérieurs : RDF pour l'annotation et OWL pour le peuplement d'ontologie. Dans la section suivante, nous allons donc étudier plus en détail la structuration des arbres de concepts générés afin de comprendre comment ils peuvent être transformés en une représentation plus formelle du contenu.

2.3 Réflexion sur la représentation en arbre conceptuel

Nous allons illustrer les différentes caractéristiques d'un arbre conceptuel à l'aide de la Figure 26. Cet exemple d'arbre conceptuel a été généré à partir d'un extrait de l'article sur la biographie de « Francis Ford Coppola », déjà présenté au chapitre précédent. Une vue complète ainsi qu'une analyse phrase par phrase de cet arbre conceptuel généré par l'IDE est fournie dans l'annexe II de ce mémoire.

Nous adoptons ce terme « **arbre conceptuel** » pour décrire les résultats des moteurs d'extraction bien qu'ils ne correspondent pas véritablement à un arbre « de concepts », au sens ontologique du terme. Néanmoins, l'étiquetage sémantique opéré par les règles d'extraction apporte un premier niveau de conceptualisation du contenu représenté hiérarchiquement sous la forme de cet arbre. En fait, l'arbre conceptuel se trouve à mi-chemin entre un arbre syntaxique et un graphe conceptuel : il a été construit à partir de l'arbre syntaxique issu de l'analyse morpho-syntaxique auquel ont été rajoutées les étiquettes sémantiques produites au fur et à mesure du déclenchement des patrons d'extraction des différents niveaux de l'analyse sémantique.

Sa racine représente généralement le document analysé par le moteur d'extraction. Chaque nœud de l'arbre est constitué d'une étiquette sémantique (préfixée par le symbole « / ») et de la valeur (indiquée entre parenthèses) de l'unité textuelle du texte à laquelle cette étiquette a été affectée. D'autres informations, telles que le lemme de cette unité textuelle ou sa position dans le texte, peuvent aussi être associées aux étiquettes sémantiques si besoin est. Dans la Figure 26, la racine du document porte l'étiquette sémantique « /article » et a pour valeur son titre : « Famille Coppola, l'esprit de clan ». Chaque sous-arbre correspond soit à une entité nommée isolée soit à une phrase ou même plus généralement à une proposition du document analysé.

Premièrement, les sous-arbres représentant les entités nommées isolées sont généralement formés à partir de deux schémas d'extraction :

1) les entités nommées identifiées de manière « certaine » à partir d'un lexique ou de toute autre ressource linguistique utilisée par le moteur d'extraction ;
2) les entités nommées identifiées de manière « potentielle » à partir de l'application d'un patron d'extraction compilé dans le moteur d'extraction.

Dans le premier sous-arbre de la Figure 26, l'étiquette « /REFERENCE-ACTEUR » représente l'entité nommée isolée dont la valeur est « Francis Ford Coppola ». Ce nœud possède un nœud fils « /ActorNamed », qui signifie pour IDE qu'il a été repéré à partir d'un lexique, lui-même ayant pour nœud fils « /Personnalite » qui indique le lexique dont a été extraite l'entité nommée « Francis Ford Coppola ». A l'inverse, dans le deuxième sous-arbre de la Figure 26, si l'entité nommée « Spike Jonze » a pour racine la même étiquette que l'entité nommée précédente, son nœud fils « /NomDePersonnePotentiel » indique que son étiquette sémantique a été calculée à partir du déclenchement d'un certain patron d'extraction pour la reconnaissance des noms de personnes construit à partir des étiquettes « Prenom » et « ProperName ». Le mélange des étiquettes ayant un label français ou anglais n'est pas un hasard. Les ingénieurs linguistes réutilisent souvent des ressources linguistiques d'applications différentes pour éviter de réécrire tous les patrons à chaque nouvelle application. Dans notre cas, l'étiquette « ProperName » a surement été réutilisée à partir d'une précédente cartouche linguistique.

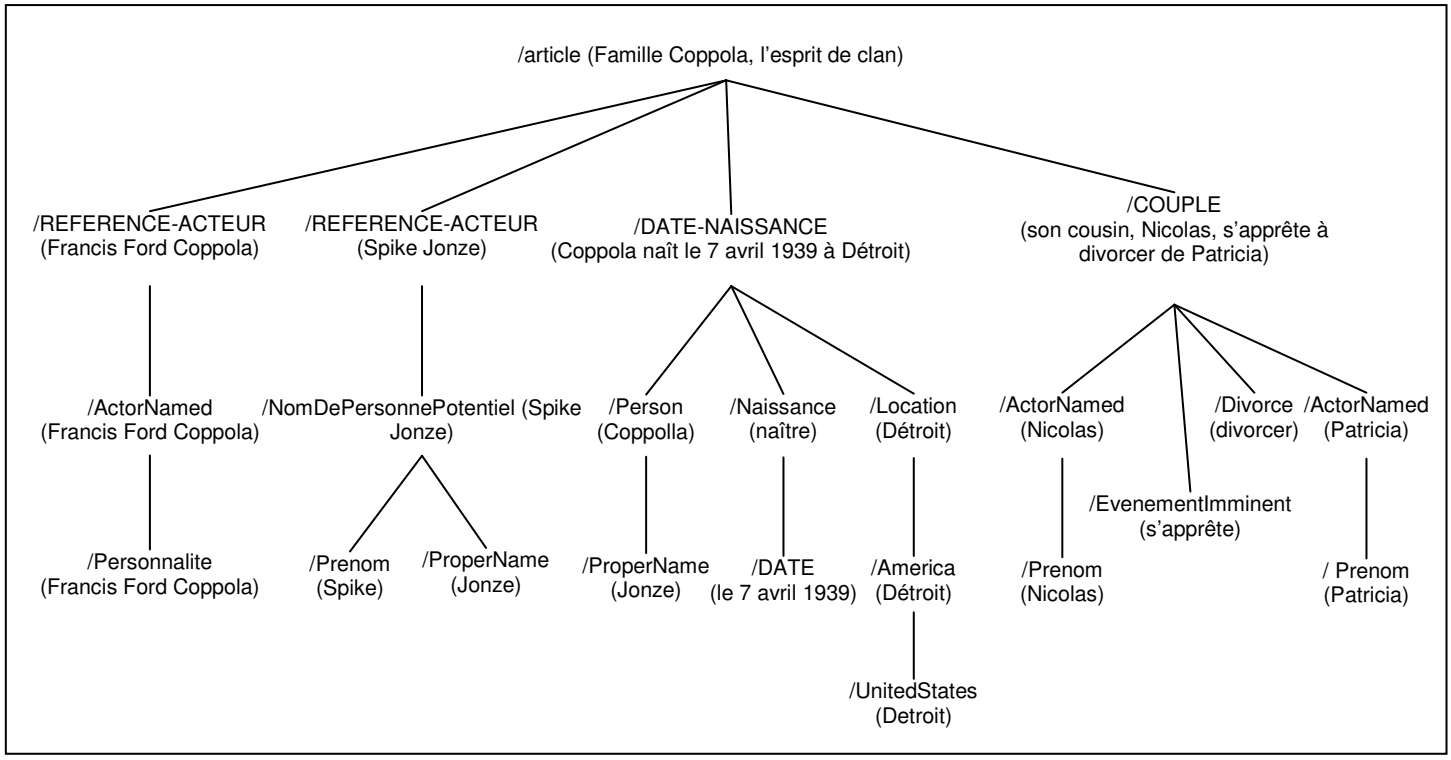


Figure 26. Extrait de l'arbre conceptuel produit par l'analyse linguistique de l'article "La tribu Coppola" publié dans le magasine Elle

Deuxièmement, une proposition n'est représentée sous la forme d'un sous-arbre que si elle est composée d'unités textuelles ayant donné lieu à une extraction. De la même manière, certaines unités textuelles de la proposition peuvent ne pas apparaître dans le sous-arbre : seules celles ayant été étiquetées sémantiquement par les patrons seront représentées par un nœud du sous-arbre. Ces sous-arbres correspondent à l'interprétation de la proposition pour répondre aux tâches de reconnaissance d'un attribut, d'une relation, voire d'un scénario, comme définies par les conférences

MUC (cf. §2.1.1). D'après l'analyse des sous-arbres, on peut résumer les différentes situations comme suit :

- Lorsqu'ils contiennent une seule référence à une entité nommée, alors la tâche correspond à l'identification d'un attribut pour celle-ci.
- Lorsqu'ils contiennent deux étiquettes correspondant à des entités nommées, alors il s'agit de la tâche d'instanciation d'une relation entre ces entités nommées.
- Lorsqu'ils contiennent plusieurs entités nommées, cela signifie qu'un scénario a été clairement identifié au sein de la proposition.

Par exemple le sous-arbre portant l'étiquette « /DATE-NAISSANCE » dans la Figure 26 correspond à l'extraction d'un scénario complet correspondant à la naissance d'une personnalité. Ce scénario est constitué des entités nommées « Personnalité », « Date » et « Lieu », respectivement représentées par les étiquettes « /Person », « DATE » et « Location ». Il faut savoir également que, généralement, la racine du sous-arbre est identifiée par une étiquette correspondant à la sémantique verbale de la proposition. Comme le soulignent Enjalbert et Victorri dans [ENJ 05c], ce sont avant tout les constructions verbales qui définissent et relient les entités et événements d'une phrase tout en leur assignant des rôles précis. Mais les constructions basées sur les syntagmes nominaux sont aussi vecteurs de relations entre entités [ENJ 05c] et dans ce cas, l'étiquette identifiant le sous-arbre ainsi généré sera liée à la sémantique du noyau nominal du syntagme. Le dernier sous-arbre de la Figure 26 représente une relation entre deux entités nommées. Il porte l'étiquette « /COUPLE » car le verbe extrait « divorce » appartient à la catégorie des verbes indiquant une relation de couple, comme « marier, unir, fiancer, rompre, divorcer, … ».

On peut dire que les propositions s'organisent en « Faits », c'est-à-dire, comme le décrit Van Dijk, que les « *représentations cognitives des faits que nous identifions dans la perception et la compréhension du monde* » [VAN 85]. Autrement dit, chaque proposition, ou sous-arbre, constitue un fait qui représente la structure schématique d'un événement, d'une action, d'un état, etc. En fait, on pourrait rapprocher cette structure de la grammaire des cas proposée par Fillmore [FIL 68] même si celle-ci a depuis montré ses limites. Fillmore s'est intéressé à l'identification de « *l'ensemble des cas sémantiques permettant de mettre en évidence, à la manière des cas syntaxiques, les relations de sens qui existent entre les noms (ou les groupes nominaux) et le verbe dans une phrase simple* » [SAB 90]. Ainsi, chaque phrase peut être représentée en distinguant la proposition et ses modalités. La proposition est constituée d'un verbe d'où partent des relations étiquetées sémantiquement et correspondant aux différents « cas ». Les modalités concernent les informations qui situent et spécifient la proposition comme la négation, le mode, le temps, l'aspect, etc. Il suppose que pour chaque verbe, il existe seulement un nombre restreint de cas parmi les suivants [SAB 90] :

- AGENT : l'instigateur animé d'une action
- INSTRUMENT : la force inanimée ou l'objet affecté
- DATIF : l'animé affecté par l'action
- FACTITIF : l'objet résultat de l'action
- LIEU : le lieu ou l'orientation

- OBJET : l'entité qui bouge, change ou dont la position ou l'existence est en question

Par la suite, Fillmore complètera ces cas avec d'autres comme le contre-agent (la force contre laquelle l'action est exécutée), le résultat (l'entité créée), la source (lieu de départ de quelque chose qui bouge), le but (lieu d'arrivée de quelque chose qui bouge) ou encore le patient (l'entité qui reçoit, accepte ou subit les effets d'une action) [SAB 90].

```
/CI_Acquisition (France Telecom said it would acquire a 54.3% stake in
Equant)
      who (FranceTelecom)
            /Telecom_Operator (FranceTelecom)
      which_information (say it would acquire a 54.3 stake)
            Announcement (said)
            Rumor (would)
            /CI_Financial(acquire a 54.3 stake)
                  /buying_stake (acquire a 54.3 stake)
                        /buying_acquisition (acquire)
                        /CI_Financial (a 54.3% stake)
                              /stake (a 54.3% stake)
                                    how_much_percent (a 54.3%)
      whom (Equant)
            /Telecom_Operator (Equant)
```

Figure 27. Exemple d'un arbre conceptuel au sujet d'une prise de participation de France Telecom dans la société Equant

La relation entre grammaire des cas et arbre conceptuel est encore plus évidente dans le domaine de la veille économique, bien connu des moteurs d'extraction d'information [WER 05]. Il a notamment fait l'objet des évaluations des MUC-5 et MUC-6 [GRI 96]. Les scénarios de ce domaine sont généralement bien identifiés et structurés en fonction des différentes informations à extraire correspondant aux questions suivantes [GRI 01b] :

- Qui sont les acteurs du secteur d'activité ou du domaine économique étudié ?
- Quels sont les objets considérés relatifs au domaine décrit ?
- Quelles sont les actions de ces acteurs, sur quels objets portent-elles et comment s'effectuent-elles ?
- Où ont lieu les actions en question ?
- Quand ont-elles eu lieu ?
- Quel est le montant de ces actions ?

La Figure 27 nous montre l'exemple d'un sous-arbre conceptuel construit automatiquement à partir d'une proposition dans laquelle ont été clairement identifiés : les acteurs par les étiquettes « /who » et « /whom », l'action par l'étiquette « /buying_acquisition », l'objet par l'étiquette « /stake » et enfin le montant par l'étiquette « /how_much_percent ». Certaines modalités sont également identifiées par des étiquettes sémantiques comme le fait de préciser que ce scénario extrait est une rumeur, cf. « /Rumor », d'annonce, cf. « /Announcement ». Cette proposition peut également être analysée du point de vue de la grammaire des cas et être représentée comme à la Figure 28.

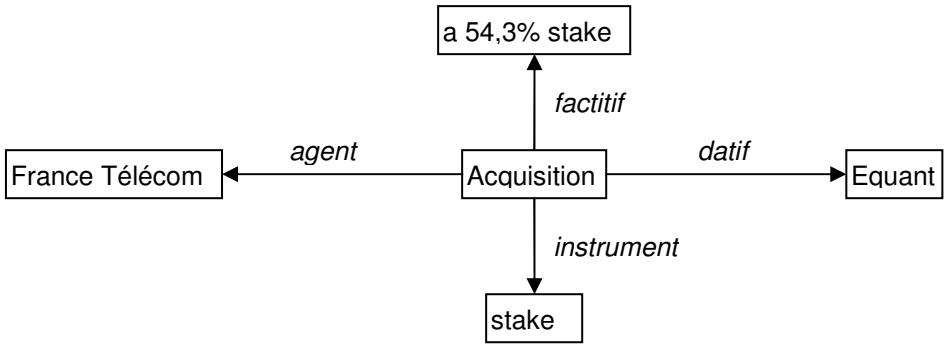


Figure 28. Application de la grammaire des cas à la proposition de la figure précédente.

Mais cette grammaire des cas ne peut correctement représenter les phrases plus complexes où, comme-ci-dessus, on a des emboîtements de propositions les unes dans les autres. Elle convient surtout aux phrases dites simples [SAB 90]. L'arbre conceptuel est capable de représenter hiérarchiquement des constructions plus complexes comme par exemple dans le dernier sous-arbre de la Figure 26 où le syntagme nominal « son cousin, Nicolas » se décompose en deux syntagmes nominaux : le premier, « mon cousin », est représenté par l'étiquette sémantique « /Famille » et le second « Nicolas » identifié par l'étiquette « /Prenom ».

De leur côté, Kintsch et Van Dijk [VAN 83] se sont inspirés de la grammaire des cas pour étudier la construction de propositions plus complexes dans le cadre d'un modèle d'organisation de la mémoire sémantique [COI 96]. La proposition reste l'unité de base mais Kintsch la définit comme une *« unité composite », i.e. une unité de signification qui contient un ou plusieurs arguments* » [VAN 83]. Parmi ces arguments, on retrouve les entités référentielles, comme les êtres, les objets physiques ou abstraits, etc. mais également d'autres propositions elles-mêmes. Des prédicats, comme « acquérir(France Télécom, Equant) », assignent des propriétés aux différents arguments ou définissent des relations entre eux. La Figure 29 montre la proposition de la Figure 27 sous la forme « *schéma propositionnel* » d'après le modèle proposé par Kintsch et Van Dijk [VAN 83].

```
PROP
       PRED acquire (a, b, c)
              MOD would
       ARG
              AGENT France Telecom = a
              OBJECT stake(b)
                     MOD 54,3%
              GOAL   Equant = c
       CIRC
              TIME
              PLACE
```

Figure 29. Analyse de la proposition précédente par le modèle propositionnel de Kintsch & Van Dijk

Contrairement à l'arbre conceptuel qui conserve l'ordre imposé par la structure syntaxique du texte analysé, ce formalisme réorganise les différents nœuds de l'arbre représentants la proposition (PROP) en prédicat (PRED), arguments (ARG) et circonstanciels (CIRC). L'ensemble de ces schémas

constituent une « *microstructure* » [VAN 83] qui décrit la signification du texte par un réseau de propositions hiérarchisées. Dans cette microstructure, chaque phrase peut être traitée de manière relativement indépendante, les éventuelles liaisons étant établies par la référence à des arguments partagés entre prédicats. Cette microstructure constitue une représentation intermédiaire entre l'arbre conceptuel, qui organise les propositions indépendamment les unes des autres dans une structure hiérarchisée, et une représentation logique utilisant des prédicats qui déclarent et référencent les mêmes arguments entre des propositions différentes.

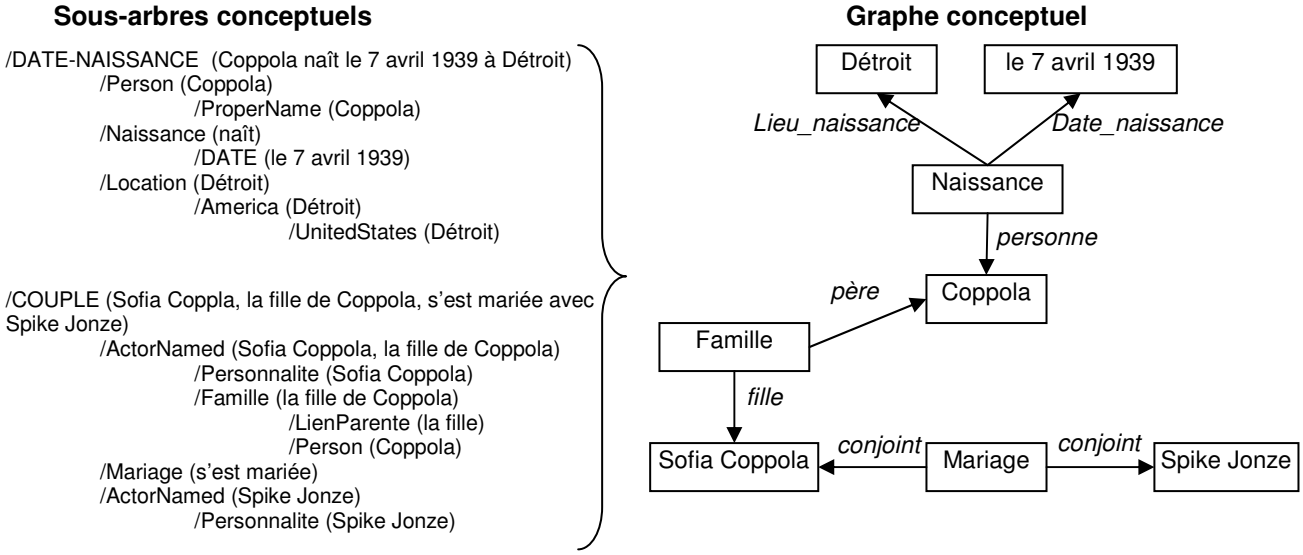


Figure 30. Transformation de deux sous-arbres conceptuels en un graphe conceptuel où la même entité nommée « Coppola » fait le lien entre les deux propositions

Par conséquent, pour qu'un arbre conceptuel issu de l'analyse linguistique produise une représentation plus globale de la signification du texte originel, il faut pouvoir identifier et référencer les entités nommées identiques entre elles au delà des différents sous-arbres des propositions. Comme illustré à la Figure 30, nous obtiendrons alors une représentation sous la forme d'un graphe conceptuel qui est un premier pas vers une représentation ontologique comme nous l'avons vu dans la section 1.2.1.

Or, le référencement inter-entités nommées soulève le problème de la coréférence entre les différentes propositions [ENJ 05]. Dans l'exemplede la Figure 26, le syntagme nominal « son cousin » fait référence à la phrase précédente dans laquelle Coppola est clairement identifié. Ainsi, le lecteur peut comprendre que Nicolas est le cousin de Coppola, d'où une relation de parenté entre ces deux entités nommées. Or, sans l'ajout dans l'arbre d'une information identifiant clairement la coréférence par le moteur d'extraction, il sera impossible de relier les deux entités dans l'état actuel du sous-arbre présenté à la Figure 26. Par ailleurs, les entités nommées non identifiées (comme les pronoms « il », « elles » ou les syntagmes nominaux « l'acteur », « le jeune réalisateur américain », etc. ») sont également potentiellement intéressantes pour reconstituer la représentation globale du texte entre propositions et permettre la construction d'un graphe conceptuel associé. Il s'agit des anaphores pronominales ou nominales dont le calcul est un des problèmes majeur en TALN [VIC 05]. Sur les

deux outils présentés précédemment, seul GATE possède quelques ressources pour résoudre les problèmes soulevés par la coréférence et il est, de manière générale, très difficile de trouver des moteurs d'extraction sachant résoudre ces situations avec un taux de réussite acceptable [CUN 00].

2.4 Conclusion

Après un rappel succinct de la problématique de l'Extraction d'Information, nous avons présenté deux moteurs d'extraction, l'un orienté vers la recherche, GATE, et l'autre vers les applications industrielles, l'IDE. Ces outils ne reposent pas sur une analyse linguistique complète du texte ou du document mais plutôt sur le repérage de fragments de phrases jugées pertinentes pour le domaine concerné de par la présence de certaines entités nommées ou expressions régulières clefs. L'analyse de la coréférence reste difficile à mettre en œuvre dans les applications car les outils actuels ne produisent pas des résultats acceptables ou exploitables par ailleurs.

Puis, nous avons détaillé les caractéristiques relatives aux arbres conceptuels, résultats des moteurs d'extraction d'information. S'il est possible de transformer la structure hiérarchique de ces arbres en prédicats logiques ou en graphes conceptuels grâce au référencement inter-propositions des entités nommées identiques, il nous faut également faire correspondre les différentes étiquettes sémantiques de l'arbre conceptuel avec les éléments (concepts, relations et attributs) de l'ontologie de référence. Ces étiquettes sont habituellement créées par les ingénieurs linguistes lors du développement des patrons d'extraction alors que les éléments de l'ontologie préexistent ou sont modélisés par un expert en représentation des connaissances et notamment en ingénierie ontologique. Par conséquent, non seulement il est nécessaire de correctement interpréter la sémantique fournie par les arbres conceptuels mais également de prendre en considération le décalage qui peut exister entre les deux modes de représentation de la connaissance.

Dans le prochain chapitre, nous allons exposer notre solution : les Règles d'Acquisition de Connaissance. Ces règles permettent notamment de coupler une ou plusieurs étiquettes sémantiques d'un arbre conceptuel avec un élément de l'ontologie de référence. Elles s'appuient notamment sur le contexte des étiquettes sémantiques dans l'arbre conceptuel afin de résoudre un certain nombre d'ambiguïtés que nous détaillerons.

Deuxième partie.

Notre Démarche,

OntoPop

Chapitre 3. Au cœur d'OntoPop : les Règles d'Acquisition de Connaissance

Nous venons de voir que les arbres conceptuels ne constituent pas une représentation suffisamment « sémantisée » du contenu textuel. Malgré la conceptualisation de certaines formes, ils ne résultent que d'une analyse de surface du texte qui ne représente pas la signification globale du contenu mais plutôt une vue linéaire, fragmentée en fonction de certaines unités textuelles, comme les propositions ou les syntagmes nominaux. Il est donc absolument nécessaire de mettre en place un mode d'interprétation de ces arbres conceptuels afin de pouvoir proposer une représentation formelle du contenu, basée sur une ontologie de domaine. Nous proposons donc une passerelle entre ces deux modes de représentation qui puisse être indépendante des outils utilisés, que ce soit les moteurs d'extraction ou l'outil de représentation des connaissances. Cette passerelle constitue le cœur de la démarche OntoPop, de sa méthodologie et de sa plateforme logicielle, présentés dans les prochains chapitres de ce mémoire. La passerelle prend la forme d'un format intermédiaire de représentation couplé aux technologies issues du Web Sémantique. Ce format permet de définir un ensemble de Règles d'Acquisition de Connaissance.

Dans la suite de ce chapitre, nous présentons ces Règles d'Acquisition de Connaissance (RAC). Nous allons d'abord expliquer la nécessité de cette passerelle entre les outils, puis montrer que les extractions ne sont pas si simples à appréhender et que le contexte des étiquettes linguistiques est primordial à l'analyse. Nous définirons alors la notion de Règles d'Acquisition de Connaissance, leur formalisme par la grammaire du langage OPAL (Ontology Population and Annotation Langue) et leur implémentation dans la suite logicielle d'OntoPop.

3.1 Une passerelle pour l'annotation sémantique et le peuplement d'ontologie

Dans les outils actuels d'annotation sémantique (cf. section 1.4), l'automatisation, complète ou partielle, du traitement se fait grâce à l'intégration d'un outil d'extraction d'information. Dans les solutions proposées, ces outils sont étroitement liés et dépendants du couplage réalisé entre les deux modes de représentation du contenu. Par exemple, OntoMat dans sa version S-CREAM [HAN 05], reconnaît que son couplage avec l'outil Amilcare est réalisé de manière « ad hoc » et spécifique à la sortie produite par Amilcare. Ce couplage ne pourrait être ré-exploité pour l'intégration avec un autre outil d'extraction d'information.

Or, il nous semble qu'un outil d'annotation sémantique et/ou de peuplement d'ontologie devrait pouvoir facilement changer d'outil d'extraction d'information en fonction des besoins de l'application cible. Par exemple, dans un cas nécessitant l'exploitation des coréférences, l'outil d'annotation

pourrait être couplé avec GATE alors que pour l'exploitation de scénarios complexes dans un domaine d'application précis, l'outil d'annotation serait plutôt à coupler avec l'IDE.

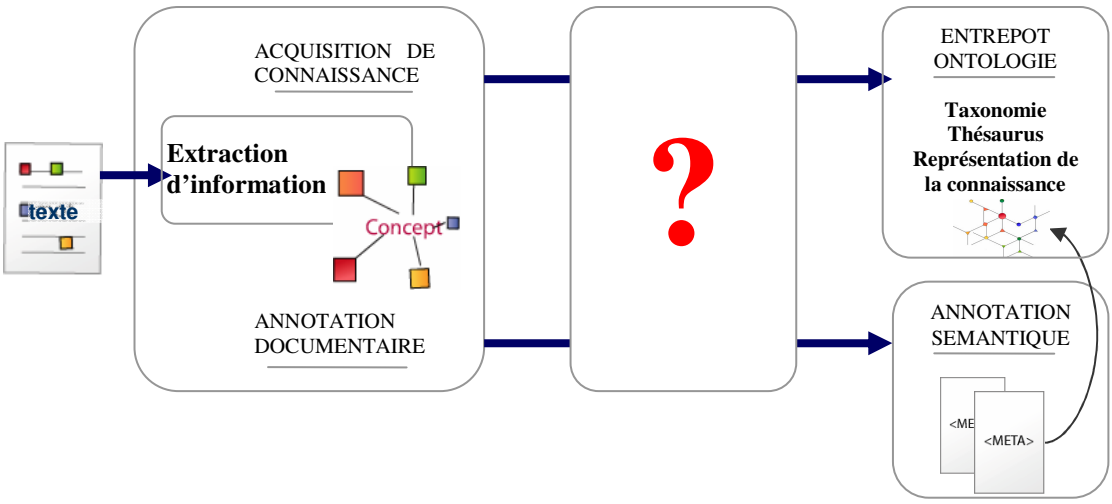


Figure 31. Le fossé entre la représentation textuelle et la représentation sémantique

Je pense donc que l'outil d'annotation sémantique, ou de peuplement d'ontologie, doit rester indépendant du système d'extraction d'information utilisé afin de proposer aux futures applications davantage de flexibilité et de modularité. Mais pour cela, il s'agit de trouver une solution générique au fossé existant entre les arbres conceptuels (tels que présentés au chapitre précédent) et la représentation formelle du contenu comme modélisée dans les ontologies (cf. Figure 31). Par conséquent, il est nécessaire de concevoir un niveau intermédiaire entre ces deux représentations qui permette le couplage des outils d'EI avec les outils d'annotation et/ou de peuplement d'ontologie. En d'autres termes, comment faire correspondre une certaine représentation du contenu d'un texte avec une représentation sémantique de la connaissance, qu'elle soit une annotation sémantique (énoncés RDF) ou une nouvelle instance de l'ontologie (OWL, XTM) ?

Une des spécificités de notre démarche, appelée OntoPop pour « Ontology Population », consiste à réaliser cette correspondance grâce à la déclaration d'un ensemble de Règles d'Acquisition de Connaissance (RAC). Elles s'appuient à la fois des étiquettes sémantiques produites par les arbres conceptuels des outils d'extraction d'information et des éléments (concepts, attributs et relations) modélisés dans l'ontologie. Elles décrivent la manière dont un concept de l'ontologie va être instancié ou utilisé pour l'annotation documentaire. Pour ce faire, elle identifie l'étiquette sémantique qui déclenchera le processus d'instanciation ou d'annotation sur ce concept. Ces RAC constituent le cœur de la démarche OntoPop.

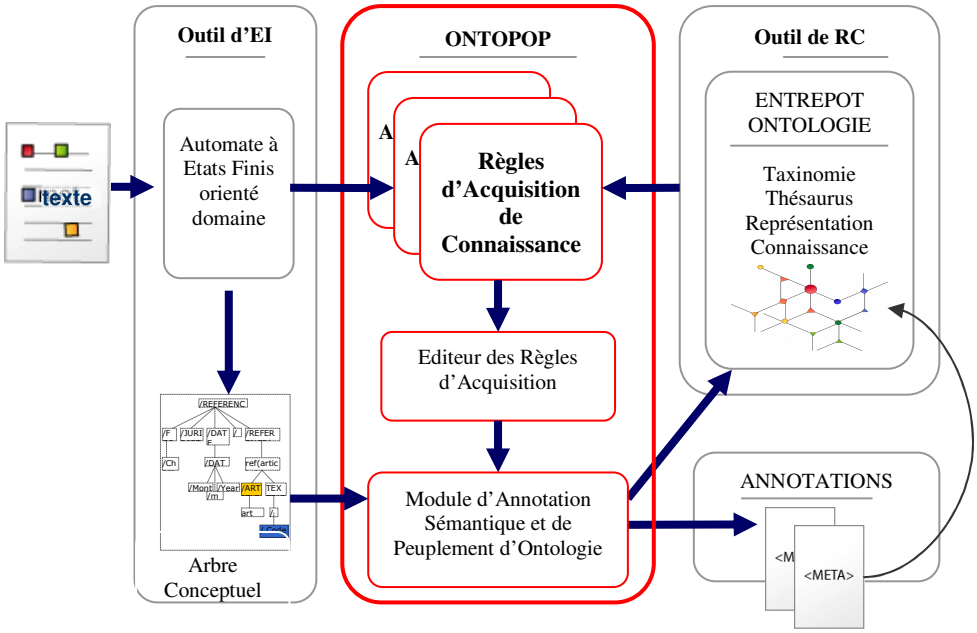


Figure 32. La passerelle proposée par OntoPop

La démarche OntoPop propose également une méthodologie pour la mise en place d'une application d'annotation sémantique et/ou de peuplement d'ontologie basée sur la définition des RAC (cf. Chapitre 5) et une plate-forme logicielle (cf. Chapitre 6). Comme illustré dans la Figure 32, cette plate-forme se compose d'un Editeur des RAC et d'un Module d'Annotation Sémantique et de Peuplement d'Ontologie. L'Editeur compile les RAC, avant tout destinées à des utilisateurs humains, pour les traduire dans un langage orienté machine afin qu'elles puissent être traitées par l'application cible. Le Module procède à la transformation des arbres conceptuels en représentations formelles, que ce soit les annotations sémantiques de chaque document analysé ou les nouvelles instances à insérer dans la base de connaissance. Le chapitre suivant présente l'ensemble du processus d'annotation sémantique et de peuplement d'ontologie. Mais à présent, nous allons définir plus en détail ce que nous entendons par « Règles d'Acquisition de Connaissance ».

3.2 La formalisation des Règles d'Acquisition de Connaissance

3.2.1 L'importance du contexte dans les arbres conceptuels

Afin de pouvoir déterminer le format intermédiaire qui permettra de passer d'une représentation du texte sous la forme d'un arbre conceptuel à une représentation formelle basée sur la modélisation d'une ontologie, il faut comparer l'ensemble des étiquettes sémantiques constituant les arbres conceptuels d'un corpus avec l'ensemble des éléments (concepts, relations, attributs) de l'ontologie du domaine. Prenons par exemple le domaine de la presse « People ». Parmi tous les éléments de l'ontologie (cf. §7.2.1.2), seuls dix classes (4 correspondant aux entités nommées et 6 correspondant

aux événements), 14 relations et 13 attributs doivent être couplés aux étiquettes sémantiques. Par exemple, la classe « Personnalité », sous-classe de « Personne », possède les attributs « date de naissance », « alias », etc. comme illustré dans la Figure 33. La classe « Article » possède les attributs « date de publication », « source », « auteur »… La classe « Article » est aussi reliée à la classe « Personnalité » à travers la relation « indexation personnalité », qui sera utilisée comme annotation sur le document.

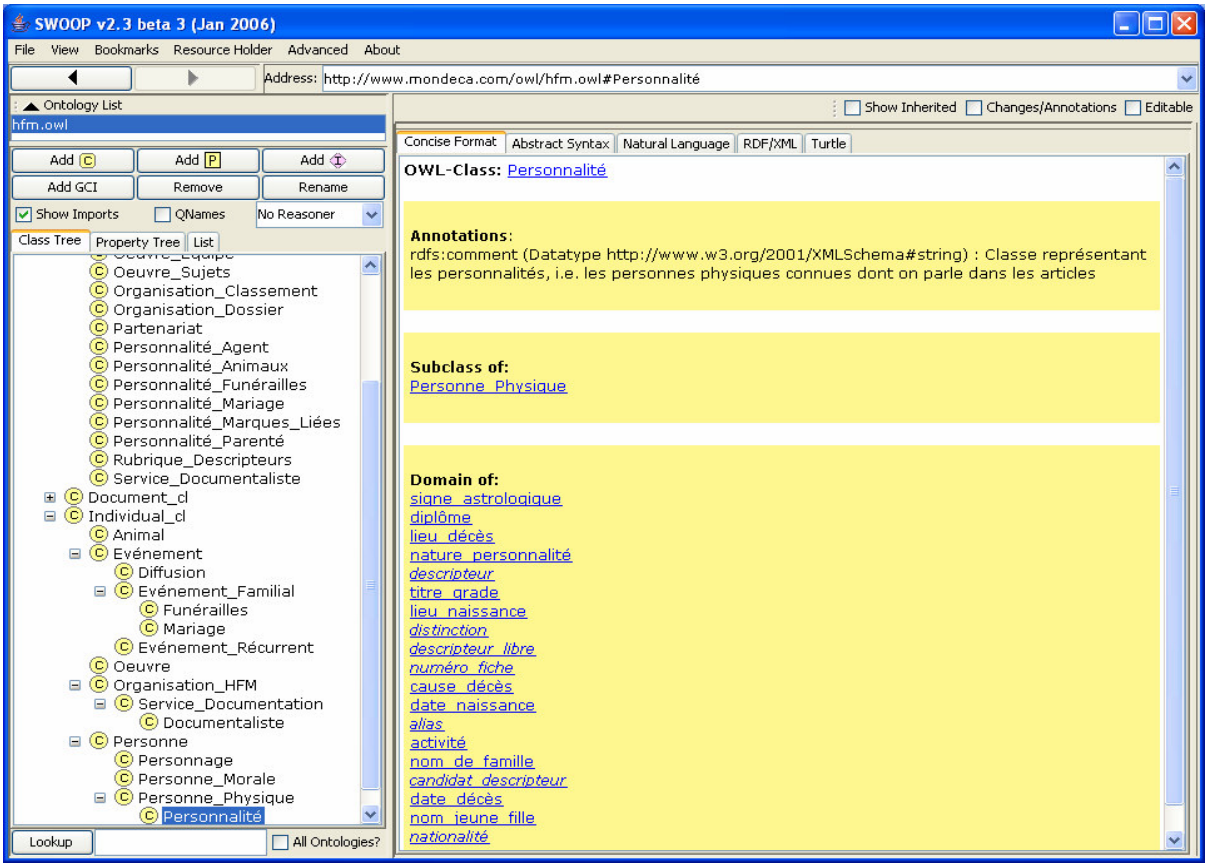


Figure 33. Extrait de l'ontologie concernant le domaine de la presse « People »

Durant la tâche de comparaison, toutes les informations nécessaires au couplage sont centralisées, comme dans le Tableau 2. Les différents cas de couplage à traiter sont les suivants :

- une seule étiquette sémantique correspond à un seul élément, cf. la classe « /Film ».
- Plusieurs étiquettes sémantiques correspondent au même élément, cf. la classe « Personnalité ».
- Une étiquette sémantique correspond à plusieurs éléments du même type, cf. l'étiquette « /COUPLE ».
- Une étiquette sémantique correspond à plusieurs éléments, de types différents, cf. l'étiquette « /ActorNamed »
- Une étiquette sémantique ne correspond à aucun élément, cf. l'étiquette « Evenement_Imminent »
- Un élément ne correspond à aucune étiquette sémantique, cf. l'attribut « Signe Zodiacal ».

Elément dans l'ontologie	Type dans l'ontologie	Etiquette Sémantique	Contexte des étiquettes dans l'arbre conceptuel
Film	Classe	/OeuvreFilm	
Personnalité	Classe	/Personnalité	
		/NomPotentielDePersonne	
Mariage	Classe	/COUPLE	∃ Enfant = /Mariage
Divorce	Classe		∃ Enfant = /Divorce
Personnalité	Classe	/ActorNamed	∃ Enfant = /Personnalité
Conjoint	Relation		∃ Enfant = /Personnalité **and** ∃ Parent = /COUPLE
Indexation Personnalité	Relation		
Lieu de naissance	Attribut	/Location	∃ Parent = /DATE-NAISSANCE
Lieu du mariage	Attribut		∃ Parent = /COUPLE **and** ∃ Frère = /Mariage
Indexation lieu	Attribut		
Date de naissance	Attribut	/DATE	∃ Oncle = /Personne **and** ∃ Père = /Naissance **and** ∃ Ancêtre = /DATE-NAISSANCE
		/Evenement_Imminent	
Signe Zodiacal	Attribut		

Tableau 2. Tableau de comparaison éléments ontologiques versus étiquettes sémantiques

Dans le cas où une étiquette sémantique correspond à plusieurs éléments de l'ontologie, le contexte de cette étiquette dans l'arbre conceptuel devient alors très important pour résoudre les ambiguïtés. Le contexte d'une étiquette sémantique comprend ses ancêtres (parents, grands-parents et autres aïeux), ses descendants (enfant, petits-enfants, etc.), ses frères, ses oncles, etc. En fait, on peut résumer le contexte comme étant l'ensemble des nœuds composant le sous-arbre auquel appartient l'étiquette sémantique étudiée, comme illustré dans la Figure 34. La présence (« ∃ ») ou l'absence (« not(∃) ») d'un ou de plusieurs nœud(s) dans ce contexte permet de déterminer l'élément de l'ontologie qui peut être couplé à l'étiquette sémantique.

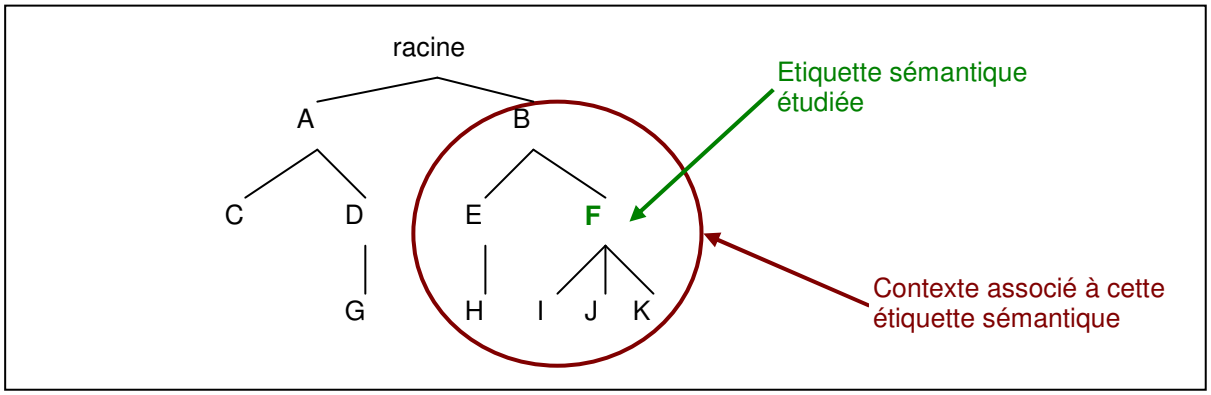


Figure 34. Contexte d'une étiquette sémantique dans un arbre conceptuel

Par exemple, si l'étiquette sémantique "/COUPLE" possède un nœud enfant "/Divorce", comme cela est le cas dans notre exemple concernant la famille Coppola (cf. Figure 26), un événement de la classe "Divorce" sera instancié. Par contre, si ce nœud enfant est le nœud « Mariage », alors il s'agira d'un événement de la classe « Mariage ». La notion de contexte est donc cruciale car le couplage entre les étiquettes sémantiques et les éléments de l'ontologie n'est que rarement une simple bijection. Ainsi, ce contexte doit être facilement appréhendable par les Règles d'Acquisition de Connaissance.

3.2.2 La méthode d'exploration contextuelle

Afin de construire le meilleur formalisme pour les RAC qui permettrait de rendre compte du contexte des étiquettes sémantiques utilisées, je me suis inspirée des travaux menés sur l'exploration contextuelle telle que définie par Desclés [DES 91] [DES 93]. Cette méthode d'exploration contextuelle permet de parcourir, non pas un arbre conceptuel, mais un document textuel non structuré et de l'étiqueter en se basant uniquement sur la présence de marqueurs linguistiques (indices déclencheurs et indices complémentaires). Elle repose entièrement sur les connaissances linguistiques présentes dans les textes, sans recourir à des connaissances encyclopédiques par exemple, et se refuse à concevoir l'analyse linguistique comme un enchaînement des analyses lexicales, syntaxiques et sémantiques. D'après Desclés [DES 93], la méthode repose sur l'identification d'indicateurs linguistiques qui explicitent une valeur sémantique à repérer dans une application donnée. Ces indicateurs sont relatifs à un champ grammatical ou discursif précis comme par exemple les indicateurs discursifs d'annonces thématiques, de relations de causalité ou temporelles entre événements, etc.

Or, le repérage de ces indicateurs linguistiques n'est pas toujours suffisant à l'identification de la valeur sémantique car comme le précise Minel, « *le rapport entre signifiants et signifiés n'est pas bijectif dans les langues, tout particulièrement pour les champs grammaticaux et discursifs* » [MIN 02]. Ainsi, outre l'identification des indicateurs linguistiques, il est nécessaire de repérer d'autres indices linguistiques, i.e. les indices complémentaires. Ces derniers conditionnent alors l'affectation ou non de la valeur sémantique à l'unité textuelle analysée. Ces indices complémentaires peuvent être de nature très diverse comme une ponctuation, une position donnée dans l'unité textuelle analysée, un élément structurel du document comme un titre ou encore un type d'acte discursif comme une conclusion [DES 93].

La méthode d'exploration contextuelle fait donc intervenir, pour l'attribution d'une valeur sémantique donnée, un indice déclencheur et des indices complémentaires qui analysent le contexte de l'indice déclencheur pour déterminer l'attribution ou non d'une valeur sémantique donnée. Cette méthode, issue des travaux en systèmes de décision, est basée sur l'implémentation de règles d'exploration contextuelle formellement définies comme suit : « *une règle R_k est composée d'une classe d'indicateur K, d'un ensemble fini de couples (I_p, C_p) où I_p représente la p-ième classe d'indices à*

rechercher dans le contexte linguistique C_p, et d'une décision D_k » [MIN 02]. L'objectif de ces règles consiste à formaliser la démarche proposée dans la méthode sous la forme : « *si l'indicateur u_i est identifié dans un texte T et si l'on constate la présence des indices I_p dans les contextes C_p alors prendre la décision D_j'* » [CRI 03]. Par conséquent, une règle d'exploration contextuelle se divise en trois parties : 1) la partie « déclenchement » qui contient l'indice déclencheur et l'étiquette sémantique associée, 2) la partie « conditions » qui décrit les indices complémentaires à repérer ou non dans le contexte de l'indice déclencheur et 3) la partie « actions » qui n'est exécutée que si toutes les conditions sont vérifiées [CRI 03].

Une certaine analogie apparaît naturellement entre la méthode d'exploration contextuelle et notre besoin de repérer les étiquettes sémantiques d'un arbre conceptuel. Dans notre cas, nous explorons des ressources étiquetées, les arbres conceptuels, afin de repérer des marqueurs, non pas linguistiques mais sémantiques, pour les rapprocher des éléments modélisés dans l'ontologie du domaine concerné. La notion de contexte dans ces arbres y est également importante et décisive puisque c'est la présence ou non de certaines étiquettes sémantiques dans l'entourage du nœud déclencheur qui va permettre l'activation de la règle. Ce déclenchement provoque la création d'annotations sémantiques ou de nouvelles instances dans la base de connaissance. Nous avons donc adapté les différentes notions vues précédemment à la définition de nos propres Règles d'Acquisition de Connaissance.

3.2.3 Les constituants d'une Règle d'Acquisition de Connaissance

Plus formellement, nous pouvons définir une Règle d'Acquisition de Connaissance comme composée de la classe du nœud indicateur N, d'un ensemble fini de couples (I_p, C_p) où I_p représente la p-ième classe d'indices à rechercher dans le contexte sémantique des arbres conceptuels C_p, d'une décision D_k ainsi que d'un ensemble fini de triplets (S, L, T) où S représente la valeur de la décision D_k, L la position de cette valeur dans le texte d'origine et T la valeur de confiance accordée à la décision.

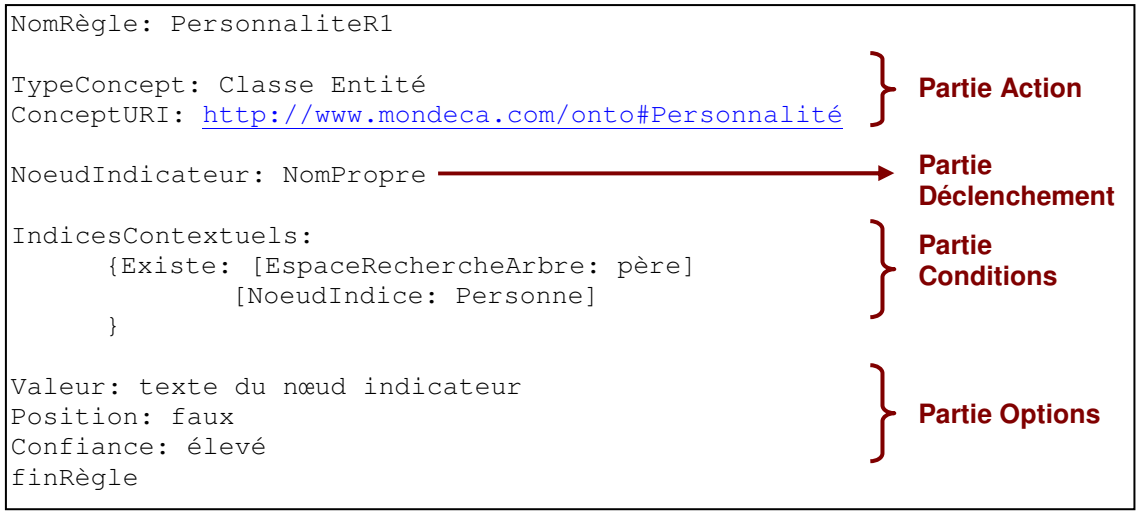


Figure 35. Exemple d'une Règle d'Acquisition de Connaissance en langage OPAL

La partie Déclenchement d'une RAC définit la classe N, la partie Conditions définit les couples (I_p, C_p) et la partie Action définit la décision D_k. Une RAC possède aussi une partie Options, contrairement aux règles d'exploration contextuelle, qui définit le triplet (S, L, T). Nous allons à présent détailler chacun de ces composants. La Figure 35 présente un exemple de Règle d'Acquisition de Connaissance, formalisée dans le langage OPAL que nous présentons dans la prochaine section de ce chapitre. Nous voyons clairement les quatre parties constituantes d'une règle que nous allons à présent détailler.

3.2.3.1 La partie Déclenchement

Cette partie concerne l'information liée au déclenchement de la règle, i.e. l'étiquette sémantique qui va activer la règle. Cette étiquette sémantique est également appelée nœud indicateur ou nœud déclencheur. L'algorithme d'annotation et de peuplement d'ontologie d'OntoPop, décrit au prochain chapitre, parcourt chaque nœud de l'arbre conceptuel généré par l'outil d'extraction d'information à la recherche du prochain nœud porteur d'une étiquette sémantique paramétrée pour activer une Règle d'Acquisition de Connaissance.

3.2.3.2 La partie Conditions

La partie Conditions est optionnelle. Elle ne doit être déclarée que lorsqu'il est nécessaire de vérifier un ensemble de conditions auprès de certaines étiquettes sémantiques, i.e. les indices contextuels complémentaires, dans l'environnement du nœud déclencheur. Chaque condition est constituée de trois composants : l'opération à vérifier (existence, absence, autre, …), l'espace de recherche dans l'arbre auquel cette opération se limite et le nœud indice, i.e. la valeur de l'étiquette sémantique sur laquelle la condition porte.

Dans le cas où plusieurs conditions sont déclarées dans la règle, celles-ci sont liées par l'opérateur logique « ET », c'est-à-dire qu'elles doivent toutes être vérifiées pour que l'action puisse être exécutée. Il est également possible d'imbriquer plusieurs conditions les unes dans les autres comme nous le montrerons plus loin.

3.2.3.3 La partie Action

La partie Action est exécutée si et si seulement toutes les conditions de la partie précédente sont vérifiées. Cette partie précise l'élément de l'ontologie à instancier ou servant à créer l'annotation désirée ainsi que son type dans l'ontologie du domaine.

L'élément de l'ontologie à instancier ou servant à créer l'annotation doit être identifié de manière unique par la Règle d'Acquisition de Connaissance. Il ne doit pas y avoir d'ambiguïté possible. Comme l'identification des éléments dans une ontologie se fait par l'utilisation des « Uniform Resource Identifier » (URI), chaque Règle d'Acquisition de Connaissance utilisera l'URI de l'élément concerné pour l'identifier de manière univoque.

Dans une ontologie, le type d'un élément est normalement l'un des trois suivants : « classe », « attribut » ou « relation ». Mais nous avons décidé d'être entièrement compatibles avec les différents langages de représentation des connaissances présentés à la section 1.3 : RDF, OWL et XTM. Rappelons que d'un côté, RDF et OWL reposent sur des triplets dans lesquels les relations ne sont que binaires. De l'autre, XTM fait intervenir une autre notion, celle de « rôle » d'une classe dans une relation donnée qui permet de modéliser des relations n-aires. Nous pensons qu'il est intéressant de conserver cette façon de modéliser les relations n-aires car elles sont plus proches des formulations langagières trouvées dans les textes non structurés. Comme argumenté dans [VAT 02], une des manières de procéder consiste à assembler des relations binaires RDF ou OWL en fonction d'une classe représentant l'événement de la relation. Chaque relation binaire devient un rôle entre cette classe et les autres classes participant à la relation n-aire. Il devient alors possible de définir des attributs pour la relation n-aire modélisés au niveau de la classe de réification. Par exemple, dans une relation de mariage, la classe de réification représentant cette relation aura des attributs tels que « date mariage » et « lieu mariage ».

Pour en revenir à la formalisation des Règles d'Acquisition de Connaissance, nous avons décidé de tenir compte de la possibilité de réifier en OWL et en RDF les relations n-aires de la même manière que XTM en distinguant deux sortes de classes : les classes usuelles comme celles modélisant les entités nommées désormais appelées « classes entités » dans la suite de ce mémoire et les classes représentant les réifications de relations que nous appellons « classes relations ». Par conséquent, dans la partie Actions d'une RAC, le type de l'élément de l'ontologie concerné par l'annotation ou le peuplement doit correspondre à l'une de ces quatre catégories : classe entité, classe relation, attribut et rôle.

Enfin, il est important de préciser qu'un attribut ou un rôle ne peuvent être instanciés que par rapport à la classe pour laquelle ils ont été définis, i.e. leur domaine. Par exemple, si dans l'ontologie une classe « Personnalité » possède un attribut « Date naissance », il ne sera pas possible de créer une instance de cet attribut sans connaître au préalable l'instance de la classe « Personnalité » à laquelle elle est rattachée. Les règles d'acquisition fonctionnent par binômes : (classe entité, attribut) ou (classe relation, rôle) ou encore (classe relation, attribut). Lorsque le type de l'élément de la partie Action est soit « Attribut » soit « Rôle », il faut indiquer l'URI de la classe représentant le domaine de ces propriétés. Ainsi, lorsqu'une instance de classe est repérée dans l'arbre conceptuel, les règles ayant comme type « Attribut » et comme domaine l'URI de la classe instanciée sont déclenchées et la valeur de l'attribut rattachée à l'instance de la classe (cf. Figure 41).

3.2.3.4 La partie Options

Au cours des premières expérimentations des RAC [AMA 04], il est apparu que les trois parties précédentes, bien qu'essentielles, n'étaient pas suffisantes pour capturer toute la complexité existante dans la réalisation de la passerelle OntoPop. Nous avons donc complété les RAC en ajoutant trois options : la valeur générée par la règle, la position de cette valeur dans le document source et le niveau de confiance accordé à la règle.

Premièrement, la valeur à générer par la règle en sortie ne correspond pas toujours à la valeur textuelle contenue dans le nœud indicateur, surtout lorsque l'application cible requiert une forte normalisation des connaissances extraites des documents. Il est donc parfois nécessaire de créer la valeur de l'instance ou de l'annotation en fonction de diverses informations comme le nom d'un nœud, la concaténation de plusieurs valeurs de nœuds de l'arbre, une constante quelle que soit la valeur textuelle du nœud indicateur, etc. Par exemple, dans notre ontologie « People » existe un plan de classement des articles pour chacune des personnalités. Si les rubriques de ce plan de classement sont les mêmes pour toutes les personnalités, leur instance doit porter le nom de la rubrique accolé au nom de la personnalité. Dans la Figure 36, est présenté un arbre conceptuel concernant une extraction liée au mariage de l'instance « Johnny Halliday » de la classe entité « Personnalité ». Or dans le plan de classement des personnalités, il existe une rubrique « Mariage ». Par conséquent, l'instance de cette rubrique pour la personnalité « Johnny Halliday » doit porter le nom « Mariage – Johnny Halliday ». Mais afin de générer automatiquement le libellé de cette instance, il est nécessaire de concaténer deux sortes d'informations différentes : celle du nom de l'étiquette sémantique « /Mariage » avec la valeur textuelle de l'étiquette sémantique « /Personnalite ». Bien qu'apparemment simple, cette situation nous a posé beaucoup difficultés lors des premières implémentations des RAC.

```
/COUPLE(Johnny Hallyday s'est marié le 24 juin 1962 à Paris.)
    /ActorNamed(Johnny Hallyday)
       /Personnalite(Johnny Hallyday)
    /Mariage(s'est marié)
    /Date(24 juin 1962)
 /Location(Paris)
     /France(Paris)
```

Figure 36. Exemple d'arbre conceptuel représentant un événement mariage

Deuxièmement, il est potentiellement intéressant de vouloir conserver les informations de position des informations extraites ou annotées dans le document d'origine, notamment à des fins de présentation pour l'utilisateur final. Il sera alors possible de créer une interface lui présentant l'emplacement exact des différentes informations retenues comme annotations ou comme nouvelles instances de la base de connaissance.

Troisièmement, chaque Règle d'Acquisition de Connaissance ne possède pas le même niveau de confiance. Par exemple, nous avons vu au chapitre précédent qu'il existait deux sortes de méthodes permettant de repérer les entités nommées dans un document : soit à partir de lexiques contenant les entités nommées connues d'un domaine en particulier, soit à partir de patrons d'extraction qui sont capables de détecter de nouvelles entités nommées potentielles pour ce domaine. Ainsi, dans le cas d'une application permettant de peupler une ontologie automatiquement à partir de ces deux méthodes, il est important pour elle de savoir si la nouvelle instance d'entité nommée provient d'un lexique, auquel cas le taux de confiance dans l'information extraite est élevé, ou bien si elle a été déduite à partir d'un patron d'extraction, auquel cas le taux de confiance accordé sera plus faible. Par exemple, on dispose de deux sortes d'étiquettes sémantique dans notre application « People » :

l'étiquette sémantique « /Personnalite » permet de savoir que la personnalité reconnue provient d'un lexique contrôlé et l'étiquette sémantique « /NomDePersonnePotentiel » que le nom de la personnalité a été déduite par un patron. Ainsi, deux Règles d'Acquisition de Connaissance sont définies à partir de ces étiquettes, l'une possédant un taux de confiance plus élevé que l'autre. Cette information peut être utilisée pour effectuer un nettoyage de la base de connaissance ou bien pour signaler à l'utilisateur le niveau de pertinence de l'information.

Nous allons à présent décrire comment ces RAC sont implémentées dans OntoPop et notamment présenter le formalisme utilisé pour les représenter.

3.3 L'implémentation des Règles d'Acquisition de Connaissance

3.3.1 Le langage OPAL

Crispino a défini un langage, LangText [CRI 03], à la fois destiné aux linguistes et aux informaticiens, pour modéliser les règles d'exploration contextuelle chargées de résoudre l'apposition d'étiquettes sémantiques dans le texte analysé. Ce formalisme répond aux critères de la méthode d'exploration contextuelle énoncés ci-dessus. Voici dans la Figure 37 un exemple de règle d'exploration contextuelle définie en LangText :

```
Nomregle = RCenthe111 ;
IndicateursDeclencheurs = &Cobjectif
Etiquette = Thematique_2 ;
{Existe:
        [EspaceRecherche : voisinage(i, 5)]
        [indice: y]
        [&document ; &personne ;] }
{Existe:
        [EspaceRecherche : voisinage(i, 10)]
        [indice : z]
        [&Centhe1 ;] }
{Precede:
        [y] [z] }
{Distance:
        [y] [z] [distance : 5] }
attribuerEtiquette : Phrase
finRegle
```

Figure 37. Exemple d'une règle d'exploration contextuelle formalisée en LangText, tiré de [CRI 03]

Dans LangText, les « IndicateursDéclencheurs » correspondent aux marqueurs linguistiques déclenchants l'insertion d'une nouvelle « Etiquette » dans le document. Mais afin de valider cette insertion, une ou plusieurs conditions peuvent être optionnellement définies. Par exemple, la première condition de la règle de la Figure 37 déclare rechercher un indice « y » qui correspond soit au marqueur « &document » soit au marqueur « &personne ». Puis, cet indice doit apparaître dans un

voisinage de 5 mots au plus par rapport à l'indice déclencheur. Si l'ensemble des conditions sont vérifiées, alors une étiquette sémantique sera attribuée à un segment textuel, précisé dans le paramètre « attribuerEtiquette ».

Formalisme LangText	**Formalisme OPAL**
Indicateurs Déclencheurs	NoeudIndicateur
Etiquette	ConceptURI
Indices	IndicesContextuels
attribuerEtiquette	TypeConcept (Classe, Relation, Attribut, Rôle)

Tableau 3. Tableau de rapprochement des concepts de chacun des formalismes LangText et OPAL

De la même manière que je me suis inspirée de l'analyse contextuelle réalisée par la méthode d'exploration contextuelle pour la formalisation des RAC, j'ai décidé d'adapter le formalisme du langage LangText à l'écriture des RAC pour OntoPop, comme illustré dans le Tableau 3. J'ai baptisé ce nouveau langage **OPAL**, signifiant « **Ontology Population and Annotation Language** ». La grammaire de notre formalisme déclaratif des règles d'acquisition dans le langage OPAL est constituée des différents éléments suivants, représentés en notation Extended BNF[32] :

<**Règle**> ::= <**NomRègle**>+ <**TypeConcept**>+ <**ConceptURI**>+ <**DomainURI**>?
<**Noeudindicateur**>+ <**IndicesContextuels**>+ <**Valeur**>? <**Position**>? <**Confiance**>?
'**finRègle**'
<**NomRègle**> ::= ('A-Z' | 'a-z' | '0-9')
<**TypeConcept**> ::= 'Classe entité' | 'Classe relation' | 'Attribut' | 'Role'
<**ConceptURI**> ::= 'http ::' ('A-Z' | 'a-z' | '0-9')
<**DomainURI**> ::= 'http ::' ('A-Z' | 'a-z' | '0-9')
<**Noeudindicateur**> ::= ('A-Z' | 'a-z' | '0-9')
<**IndicesContextuels**> ::= <Opérateurs> <EspaceRechercheArbre> <ObjetCondition>
<Opérateurs> ::= 'existe' | 'nonExiste' | 'contient' | 'valeur' | 'position'
<EspaceRechercheArbre> ::= 'père' | 'fils' | 'grand-père' | 'petit-fils' | 'ancêtre' | 'descendant' | 'nom nœud' | 'texte'
< ObjetCondition> ::= <NoeudIndice> | <ValeurCondition>
<NoeudIndice> ::= ('A-Z' | 'a-z' | '0-9')
<ValeurCondition> ::= ('A-Z' | 'a-z' | '0-9')
<**Valeur**> ::= <Constante> | <FonctionXpath> | <ExpressionXML>
<Constante> ::= ('A-Z' | 'a-z' | '0-9')
<FonctionXpath> ::= 'text()' | 'name()' | …
<ExpressionXML> ::= <elementXML> | <attributXML> | …
<**Position**> ::= 'Vrai' | 'Faux'
<**Confiance**> ::= 'Elevée' | 'Moyenne' | 'Faible'

Figure 38. Grammaire EBNF du langage OPAL

La signification des principaux éléments constitutifs d'une Règle d'Acquisition de Connaissance en OPAL est la suivante :

[32] Cf. http://en.wikipedia.org/wiki/Extended_Backus_Naur_Form

NomRègle: nom donné par l'utilisateur

TypeConcept: nature de l'élément de l'ontologie

ConceptURI: URI de l'élément dans l'ontologie du domaine

DomaineURI (optionel): URI de la classe correspondant au domaine d'une propriété dans l'ontologie (à renseigner uniquement lorsque le « TypeConcept » a pour valeur attribut ou rôle)

NoeudIndicateur: étiquette sémantique déclenchant le processus d'annotation ou de peuplement d'ontologie

IndicesContextuels (optionnel): conditions sur l'existence ou non de certaines étiquettes sémantiques dans le contexte du NoeudIndicateur ou encore sur la position de ces étiquettes dans l'arbre conceptuel

Valeur (optionel): par défaut, la valeur des annotations ou des instances créées à partir des règles d'acquisition de connaissance est celle du texte issu du document original qui est associé au nœud déclencheur. Mais il est parfois nécessaire, notamment à des fins de normalisation, de construire une valeur à partir du nom d'un nœud de l'arbre, ou de plusieurs autres valeurs textuelles associées à d'autres nœuds, etc. Ce champ permet donc de paramétrer lorsque cela est nécessaire la manière de calculer la valeur en sortie de la règle d'acquisition

Position (optionel): indique si la position de l'information extraite dans le document original doit être conservée comme métadonnée des nouvelles instances (par défaut la valeur est 'faux')

Confiance (optionel): indique le niveau de confiance à accorder à cette règle (par défaut le niveau de confiance est positionné à 'élevée')

finRègle

Figure 39. Description des éléments d'une Règle d'Acquisition de Connaissance

D'après la grammaire EBNF, on voit que chaque indice contextuel est composé de trois paramètres :

1) une « Opération » qui correspond à la condition à satisfaire, comme « existe », « nonExiste », « position », etc.,
2) l'« EspaceRechercheArbre », i.e. la localisation de l'indice contextuel dans l'arbre (fils, père, petit-fils, grand-père, descendant, ancêtre sont les seules constructions utilisées à ce jour, mais on pourrait aisément étendre cette liste avec les frères, les oncles, etc.) et,
3) l'« ObjetCondition » qui correspond soit au « NoeudIndice », i.e. au nom du nœud sur laquelle porte l'opération de la condition, soit à une « ValeurConfition », celle-ci pouvant être une valeur numérique ou textuelle.

Par ailleurs, plusieurs IndicesContextuels peuvent être juxtaposés comme dans la Figure 40. Dans ce cas, les deux conditions doivent être réunies pour que la règle produise l'action désirée. Cette figure montre aussi qu'un IndiceContextuel peut lui-même être imbriqué dans un autre IndiceContextuel. Ceci permet de déterminer par exemple que l'indice contextuel sur l'ancêtre « DATE-NAISSANCE » doit lui-même posséder un nœud enfant « Personne ».

```
NomRègle: DateNaissanceR1
TypeConcept: Attribut
ConceptURI: http://www.mondeca.com/onto#Date_Naissance
DomaineURI : http://www.mondeca.com/onto#Personnalité
NoeudIndicateur: DATE
IndicesContextuels:
      {Existe: [EspaceRechercheArbre: père]
               [NoeudIndice: Naissance]
      }
      {Existe: [EspaceRechercheArbre: ancêtre]
               [NoeudIndice: DATE-NAISSANCE]
               {Existe: [EspaceRechercheArbre: enfant]
                        [NoeudIndice: Personne]
               }
      }
Valeur: text()
Position: faux
Confiance: élevé
finRègle
```

Figure 40. Exemple d'une Règle d'Acquisition de Connaissance en langage OPAL permettant d'instancier un attribut de type « Date_Naissance » lié à une instance de classe « Personnalité »

Par exemple, si l'analyse de la phrase « Coppola est né le 7 avril 1939 à Detroit » produit l'arbre conceptuel situé en haut de la Figure 41, alors l'application des deux Règles d'Acquisition de Connaissance définies ci-dessus (Figure 35 & Figure 40) créera le réseau sémantique, situé en bas de la Figure 41. Ce réseau associe l'attribut « Date de Naissance » ayant pour valeur « 7 avril 1939 » à l'instance « Coppola » de la classe « Personnalité » à laquelle il fait référence.

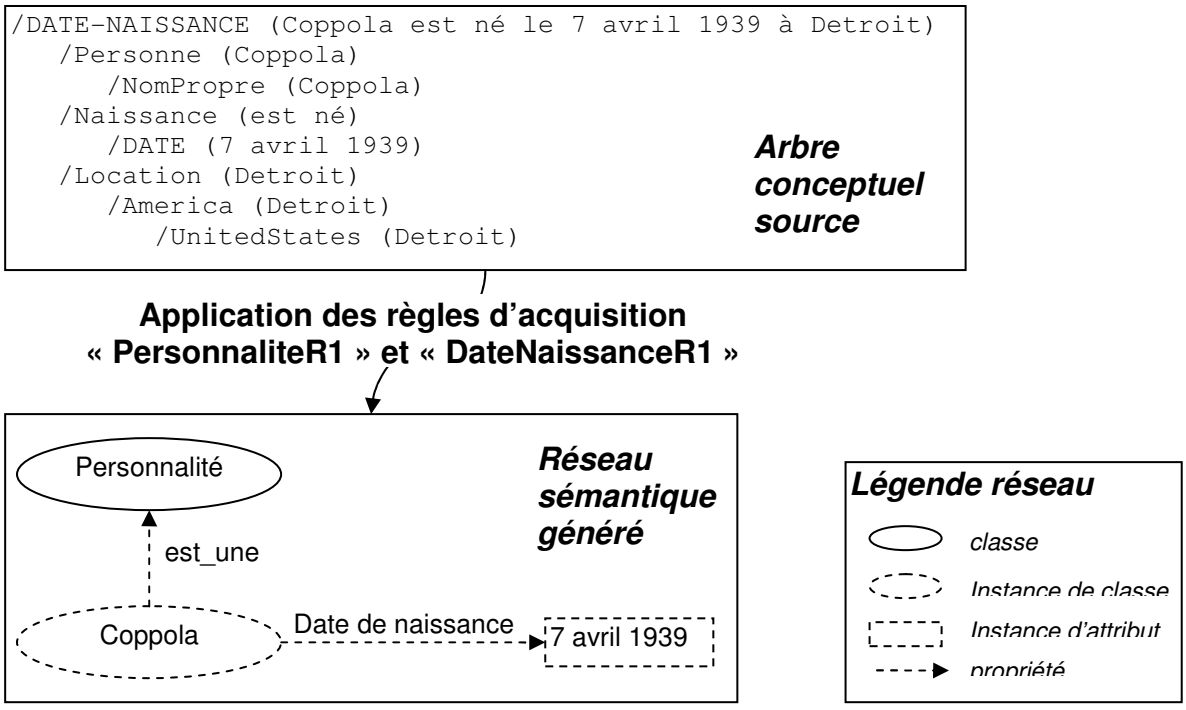


Figure 41. Application de règles d'acquisition sur un arbre conceptuel pour produire le réseau sémantique associé

3.3.2 Edition et compilation des Règles d'Acquisition de Connaissance

Les Règles d'Acquisition de Connaissance sont au cœur d'OntoPop. Au niveau de la méthodologie, elles représentent le socle de la passerelle entre le résultat des extractions linguistiques d'un côté et la représentation sémantique de la connaissance de l'autre. Au niveau de la solution logicielle proposée par OntoPop, elles composent l'ingrédient indispensable au bon fonctionnement du processus de peuplement d'ontologie et d'annotation documentaire.

Parmi la suite logicielle fournie par OntoPop, l'utilisateur dispose d'un Editeur de Règles qui lui permet de saisir l'ensemble des Règles d'Acquisition de Connaissance d'une application cible donnée. L'interface utilisateur est très simple : elle propose un formulaire composé des champs nécessaires à la création d'une nouvelle règle comme spécifié par le langage OPAL. Une description plus technique de cet éditeur est fournie à la section 6.1. Mais ce langage OPAL est avant tout destiné à des utilisateurs humains. Par conséquent, les RAC doivent être traduites dans un langage informatique pour être comprises et manipulées par les agents logiciels. Ce langage informatique doit lui-même être assez puissant et flexible pour traduire l'expressivité et la complexité de chacune des Règles d'Acquisition de Connaissance. Il doit également servir les objectifs de ces règles d'acquisition, à savoir :

1) le passage d'un format de représentation à un autre, que ce dernier représente un réseau sémantique de connaissance au format XTM ou OWL ou un ensemble d'annotations sémantiques au format RDF,
2) et la gestion du contexte particulièrement importante comme nous l'avons vu.

Au chapitre précédent, nous avons déjà précisé qu'un arbre conceptuel peut être représenté comme un document XML. Il nous a donc semblé tout à fait naturel de nous appuyer sur l'ensemble des langages et formats issus de la technologie XML. Parmi cette offre, se dégage tout naturellement le langage XSLT[33] qui opère des actions de transformations sur les éléments sélectionnés d'un arbre. Les feuilles de transformations XSLT reposent sur l'exploitation des chemins XPath[34] afin de sélectionner les nœuds de l'arbre XML à transformer. Le langage XPath définit en effet des chemins à travers l'arbre XML : il permet de naviguer dans un arbre XML, de sélectionner n'importe quel nœud de cet arbre et d'effectuer des opérations sur ces nœuds. Or pour appréhender la complexité des Indices Contextuels des RAC, il est particulièrement intéressant de disposer d'un tel langage de parcours des arbres.

Rappelons brièvement que le déclenchement d'une règle d'acquisition repose sur la présence dans l'arbre de concept du nœud indicateur contenant l'information à récupérer pour instancier un élément de l'ontologie ou pour annoter le document en question. L'utilisation de ce nœud indicateur est dépendante de l'ensemble des contraintes exprimées sur le contexte de ce nœud dans l'arbre de concept. Ce contexte peut être ascendant comme descendant. Les contraintes peuvent être

33 Site web de XSLT: http://www.w3.org/TR/xslt
34 Site web de XPath: http://www.w3.org/TR/xpath

formulées sur l'existence ou non d'un nœud ou d'une valeur particulière. D'autres sortes de contraintes peuvent porter sur des enchaînements d'opérations diverses, des calculs à partir de chaînes de caractères ou de valeurs numériques, etc.

Ces langages, et tout particulièrement XPath, possèdent a priori toute l'expressivité et la flexibilité nécessaires à la transcription des Règles d'Acquisition de Connaissance dans un langage compréhensible et manipulable par les machines. A eux deux, XSLT et XPath remplissent les objectifs des RAC : XPath va être utilisé pour sélectionner les nœuds indicateurs dans l'arbre de concept et vérifier les contraintes liées au contexte de ce nœud et XSLT va réaliser la transformation de ces arbres de concepts en nouvelles instances de la connaissance ou en nouvelles annotations sémantiques. Par ailleurs, nombre d'implémentations techniques et logicielles ont d'ors et déjà été développées et sont disponibles en libre accès. OntoPop peut directement les intégrer et les exploiter dans ses divers modules. L'utilisation de ces langages implique néanmoins que l'arbre de concept généré par l'outil d'extraction d'information soit transformé au format XML, si cela n'est déjà pas la sortie en standard de l'outil.

Par exemple, la Règle d'Acquisition de Connaissance concernant l'attribut « Date de naissance » (cf. Figure 40) sera définie par la règle Xpath suivante : « Naissance/DATE[ancestor::DATE-NAISSANCE/Personne] ». Cette règle XPath signifie : *trouver un nœud « DATE » dont le père est le nœud « Naissance » et qui possède un ancêtre « DATE-NAISSANCE », ayant lui-même un nœud fils nommé « Personne »*.

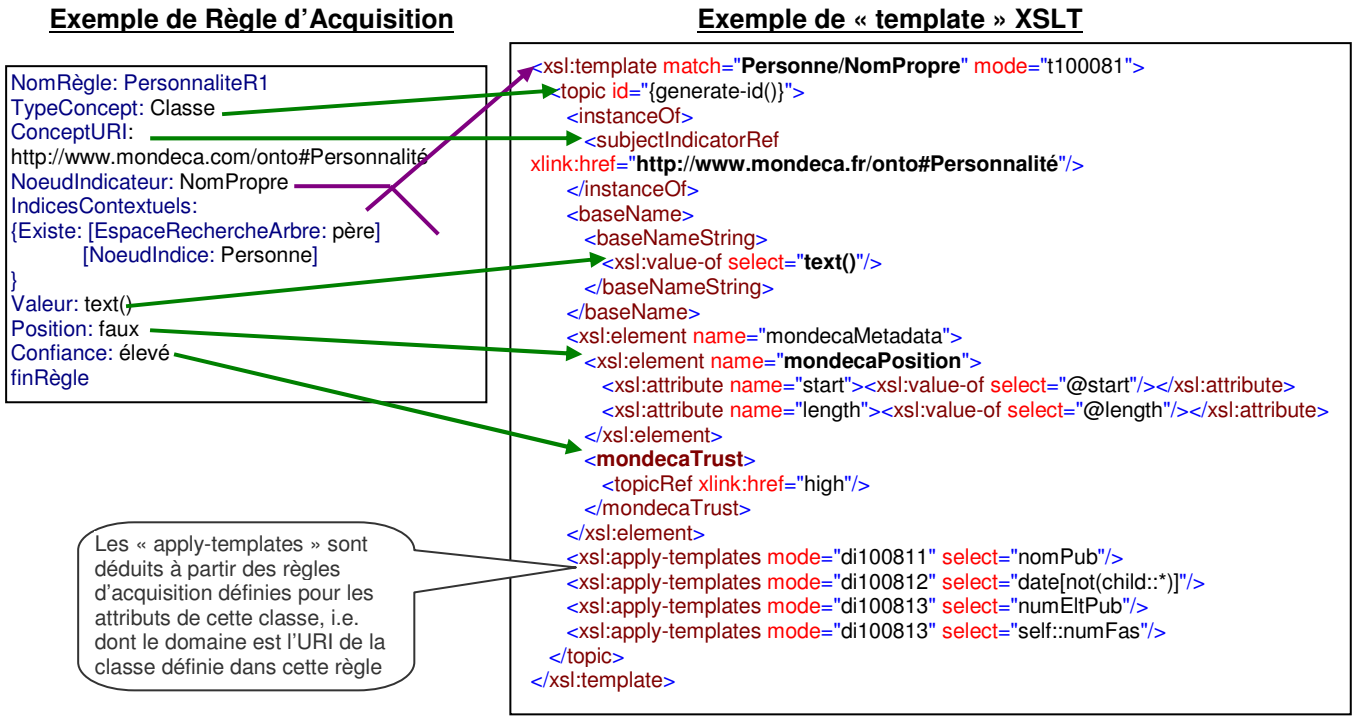


Figure 42. Transcription d'une Règle d'Acquisition de Connaissance en template XSLT pour le peuplement d'ontologie.

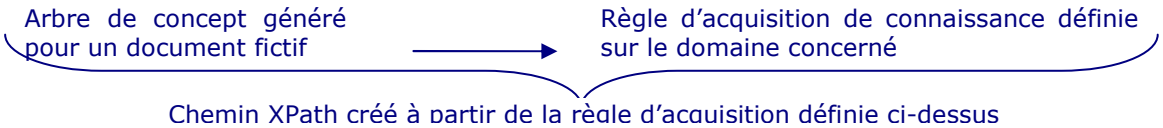
Par conséquent, lorsque l'utilisateur humain saisit une nouvelle règle d'acquisition dans l'Editeur des Règles, celle-ci est immédiatement compilée avec les règles déjà existantes pour générer une feuille de transformation au format XSLT. En réalité, la compilation des règles d'acquisition ne génère pas une seule feuille de transformation XSLT, mais deux : la première destinée au peuplement de l'ontologie du domaine et la seconde pour l'annotation documentaire.

Chaque élément d'OPAL est utilisé pour construire un « template » XSLT. Dans la Figure 42, le template généré par le compilateur de l'Editeur de Règle correspond à la création d'une instance de classe, i.e. un « topic » dans le langage XTM. Rappelons que dans OntoPop ce langage est celui utilisé pour la modélisation de la base de connaissance, et par conséquent pour le processus de peuplement d'ontologie. La partie la plus complexe dans la génération des templates à partir des RAC écrites en OPAL concerne la création du chemin XPath à partir du nœud indicateur et de ses indices contextuels.

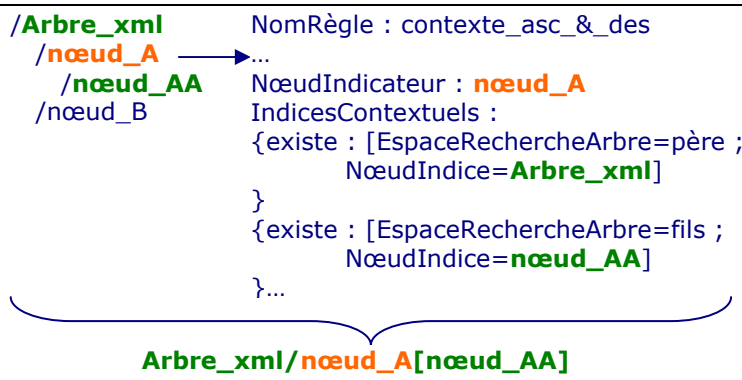
Chaque règle d'acquisition, quelle que soit sa complexité, doit pouvoir être traduite dans le langage XPath. Et cette complexité détermine forcément le niveau de difficulté de la transcription à opérer. Nous avons donc analysé les différentes constructions utilisées pour l'écriture des conditions contextuelles des RAC et nous avons dégagé la graduation suivante :

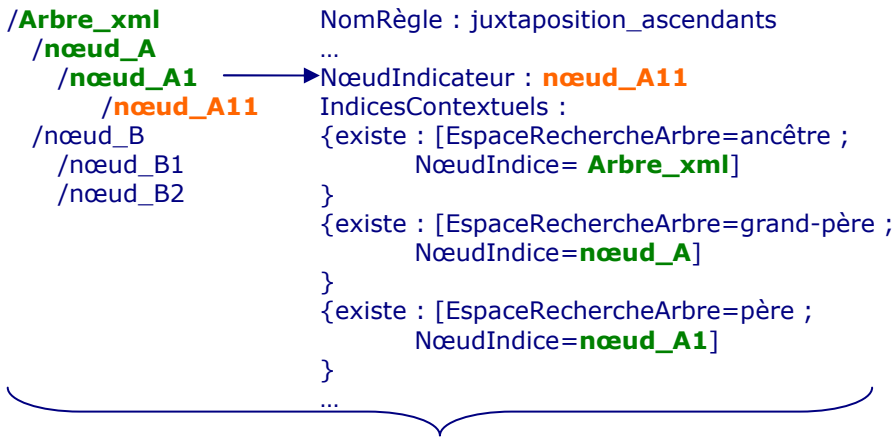
- sans contexte : aucune condition contextuelle n'a été précisée dans la RAC, par conséquent cette règle sera déclenchée quelle que soit la position du nœud déclencheur dans l'arbre conceptuel ou autrement dit, quels que soient ses ancêtres.
- avec un contexte ascendant : la RAC contient une seule condition contextuelle qui porte sur l'existence d'un nœud ancêtre (parent, grand-parent ou autre aïeul)
- avec un contexte descendant : la RAC contient une seule condition contextuelle qui porte l'existence d'un nœud descendant (fils, petit-fils, etc.)
- avec un contexte ascendant et un contexte descendant : la RAC contient deux conditions contextuelles dont une sur l'existence d'un nœud ascendant et l'autre sur l'existence d'un nœud descendant, à quelque niveau que ce soit de la hiérarchie
- avec juxtapositions de contextes ascendants : la RAC contient plusieurs conditions contextuelles, toutes liées à l'existence de plusieurs ascendants
- avec juxtapositions de contextes descendants : la RAC contient plusieurs conditions contextuelles, toutes liées à l'existence de plusieurs nœuds descendants
- avec imbrications de contextes : la RAC contient une ou plusieurs conditions contextuelles, qui elles-mêmes peuvent contenir une ou plusieurs conditions
- avec utilisation de fonctions : la condition contextuelle ne porte plus seulement sur l'existence ou non d'un nœud dans l'arbre mais sur des conditions nécessitant l'utilisation de fonctions comme le calcul de sa position dans un sous-arbre ou d'une partie de sa valeur textuelle par exemple.

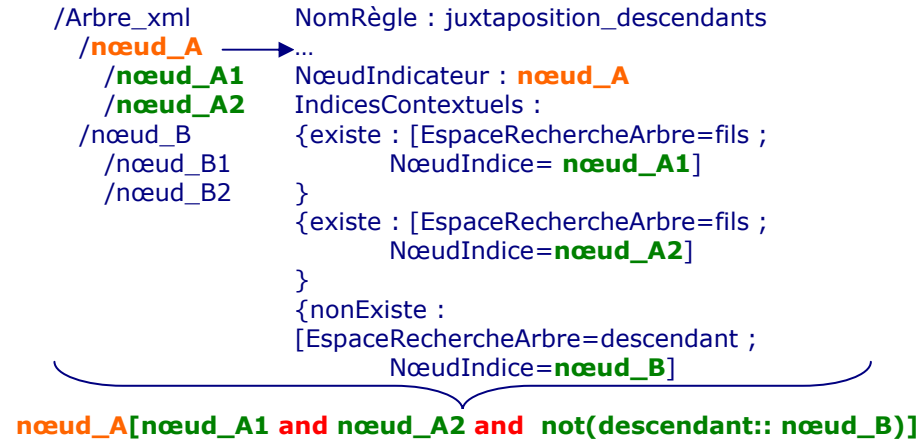
Nous avons analysé ces niveaux de construction des RAC afin de pouvoir dégager pour chaque cas un algorithme nous permettant de générer le chemin XPath associé. Le tableau ci-dessous présente chaque opération de transcription, illustrée par un exemple précis à l'aide du schéma représentant les éléments suivants :



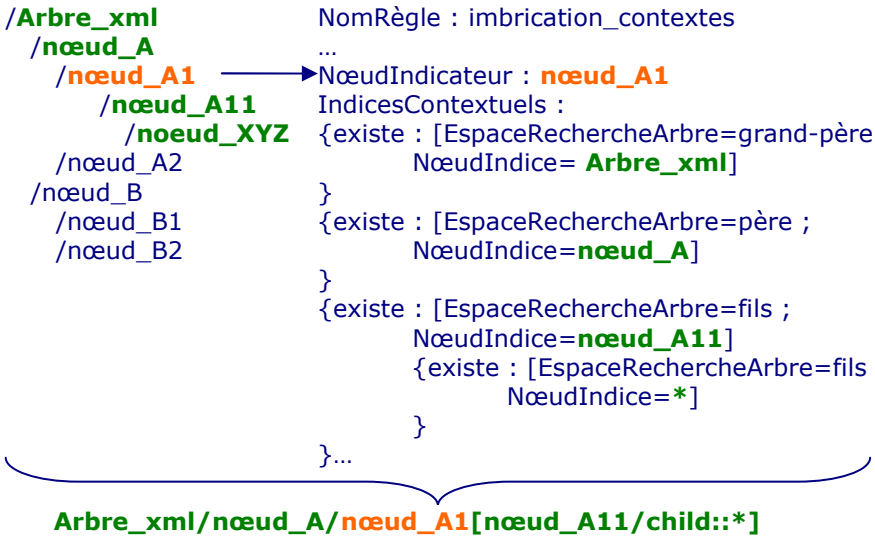
Constructions	Opérations de transcription
sans contexte	En Xpath, l'axe « ancestor ::* » et son raccourci « // » sont utilisés pour indiquer qu'un nœud peut avoir n'importe quel ancêtre. /Arbre_xml /nœud_A /**nœud_AA** /nœud_B NomRègle : sans_contexte … NœudIndicateur : **nœud_AA** IndicesContextuels : {} … **//nœud AA**
Avec un contexte ascendant	L'axe « parent :: » et son raccourci « / » sont utilisés pour indiquer le nœud père du nœud concerné. Nous insérerons donc ce raccourci ou celui représentant les ancêtres, i.e. « // », entre le nœud indice et le nœud déclencheur de la RAC. /Arbre_xml /**nœud_A** /**nœud_AA** /nœud_B Nom Règle : contexte_ascendant … NœudIndicateur : **nœud_AA** IndicesContextuels : {existe : [EspaceRechercheArbre=père ; NœudIndice=**nœud_A**] } … **nœud A/nœud AA**
avec un contexte descendant	Les axes « child:: » et « descendant:: » sont utilisés pour respectivement indiquer un nœud fils ou un nœud descendant du nœud concerné. La fonction « child :: » peut être omise, il s'agit d'un raccourci XPath. Ces fonctions concernant le nœud indice de la RAC seront insérées entre crochets « [] » après le nœud déclencheur. /Arbre_xml /**nœud_A** /**nœud_AA** /nœud_B NomRègle : contexte_descendant … NœudIndicateur : **nœud_A** IndicesContextuels : {existe : [EspaceRechercheArbre=fils ; NœudIndice=**nœud_AA**] }… **nœud_A[child::nœud_AA]**
Avec un contexte ascendant et descendant	Il s'agit dans ce cas de combiner les deux opérations de transcriptions précédentes : ajout des raccourcis « / » ou « // » avant le nœud déclencheur de la RAC pour indiquer le nœud indice ascendant et insertion d'une des fonctions descendantes entre « [] » pour indique le nœud indice descendant.

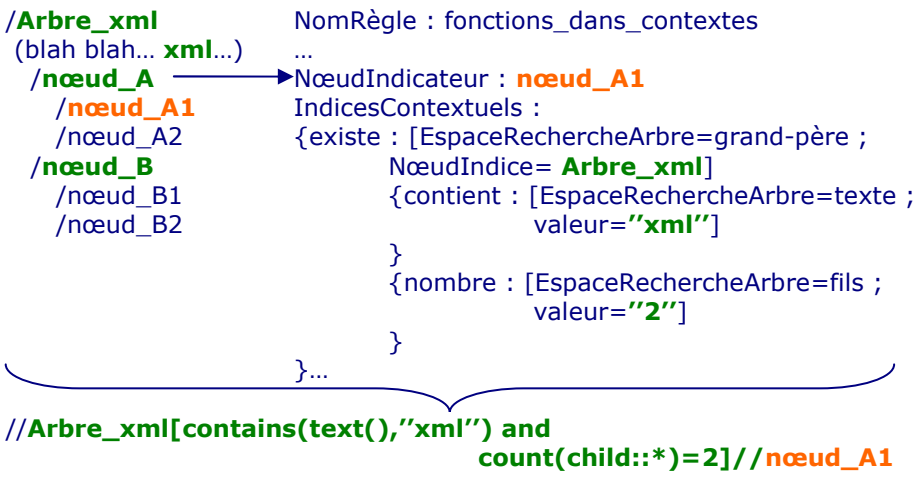
<table>
<tr>
<td></td>
<td>


</td>
</tr>
<tr>
<td>avec juxtapositions de contextes ascendants</td>
<td>
La juxtaposition de ces indices contextuels se caractérise par la réinterprétation de la hiérarchie des ascendants grâce à l'utilisation des axes : « ancestor :: » et « parent :: », traduit par leurs raccourcis « // » et « / ».
Dans l'exemple ci-dessous, le nœud déclencheur « nœud_A11 » possède un père dénommé «nœud_A1 » et un grand-père « nœud_A » lui-même étant le père du « nœud_A1 » par transitivité. Enfin, l'ancêtre de ces nœuds est le nœud dénommé « Arbre_xml ». Comme sa place dans la hiérarchie des ascendants n'est pas connue, le raccourci « // » entre ce nœud et le nœud grand-père indique qu'il peut optionnellement y avoir d'autres nœuds ascendants entre eux.


</td>
</tr>
<tr>
<td>avec juxtapositions de contextes descendants</td>
<td>
La juxtaposition d'indices contextuels descendants se caractérise par la coordination des axes « child :: » et « descendant :: » avec l'opérateur « and » dans la partie située entre « [] » après l'indice déclencheur. Précisons ici que la non-existence d'un nœud se définit par l'utilisation de la fonction « not(axe ::nom_noeud) ».


</td>
</tr>
</table>

<table>
<tr>
<td>avec imbrications de contextes</td>
<td>La complexité repose ici sur la reconstitution de ces imbrications, selon si elles sont liées aux ascendants ou aux descendants.
Par exemple ci-dessous, le nœud indicateur « nœud_A1 » possède un nœud fils « nœud_A11 », lui-même ayant au moins un nœud fils, quel qu'il soit, symbolisé par le caractère « * ». Il faut donc reconstruire la hiérarchie descendante dans la condition entre « [] ».


</td>
</tr>
<tr>
<td>avec l'utilisation de fonctions</td>
<td>Les RAC utilisent aussi contraintes liées au nombre de nœuds, à leur position dans le sous-arbre, à leur nom, leurs valeurs textuelles, etc. Les chemins XPath doivent donc être en mesure d'implémenter ces conditions particulières à l'aide des fonctions prédéfinies dans ce langage. Dans l'exemple ci-dessous deux fonctions Xpath sont utilisées : la première « contains(text(), ''xml'') » pour tester si la valeur textuelle du nœud contient une certaine chaîne de caractères et la seconde « count(child ::*)=2 » pour savoir si il possède exactement deux nœuds fils.


</td>
</tr>
</table>

Tableau 4. Opérations de transcription des RAC en chemins XPath

Jusqu'à présent, l'algorithme de transcription des RAC fournit une solution générique permettant générer automatiquement les chemins XPath à partir des RAC. Il est capable de gérer toutes les formes de constructions des conditions contextuelles rencontrées, quel que soit leur niveau de

complexité, grâce notamment à la richesse du langage XPath. Néanmoins, les RAC sont appelées à évoluer en fonction des besoins de chacune des applications pour lesquelles elles sont créées. Il se peut donc que l'algorithme actuel ne puisse résoudre les nouvelles constructions, même si nous avons tenté de les anticiper.

Une fois l'ensemble des RAC saisies dans l'Editeur des Règles et compilées en feuilles de transformation XSLT, le processus d'annotation sémantique et de peuplement d'ontologie peut être lancé sur un corpus documentaire. Nous allons voir dans la suite quelles sont les tâches et les spécificités de ce processus dans le cadre de la démarche adoptée par OntoPop.

3.4 Conclusion

Les Règles d'Acquisition de Connaissance forment le cœur de la démarche OntoPop. Elles permettent de passer d'une représentation du texte sous la forme d'un arbre conceptuel à une représentation formelle du contenu sous la forme d'annotations sémantiques et/ou d'instances de la base de connaissance selon les besoins. Elles fournissent une solution flexible, facilement adaptable et sachant combler le fossé qui sépare le plus souvent ces deux niveaux de représentation du contenu.

Dans ce chapitre, nous avons défini les composants nécessaires à la formalisation d'une Règle d'Acquisition de Connaissance et décrit le langage OPAL qui permet de les implémenter dans un système logiciel. Enfin, nous avons présenté la manière dont ces règles, destinées à des utilisateurs humains, sont compilées en feuilles XSLT. A présent, nous allons présenter le reste de la démarche OntoPop dans laquelle s'inscrivent ces RAC. Nous allons notamment nous intéresser au cycle de vie des ressources terminologiques et ontologiques au cours du processus d'annotation sémantique et de peuplement d'ontologie.

Chapitre 4. Cycle de vie des ressources terminologiques ou ontologiques

La démarche OntoPop se compose non seulement des Règles d'Acquisition de Connaissance décrites au chapitre précédent, mais également de propositions pour la mise en place de composants logiciels. Ces composants permettent d'opérationnaliser la démarche et de lancer les traitements nécessaires à l'analyse de corpus documentaires en vue des tâches d'annotation sémantique et de peuplement d'ontologie. Ces traitements s'effectuent en fonction du cycle de vie des ressources terminologiques ou ontologiques (RTO) dans OntoPop, à savoir les instances de l'ontologie et les descripteurs des thésaurus. Dans un premier temps, nous expliciterons en quoi ils constituent un véritable cercle vertueux dans OntoPop. Puis, nous présenterons plus en détail chacun de ces traitements à l'aide d'un exemple concret toujours issu du domaine de la Presse People.

4.1 OntoPop, un cercle vertueux

La démarche proposée par OntoPop constitue la passerelle nécessaire entre l'outil de représentation de la connaissance et les outils linguistiques d'extraction d'information utilisés par une application donnée, cf. Figure 43. Cette passerelle n'est rendue possible que par la définition des Règles d'Acquisition de Connaissance. Le cycle de vie des ressources terminologiques et ontologiques dans OntoPop s'organise donc autour de ces RACs, représentées sous la forme de pointillés dans le cycle de la Figure 43.

4.1.1 L'analyse linguistique

La première étape du cycle de vie des RTO dans OntoPop consiste à extraire d'un document soumis les informations pertinentes relatives au domaine étudié. Le moteur d'extraction analyse le document soumis en fonction de ses lexiques et de ses patrons d'extractions. Il repère les informations à extraire dans le document et les étiquettes afin de générer en sortie un arbre conceptuel comme vu au chapitre 2.

4.1.2 L'application des Règles d'Acquisition de Connaissance

La deuxième étape est opérée par le Module d'Annotation et d'Acquisition d'OntoPop qui se subdivise en deux composants : celui dédié au peuplement de l'ontologie et celui dédié à l'annotation documentaire. Ce module dispose d'un ensemble de RACs définies préalablement pour l'application

cible et les applique à l'arbre conceptuel du document analysé. Comme nous l'avons vu, les RACs sont déclenchées en fonction des nœuds indicateurs rencontrés dans l'arbre et de la résolution des indices contextuels relatifs à ces nœuds indicateurs.

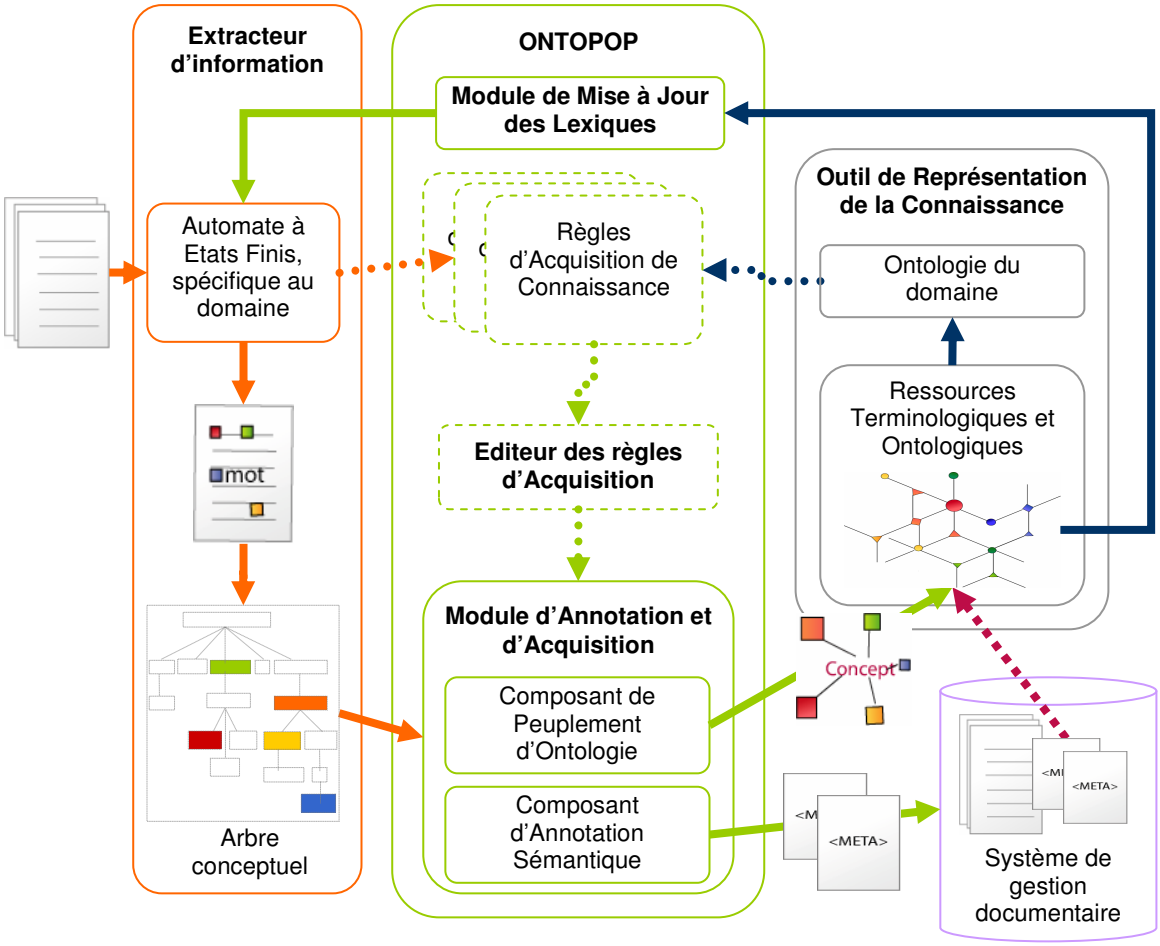


Figure 43. Le cercle vertueux d'OntoPop

L'application de ces RACs produit deux sortes de résultats. En fait, la création de ces résultats est dépendante des objectifs de l'application cible : il est possible de paramétrer OntoPop pour que l'application cible ne réalise que le peuplement d'ontologie ou que l'annotation documentaire ou bien les deux, en fonction des besoins. Dans le premier cas, les règles de connaissances produisent un réseau sémantique de connaissance dit « potentiel » car en attente de consolidation par le Composant de Peuplement d'Ontologie. Dans le second cas, elles produisent un ensemble d'annotations sémantiques également « potentielles » devant être contrôlées par le Composant d'Annotation Sémantique. Enfin, dans le dernier cas, les règles d'acquisition produisent un réseau sémantique potentiel conjointement avec les annotations documentaires potentielles.

Chacun des composants contrôle donc la validité des résultats produits ainsi que l'absence de redondance par rapport au référentiel terminologique et ontologique de l'outil de représentation des connaissances. Le réseau sémantique de connaissance est le premier à être contrôlé. En effet, si de nouvelles instances ou descripteurs sont créés, respectivement dans la base de connaissance ou dans les thésaurus du référentiel, ils pourront alors être utilisés par le Composant d'Annotation

Sémantique pour consolider les annotations potentielles. Les deux composants ne sont donc pas déclenchés simultanément mais en séquence.

Une fois le réseau sémantique contrôlé par le Composant de Peuplement d'Ontologie, il devient « valide » et il est importé dans l'outil de représentation des connaissances. De même, les annotations sémantiques contrôlées et « valides » vis-à-vis du Composant d'Annotation Documentaire sont mises à disposition d'un système de gestion de contenu documentaire. Elles y seront stockées en mode embarqué ou débarqué selon le mode de fonctionnement de ce système. Par contre, dans les deux cas, elles font référence aux RTO contenues dans l'outil de représentation des connaissances.

4.1.3 L'enrichissement des lexiques linguistiques

La troisième étape du cycle de vie des RTO dans OntoPop concerne l'enrichissement des lexiques de l'outil d'extraction d'information à partir des instances ou des descripteurs nouvellement créés dans l'outil de représentation de connaissance. Cette étape est réalisée par le Module de Mise à Jour des Lexiques. Il « écoute » en quelque sorte les créations, modifications ou suppressions qui sont opérées automatiquement par le Module d'Annotation et d'Acquisition, mais aussi manuellement par les utilisateurs humains de l'outil de représentation des connaissances. En effet, un utilisateur peut intervenir de deux manières. Premièrement, lors de la validation manuelle des instances ou descripteurs importés automatiquement par le Module d'Annotation et d'Acquisition. Cette validation a lieu dans le cas où l'application cliente requiert un processus semi-automatisé afin de pouvoir contrôler humainement la qualité de la connaissance ou des annotations produites. En revanche, certaines de nos applications fonctionnent de manière entièrement automatisée. Deuxièmement, cet utilisateur peut, à tout moment, en fonction de son statut et de ses droits dans l'outil de représentation des connaissances, ajouter, modifier ou supprimer de la connaissance par le biais des interfaces de saisie et d'édition standard de cet outil.

Par conséquent, à chaque création, modification ou suppression d'une ressource terminologique ou ontologique servant à alimenter les lexiques de l'outil linguistique, le Module de Mise à Jour récupère une copie de cette ressource et la transfère à l'outil d'extraction d'information. Ce dernier peut alors créer une nouvelle entrée dans le lexique adéquat, ou bien mettre à jour l'entrée correspondante à cette ressource dans le cas d'une modification, ou encore supprimer cette entrée du lexique.

Ces trois principales étapes du cycle de vie des RTO dans OntoPop définissent un cercle vertueux [BUI 03] : ces ressources, extraites à partir du contenu d'un document, sont tout d'abord utilisées pour enrichir le référentiel commun à la gestion de la connaissance et de la terminologie avant d'être de nouveau exploitées pour enrichir les lexiques initiaux de l'outil d'extraction d'information. Comme ces lexiques sont en phase avec le référentiel commun de l'application, les ressources concernées par l'enrichissement sont celles qui ont été **déduites** à partir des patrons d'extractions et non pas celles qui ont été tout simplement **reconnues** à partir d'entrées existantes d'un de ces lexiques.

Par exemple, un patron d'extraction est défini pour déduire un nom de société à partir d'un mot commençant par une lettre majuscule suivie d'un terme appartenant à l'ensemble suivant {SA, Ltd, Gmbh, …}. Il existe par ailleurs un lexique des noms de sociétés comprenant par exemple {Bouygues, France Télécom, PSA, Renault, Sanofi, SFR, …}. Prenons la phrase suivante : « Hier, France Télécom a racheté Orange SA pour un montant de ... ». Le premier nom de société « France Télécom » est **reconnu** par l'automate à partir de son entrée dans le lexique des sociétés. Une instance de la classe « Société » portant ce libellé est trouvée dans l'outil de représentation des connaissances. Il n'y a donc pas d'enrichissement de la base de connaissance et donc du lexique, même si cette instance est exploitée pour annoter le document d'origine.

Par contre, le deuxième nom de société « Orange » (absent du lexique des sociétés) est **déduit** à partir du patron d'extraction. Le Composant de Peuplement d'Ontologie vérifie l'existence d'une instance de la classe « Société » portant ce libellé dans le référentiel. Si cette instance n'existe pas, elle est créée dans la base de connaissance, et optionnellement validée manuellement par un utilisateur final. Puisqu'une nouvelle instance a été créée dans le référentiel, le Module de Mise à Jour envoie ce nouveau libellé, ainsi que sa classe d'appartenance, à l'outil d'extraction. Grâce à ces informations, ce dernier ajoute l'entrée « Orange » à son lexique des noms de sociétés. Par conséquent, au prochain document analysé, si le nom de la société « Orange » est mentionné, il sera **reconnu** à partir de son entrée dans le lexique.

Nous supposons qu'après un certain temps d'utilisation de l'application cliente, l'essentiel de l'information clef du domaine concerné sera intégré dans le référentiel de l'outil de représentation des connaissances, et par conséquent transmise aux ressources linguistiques utilisées par l'outil d'extraction d'information. Ainsi, les utilisateurs finaux n'auront pas à valider autant d'information qu'initialement. Par conséquent, ils disposeront de plus de temps pour des activités telles que la recherche d'information ou la publication. En conclusion, plus le système fonctionnera, plus le gain de productivité sera conséquent. Nous allons à présent détailler le fonctionnement de chacun des deux modules qui constituent la démarche OntoPop, i.e. le Module d'Annotation et d'Acquisition et le Module de Mise à Jour des Lexiques.

4.2 L'annotation sémantique et le peuplement ontologique

Le processus d'annotation sémantique et de peuplement d'ontologie, opéré par le Module d'Annotation et d'Acquisition d'OntoPop, se décompose en trois grandes étapes :

1) **Transformer** les arbres conceptuels en des formats correspondant aux tâches de peuplement d'ontologie ou d'annotation sémantique ;
2) **Consolider** ces formats en fonction de la modélisation imposée par l'ontologie et du contenu du référentiel terminologique et ontologique ;
3) **Faire valider** les annotations ou les instances nouvellement créées par un utilisateur humain, dans le cas d'un processus semi-automatisé.

Le Module d'Annotation et d'Acquisition est entièrement paramétrable, comme nous le verrons au chapitre 6, afin de s'adapter aux besoins de chaque application. Nous allons détailler le fonctionnement de chacune des trois étapes dans la suite de cette section.

4.2.1 La transformation

Une fois l'ensemble des Règles d'Acquisition de Connaissance définies et compilées pour former les deux feuilles de transformation XSLT (cf. section 3.3.2), le processus peut être déclenché sur un document ou un corpus de documents. Chaque document analysé produit un arbre conceptuel qui est « xml-isé » s'il n'a pas déjà été généré au format XML. Chacune des feuilles de transformation est appliquée à l'arbre conceptuel créant, d'une part un réseau sémantique de connaissance pour le peuplement de l'ontologie (au format XTM), et d'autre part un ensemble d'annotations sémantiques pour l'annotation documentaire (au format RDF). Nous allons illustrer chacune des transformations et des formats obtenus à partir de notre exemple issu de l'article « Le Clan Coppola ».

La Figure 44 représente, sous la forme de flèches, l'application des Règles d'Acquisition de Connaissance définies pour la tâche de peuplement d'ontologie entre :

- certaines étiquettes sémantiques de l'arbre conceptuel situé à gauche, et
- certains éléments (classes entitiés en orange, attributs en vert, classes relations en bleu et rôles en violet) de l'ontologie du domaine de la presse « People » (présentée au chapitre précédent) située à droite.

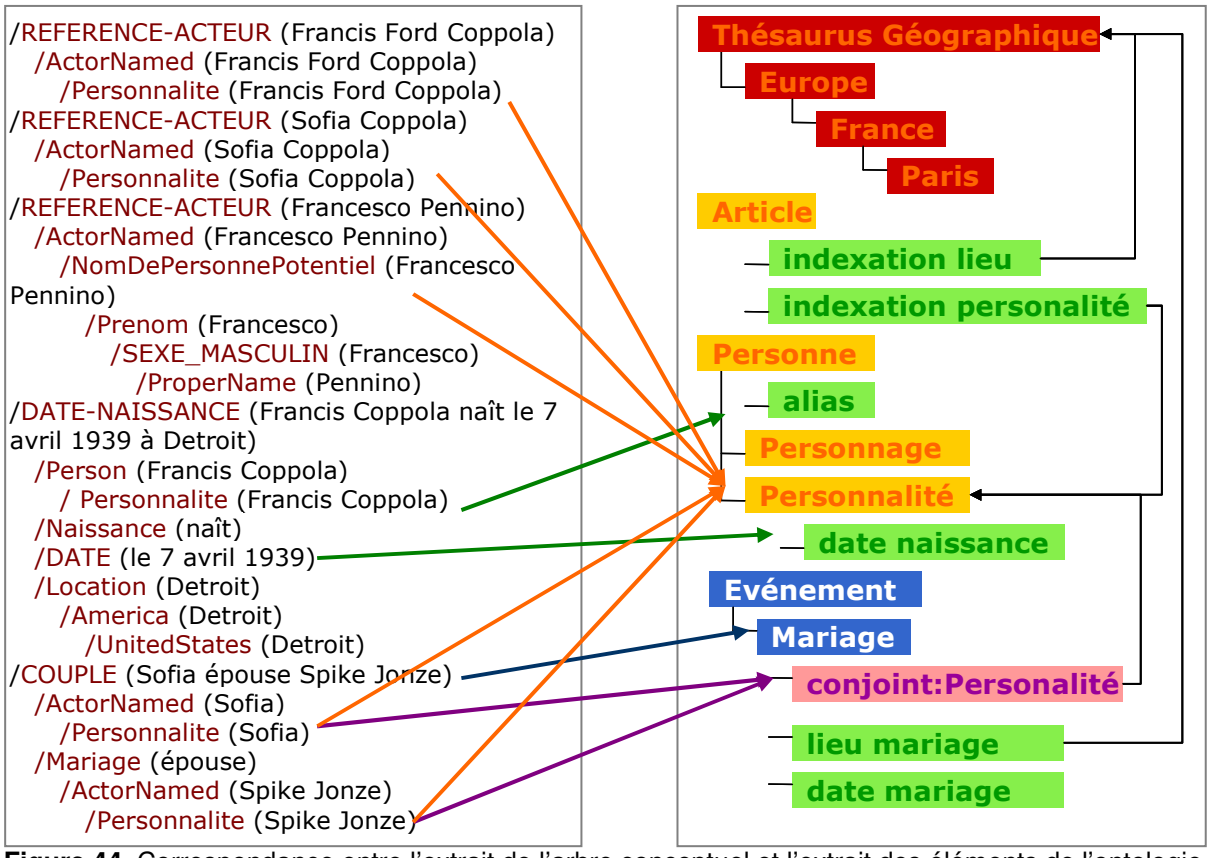


Figure 44. Correspondance entre l'extrait de l'arbre conceptuel et l'extrait des éléments de l'ontologie pour la tâche de peuplement d'annotation

L'application de ces règles, via la feuille XSLT, à l'arbre conceptuel crée le réseau sémantique présenté dans la Figure 45. La transformation XSLT correspond à une simple projection de l'arbre conceptuel vers un nouveau format. Cette projection induit un certain nombre de problèmes qu'il est nécessaire de résoudre avant de pouvoir importer ce réseau sémantique dans la base de connaissance. Nous étudierons ces problèmes dans la prochaine section mais mentionnons d'ors et déjà la redondance de l'information et la non-résolution des références directement liées à la transformation.

En effet, si une entité nommée est citée plusieurs fois dans le document d'origine, elle est à chaque fois extraite par le moteur d'extraction et apparaît donc autant de fois dans l'arbre conceptuel. Du coup, le réseau sémantique contient également autant d'instances identiques. Par exemple, dans l'arbre conceptuel de la Figure 44, « Sofia Coppola » apparaît deux fois et l'on retrouve deux instances de la classe « Personnalité » ayant le même label « Sofia Coppola » dans le réseau sémantique de la Figure 45. Or il s'agit bien de la même personnalité et ces deux instances doivent être fusionnées.

Par ailleurs, la simple projection de l'arbre en réseau sémantique ne permet par d'analyser les informations par rapport à un quelconque référentiel, ni même par rapport aux informations déjà présentes dans le réseau sémantique. Dans notre ontologie, la classe relation « Mariage » possède un rôle « conjoint » dont la valeur doit être une instance de la classe Personnalité. Or, dans le réseau sémantique de la Figure 45, les conjoints de l'instance de cette relation sont représentés par les valeurs textuelles « Sofia Coppola » et « Spike Jonze » et non par les instances « Sofia Coppola » et « Spike Jonze » de la classe Personnalité déjà existantes dans le réseau.

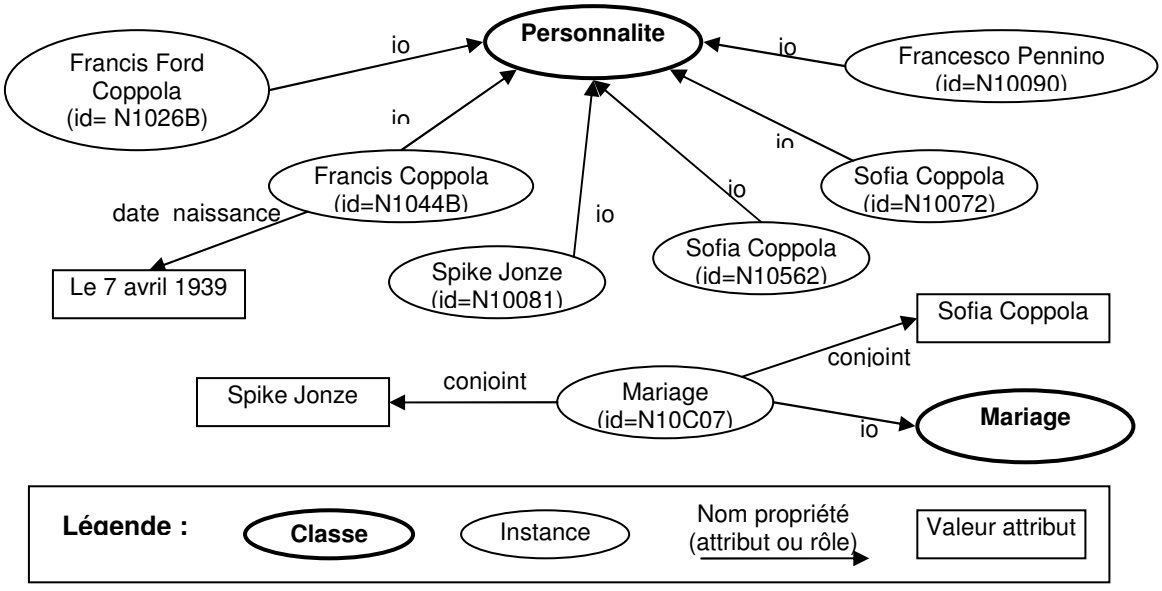


Figure 45. Extrait du réseau sémantique de connaissance généré

Nous allons maintenant présenter, à travers le même exemple, la création des annotations sémantiques. La Figure 46 représente, sous la forme de flèches, les Règles d'Acquisition de Connaissance qui donnent lieu à la création des annotations sémantiques présentées Figure 47. Ces

annotations sémantiques sont représentées dans le format RDF afin d'être exploitables par la plupart des outils de gestion de contenu documentaire.

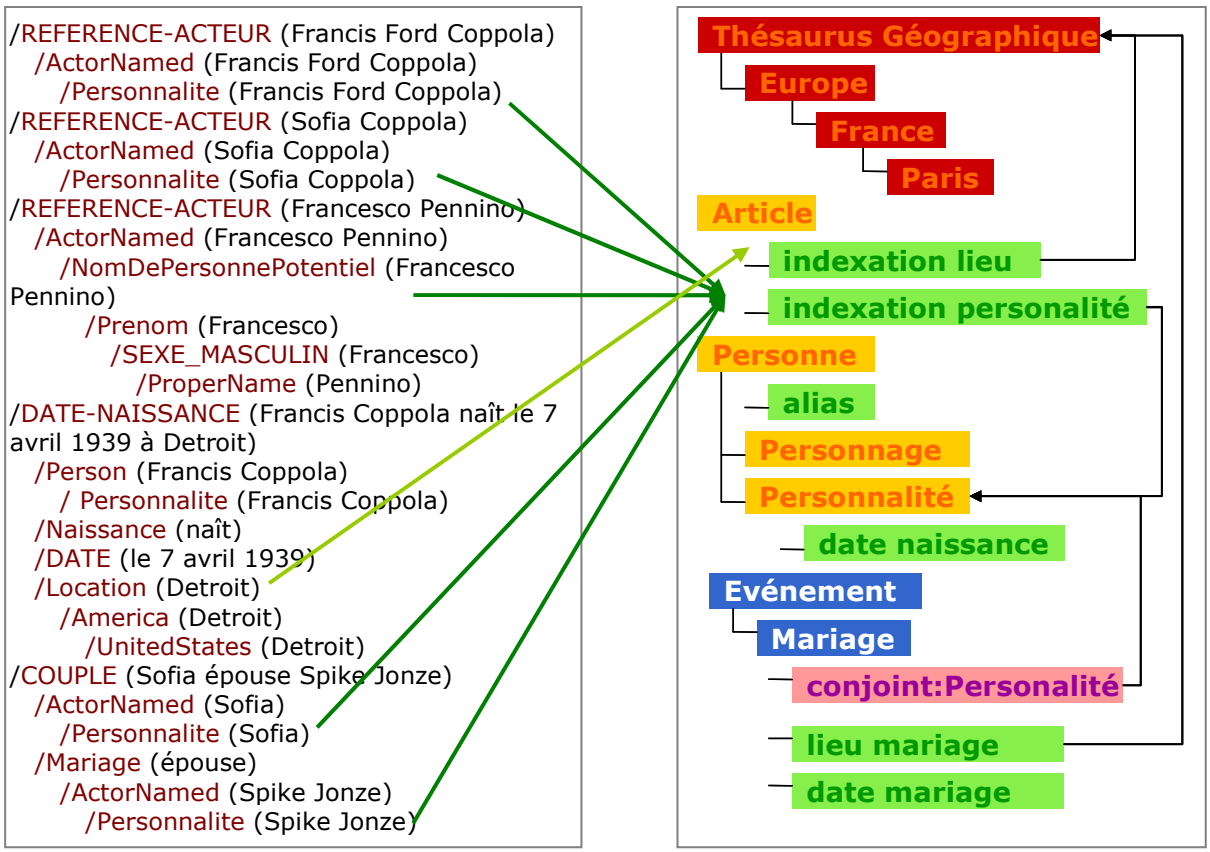


Figure 46. Correspondance entre l'extrait de l'arbre conceptuel et l'extrait des éléments de l'ontologie pour la tâche d'annotation sémantique

Le problème de la redondance des informations liées à la transformation de l'arbre conceptuel soulevé précédemment pour le réseau sémantique est automatiquement résolu dans le cas des annotations sémantiques. En effet, la représentation des annotations au format RDF par l'intermédiaire de l'API JENA[35] élimine automatiquement toute redondance dans les triplets RDF. Par contre, le problème de la non-résolution des références est également présent dans le cas des annotations sémantiques. Chaque annotation de la Figure 47 a pour valeur la chaîne de caractères extraite par l'outil linguistique alors que dans l'ontologie de la Figure 46, ces annotations sont modélisées comme des prédicats dont la valeur correspond à une ressource terminologique ou ontologique. La feuille de transformation XSLT n'est pas en mesure de résoudre ces références, puisqu'elle ne fait que reproduire l'information présente dans l'arbre conceptuel d'origine.

[35] Site web API JENA : http://jena.sourceforge.net/

```
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:onto="http://www.mondeca.fr/onto#">

<rdf:Description rdf:about="file:/E:/projets/people/Corpus/tests/coppola.xml">
        <rdf:type rdf:resource="http://www.mondeca.fr/onto#Article"/>
        <onto:indexation_personnalite>Francis Ford Coppola</onto:indexation_personnalite>
        <onto:indexation_personnalite>Sofia Coppola</onto:indexation_personnalite>
        <onto:indexation_personnalite>Francesco Pennino</onto:indexation_personnalite>
        <onto:indexation_personnalite>Francis Coppola</onto:indexation_personnalite>
        <onto:indexation_personnalite>Spike Jonze</onto:indexation_personnalite>
        <onto:indexation_lieu>Detroit</onto:indexation_lieu>
<rdf:Description>

</rdf:RDF>
```

Figure 47. Extrait des annotations sémantiques créés à partir de l'article « Le Clan Coppola »

En conclusion, la projection de l'arbre conceptuel vers un nouveau format, quel qu'il soit, à partir d'une simple feuille de transformation XSLT, nécessite de retravailler les résultats obtenus. La résolution des références vis-à-vis du contenu du référentiel terminologique et ontologique est complétée par le contrôle du respect des contraintes imposées par la modélisation de l'ontologie du domaine. Il s'agit de l'opération de consolidation des informations afin de les rendre « valides ». C'est l'objectif poursuivi par la prochaine étape du processus d'Annotation et d'Acquisition.

4.2.2 La consolidation

Comme indiqué dans [ALA 03], rares sont les outils de peuplement d'ontologie ou d'annotation sémantique qui explicitent, ou même mentionnent, l'étape de consolidation dans le flux de leurs traitements. Or, cette étape est extrêmement importante pour maintenir l'intégrité et la qualité du référentiel de l'application. En fait, la plupart des outils font surtout appel à une validation manuelle pour vérifier les annotations ou les instances générées par leur système. Certains outils d'annotation comme OntoMat [HAN 01] [HAN 02] ou SMORE [KAL 03b] intègrent un éditeur d'ontologie qui permet aux utilisateurs de vérifier les contraintes de domaine et de portée sur les annotations créées. Du côté du peuplement d'ontologie, l'un des seuls outils à s'être intéressé à la consolidation et à l'expliciter de manière détaillée est le projet ArtEquAkt [ALA 03].

Dans ce projet, Alani et al. répertorient quatre problèmes liés à l'intégration de nouvelles instances dans une base de connaissance : l'information dupliquée, la consolidation géographique, la consolidation temporelle et l'information inconsistante. Pour chaque problème, ils recensent les opérations de consolidation nécessaires à leur résolution :

- **Information dupliquée** : fusion des instances ayant le même libellé, fusion des instances si croisement de certains attributs positifs, fusion des attributs lorsqu'identiques (nom, valeur) ;
- **Consolidation géographique** : utilisation des relations de synonymie et de spécialisation d'un thésaurus géographique tel que le Thesaurus of Geographic Names (TGN) [HAR 97], désambiguïsation des noms de lieux par analyse du contexte (contenu du document ou du réseau sémantique associé) ;

- **Consolidation temporelle** : raisonnement sur les dates pour les identifier de manière précise, désambiguïsation des dates par analyse du contexte (contenu du document ou du réseau sémantique associé) ;
- **Information inconsistante** : fréquence d'extraction comme preuve de précision.

Leur approche pour résoudre ces problèmes consiste à instancier la base de connaissance avec les informations extraites des documents puis à appliquer un algorithme de consolidation basé sur un ensemble d'heuristiques et de méthodes d'expansion terminologique. Cet algorithme utilise la base lexicale WordNet [VOO 98] afin d'automatiser le traitement effectué sur les instances de la base de connaissance.

4.2.2.1 Les Tests de consolidation proposés par OntoPop

A contrario, je pense que pour préserver l'intégrité de la base de connaissance, cette étape de consolidation doit être réalisée avant la création des instances dans le référentiel. Cette opération s'effectue donc à partir des résultats obtenus à l'étape de transformation. Ces résultats sont analysés dans les moindres détails pour lever toute ambiguïté, toute inconsistance ou tout conflit avec des informations déjà existantes. Ainsi, sommes-nous à peu près sûrs que toute nouvelle information créée, annotation ou instance, préserve l'intégrité des référentiels et améliore ainsi la qualité de notre application.

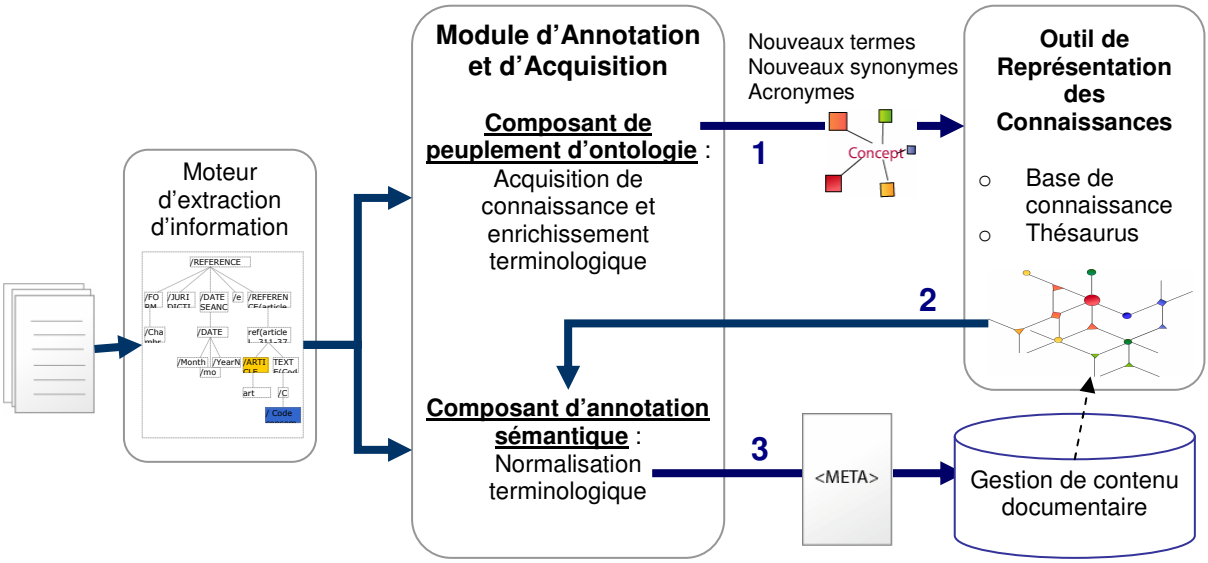


Figure 48. Processus de consolidation des informations extraites (instances et annotations)

Pour cela, le Module d'Annotation et d'Acquisition d'OntoPop va procéder à une batterie de tests qui vérifient la validité et la consistance de chacune des informations présentes soit dans le réseau sémantique, soit dans les annotations sémantiques résultant de l'étape précédente. Par ailleurs, comme les annotations peuvent faire référence à des instances du réseau sémantique non encore importées dans le référentiel terminologique et ontologique, l'étape de consolidation débute par la résolution des problèmes soulevés par les futures instances. Si ces dernières passent avec succès les

différents contrôles, elles seront automatiquement importées dans la base de connaissance ou bien dans un thesaurus, selon s'il s'agit d'instances de classes, d'attributs, de relations, ou bien de descripteurs candidats (1). Ainsi, comme illustré dans la Figure 48, lorsque vient le tour de la vérification des futures annotations sémantiques, ces instances et descripteurs sont accessibles dans le référentiel associé (2) pour être utilisés comme référence par les annotations (3).

Nous avons étudié les différents cas possibles de création d'instances et d'annotations sémantiques par les Règles d'Acquisition de Connaissance. Nous en avons déduit deux axes de consolidation :

- le premier axe définit l'élément concerné, i.e. une instance d'une classe entité, d'un attribut, d'une classe relation, un descripteur candidat du thésaurus ou une annotation sémantique ;
- le deuxième axe définit les contraintes devant être vérifiées, i.e. la non duplication, les restrictions de domaine, de portée et de cardinalité.

Le deuxième axe doit être adapté en fonction de l'élément concerné. En effet pour une instance de classe comme pour un descripteur du thésaurus, il n'y a pas lieu de contrôler les restrictions de domaine, de portée et de cardinalité. Mais, nous pouvons interpréter certaines de ces restrictions en fonction de l'élément : par exemple, la restriction de domaine sur une instance de classe pourra être considérée comme l'appartenance de cette instance à la bonne classe de l'ontologie. De même, la restriction de portée pour un attribut peut être comprise comme la vérification du type de donnée attendu par la base de connaissance, comme une chaîne de caractères, un numérique, une adresse URL ou encore une date.

En fonction de ces deux axes, nous avons répertorié l'ensemble des opérations de consolidation réalisées par le Module d'Annotation et d'Acquisition dans le Tableau 5. En plus de ces opérations de consolidation effectuées en fonction de chacun des éléments traités par OntoPop, il faut également procéder à l'évaluation du niveau de confiance accordée à chaque nouvelle instance ou annotation. En effet, nous avons vu au chapitre précédent qu'un niveau de confiance (par défaut « élevé ») est attribué à chacune des Règles d'Acquisition de Connaissance. Ainsi, si deux règles concurrentes se déclenchent sur deux étiquettes ayant des valeurs différentes pour instancier le même élément, seule celle possédant le niveau de confiance le plus élevé sera conservée. L'autre sera tout simplement supprimée.

Ceci est le seul cas où de l'information est supprimée. Dans tous les autres cas de consolidation cités, si l'instance, le descripteur ou l'annotation est rejeté par l'étape de consolidation, il est conservé dans un « tampon » afin d'être ultérieurement proposé à l'utilisateur pour correction et validation. Nous considérons que toute connaissance est exploitable même si cela requiert l'intervention d'un utilisateur humain. Par contre, l'information non conforme à l'existant ou à la modélisation de l'ontologie ne doit pas rendre le référentiel inconsistant. C'est pourquoi elle est sauvegardée à part des instances ou des annotations jugées valides par ces contrôles.

Eléments vs. Contraintes	Instance de classe entité	Instance d'attribut	Instance de classe relation	Descripteur candidat	Annotation sémantique
Information dupliquée	Contrôle de l'existence de l'instance dans la base de connaissance par : → recherche de son libellé ou de ses alias → recherche de ses propriétés identificatoires (attributs obligatoires, i.e. dont les cardinalités sont non nulles)	Contrôle de l'existence d'un attribut pour une instance donnée par : → recherche de ce type d'attribut sur l'instance donnée et vérification de sa valeur	Contrôle de l'existence d'une relation entre des instances données par : → recherche de ce type de relation sur chacune des instances données et vérification de leurs valeurs	Contrôle de l'existence du descripteur dans un thésaurus de l'application par : → recherche de son libellé ou de ses synonymes, variantes orthographique ou traductions	Contrôle de l'existence d'une annotation liée à un document donné par : → recherche de ce type d'annotation attaché au document donné et vérification de sa valeur (texte ou référence)
Domaine ou Classe	Contrôle de l'appartenance de l'instance à la bonne classe de l'ontologie ou à l'une de ses sous-classes	Contrôle de l'adéquation de la classe de l'instance à laquelle cet attribut est rattaché avec la modélisation de son domaine dans l'ontologie	Contrôle de l'adéquation de la classe de l'instance à laquelle cette relation est rattachée avec la modélisation de son domaine dans l'ontologie	Pas de contrôle, nouveau descripteur ajouté par défaut dans la classe « Descripteurs candidats »	Contrôle de l'adéquation de la classe de l'instance à laquelle cette annotation est rattachée avec la modélisation de son domaine dans l'ontologie
Portée ou Format de Données	Pas de contrôle	Contrôle de l'adéquation de la valeur de cet attribut avec la modélisation de son format de données dans l'ontologie (texte, date, numérique, etc.)	Contrôle de l'adéquation de la valeur (référence) de cette relation avec la modélisation de sa portée dans l'ontologie (classe autorisée et ses sous-classes)	Pas de contrôle	Contrôle de l'adéquation de la valeur (référence) de cette annotation avec la modélisation de sa portée dans l'ontologie (classe autorisée et ses sous-classes)
Cardinalité	Pas de contrôle	Contrôle du nombre d'attribut de ce type existants sur l'instance donnée	Contrôle de l'arité de la relation : les relations unaires ne sont pas considérées comme une vraie relation	Pas de contrôle	Pas de contrôle

Tableau 5. Opérations de consolidation réalisées par OntoPop en fonction des deux axes traités

Enfin, lorsqu'un nouveau descripteur doit être créé dans un des thésaurus de l'application, le processus de consolidation ne peut savoir avec précision à quel emplacement du thésaurus concerné ce nouveau descripteur doit être enregistré : quelle sera sa hiérarchie parmi la taxonomie des descripteurs existants ? quels seront ses termes associés ? etc. Par conséquent, tout nouveau descripteur est enregistré dans une classe nommée « Descripteur Candidat » du thésaurus concerné. Son véritable emplacement est ensuite confirmé par un utilisateur humain ayant la compétence et les droits d'édition sur le thésaurus en question.

4.2.2.2 Illustration des opérations de consolidation

Nous allons à présent détailler chacun des contrôles effectués en fonction du premier axe ci-dessus en les illustrant à partir des exemples fournis par le réseau sémantique de la Figure 45 et les annotations sémantiques de la Figure 47.

- **Les instances :**

Les instances proposées pour la classe entité Personnalité sont recherchées dans la base de connaissance en fonction de leur libellé principal et de leurs alias. Les instances « Francis Ford Coppola », « Sofia Coppola » et « Spike Jonze » sont trouvées dans la base de connaissance. La proposition « Francis Coppola » n'est pas retrouvée comme une instance mais comme l'alias de l'instance « Francis Ford Coppola ». Elle est donc fusionnée avec la référence de cette instance. La proposition « Fransisco Pennino » n'est pas retrouvée dans la base de connaissance : ni dans la classe Personnalité, ni dans une de ses sous-classes. Elle est donc mise de côté dans le tampon à moins que dans l'ontologie ne soit précisé que les instances de la classe Personnalité puissent être créées automatiquement. Dans ce cas, cette proposition d'instance est créée avec le statut « à valider ».

- **Les attributs :**

Le réseau sémantique contient un attribut de type « date naissance » à ajouter à l'instance « Francis Ford Coppola » de la classe entité Personnalité. Le processus vérifie tout d'abord que ce type d'attribut est autorisé sur cette classe (son domaine). Puis il recherche l'existence d'une instance d'attribut du type « date naissance » liée à l'instance « Francis Ford Coppola » :

- si **existe_pas**(date naissance(Francis Ford Coppola)) ou { si (**existe**(date naissance(Francis Ford Coppola)) et **nombre**(date_naissance(Francis Ford Coppola)) < **cardinalité**(date naissance(Personnalité)) },
- alors **création**(date_naissance(Francis Ford Coppola, 7 avril 1939)),
- sinon **rejet**(date_naissance(Francis Ford Coppola, 7 avril 1939)). Cette instance d'attribut est ajoutée au tampon avec la copie de l'instance à laquelle il est rattaché pour que le contexte soit préservé et restitué à l'utilisateur lors de la validation manuelle.

Lors de la création de l'instance d'attribut, le processus contrôle aussi que le type de données de la valeur respecte le type de données modélisé dans l'ontologie. Ainsi, l'attribut « date naissance » est de type « Date » avec un format tel que « JJ/MM/AAAA ». Or, dans notre exemple, « le 7 avril 1939 »

est une valeur textuelle qu'il est nécessaire de traiter pour pouvoir obtenir la valeur de type Date « 07/04/1939 ». L'instance d'attribut est donc rejetée et ajoutée au tampon pour validation manuelle.

- **Les relations :**

Le réseau sémantique contient une classe relation de type « Mariage » composée de deux rôles « conjoint » ayant pour valeurs les libellés « Sofia Coppola » et « Spike Jonze ». Le processus vérifie tout d'abord que ce type de rôle est autorisé sur la relation Mariage. Puis il regarde dans la base de connaissance s'il existe déjà une telle relation.

- si **existe_pas**(conjoint(Mariage, Spike Jonze) et conjoint(Mariage, Sofia Coppola)),
- alors **contrôle**(**portée**(conjoint, Personnalité))
    - si **existe**(Personnalité(Sofia Coppola)) et **existe**(Personnalité(Spike Jonze)),
    - alors **création**(conjoint(Mariage, Sofia Coppola) et **création**(Mariage(conjoint(Spike Jonze),
    - sinon **rejet**(conjoint(Mariage, Sofia Coppola) et **rejet**(Mariage(conjoint(Spike Jonze)

Dans le cas où il ne reste plus qu'un rôle valide dans la relation, elle sera également ajoutée au tampon. Elle correspond à une relation unaire et donc n'a pas beaucoup de sens à être importée telle quelle dans la base de connaissance.

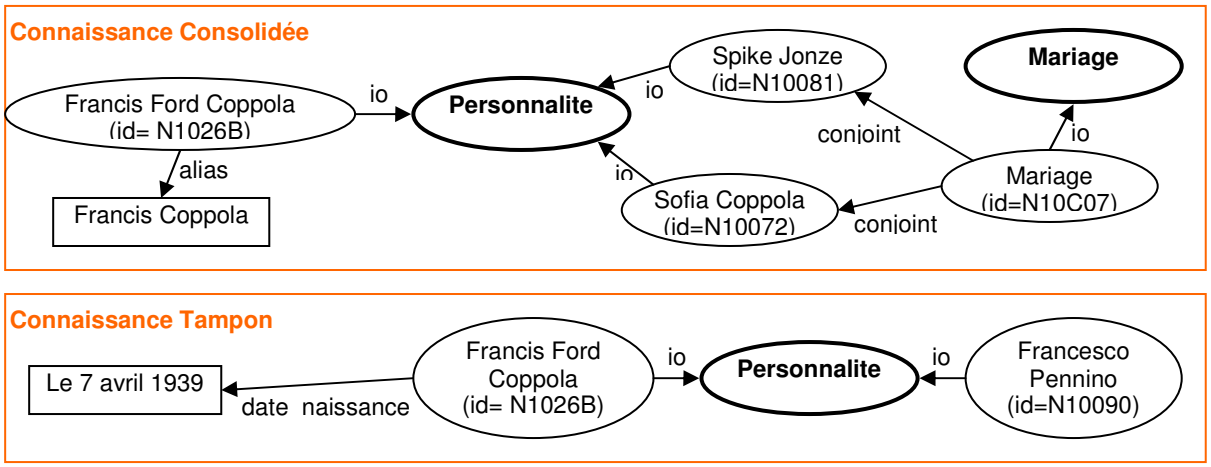


Figure 49. Exemple de consolidation du réseau sémantique pour le peuplement d'ontologie

- **Les descripteurs :**

Dans notre exemple, il existe une proposition de descripteur géographique, « Detroit ». Ce descripteur est recherché dans le thésaurus géographique dans le libellé des descripteurs, leurs synonymes, leurs variantes orthographiques, leurs traductions et tout autre terme décrivant un descripteur. Si ce descripteur n'est pas retrouvé dans le thésaurus géographique, il est ajouté au tampon, à moins que la création automatique ne soit paramétrée. Auquel cas, ce descripteur est ajouté à la classe « Descripteur Candidat ».

- **Les annotations sémantiques :**

Les annotations sémantiques proposées par OntoPop pour la biographie Coppola se divisent en deux types : « indexation_personnalité » et « indexation_lieu ». Le processus vérifie d'abord s'ils sont autorisés sur la classe Article puis si la portée de chacune des annotations proposées respecte les restrictions modélisées dans l'ontologie. Ainsi, « indexation_personnalité » doit référencer une instance de la classe Personnalité ou de ses sous-classes alors que « indexation_lieu » doit être lié à un descripteur du thésaurus géographique. Dans les propositions de la Figure 47, l'instance « Francesco Pennino » de la classe Personnalité n'a pas été créée dans la base de connaissance et ne peut donc servir de référence. Il en est de même avec le descripteur géographique « Detroit ». Ces deux propositions sont transférées dans le tampon de l'annotation alors que les autres propositions sont créées après avoir récupéré les références existantes dans le référentiel.

Annotations Sémantiques Consolidées	Annotations Sémantiques Tampon
Index_Personnalité Francis Ford Coppola **Index_Personnalité :** Sofia Coppola **Index_Personnalité :** Spike Jonze	**Index_Personnalité:** Francesco Pennino **Index_Lieu :** Detroit

Figure 50. Exemple de consolidation des annotations sémantiques pour l'annotation documentaire

OntoPop résout aussi partiellement les problèmes de consolidation temporelle et géographique décrit dans [ALA 01]. En effet, le contrôle imposé sur les formats de dates ou sur les références vers un thésaurus géographique sont une première étape vers la consolidation temporelle ou géographique. Mais nous considérons que le contexte de chaque information temporelle ou géographique à consolider est rarement porteur d'information co-occurrente permettant de résoudre les ambiguïtés comme cela est le cas dans le corpus utilisé par [ALA 01]. De même, le problème de l'information inconsistante est laissé à l'appréciation de l'utilisateur humain lors de l'étape de validation. Lui, mieux que tout système informatisé, saura en quoi l'information est inconsistante et comment résoudre ce problème par la modification, la création, ou même la fusion de d'entités du référentiel (instances ou descripteurs).

En résumé, l'étape de consolidation implémentée dans le Module d'Annotation et d'Acquisition d'OntoPop consiste à :

- contrôler les instances du réseau sémantique puis les annotations sémantiques en fonction du modèle de l'ontologie (restrictions de domaine, de couverture, de cardinalité), des vocabulaires contrôlés, comme les thésaurus ou les tables de références, et de la base de connaissance ;
- importer les nouvelles instances valides dans la base de connaissance ou dans les vocabulaires contrôlés et créer les annotations sémantiques valides ;
- enregistrer les instances et les annotations inconsistantes dans un tampon pour une validation manuelle de l'utilisateur final ultérieure.

4.2.3 La validation

Dans le cas d'une application semi-automatisée, l'utilisateur doit valider les résultats générés par le Module d'Annotation et d'Acquisition afin d'en contrôler la qualité. Pour cela, il dispose d'une seule et unique interface, cf. 6.2.5, pour valider à la fois les annotations sémantiques et les instances créées. Il peut les éditer, les modifier ou encore les supprimer. Les annotations et instances ayant été rejetées à l'étape de consolidation sont aussi présentées à l'utilisateur. Celui-ci pourra « repêcher » les informations jugées essentielles et que le processus de consolidation n'a pas réussi à résoudre automatiquement. Elles peuvent aussi être fusionnées avec des instances existantes. Une fois l'ensemble des résultats validés par l'utilisateur, celui-ci peut alors les exploiter à travers les activités relatives à son métier : la recherche d'information, la publication, l'édition, etc. En guise d'exemple, la Figure 51 présente le résultat d'une validation manuelle opérée à partir de la connaissance consolidée et de la connaissance tampon. Nous obtenons donc un nouveau réseau sémantique qui aura le statut « valide » dans la base de connaissance.

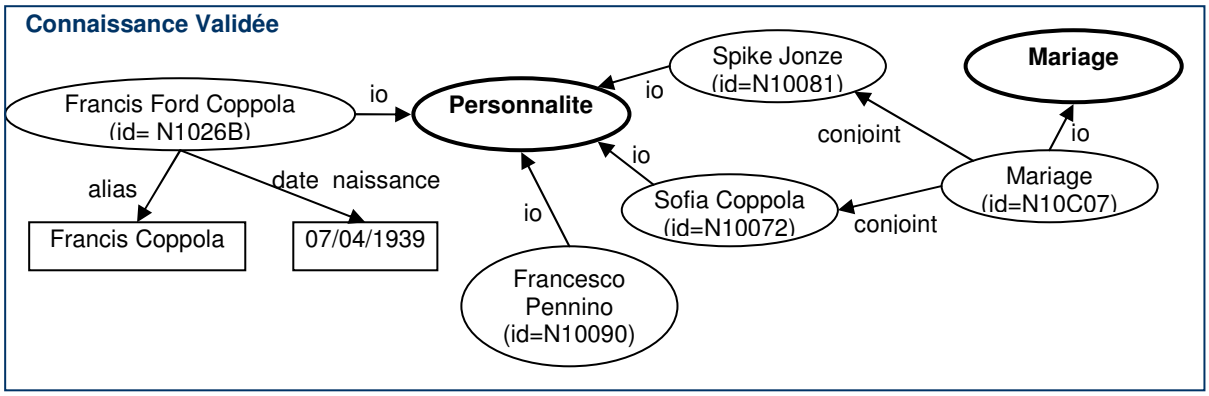


Figure 51. Exemple de réseau sémantique validé pour le peuplement d'ontologie

Le Module d'Annotation et d'Acquisition proposé par OntoPop implémente les trois étapes principales du processus de peuplement d'ontologie et d'annotation sémantique : la transformation, la consolidation et la validation. Il offre une solution à la fois simple dans sa mise en œuvre et pourtant robuste, flexible, évolutive et adaptable aux divers besoins des applications, grâce notamment à un paramétrage fin du processus.

A présent que l'outil de représentation des connaissances a été enrichi par de nouvelles ressources terminologiques et ontologiques extraites des documents, il est intéressant de se pencher sur le phénomène inverse : l'enrichissement des ressources linguistiques de l'outil d'extraction d'information à partir de ces mêmes RTO du référentiel de l'application.

4.3 La maintenance des lexiques et autres ressources linguistiques

L'objectif principal d'OntoPop est d'assister les utilisateurs dans les tâches fastidieuses d'annotation sémantique et de peuplement d'ontologie de leurs applications métier. Ils pourront ainsi consacrer leur temps et leurs compétences aux activités propres à leur métier. Or, le gain en qualité et en productivité procuré par le Module d'Annotation et d'Acquisition ne doit pas être perdu lors de l'étape de validation décrite précédemment. En d'autres termes, si l'utilisateur se voit contraint de constamment corriger et valider la plupart des propositions remontées par le Module d'Annotation et d'Acquisition, tout le bénéfice apporté par les étapes précédentes devient alors caduc. Par conséquent, il est nécessaire de capitaliser sur ces corrections et validations entrées par l'utilisateur.

Nous avons choisi d'intégrer ces corrections et validations au processus grâce à la maintenance des ressources terminologiques et ontologiques. Cette maintenance s'applique aux nouvelles entités (instances et descripteurs) créées et validées dans le référentiel dans le cadre de la tâche de peuplement de l'ontologie. Comme nous l'avons vu au chapitre 2, le document est analysé en fonction de la cartouche linguistique spécifique au domaine étudié. Chaque entrée du lexique est extraite puis les patrons d'extraction sont appliqués au fur et à mesure jusqu'à l'obtention de l'arbre conceptuel final. Parmi les informations extraites, il y a donc de la connaissance reconnue provenant des lexiques et de la connaissance déduite provenant des patrons d'extraction (cf. section 4.1).

A priori, l'outil de représentation des connaissances contient le référentiel terminologique et ontologique de l'application, ce qui signifie que toutes les informations extraites à partir des lexiques sont également connues de ce référentiel. Par contre, dans le cas de la connaissance déduite, le référentiel peut ne pas posséder d'entité correspondant à cette nouvelle connaissance. C'est cette **nouvelle** connaissance qui est exportée vers l'outil d'extraction d'information afin qu'elle soit intégrée à ses lexiques puis reconnue.

En fait, nous pouvons pousser le raisonnement un peu plus loin car il est également intéressant de capitaliser les opérations d'ajout, de modification et de suppression réalisées par les utilisateurs humains dans le référentiel commun à travers les écrans d'édition standard de l'outil de représentation des connaissances. Toutefois, les suppressions ou modifications d'entités existantes du référentiel sont à manier avec une certaine précaution : la suppression ou même la modification d'une entité peut entraîner une incohérence au niveau des annotations sémantiques qui reposent sur la référence à cette entité [MAG 05].

Par maintenance des ressources terminologiques ou ontologiques, nous entendons donc la mise à jour des lexiques, et autres ressources linguistiques, des outils d'extraction d'information en fonction des entités (instances de la base de connaissance ou descripteurs d'un thésaurus) validées, créées, modifiées ou supprimées dans le référentiel d'une application donnée. Comme illustré dans la Figure 52, ces entités enrichissent les lexiques de la cartouche linguistique conçue pour le domaine de l'application. Ainsi, ces entités seront automatiquement reconnues et interprétées par l'outil

d'extraction d'information à sa prochaine utilisation par l'application cible. Précisons que la maintenance RTO est un processus optionnel dans OntoPop car dépendant des objectifs et des besoins de l'application cible. En effet, nous désirons conserver toute la flexibilité de la solution apportée par OntoPop.

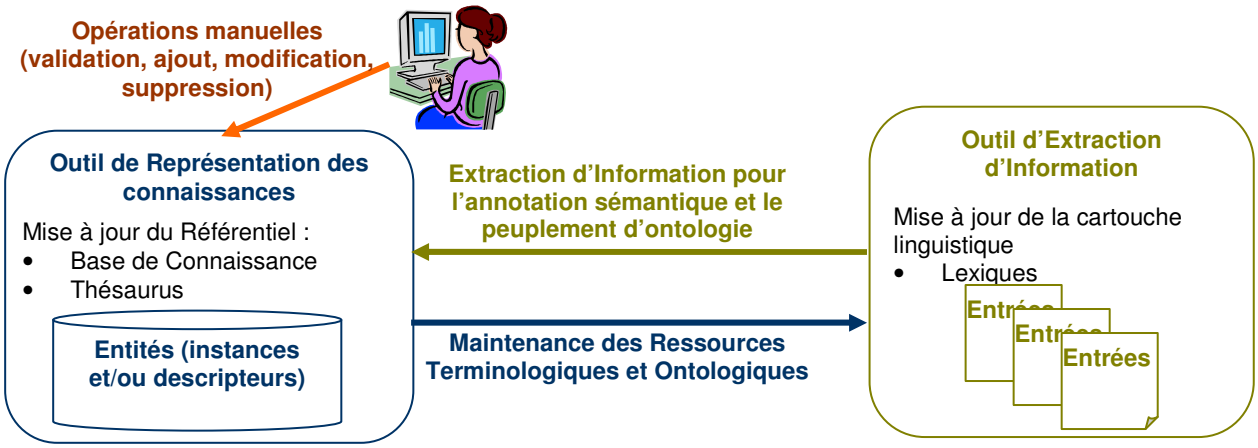


Figure 52. Capitalisation des entités du référentiel en entrées des lexiques de l'outil d'extraction

Nous allons à présent illustrer le processus de maintenance des RTO toujours à l'appui du même exemple. Comme montré dans la **Figure 53**, à l'issue de l'étape de validation :

- une nouvelle instance de la classe « Personnalité », i.e. « Francesco Pennino », a été créée dans la base de connaissance,
- une autre instance de cette classe « Personnalité », i.e. « Francis Ford Coppola », a été modifiée afin de lui ajouter l'alias « Francis Coppola »,
- et enfin un descripteur du thésaurus géographique, i.e. « Détroit », a été modifié afin de lui ajouter la variante orthographique « Detroit ».

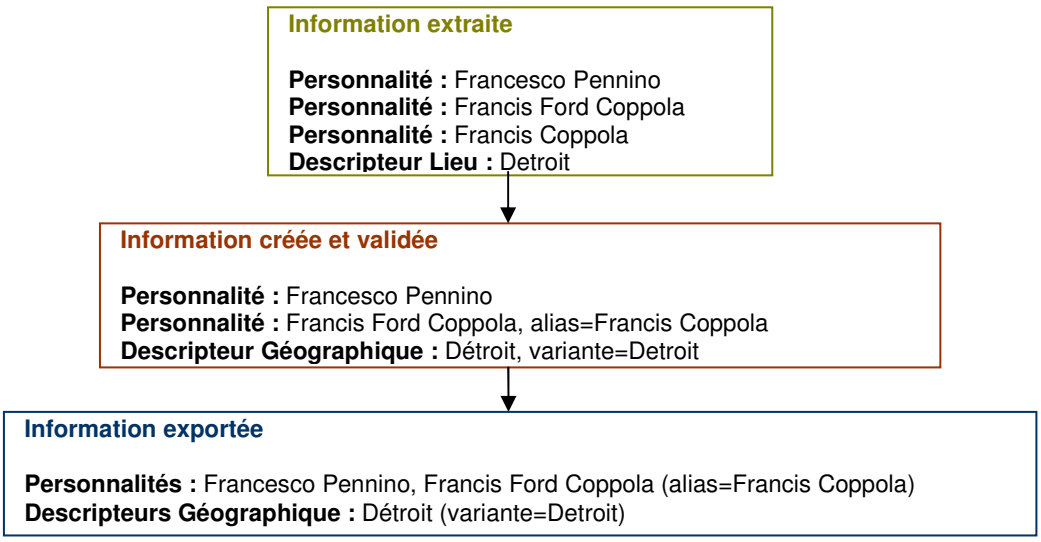


Figure 53. Exemple de mise à jour des ressources terminologiques et ontologiques

La mise à jour des lexiques du moteur d'extraction est déclenchée en fonction d'un ensemble d'alertes configurées dans l'outil de représentation des connaissances sur chacune des classes du référentiel

dont les instances doivent être exportées. Ces alertes écoutent les différentes opérations effectuées sur ces entités, notamment dans les cas de :

- création automatique ou manuelle d'une entité inconnue du référentiel ;
- modification manuelle d'un des termes (libellé, alias, synonyme, variante orthographique, traduction, etc.) représentant une entité existante ;
- validation du libellé de l'information extraite comme nouveau terme d'une entité existante ;
- suppression manuelle d'une entité existante.

Lorsqu'une alerte reconnaît l'un des cas ci-dessus au sujet d'une entité dont la classe a été configurée, elle récupère alors les différents termes de l'entité concernée, l'intitulé de sa classe ainsi que l'opération exécutée (création, modification ou suppression) :

1) La nature de l'opération effectuée dans le référentiel permet de savoir quelle action exécuter du côté de l'outil linguistique. En effet, s'il s'agit d'une création, alors l'entrée exportée est ajoutée au lexique correspondant dans l'outil d'extraction d'information. Par contre, s'il s'agit d'une modification ou d'une suppression, l'outil d'extraction doit d'abord retrouver l'entrée existante dans le bon lexique avant de mener à bien la modification de cette entrée ou sa suppression définitive.

2) L'intitulé de la classe permet à l'outil d'extraction de connaître le lexique à enrichir ou à modifier le cas échéant. En effet, il se peut que l'outil d'extraction d'information utilise plusieurs lexiques dans la même cartouche linguistique. Il est donc nécessaire d'avoir une information permettant de déduire à quel lexique doit être ajoutée la nouvelle entrée. Par contre, un couplage simple (car non contextualisé, à l'inverse des Règles d'Acquisition de Connaissance) doit pouvoir être mis en place. En effet, les noms des lexiques ne coïncident pas forcément avec les noms donnés aux classes de l'ontologie ou des thésaurus de l'application. Ainsi, l'instance « Francisco Pennino » appartient à la classe « Personnalité » dans l'ontologie du domaine de la presse people mais son entrée associée est enregistrée dans le lexique « Personne » de la cartouche linguistique d'extraction des Entités Nommées, elle-même incluse dans la cartouche du domaine de la presse People.

3) Les différents termes de l'entité transmise correspondent aux entrées du lexique concerné dans l'outil d'extraction. Selon le langage utilisé par l'outil d'extraction pour créer ces entrées, il se peut que des heuristiques doivent être appliquées afin de générer la forme correspondante à l'entrée pour que celle-ci puisse être extraite. Par exemple, dans l'outil d'extraction IDE™, un espace entre deux mots est représenté par le caractère « \ ». Ainsi, l'entrée « Francis Coppola » doit être générée par l'application d'une de ces heuristiques afin d'obtenir la forme « Francis \ Coppola ». De même, pour certaines entrées dont la forme correspond à un nom commun, comme la société « Orange » par exemple, il est nécessaire d'appliquer une heuristique spécifiant que cette entrée ne doit être repérée par l'outil que si elle débute par une lettre majuscule. Ainsi, le nom commun « orange » ne sera jamais étiqueté comme un nom de société.

En résumé, chacune des entités exportées est analysée par le moteur d'extraction qui complète ses dictionnaires d'entités nommées ou toute autre ressource lexicale avec les différents termes attachés à l'entité. Ces nouvelles ressources linguistiques sont compilées dans une nouvelle version de la cartouche linguistique de l'application afin d'être prises en compte durant la prochaine tâche d'extraction d'information. Par conséquent, la cohérence des ressources linguistiques est conservée vis-à-vis de l'ensemble du référentiel de l'application détenu par l'outil de représentation des connaissances.

4.4 Conclusion

La démarche OntoPop propose un certain nombre de modules logiciels permettant d'implémenter des applications orientées métier pour le peuplement d'ontologie et l'annotation sémantique. La démarche d'enrichissement du référentiel de l'application grâce aux RTO extraites par le moteur d'extraction d'information est habilement complétée par la capitalisation à partir du retour d'expérience prodigué par les utilisateurs humains. Ce retour d'expérience a lieu aussi bien lors de la phase de validation des propositions suggérées par le Module d'Annotation et d'Acquisition que lors de l'enrichissement manuel du référentiel à travers ses écrans standards d'édition. Il permet ainsi l'enrichissement des ressources linguistiques utilisées par le moteur d'extraction. L'ensemble du processus de peuplement d'ontologie et d'annotation sémantique peut donc être considéré comme un cercle vertueux. De plus, une attention constante est apportée à la préservation de l'intégrité du référentiel de l'application à chacune des phases du processus.

Forte des expériences acquises au cours de divers projets en entreprise, nous avons élaboré une méthodologie proposant une opérationnalisation de cette démarche étape par étape. La prochaine partie de ce mémoire présente cette méthodologie puis décrit l'implémentation technique des différents composants logiciels qui forment la plateforme proposée aux entreprises pour la mise en œuvre de la démarche OntoPop.

Troisième Partie.

L’implémentation de notre solution OntoPop

Chapitre 5. La méthodologie OntoPop

La démarche OntoPop a été implémentée et testée lors de divers projets au cours de cette thèse. Ceci a donné lieu à une réflexion sur la meilleure manière de mener à bien un projet d'application d'annotation sémantique et/ou de peuplement d'ontologie qui soit basée sur la solution que nous proposons. Nous avons donc progressivement élaboré cette méthodologie dont nous allons présenter les grandes étapes dans ce chapitre.

5.1 Présentation générale de la méthodologie

L'objectif de cette méthodologie consiste à déployer des applications d'annotation sémantique et/ou de peuplement d'ontologie selon les besoins d'un client dans un domaine donné. Autrement dit, elle doit faciliter l'opérationnalisation de la démarche vue dans la précédente partie pour des applications et des besoins concrets. Elle doit notamment permettre de créer un couplage optimal entre les arbres conceptuels fournis par le moteur d'extraction utilisé et les différents éléments du référentiel métier de l'application. Par optimal, nous entendons qu'il soit le plus riche, le plus complet et le plus pertinent possible pour une application donnée.

La méthodologie OntoPop fonctionne de manière progressive et itérative entre les différents intervenants humains dans la mise en application de la démarche. Ces acteurs peuvent être classés en fonction de leur rôle et de leur expertise dans la méthodologie :

- **L'expert** du domaine (le plus souvent le client de l'application cible) apporte ses besoins, spécifie les limites de l'application et du domaine concerné, fournit les ressources existantes et enfin valide l'ensemble de la solution.
- **Le linguiste** se charge de l'analyse textuelle en fonction d'un corpus de documents représentatifs et livre un outil d'extraction d'information adapté au domaine.
- **L'ontographe**[36] modélise l'ontologie du domaine et reprend les données existantes (bases de données, thésaurus, terminologies, etc.) de l'application source pour les enregistrer dans la base de connaissance et les thésaurus de l'application cible.
- **L'intégrateur** met en place les différents outils et implémente la solution dans sa globalité.

Comme nous l'avons vu dans la partie précédente, les règles d'acquisition composent le cœur de la démarche d'OntoPop. Elles sont un pré-requis essentiel au bon fonctionnement de la future application. C'est à l'intégrateur que revient la tâche de modéliser ces règles, sur la base des informations fournies par le linguiste, le client et l'ontographe. Mais avant de pouvoir définir ces règles, il faut, pour tout projet ou domaine donné, savoir évaluer les charges de spécifications et de développement inhérentes à chacun des outils utilisés. Il est pour cela nécessaire :

[36] Merci à Philippe Laublet pour l'apport de ce terme permettant de décrire la personne chargée d'implémenter une ontologie

a) d'estimer la capacité d'extraction d'information sur un nouveau domaine et la couverture pouvant être atteinte sur des corpus documentaires représentatifs, et

b) de modéliser l'ontologie en tenant compte à la fois de la formalisation, de la couverture ciblée pour la nouvelle application et de la reprise de l'existant.

Il est également important de pouvoir, au plus tôt du déroulement d'un projet, mettre en commun le besoin client et ses problématiques spécifiques. Il faut aussi identifier rapidement la couverture commune au moteur d'extraction et à l'ontologie du domaine. Pour ce faire, des documents d'échange, comme la structure des ressources linguistiques et de l'ontologie, doivent être partagés entre les différents intervenants afin de réaliser une première intégration dans les plus courts délais. Cette dernière sera ensuite itérativement développée, testée et affinée avant d'être finalement validée par le client. Ces quelques principes de base m'ont permis de définir la méthodologie OntoPop, inspirée également des recommandations du génie logiciel [SCH 97] autour des étapes d'évaluation des besoins, de construction de la solution logicielle, de sa validation et de son utilisation dans une application donnée. Ce cycle de vie se décline de la manière suivante dans la méthodologie OntoPop :

1) **La Phase d'Etude** : Etude sur les données à traiter par l'application cible ainsi que leurs sources et définition de la couverture du domaine ;
2) **La Phase de Structuration** : Définition de la structure des arbres conceptuels, résultats des outils de TAL, et modélisation de l'ontologie du domaine ;
3) **La Phase de Couplage** : Couplage des arbres conceptuels avec les éléments de l'ontologie et définition des Règles d'Acquisition de Connaissance ;
4) **La Phase de Validation/Qualité** : Validation des annotations et des instances créées et réitération si besoin est à partir de la Phase de Structuration.
5) **La Phase de Livraison** : Livraison et maintenance auprès du client.

Nous allons à présent détailler chacune de ces cinq étapes constituant la méthodologie OntoPop en mettant l'accent sur le rôle de chacun des intervenants, les actions qu'ils doivent mener et les moyens qu'ils ont à leur disposition pour atteindre les objectifs.

5.2 La Phase d'Etude

Cette phase consiste à déterminer les matériaux de base nécessaires à l'analyse linguistique mais aussi à la représentation des connaissances destinées à l'application cible. Au cours de cette étape de familiarisation, l'ontographe et le linguiste cherchent à comprendre à la fois le domaine concerné, le vocabulaire utilisé pour décrire ce domaine et les problématiques que la future application doit résoudre. L'expert leur fournit tout le support nécessaire à la compréhension des points clefs du domaine, de ses frontières ainsi que des objectifs de l'application. Il doit pour cela formuler des réponses claires et précises à un ensemble de questions, parmi lesquelles :

- des questions d'ordre général, comme

  - l'application est-elle axée sur le peuplement d'ontologie, l'annotation documentaire ou les deux ?
  - l'application doit-elle être entièrement automatisée ou semi-automatisée avec validation des résultats par un utilisateur final ?
  - existe-t-il des spécifications fonctionnelles et techniques de l'application cible ?
  - quels sont les points sensibles dans les processus de l'application existante devant être résolus dans l'application cible ?
- des questions d'ordre documentaire, comme
  - quels sont les flux documentaires existants et à prévoir dans la future application ?
  - quels sont les documents textuels ou autres représentatifs du domaine ?
  - quels sont les documents textuels ou autres manipulés par l'application existante et devant l'être par l'application cible ?
  - quels sont les formats documentaires des ressources textuelles (Word, PDF, XML, HTML, …) ?
- des questions d'ordre de la gestion des connaissances, comme
  - existe-t-il une ou plusieurs bases de données, ou mieux encore une ou plusieurs bases de connaissances, liées à l'application existante et devant être reprises dans l'application cible ?
  - existe-t-il des ressources terminologiques, sous la forme de lexiques, de taxonomies ou de thésaurus, utilisés par l'application existante ou externes mais correspondant au domaine concerné ?
  - les données existantes doivent-elles être conservées dans la future application ?
  - leur structure (schéma ou modèle) doit-elle être également préservée ou est-il possible de repenser complètement cette structure ?
  - les documents déjà annotés, s'il existe une base documentaire existante, doivent-ils être ré-annotés en fonction du nouveau modèle de représentation des connaissances ?

En fonction des données existantes ou non, le linguiste et l'ontographe vont étudier le domaine afin de cerner les possibilités tant au niveau de la couverture du moteur d'extraction sur le corpus documentaire qu'au niveau de la modélisation d'une ontologie de domaine, cf. Figure 54.

Le linguiste procède par une approche en mode découverte [ENJ 05b], c'est-à-dire qu'il part des données textuelles présentes dans les documents du corpus représentatif fourni par l'expert afin d'évaluer le degré d'analyse linguistique nécessaire. Il se familiarise avec le vocabulaire utilisé, avec la structure des phrases mais aussi avec la structure même des documents qui peuvent aussi apporter des éléments importants pour la construction des patrons d'extraction. En ce qui concerne l'élaboration des ressources linguistiques, et notamment des différents lexiques utilisés par les patrons d'extraction, il va recueillir et analyser toutes les ressources terminologiques mises à sa disposition par l'expert. Ces ressources terminologiques comprennent aussi bien des lexiques, des taxonomies, des thesaurus que des listes d'entités nommées exportées des bases de données ou des bases de connaissances existantes. Enfin, afin de cerner la couverture de la cartouche linguistique du domaine et les informations essentielles à extraire pour l'application finale, le linguiste procède en plus à une série d'entretiens avec l'expert. Celui-ci va notamment détailler la manière dont les informations

extraites vont être exploitées par l'application cible. Il peut aussi lui préciser les besoins spécifiques à l'application cible, comme la normalisation des informations extraites.

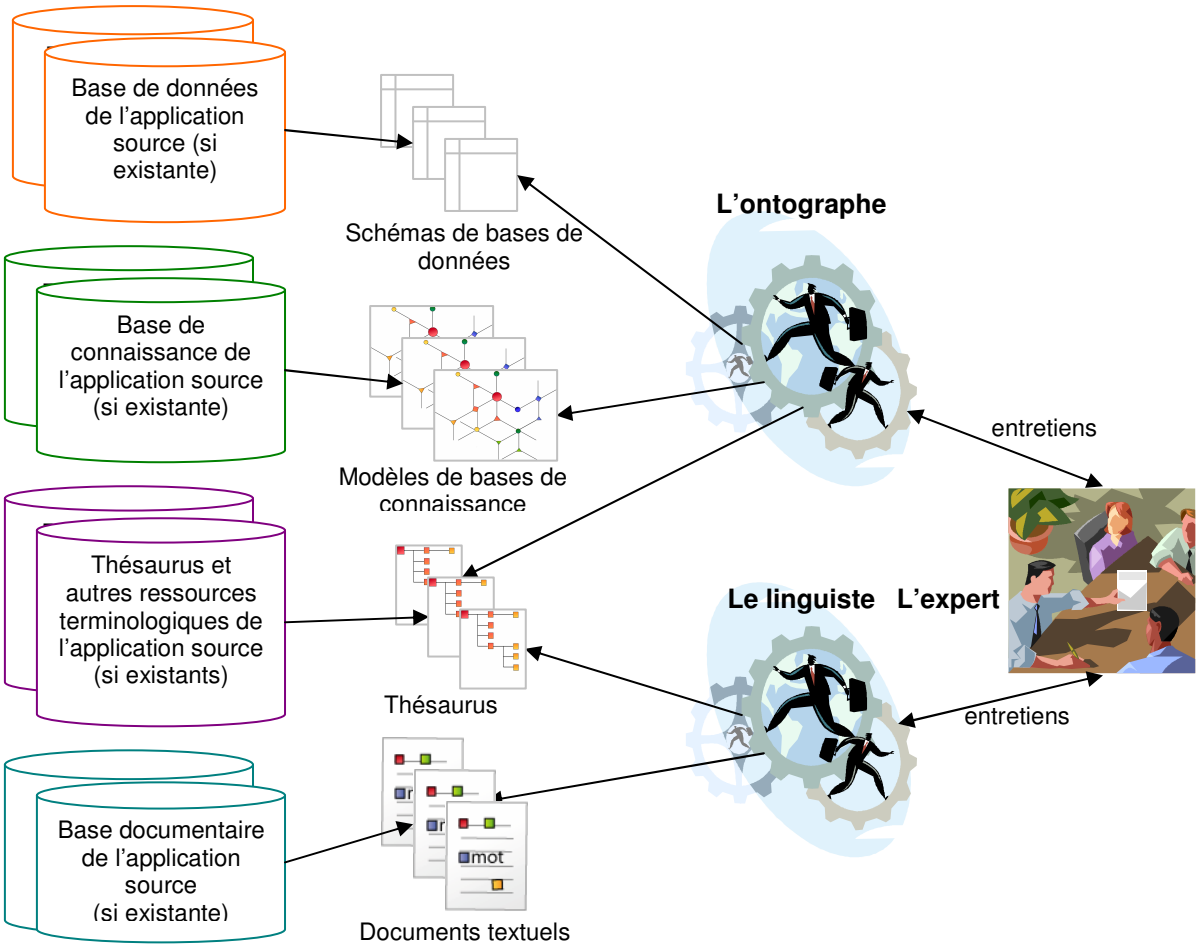


Figure 54. Sources de données pour le linguiste et l'ontographe dans la phase d'Etude

Concernant la future ontologie, l'ontographe procède par une approche métier, c'est-à-dire qu'il va partir de l'étude de l'existant (tant au niveau des bases de données, que des bases de connaissances ou encore des ressources terminologiques), des processus mis en place et des habitudes métier des utilisateurs. Il cible avec l'expert les spécificités métier, les référentiels à intégrer, les concepts clefs de l'application et plus généralement du domaine, ainsi que leurs attributs et leurs interactions. Il analyse également les schémas des bases de données et les modèles des bases de connaissances à reprendre, lorsqu'elles existent. Il peut aussi s'inspirer de sources de connaissances externes correspondant au domaine concerné par la future application, comme l'intégration d'un thésaurus géographique.

Cette étape est validée par la confrontation des différents intervenants afin que le linguiste et l'ontographe puissent divulguer les conclusions de leurs études et surtout, expliquer à l'expert ce qui est réalisable ou non. Chacun participe alors à l'écriture des spécifications générales qui permettent

d'une part de délimiter le périmètre de l'application finale et d'autre part de se mettre d'accord sur les objectifs à remplir concernant l'application finale. Ces spécifications générales valident aussi les différents cas suivants :

- l'outil linguistique extrait toutes les informations nécessaires au peuplement de l'ontologie et/ou à l'annotation sémantique
- l'outil de représentation des connaissances modélise le domaine de manière à ce que chaque information extraite par l'outil linguistique puisse être enregistrée dans la base de connaissance ou servir à créer de nouvelles annotations documentaires

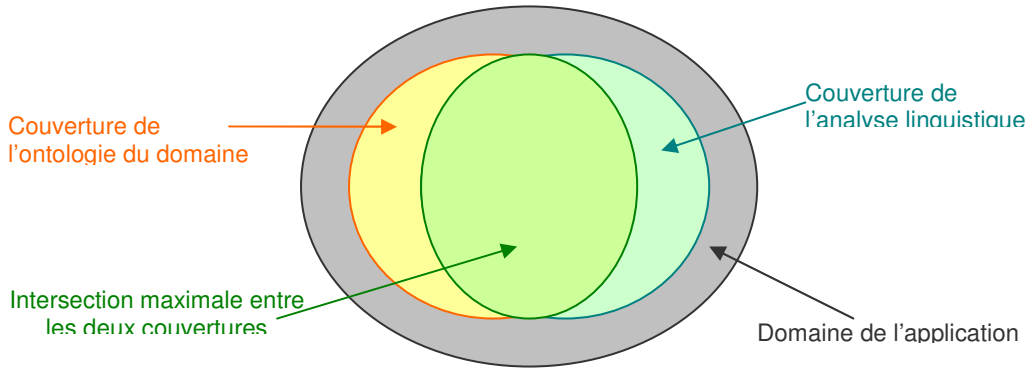


Figure 55. Intersections entre les couvertures linguistique et ontologique par rapport à un domaine

Il est important de préciser qu'il est rare que la couverture de l'outil de représentation des connaissances vis-à-vis du domaine soit exactement égale à la couverture de l'outil d'analyse linguistique pour le même domaine (cf. Figure 55). En effet, les noms des futurs concepts de l'ontologie ne correspondent pas forcément aux labels des futures étiquettes linguistiques. Certains éléments de l'ontologie sont destinés à d'autres usages métiers de l'application finale (recherche d'information, publication, autres fonctionnalités). Et la réutilisation de cartouches linguistiques génériques, comme celle des entités nommées, ou la nécessité de construire des patrons d'extraction intermédiaires, créent dans l'arbre conceptuel généré des étiquettes linguistiques dont les valeurs ne sont pas forcément exploitées ni exploitables pour le peuplement d'ontologie ou l'annotation sémantique. Malgré tout, il est fortement souhaitable pour l'application cible que l'intersection entre ces deux couvertures soit maximale, avec le plus de précision et de pertinence possible et cela est l'objectif de la phase suivante.

5.3 La Phase de Structuration

Cette phase a pour objectif d'une part de modéliser l'ontologie du domaine et d'autre part de construire la cartouche linguistique appliquée à ce domaine.

5.3.1 Modélisation de l'ontologie du domaine

Cette tâche consiste à structurer et organiser les données obtenues à l'étape précédente. Pour cela, l'ontographe doit acquérir la connaissance du domaine, que cette connaissance soit explicite dans les

bases de données ou les thésaurus disponibles ou implicite dans le savoir et le savoir-faire de l'expert du domaine. D'après John F. Sowa [SOW 00], l'acquisition de la connaissance s'effectue en trois étapes: « *Knowledge Acquisition is the process of eliciting, analysing and formalizing the patterns of thought underlying some subject of matter. In elicitation, the knowledge engineer must get the expert to articulate tacit knowledge in natural language. In formalisation, the knowledge engineer must encode the knowledge elicited from the expert in the rules and facts of some AI language. Between those two stages lies conceptual analysis: the task of analysing the concepts expressed in natural language and making their implicit relationships explicit* ».

En fait, il n'existe pas de méthodologie générique, standardisée et consensuelle ayant pour but d'acquérir cette connaissance du domaine afin de la modéliser sous la forme d'une ontologie. Plusieurs méthodologies ont plutôt vu le jour, notamment celles formulées par Uschold [USC 95], Gruninger [GRU 95], Fernandez & Gomez-Perez dans Methontology [FER 97] [BLA 98] et York Sure et al. dans On-To-Knowledge [SUR 03]. Le cycle de vie de ces méthodologies est fortement inspiré du génie logiciel, tout comme notre propre méthodologie OntoPop, et nous pouvons identifier des étapes communes telles que :

- la spécification / l'évaluation du besoin
- la conceptualisation, i.e. la capture des connaissances
- la formalisation ou « l'ontologisation », i.e. le codage de l'ontologie
- l'intégration d'ontologies existantes, par alignement ou par fusion entre ces ontologies
- l'opérationnalisation, i.e. l'implémentation de l'ontologie
- l'évaluation, la documentation et la maintenance de l'ontologie

Diverses méthodes, techniques et outils ont été proposés pour aider dans les différentes tâches du cycle de vie, et notamment au niveau de la conceptualisation, comme :

- l'extraction d'ontologies à partir de textes [BAC 96] [BOU 04] [AUS 00] ;
- la structuration des hiérarchies de concepts et de relations [GUA 92] [GUA 00] [BAC 01] [KAS 02] ;
- la fusion et l'adaptation d'ontologies existantes par l'utilisation des systèmes Onions [GAN 99] et Prompt [NOY 03] par exemple ;
- le développement collaboratif [DOM 98], [TOL 05].

L'étape de formalisation de l'ontologie dans un langage de modélisation peut être effectuée à l'aide des éditeurs d'ontologie tels que Protégé [NOY 00] [NOY 01], Swoop [KAL 05], WebODE [ARP 01], DOE [TRO 02] [BAC 02] ou encore OntoEdit [SUR 02],...

Nous n'avons pas la prétention de préconiser ici une méthodologie pour la construction d'ontologie. A l'ontographe d'opter pour la méthodologie la plus adaptée à ses données et à sa pratique. Par ailleurs, les spécificités du domaine et de l'application cible obligent souvent l'ontographe à faire certains choix de modélisation discutables. Par exemple, est-il préférable de modéliser un lieu de naissance comme un attribut ou comme une relation entre concepts ? La décision d'opter pour telle ou telle

représentation de la connaissance va impacter les modes d'utilisation de l'application. Par exemple, un moteur de recherche reposant sur ce modèle de l'ontologie n'offrira pas les mêmes fonctionnalités à l'utilisateur suivant que le champ d'interrogation du lieu de naissance a été représenté comme un attribut prenant une chaîne de caractère quelconque en entrée ou une relation entre classes conditionnant les valeurs possibles parmi les instances de celles-ci. Mais quelle que soit l'approche adoptée, le processus de construction d'une ontologie reste basé sur une collaboration étroite avec l'expert du domaine qui doit valider le modèle choisi.

5.3.2 Construction des cartouches linguistiques

Concernant la construction des cartouches linguistiques, nous avons déjà vu au chapitre 2 qu'elles s'appuient le plus souvent sur différents lexiques du domaine ainsi que sur un ensemble de patrons d'extraction composants l'automate à états finis du moteur d'extraction. Pour déterminer les patrons d'extraction, le linguiste étudie minutieusement chaque document du corpus représentatif du domaine, et par extension de l'application, afin d'identifier le vocabulaire spécifique utilisé par les auteurs de ce domaine, la structure des phrases, celle des documents, etc. [ENJ 05b]. Par exemple, le contenu d'une décision de jurisprudence dans le domaine juridique est bien différent de celui d'un article journalistique du domaine de la presse dite « people » :

- le vocabulaire de la décision est constitué de termes juridiques très précis alors que celui de la presse people est constitué de termes du langage commun, voire familier ;
- une décision de jurisprudence est souvent décrite par une seule phrase de plusieurs pages alors que l'article « people » se compose de phrases simples et courtes ;
- le document représentant la décision de jurisprudence comporte une structure bien particulière avec un en-tête composé de tous les éléments identificatoires de cette décision (cour de justice, date de la décision, la juridiction, la formation, etc.) suivi d'un corps de document narrant les argumentaires des différentes parties jusqu'au rendu de décision par la cour. Dans ce cas, le linguiste peut plus facilement cerner l'emplacement de telle ou telle information à extraire et la manière dont elle peut être extraite comparé à un texte non structuré comme l'article de la presse « people » qui n'est généralement constitué que d'un titre, parfois un chapô et du corps de l'article.

Au vu du corpus documentaire du domaine, l'une des premières tâches du linguiste consiste à élaborer les lexiques de ce domaine nécessaires à l'élaboration des patrons d'extraction. Dans certains cas, il aura à sa disposition des thésaurus, des listes d'entités nommées, ou toute autre terminologie, déjà exploités dans une version précédente de l'application et fournis par l'expert, ou bien qui font référence dans le domaine concerné. Une autre étape consiste à définir les différents traitements linguistiques nécessaires à l'analyse des documents du corpus : est-il nécessaire de découper le document en unités textuelles ? Ces unités textuelles sont-elles les paragraphes, les phrases, les mots ou autres ? A-t-on besoin d'un lemmatiseur ? L'analyse morpho-syntaxique est-elle absolument requise ? Et ainsi de suite... La décision d'inclure tel ou tel traitement dans la solution finale du moteur d'extraction est généralement dictée par la langue dans laquelle est rédigé le document, leur niveau de structuration, la nature des informations à extraire, etc.

a) « isManagerOf »

- **Objectif**

Relation entre une organisation (reconnue ou déduite) et une personne (reconnue ou déduite).

- **Lexique utilisé**

Le lien « isManagerOf » (concept au sein de la **Skill Cartridge™**) s'effectue au moyen d'un lexique existant (Module NER) contenant des termes du type « *manager, pdg, président directeur général*, etc. ».

- **Implémentation**

La personne identifiée par cette relation porte l'étiquette « Manager » (rôle au sein de la **Skill Cartridge™**). La relation elle-même porte l'étiquette « isManagerOf » (concept au sein de la **Skill Cartridge™**).

- **Exemples d'extraction**

Monsieur Thierry, pdg des Agf.

isManagerOf → *Monsieur Thierry, pdg des Agf.*
Manager → *Monsieur Thierry*
GUESSED Personnalite → *Monsieur Thierry*
Title → *Monsieur*
FamilyName → *Thierry*
Board → *pdg*
Organisation → *Agf*

Olivier Maschino PDG de la société Carlogo.

isManagerOf → *Olivier Maschino PDG de la société Carlogo.*
Manager → *Olivier Maschino*
GUESSED Personnalite → *Olivier Maschino*
FirstName → *Olivier*
FamilyName → *Maschino*
Board → *PDG*
GUESSED Organisation → *la société Carlogo*

Figure 56. Extrait d'un véritable document de spécifications détaillées des arbres conceptuels délivré par un linguiste de Temis pour le domaine de la veille économique, ici l'exemple est « isManagerOf »

Une fois les traitements définis, reste à écrire les patrons d'extraction. Pour cela, il est parfois possible de réutiliser certaines cartouches linguistiques comportant des patrons d'extraction génériques pour certains éléments particuliers à un langage. L'exemple le plus courant est celui des entités nommées comme les noms de personnes, d'organisations, de lieux, etc. Par exemple, dans le domaine de la presse « People », une cartouche existante permettant d'extraire le nom des personnes, et donc des personnalités en vogue, peut être réutilisée telle quelle. Par contre, dans le domaine juridique, cette même cartouche linguistique n'est pas suffisante pour décrire les différentes personnes intervenant dans une décision de jurisprudence : le nom des parties, celui des avocats, du juge, du président, etc. Cette cartouche doit être adaptée en lui ajoutant de nouveaux patrons d'extraction permettant de repérer et de distinguer clairement ces différents acteurs. Le linguiste étudie donc dans quelle mesure il est possible de réexploiter des cartouches existantes. A partir de là, il lui faut créer les patrons

d'extraction spécifiques au domaine de l'application cible qui viennent enrichir et compléter les patrons existants le cas échéant.

Tout au long de ce processus, le linguiste est en contact permanent avec l'expert pour lui présenter ses résultats, discuter des points à améliorer, etc. L'expert valide les informations extraites par le moteur d'extraction à partir d'un document des spécifications détaillées de la cartouche linguistique remis par le linguiste. Ce document détaille la structure de chaque arbre conceptuel généré à partir d'un patron d'extraction du domaine. La Figure 56 montre un exemple du contenu d'un tel document pour un sous-arbre en particulier, i.e. « isManagerOf », issu d'une cartouche sur le domaine de la veille économique.

Ce document doit être régulièrement mis à jour en fonction des modifications opérées sur la structure des patrons. A chaque nouvelle version de la cartouche linguistique, il est de nouveau livré à l'expert pour validation. Il est aussi fourni à l'intégrateur qui l'utilise comme base de travail pour définir les Règles d'Acquisition de Connaissance dans la prochaine étape.

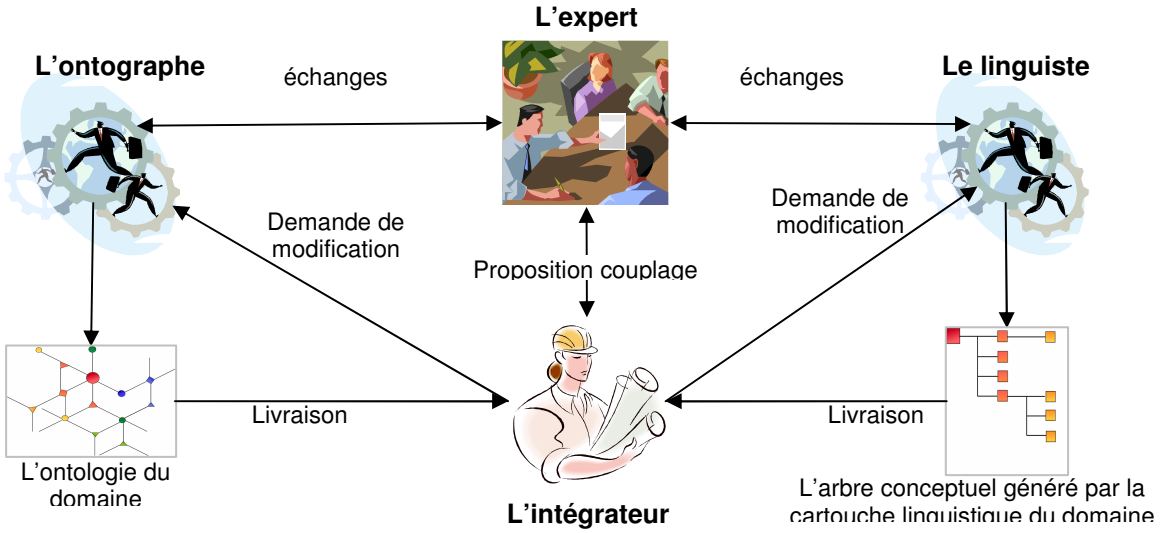


Figure 57. Echanges entre les intervenants durant la phase de Structuration

Il est important de préciser qu'à cette étape de la méthodologie, l'intégrateur est responsable de la compatibilité des deux modèles vis-à-vis du domaine, et plus particulièrement des objectifs décrits dans les spécifications générales de l'application cible. Dès les premières versions, il initie le couplage entre les concepts de l'ontologie d'un côté et les étiquettes sémantiques de l'arbre conceptuel généré par la cartouche linguistique de l'autre. Cette mise en correspondance lui permet de détecter en amont les failles et de demander les modifications nécessaires au linguiste, à l'ontographe, ou au deux si nécessaire. L'intégrateur joue donc un rôle de médiateur entre les différents intervenants, cf. Figure 57, et notamment auprès de l'expert. Même si l'expert possède une certaine connaissance du domaine et de l'application cible, il ne maîtrise pas assez le fonctionnement des outils d'extraction d'information ou de représentation des connaissances, leurs formats, leurs contraintes, etc. Seul l'intégrateur possède ce double point de vue et par conséquent les capacités de réaction associées. Afin de rendre ce double point de vue accessible aux autres intervenants et tout particulièrement à

l'expert, il rédige un document nommé la « Table de correspondance cartouche/ontologie » qui regroupe :

a) la modélisation telle quelle est implémentée dans l'ontologie du domaine, i.e. pour chaque classe (concept ou relation, i.e. le domaine), son nom, ses attributs et ses rôles le cas échéant ;
b) les contraintes des rôles, i.e. le « range » ou portée de ce rôle dans l'ontologie, et des attributs, i.e. les types de données comme chaîne de caractère, date, URL, etc. ;
c) les éléments parmi les classes, rôles et attributs dont les valeurs seront automatiquement renseignées par le moteur d'extraction linguistique ;
d) le nom de la ou des étiquette(s) sémantique(s) dont la valeur textuelle sera utilisée pour créer les valeurs des éléments concernés, i.e. les nœuds déclencheurs ;
e) la condition contextuelle implémentée dans la règle d'acquisition ;
f) les commentaires éventuels provenant des différents échanges entre les intervenants concernés pour garder une trace des problèmes soulevés et des solutions choisies.

Ce document tient une place importante dans la méthodologie OntoPop puisqu'il sert non seulement à la compréhension du fonctionnement du futur couplage, mais également à toute la gestion et maintenance de ce dernier. Il doit obligatoirement être mis à jour à chaque modification de l'ontologie ou de la structure des arbres conceptuels du domaine.

5.4 La Phase de Couplage

Cette phase consiste à définir et implémenter les Règles d'Acquisition de Connaissance pour l'application cible sur la base du modèle de l'ontologie du domaine et de la structure des arbres conceptuels générés par la cartouche linguistique du domaine. Comme nous venons de voir, chaque élément de l'ontologie utilisé pour l'annotation documentaire ou pour l'enrichissement de la base de connaissance a été répertorié dans la Table de correspondance cartouche/ontologie. En effet, il ne s'agit pas d'instancier ou d'annoter avec tous les éléments définis dans l'ontologie. Certains d'entre eux sont utilisés à d'autres fins par l'application cible, comme proposer des fonctionnalités accessibles via l'interface utilisateur de l'application. Outre la Table de correspondance, l'intégrateur dispose des spécifications détaillées des arbres conceptuels fournies par le linguiste ainsi que la documentation de l'ontologie réalisée par l'ontographe.

Avant d'initier la comparaison des éléments répertoriés de l'ontologie avec les étiquettes linguistiques des arbres, il est nécessaire de constituer un corpus de tests unitaires. Ce corpus, composé d'une sélection de ressources documentaires balayant l'ensemble des arbres conceptuels générés par la cartouche, permet de tester tous les cas de couplage possibles. L'intégrateur peut alors analyser en détail chacun des arbres conceptuels générés à partir des tests unitaires. Dans un premier temps, il identifie les étiquettes linguistiques dont les valeurs textuelles correspondent à la valeur d'une instance ou d'une annotation. Ce seront les nœuds déclencheurs d'une nouvelle Règle d'Acquisition de Connaissance. Dans les cas où il y a ambiguïté entre plusieurs étiquettes linguistiques, il analyse le contexte de ces étiquettes qui déterminera les indices contextuels de la règle. Les spécifications

détaillées fournies par le linguiste permettent à l'intégrateur d'obtenir rapidement une vue d'ensemble des différents tests unitaires et des étiquettes pouvant jouer un rôle dans la résolution des ambiguïtés.

Dans certains cas, les spécifications détaillées et les tests unitaires ne sont pas suffisants pour formuler les différentes règles d'acquisition. L'intégrateur doit également prendre en compte des règles de gestion métier spécifiées par l'expert. Par exemple, toujours dans le domaine de l'édition juridique, lors de l'identification d'une décision de jurisprudence, la date de la décision peut apparaître de plusieurs manières dans l'arbre de concept : « /DateDecision », « /DateLecture » ou encore « /DateSeance ». Or ces trois étiquettes n'ont pas la même importance. Si plusieurs de ces étiquettes sont représentées dans l'arbre de concept d'une décision de jurisprudence, une seule de ces étiquettes possède la bonne valeur de la date de la décision et seul l'expert peut apporter cette connaissance métier : l'étiquette « /DateDecision » prime sur les deux autres, ensuite l'étiquette « /DateLecture » prime sur l'étiquette « /DateSeance », la valeur de cette dernière n'est donc exploitée que si aucune autre des étiquettes représentant une date n'est présente dans l'arbre conceptuel d'une décision de jurisprudence. Trois règles d'acquisition traduisent cette règle métier, dans lesquelles la primauté d'une étiquette sur une autre est déterminée par le niveau de confiance accordée à chacune des règles. D'autres règles de gestion métier peuvent porter sur la normalisation de la valeur générée en sortie. Celle-ci peut être une concaténation de différentes valeurs d'étiquettes ou une combinaison de nom d'étiquettes et de leurs valeurs ou encore une constante, etc.

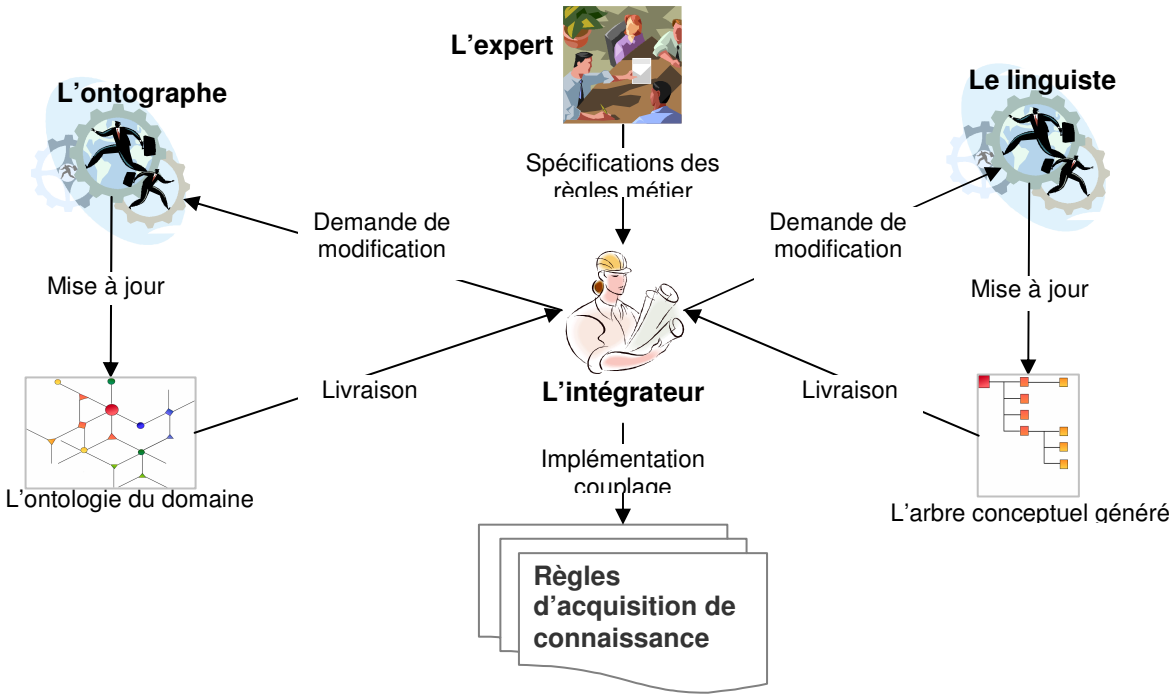


Figure 58. Echanges entre les intervenants durant la phase de Couplage

Une fois que l'intégrateur a spécifié dans sa Table de correspondance l'ensemble des nouvelles règles d'acquisition, il peut les implémenter via l'Editeur de Règle, cf. section 6.1. Pour chacune d'entre elles, il entre les informations requises à sa création (cf. section 3.3.1). Les règles peuvent

alors être testées une par une ou toutes à la fois. La première option permet de mieux contrôler que les résultats produits par chacune des règles correspondent bien aux attendus. La seconde option permet de détecter des conflits entre plusieurs règles ainsi que les incohérences. Si les règles ne produisent pas les résultats attendus, divers facteurs peuvent en être la cause tels qu'une mauvaise configuration de la règle dans l'Editeur, des incohérences dans les arbres conceptuels, des problèmes de formats avec le modèle de l'ontologie, des oublis, des conflits, etc. L'intégrateur doit être capable d'analyser l'origine du problème pour savoir s'il peut y remédier en corrigeant la règle déviante ou bien si le problème nécessite une modification de l'ontologie ou de la cartouche linguistique (cf. Figure 58). Auquel cas, il demande à l'ontographe ou au linguiste d'effectuer les modifications nécessaires dans leurs modèles. Ces modèles ne sont donc pas fixés une fois pour toute à ce stade de la méthodologie, mais plutôt constamment retravaillés jusqu'à la validation définitive effectuée à l'étape suivante.

A chaque nouvelle règle d'acquisition testée et approuvée sur le corpus de tests unitaires, l'intégrateur met à jour la Table de correspondance cartouche/ontologie. A l'issue de cette phase, cette Table de correspondance est de nouveau livrée à l'ensemble des partenaires, afin qu'ils puissent prendre connaissance des Règles d'Acquisition de Connaissance ayant été définies par l'intégrateur. La prochaine étape concerne la validation des résultats produits par l'ensemble de ces règles d'acquisition.

5.5 La Phase de Validation

Cette phase consiste à mesurer la qualité de l'ensemble de l'application, et plus particulièrement du couplage, par rapport aux objectifs fixés initialement. Pour cela, la cartouche linguistique du domaine puis le couplage de cette cartouche avec l'ontologie du domaine doivent être testés afin de contrôler la capacité d'intégration des modèles produits et la pertinence des solutions proposées. A cet effet, un corpus de validation est constitué à partir des documents non utilisés dans le corpus de tests unitaires de l'étape précédente. Ce corpus de validation est utilisé pour les deux processus de validation. Le premier mesure la performance de l'outil linguistique, la qualité de ses extractions, sa couverture vis-à-vis du domaine concerné, etc. Pour ce faire, le linguiste calcule la précision, c'est-à-dire le silence, et le rappel, c'est-à-dire le bruit, généré par les patrons d'extraction sur les documents du corpus. Il compare ces résultats aux tests unitaires réalisés précédemment et contrôle que les extractions produites sont conformes à celles attendues. Les taux de précision et de rappel doivent atteindre un seuil suffisamment important pour être acceptables dans l'application. Ce seuil est défini conjointement par le linguiste et l'expert au vu de la complexité du domaine étudié et des besoins de l'application cible.

Le second processus de validation analyse la qualité des propositions fournies par les Règles d'Acquisition de Connaissance, tant au niveau du peuplement de l'ontologie que de celui de l'annotation sémantique. Rappelons que le couplage exploite les résultats fournis par la cartouche linguistique. Si celle-ci produit trop de bruit, trop d'incohérence, trop de silence, alors les propositions fournies par le processus de peuplement d'ontologie et d'annotation sémantique seront trop pauvres ;

même si les Règles d'Acquisition de Connaissance permettent de filtrer certaines erreurs induites par le moteur d'extraction. Le processus de validation des performances du couplage considère donc que le moteur d'extraction produit des résultats optimums compte-tenu du domaine, même si cela n'est pas toujours le cas. Il prend donc pour acquis les arbres conceptuels générés par la cartouche linguistique et mesure la précision et le rappel de chacune des Règles d'Acquisition de Connaissance sur ces résultats. Il va également calculer le taux de peuplement d'annotation et celui d'annotation sémantique en comparant le nombre d'étiquettes linguistiques générées avec le nombre d'instances et d'annotations créées à partir du corpus de validation. Nous n'allons pas détailler ici la méthode d'évaluation du couplage, cf. chapitre 7. Ce processus de validation utilise le Module d'Annotation et d'Acquisition et accessoirement l'interface de validation décrite à la section 6.2.5, qui lui permet de visionner de manière conviviale les résultats produits tant au niveau du peuplement d'ontologie que de l'annotation sémantique pour chacun des documents traités.

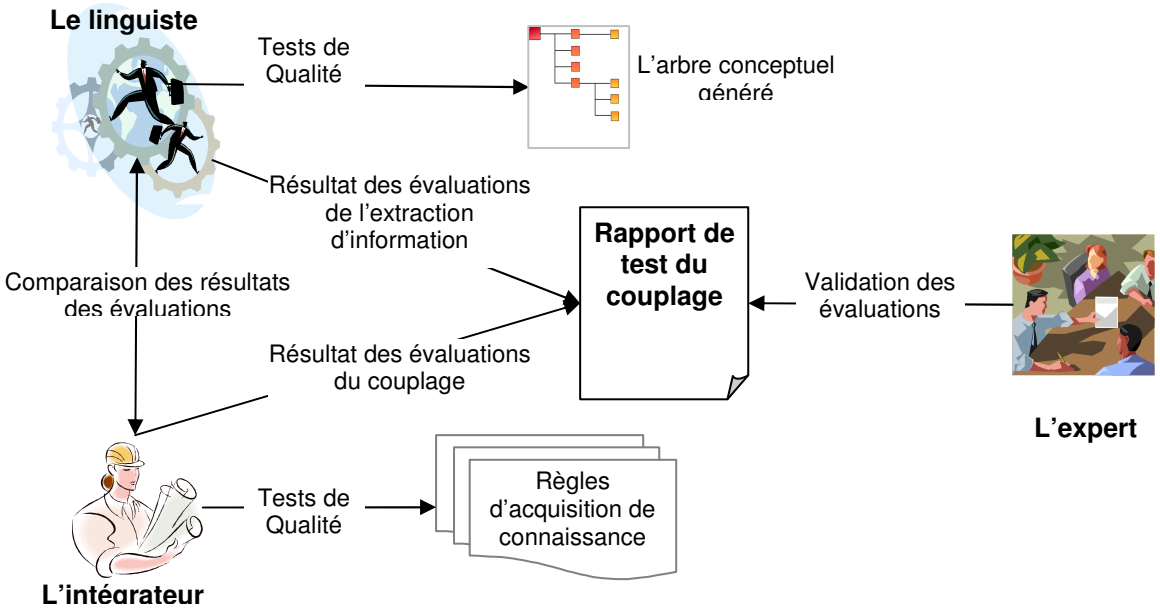


Figure 59. Echanges entre les intervenants durant la phase de Validation

La comparaison des résultats des deux processus de validation a lieu à plusieurs reprises, lors de réunions de travail entre le linguiste et l'intégrateur. Si les résultats sont jugés insuffisants vis-à-vis des objectifs fixés dans les spécifications détaillées de l'application, l'origine de la défaillance doit être réparée : les règles d'acquisition, la cartouche linguistique ou l'ontologie du domaine. Dans le premier cas, l'intégrateur modifie les règles afin de corriger les erreurs de conflits, d'oublis, de syntaxe, etc. et recommence son évaluation à partir des cas posant problème. Dans le second cas, il travaille avec le linguiste pour ajuster les patrons d'extractions ambigus. Dans le dernier cas, il demande à l'ontographe de procéder à des modifications dans la modélisation de l'ontologie du domaine. Si les modifications demandées, tant au linguiste qu'à l'ontographe, sont mineures, de nouvelles versions de la cartouche ou de l'ontologie sont livrées à l'intégrateur pour réitérer cette phase de Validation. Si ces modifications sont majeures alors l'intégrateur doit reprendre à partir de la phase de Couplage, voire même à partir de la phase de Structuration avec validation des modifications par l'expert. Par contre,

si les résultats sont jugés satisfaisants, le rapport de test du couplage et du taux de couverture entre les outils est remis à l'expert au cours d'une réunion où une démonstration de l'application est effectuée. Si l'expert valide ces résultats, alors l'application est livrée, installée et mise en service à l'étape suivante.

5.6 La Phase de Mise en Service

Cette phase consiste à livrer, installer et assurer le suivi des différents composants de l'application cible. Ces composants comprennent les outils qui composent la solution, ainsi que l'ontologie du domaine, sa base de connaissance, les thésaurus et autres ressources terminologies associées, la cartouche linguistique du domaine et enfin le couplage, i.e. l'ensemble des Règles d'Acquisition de Connaissance. L'ontologie, la base de connaissance et les thésaurus sont importés dans l'outil de représentation des connaissances. La cartouche linguistique est installée sur le serveur de l'outil linguistique. Et enfin, le couplage est installé sur le serveur de l'outil de représentation des connaissances.

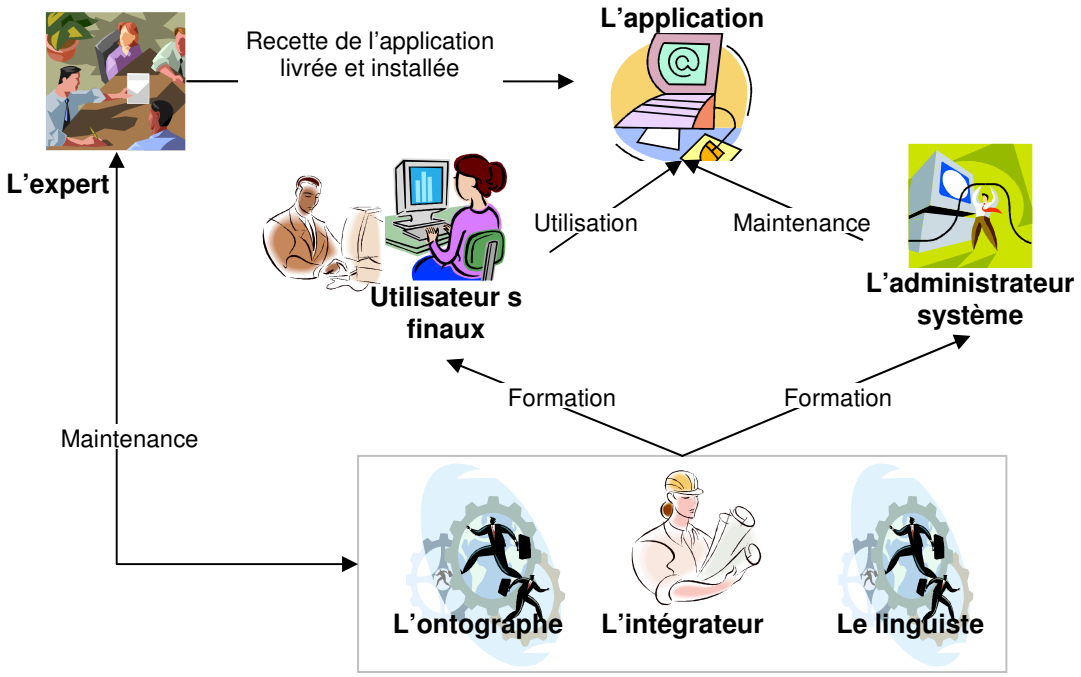


Figure 60. Echanges entre les intervenants durant la phase de Mise en Service

Une fois que toute l'installation est prête, l'expert procède à la recette de l'application en vérifiant chacune des fonctionnalités demandées dans les spécifications détaillées de l'application ainsi que la cohérence et la qualité des résultats produits, comparativement à ceux présentés à l'étape précédente. Si l'installation de l'application montre des défauts, les corrections appropriées sont apportées par les divers intervenants dans les plus brefs délais. Dans le cas contraire, l'application est mise en service auprès des utilisateurs finaux après qu'ils aient suivi une à plusieurs formations selon le degré de changement apporté par la nouvelle application. Ces formations concernent notamment la

navigation, la recherche et la publication dans le référentiel de l'application ainsi qu'à la validation des résultats fournis par l'application dans le cas d'un processus semi-automatisé. Outre les utilisateurs finaux, une formation est également dispensée au futur administrateur système de la nouvelle application. Ce dernier doit apprendre à installer, gérer et maintenir chaque outil et chaque composant formant l'ensemble de l'application.

Enfin, le linguiste, l'ontographe et l'intégrateur peuvent proposer à l'expert un ensemble d'améliorations possibles de l'application sur la base de leurs analyses et de leurs travaux développés tout au long de la méthodologie OntoPop. Par exemple, le linguiste aura peut-être identifié de nouveaux concepts à exploiter grâce à sa lecture attentive des documents du corpus représentatif du domaine. A la vue de ces nouvelles étiquettes sémantiques, l'intégrateur peut alors demander à l'ontographe la manière de modéliser ces nouveaux concepts dans l'ontologie du domaine. Cette nouvelle modélisation est alors proposée à l'expert pour savoir si le peuplement de ces nouveaux concepts ou l'annotation des documents par ces nouveaux concepts peuvent être considérés comme pertinents pour les utilisateurs finaux. Si ces ajouts sont approuvés par l'expert alors ils feront soit l'objet d'une maintenance dans le cas de modifications mineures, soit l'objet d'une nouvelle version de l'application dans le cas contraire.

5.7 Conclusion

La méthodologie OntoPop se compose de cinq phases : Etude, Structuration, Couplage, Validation et Mise en service. Pour chacune de ces phases, nous avons identifié les objectifs, les actions requises par chacun des intervenants, les moyens pour y arriver et les livrables devant être produits et échangés. Nous avons résumé chacune de ces phases dans la Figure 61, qui permet d'avoir une vue globale de la méthodologie OntoPop mise en place.

Depuis le début de ma thèse en novembre 2003, j'ai participé à neuf projets auprès de différents clients dans des domaines aussi divers que l'édition juridique [AMA 05a], la presse People [AMA 05b], la veille économique [AMA 04] ou scientifique, etc. J'intervenais en tant qu'intégrateur et à ce titre, j'ai donc réalisé tous les couplages nécessaires au développement soit de prototypes de démonstration de la solution, soit de la future application mise en service. J'ai conçu cette méthodologie OntoPop grâce à l'expérience acquise au cours de ces divers projets. Je me suis inspirée des méthodes de gestion de projet, mais aussi et surtout des réussites et des dysfonctionnements apparus tout au long du développement des solutions mises en place pour les clients. Les dysfonctionnements étaient le plus souvent dus à un manque évident de communication entre les différents intervenants de chaque projet et donc aisément résolus par la mise en place de réunions de travail régulières. Ces réunions permettaient de communiquer l'état des avancées de chacun et de pallier efficacement aux problèmes soulevés. Le fait d'établir également une liste de livrables « officiels » devant être communiqués par chacun des intervenants aux autres membres d'un projet a aussi permis d'améliorer le cycle de vie de chacun des projets.

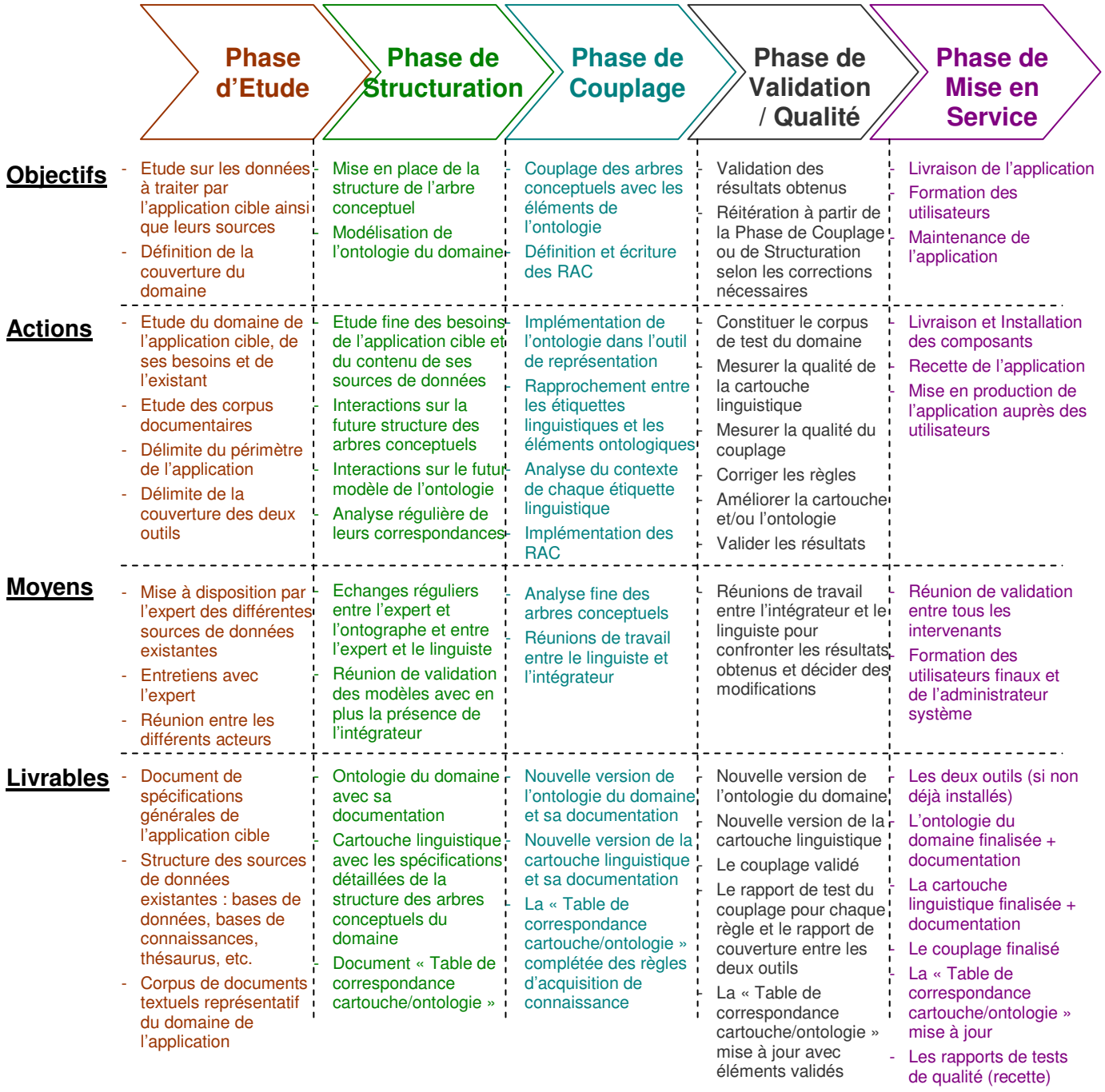


Figure 61. Vue d'ensemble de la méthodologie OntoPop

Chapitre 6. La Plateforme logicielle d'OntoPop

La méthodologie décrite au chapitre précédent permet la mise en œuvre d'une application de peuplement dontologie et d'annotation sémantique dans le cadre d'un projet particulier. Mais OntoPop propose aussi des solutions logicielles adaptables aux besoins et objectifs de chaque application. Par contre, si la méthodologie est indépendante des logiciels utilisés, on ne peut pas en dire autant de la plateforme OntoPop qui repose avant tout sur l'outil de représentation des connaissances ITM de la société Mondeca. Néanmoins, j'ai développé cette plateforme logicielle en gardant à l'esprit les exigences suivantes :

- **L'indépendance entre la structure de l'ontologie et la structure des extractions linguistiques**. Annoter une ressource documentaire et peupler une ontologie ne doivent pas induire de nouvelles contraintes sur la façon dont les ressources terminologies ou ontologiques sont modélisées ou sur le format produit par les outils d'extraction d'information.
- **La complétude.** Le système doit être capable de retrouver toute information donnée par les outils de TAL.
- **La standardisation.** Le système ne doit pas être dépendant de l'outil d'extraction utilisé.
- **La cohérence.** Les instances créées dans la base de connaissance et les annotations sémantiques produites doivent rester cohérentes avec l'ensemble du modèle de l'ontologie.
- **La facilité d'utilisation**. Le processus d'interfaçage requiert des experts de différentes branches (du domaine, de la linguistique, de la représentation des connaissances). Ainsi la solution choisie doit être facilement comprise par ces trois parties et permettre un processus itératif.
- **La capacité à évoluer.** Le système doit être capable de prendre en compte les évolutions à la fois des outils d'extraction et des ressources terminologies et ontologies.

Dans ce chapitre, nous allons décrire l'implémentation technique de cette plateforme OntoPop. Elle se compose des trois composants suivants : l'Editeur de Règles, le Module d'Annotation et d'Acquisition et enfin le Module de Maintenance des Lexiques. Chaque module est présenté en fonction de trois caractéristiques logicielles, à savoir son architecture, son processus détaillé et son implémentation technique.

6.1 L'Editeur des Règles d'Acquisition de Connaissance

Cet éditeur a pour objectif de permettre à l'intégrateur de saisir l'ensemble des Règles d'Acquisition de Connaissance d'une application donnée à partir d'une simple interface utilisateur. Les données

rentrées dans l'interface utilisateur formeront une nouvelle Règle d'Acquisition de Connaissance qui sera ajoutée à l'ensemble des règles déjà stockées sur le serveur de l'application.

6.1.1 L'architecture

Cet éditeur est la première étape nécessaire au fonctionnement de la plate-forme OntoPop. Rappelons que l'intégrateur doit manuellement configurer l'ensemble des règles d'acquisition préalablement à toute utilisation de la plateforme. L'éditeur fournit donc une interface utilisateur très simple, constituée à partir des différentes informations composant une règle d'acquisition (cf. chapitre 3). Comme montré dans la Figure 62, ce module s'interface avec les composants suivants :

- l'utilitaire Bagui qui constitue l'interface utilisateur ;
- le serveur d'ITM sur lequel est ajouté l'ensemble des règles d'acquisition au fur et à mesure de leur configuration dans un fichier XML.

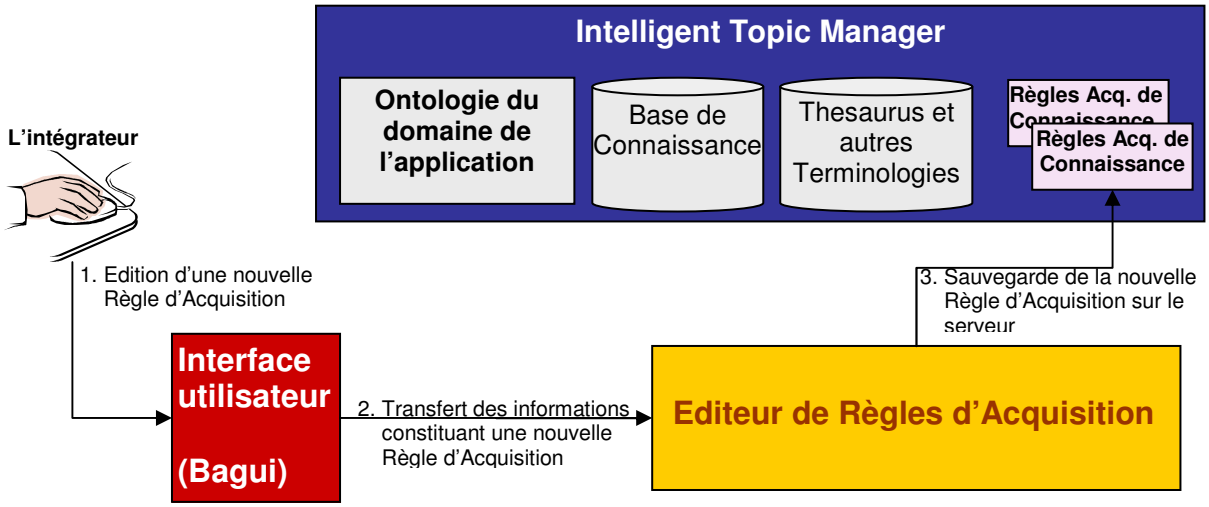


Figure 62. Architecture de l'Editeur de Règles d'Acquisition de Connaissance

6.1.2 Le processus détaillé

L'éditeur se présente sous la forme d'une interface utilisateur, cf. Figure 63, qui présente un formulaire où chaque champ correspond à une propriété spécifique d'une règle d'acquisition. Une fois les valeurs des champs du formulaire récupérés dans des objets Java, le programme procède à la sérialisation de la nouvelle règle en langage XML. Au cours de la sérialisation, le programme de l'éditeur compile chaque règle en fonction de l'algorithme de transcription en chemins XPath décrit à la section 3.3.2. Cette opération de compilation des règles est particulièrement importante pour la pertinence des résultats de la future application. Une fois la transcription accomplie, la nouvelle règle vient enrichir la liste des règles existantes sur le serveur ITM.

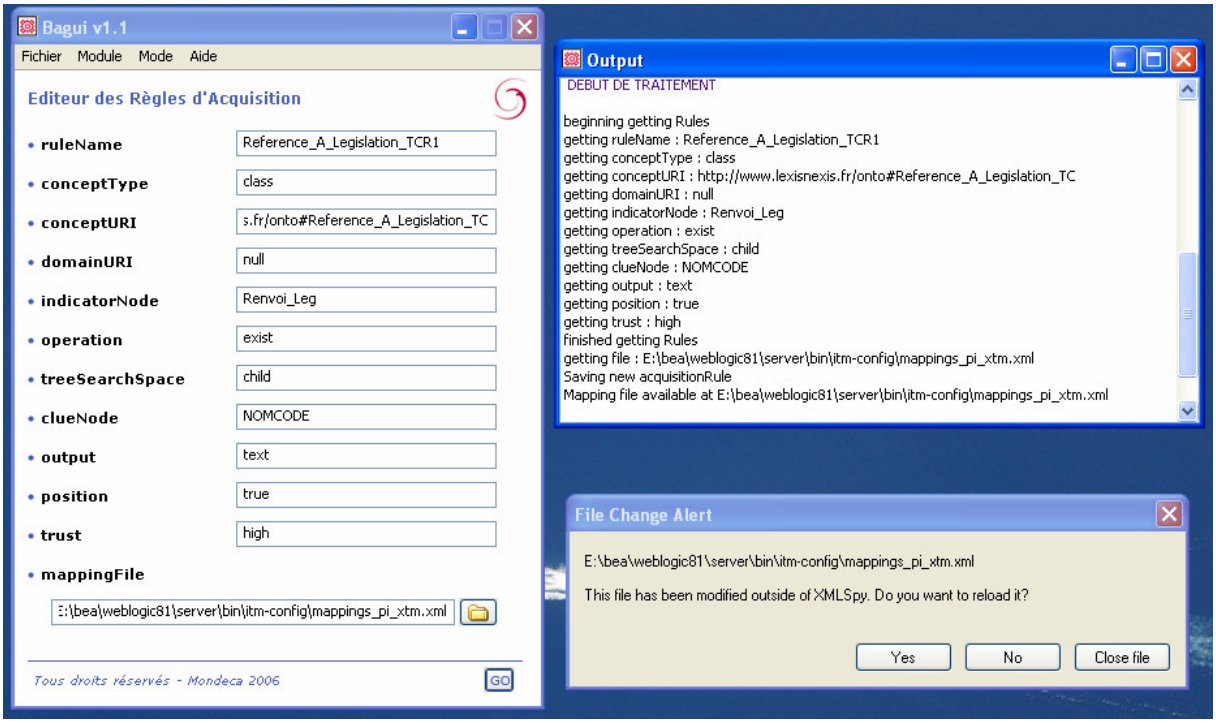


Figure 63. Interface de saisie des Règles d'Acquisition de Connaissance dans Bagui

6.1.3 L'implémentation technique

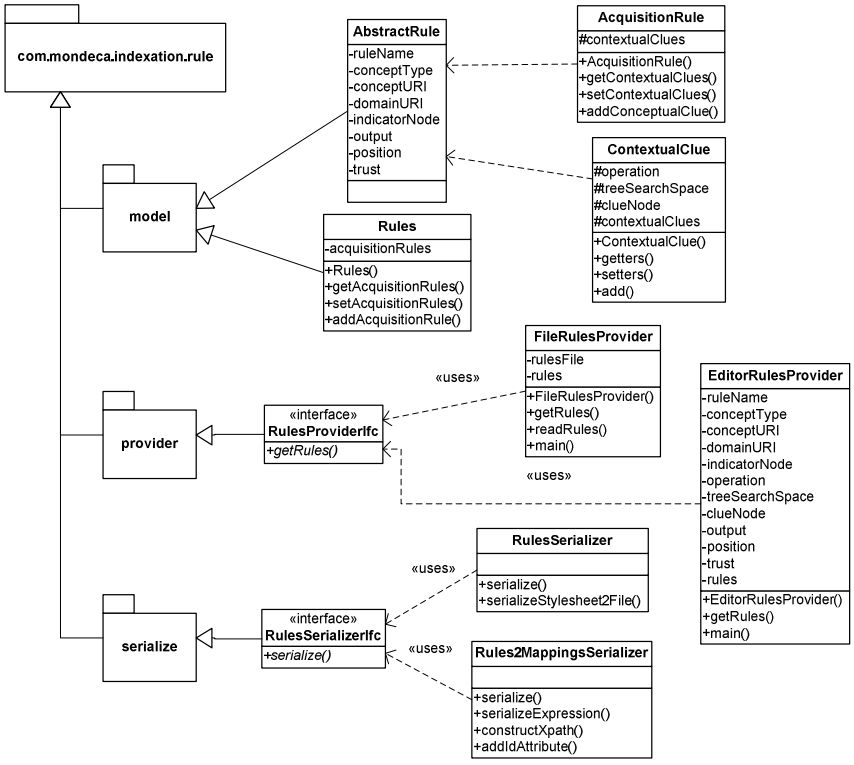


Figure 64. Architecture de l'Editeur des Règles d'Acquisition

L'utilitaire Bagui a été conçu et implémenté par un membre de l'équipe de Développement de Mondeca pour générer et lancer des programmes batch à partir d'une interface utilisateur simplifiée. J'ai donc adapté cet outil pour créer l'interface de saisie de l'Editeur des Règles. Dans cette interface, l'utilisateur doit renseigner les champs à afficher dans le formulaire de saisie d'une nouvelle règle, les valeurs par défaut de certains de ces champs ainsi que l'application Java à lancer. En ce qui concerne cette application Java, elle est implémentée dans le package `com.mondeca.indexation.rule` de la plate-forme OntoPop (cf. Figure 64).

6.2 Le Module d'Annotation et d'Acquisition d'ITM

Le Module d'Annotation et d'Acquisition défini pour ITM applique la démarche proposée par OntoPop au chapitre 4 en proposant des solutions logicielles concrètes. A terme, ce module a pour but de traiter différents types de contenus, le texte mais aussi la vidéo ou l'image, grâce au couplage de moteurs d'extraction adaptés. Pour l'instant, seules les ressources textuelles peuvent être traitées grâce à l'intégration des outils IDE et GATE (cf. 2.2).

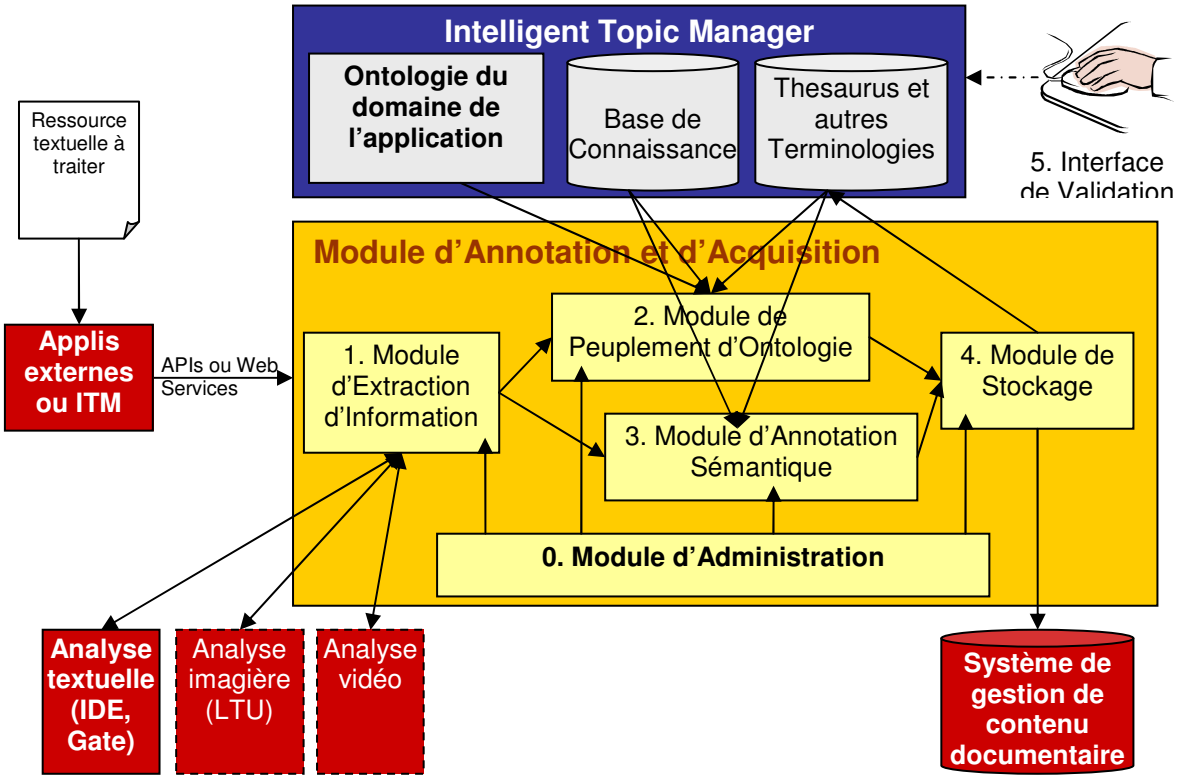


Figure 65. Architecture du Module d'Annotation et d'Acquisition

L'appel à ce module est réalisé depuis les interfaces internes à ITM ou bien depuis des applications externes, par une connexion directe aux APIs ou Web Services. Un document, ou un corpus de documents, est soumis au module qui se charge selon la configuration de l'application cliente, de se connecter au moteur d'extraction d'information spécifié, d'annoter chacun des documents et/ou d'enrichir la base de connaissance. Les phases d'annotation et d'enrichissement sont contrôlées par

rapport au modèle défini par l'ontologie de l'application et aux ressources terminologiques et ontologiques métier. En sortie, les nouvelles instances sont directement importées dans la base de connaissance ITM et les annotations enregistrées dans des fichiers ou envoyées à un gestionnaire de contenu documentaire externe pour y être stockées conjointement avec le document source. Si l'application cliente a mis en place un processus semi-automatisé, l'utilisateur humain a la charge de valider les propositions tant au niveau des annotations que des instances. Pour cela, des interfaces utilisateurs dédiées sont aussi comprises dans la solution. C'est pourquoi, comme illustré dans la Figure 65, ce module est composé des cinq composants suivants : 1) le Module d'Extraction d'Information ; 2) le Module de Peuplement d'Ontologie ; 3) le Module d'Annotation Sémantique ; 4) le Module de Stockage et 5) l'Interface de Validation.

Il est important de préciser que ces composants sont tous pilotés par le Module d'Administration. Ce module est responsable de la coordination des actions et des échanges entre les différents composants. Par ailleurs, chacun de ces composants dispose aussi de son propre module d'administration qui gère le déroulement du processus interne au composant. Enfin, les composants sont entièrement paramétrables via des fichiers de configuration indépendants, cf. Figure 66. Cette implémentation permet une modularité et une flexibilité du système particulièrement efficace pour répondre au plus près des besoins d'une application cliente. Tous les composants ont été réalisés en Java et respectent l'architecture préconisée pour le développement des APIs dans ITM. Cela permet une intégration optimum d'OntoPop tant avec les applications externes qu'avec les fonctionnalités propres à ITM. A présent, nous allons brièvement décrire les caractéristiques logicielles de chacun de ces composants.

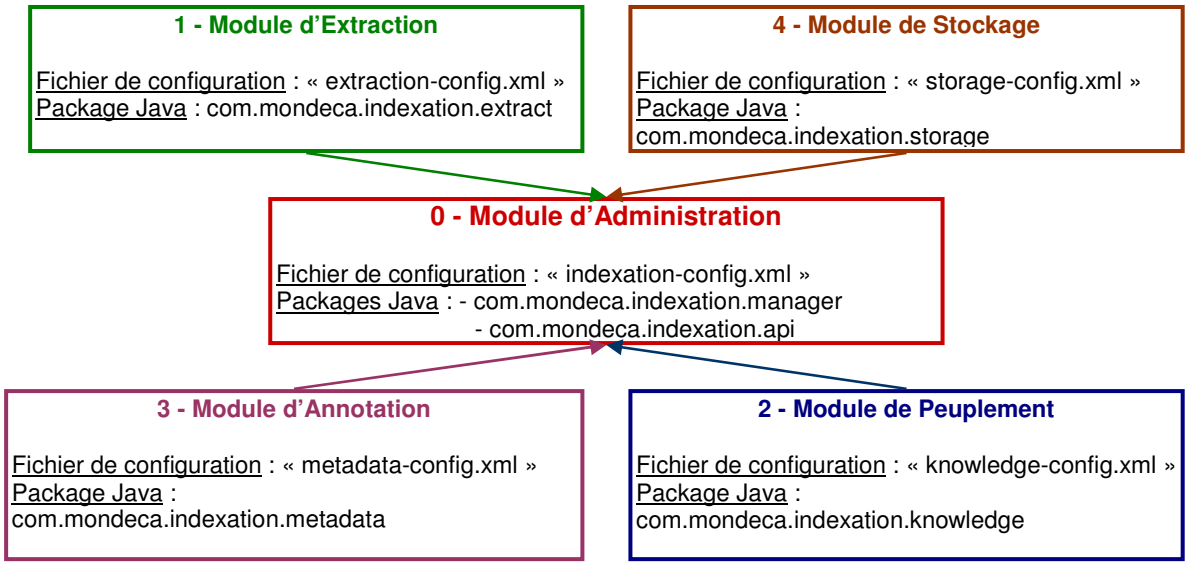


Figure 66. Architecture technique du module et de ses composants

6.2.1 Le Module d'Extraction d'Information

Ce module a pour objectif générer, à partir des arbres conceptuels produits par un moteur d'extraction, les deux représentations suivantes : le réseau sémantique de connaissance (au format XTM) et les annotations sémantiques (format RDF).

6.2.1.1 L'architecture

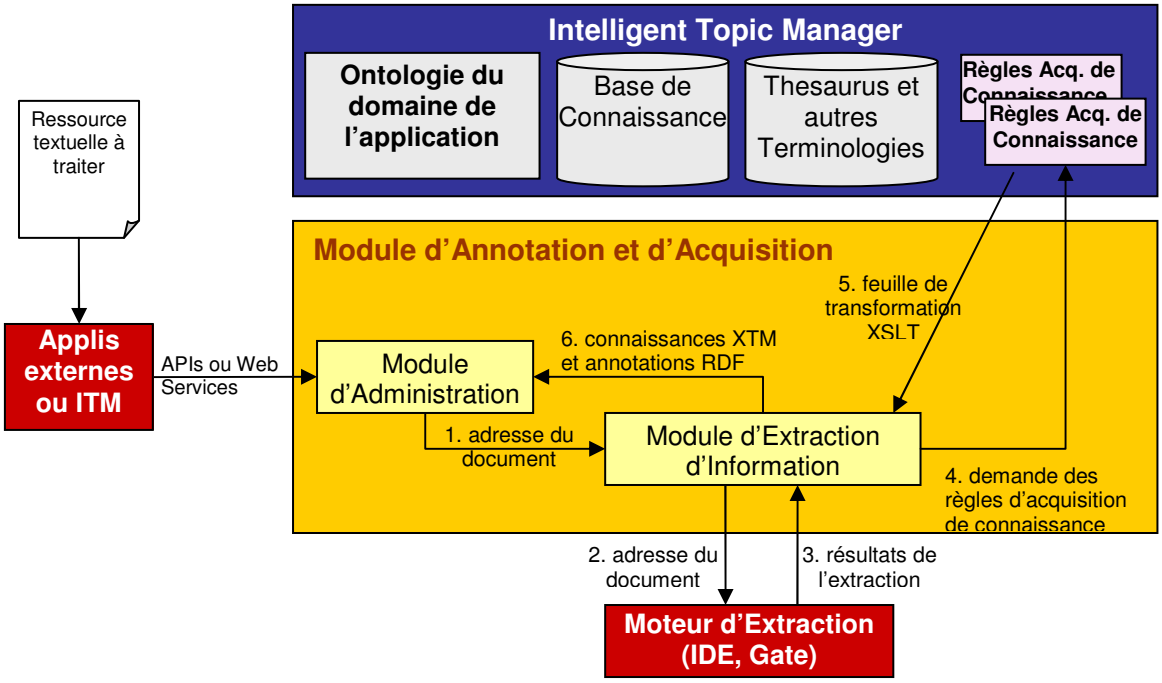


Figure 67. Architecture du Module d'Extraction d'Information

Comme le montre la Figure 67, ce module s'interface avec :

- le module d'administration du Module d'Annotation et d'Acquisition qui lui fournit la référence du document à analyser et auquel il va renvoyer le réseau sémantique et les annotations générées ;
- le moteur d'extraction chargé d'analyser le document soumis ;
- le serveur ITM qui contient les paramètres de configuration (le moteur d'extraction, la cartouche linguistique, le résultat attendu à savoir le peuplement ou l'annotation ou encore les deux, etc.) ainsi que les RAC compilées.

6.2.1.2 Le processus détaillé

Le Module d'Extraction d'Information se connecte au moteur d'extraction configuré en lui fournissant l'adresse du document à traiter. Ce moteur d'extraction analyse le contenu du document en fonction de la cartouche linguistique implémentée pour le domaine concerné et retourne l'arbre conceptuel produit. Dans le cas où de multiples extractions ont lieu sur un même document, soit parce qu'il a dû être traité par différents moteurs soit parce que différents points du vue étaient nécessaires à partir du

même moteur d'extraction, le Module d'Extraction agrège tous ces résultats dans un même arbre conceptuel. Puis il sérialise cet arbre conceptuel en un objet Java XML, implémentant les APIs du Document Object Model (DOM) développé par Sun[37]. Cette action permet d'obtenir un nouveau format dans lequel les étiquettes linguistiques deviennent des éléments XML et les autres informations disponibles comme le lemme, la position, la longueur deviennent des attributs XML. Quant à l'unité textuelle étiquetée, elle devient la valeur de l'élément XML. L'avantage de ce format est qu'il normalise en quelque sorte toutes les extractions, quel que soit le moteur qui les a produit et le format de sortie imposé par ce dernier.

Le Module d'Extraction récupère sur le serveur ITM toutes les Règles d'Acquisition de Connaissances disponibles et compilées en une ou plusieurs feuilles de transformation XSLT. En fait, le nombre de feuilles de transformation dépend des besoins de l'application et de sa configuration :

- Si l'application a besoin d'enrichir sa base de connaissance, une feuille de transformation permettant de générer un format XTM sera créée.
- Si l'application a besoin d'annoter les documents, une feuille de transformation permettant de générer un format RDF sera créée.
- Si l'application a besoin des deux solutions, alors deux feuilles de transformation permettant de générer l'un et l'autre de ces formats seront créées.

Une fois ces feuilles de transformation chargées en mémoire, le Module va les appliquer pour transformer l'arbre conceptuel au format XML en objets Java : XTM pour le réseau sémantique et RDF pour les annotations sémantiques. Dans le cas où une seule feuille de transformation a été créée et donc appliquée, les deux objets sont tout de même produits, même si l'un d'entre eux reste vide. Ceci est nécessaire pour préserver la flexibilité et l'évolutivité du système dans son ensemble.

6.2.1.3 L'implémentation technique

Ce module correspond au package `com.mondeca.indexation.extract` qui contient 37 classes ou interfaces Java réparties en 7 sous-packages (cf. la Figure 68) :

a) Le package « Manage » récupère les paramètres enregistrés dans le fichier de configuration « extraction-config.xml » et coordonne les différentes actions à enchaîner pour mener à bien la tâche de ce module.

b) Le package « Match » teste si l'URL du document à analyser est bien pris en charge pour l'application cliente donnée.

c) Le package « Extract » se connecte à un moteur d'extraction externe afin de récupérer leurs annotations concernant le document passé en argument.

d) Le package « Serialize » transforme tout résultat retourné par les outils d'extraction sous la forme d'un objet XML représenté par les APIs Document Object Model (DOM).

e) Le package « Aggregate » agrège les différents objets DOM issus de multiples extractions dans un seul.

f) Le package « Transform » applique la ou les feuilles de transformation XSLT.

g) Le package « Model » décrit les modèles dans lesquels sont enregistrés les deux nouveaux objets DOM créés, soit XTM ou RDF.

[37] Document Object Model, issu d'une API spécifique à la gestion de fichiers XML sous la forme d'objets Java.

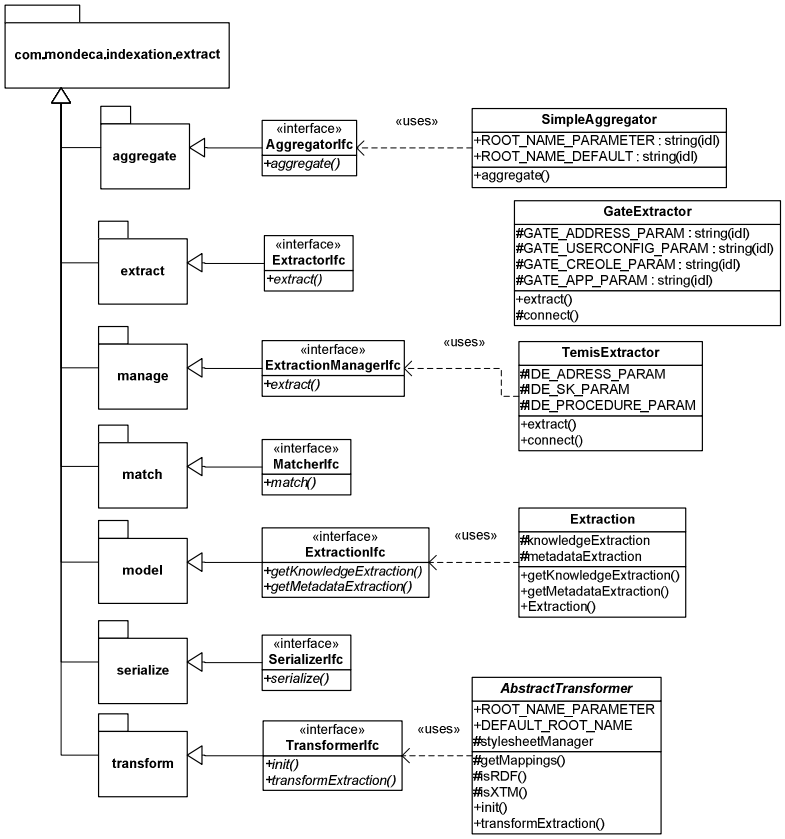


Figure 68. Extrait de l'architecture des classes Java du package « com.mondeca.indexation.extract »

6.2.2 Le Module de Peuplement d'Ontologie

Ce module a pour objectif de consolider le premier réseau sémantique de connaissance généré par le précédent module afin d'obtenir un réseau sémantique entièrement conforme à l'ontologie et à la norme du langage XTM. Les éléments ayant été rejetés au cours de ces différents contrôles sont tout de même gardés dans un réseau sémantique annexe et temporaire, dit « tampon ».

6.2.2.1 L'architecture

Ce module intervient toujours à la suite du Module d'Extraction d'Information car il prend en entrée le réseau sémantique généré par cette étape. Comme illustré dans la Figure 69, ce module s'interface avec :

- le Module d'Administration général qui lui fournit le Document XTM à contrôler et auquel il va renvoyer la connaissance consolidée et celle tampon ;
- le serveur ITM sur lequel il va récupérer les paramètres de configuration (opérations de nettoyage et de consolidation entre autres) mais aussi se connecter avec l'ontologie du domaine, la base de connaissance et les divers thésaurus et terminologies modélisés dans ITM.

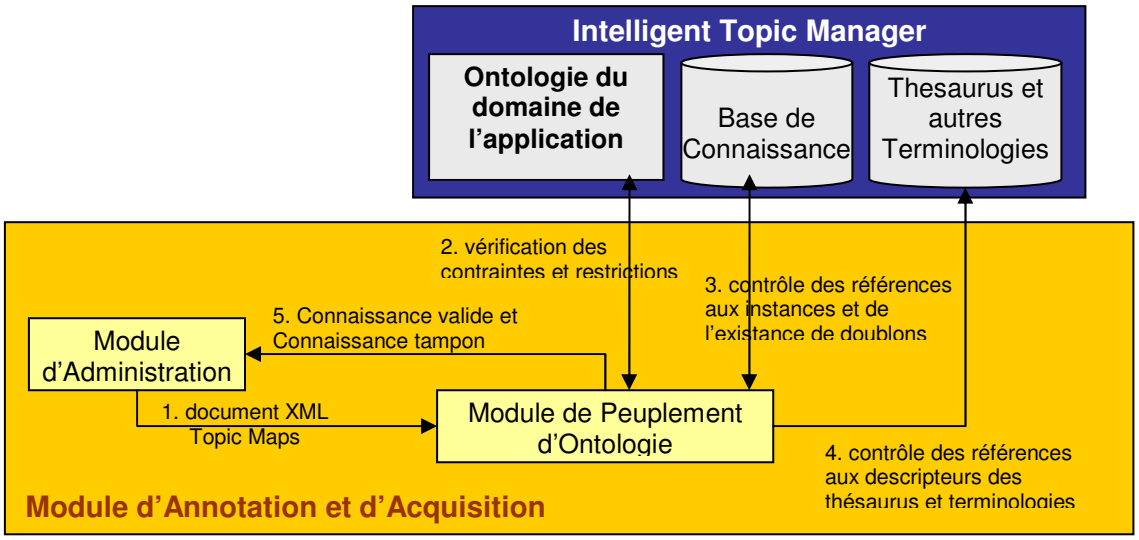


Figure 69. Architecture du Module de Peuplement d'Ontologie

6.2.2.2 Le processus détaillé

Le Module de Peuplement d'Ontologie est appelé par le Module d'Administration général qui lui transmet le réseau sémantique de connaissance, i.e. l'objet XTM, construit dans le module précédent. Le Module applique les algorithmes pour la consolidation de ce réseau sémantique vus à la section 4.2.2, en fonction des besoins de l'application pris en compte par les paramètres de configuration du module. Rappelons brièvement que les algorithmes de consolidation développés dans le cadre de ce module comprennent la résolution des problèmes suivants :

- les références aux instances existantes dans la base de connaissance de l'application,
- la redondance avec les instances présentes aussi bien dans le réseau que dans la base de connaissance ;
- les conflits entre classes ou références,
- les ambiguïtés, notamment au sujet des attributs et des rôles
- la non-conformité des éléments contenus dans le réseau sémantique au standard du langage XTM.

Les algorithmes vont également vérifier le respect des contraintes et des restrictions, notamment de domaine, de portée et de cardinalité, imposées par le modèle de l'ontologie du domaine de l'application. L'implémentation de ces algorithmes est extrèmement complexe et doit respecter un certain enchaînement dans l'ordonnancement des différents contrôles.

A l'issue de ces opérations de consolidation, deux réseaux sémantiques sont créés. Le premier conserve la connaissance contrôlée et valide vis-à-vis du modèle de l'ontologie, de la base de connaissance et des thesaurus. Le second représente la connaissance qui a été rejetée mais il est conservé pour validation ultérieure par un utilisateur.

6.2.2.3 L'implémentation technique

Ce module correspond au package Java, `com.mondeca.indexation.knowledge`, qui contient 22 classes, réparties en 6 sous-packages (cf. Figure 70) :

a) Le package « Manage » récupère les paramètres enregistrés dans le fichier de configuration « knowledge-config.xml » et coordonne les différentes actions.

b) Le package « Clean » nettoie le réseau sémantique de tous ses doublons internes.

c) Le package « IdGen » génère automatiquement les identifiants uniques des futures instances de la base de connaissance.

d) Le package « Merge » vérifie l'intégrité et la conformité de chaque instance du réseau sémantique initial. Ce package est certainement le plus complexe de tout le Module d'Annotation et d'Acquisition.

e) Le package « Control » définit les contrôleurs de formats mais aussi de contraintes sur les types de données.

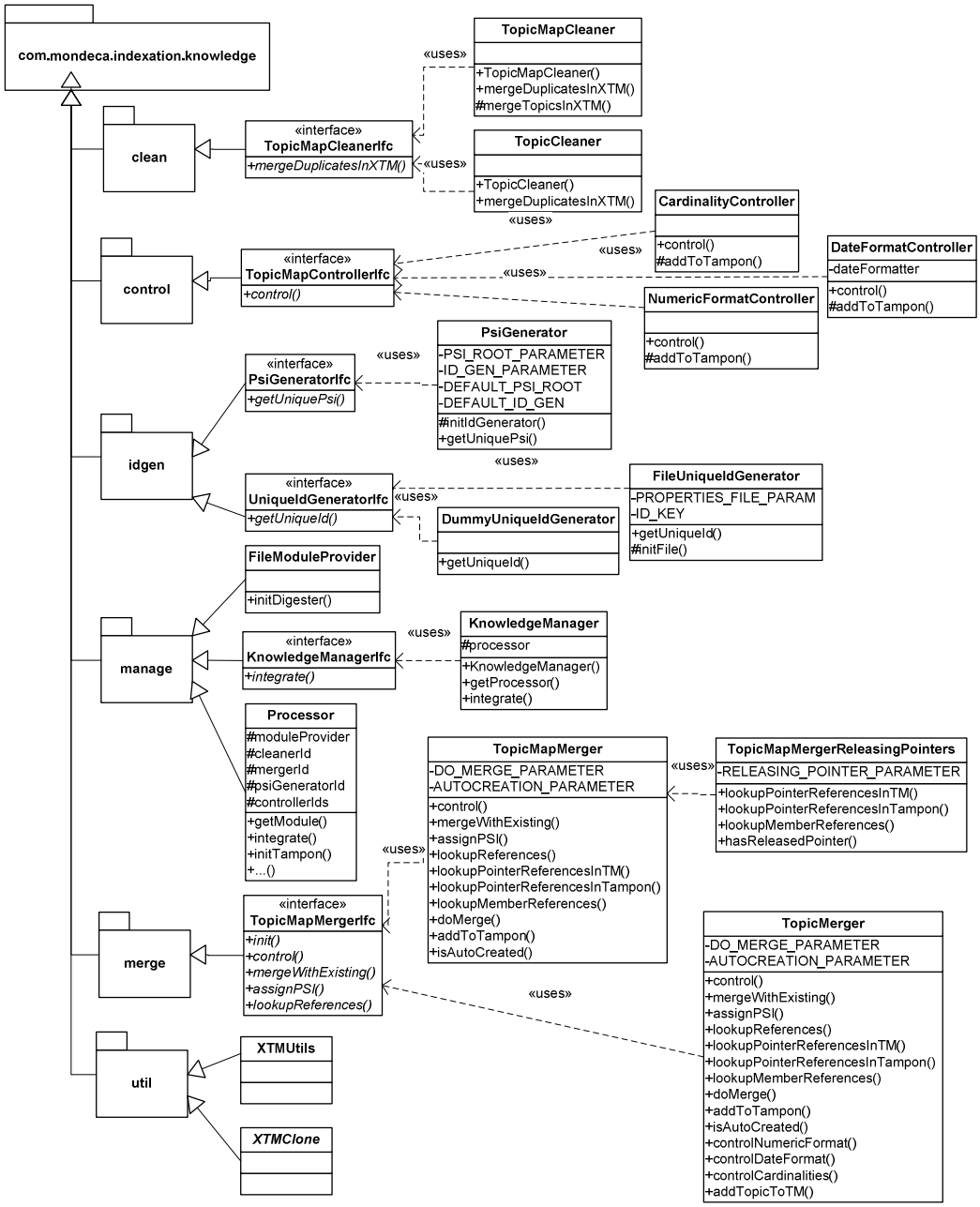


Figure 70. Architecture des classes Java du package « com.mondeca.indexation.knowledge »

6.2.3 Le Module d'Annotation Sémantique

Ce module a pour objectif de consolider les annotations sémantiques précédemment générées afin de produire des annotations qui soient conformes au contenu du référentiel terminologique et ontologique de l'application cliente. Les annotations rejetées au cours des différents contrôles sont conservées dans un document RDF annexe et temporaire.

6.2.3.1 L'architecture

Ce module intervient à la suite du Module d'Extraction d'Information, puisqu'il utilise les annotations sémantiques produites à cette étape, mais aussi à la suite du Module de Peuplement d'Ontologie. Ceci permet l'utilisation des nouvelles instances de la base de connaissance comme références pour les annotations sémantiques. D'après la Figure 71, ce module s'interface avec les composants suivants :

- le Module d'Administration qui lui fournit le Document RDF à contrôler et auquel il va renvoyer un objet `Metadata` contenant les annotations valides et tampon ;
- le serveur ITM qui contient les paramètres de configuration (opérations de consolidation à enchaîner, etc.), mais aussi l'ontologie du domaine, la base de connaissance et les divers thésaurus et terminologies métiers utilisés pour la consolidation.

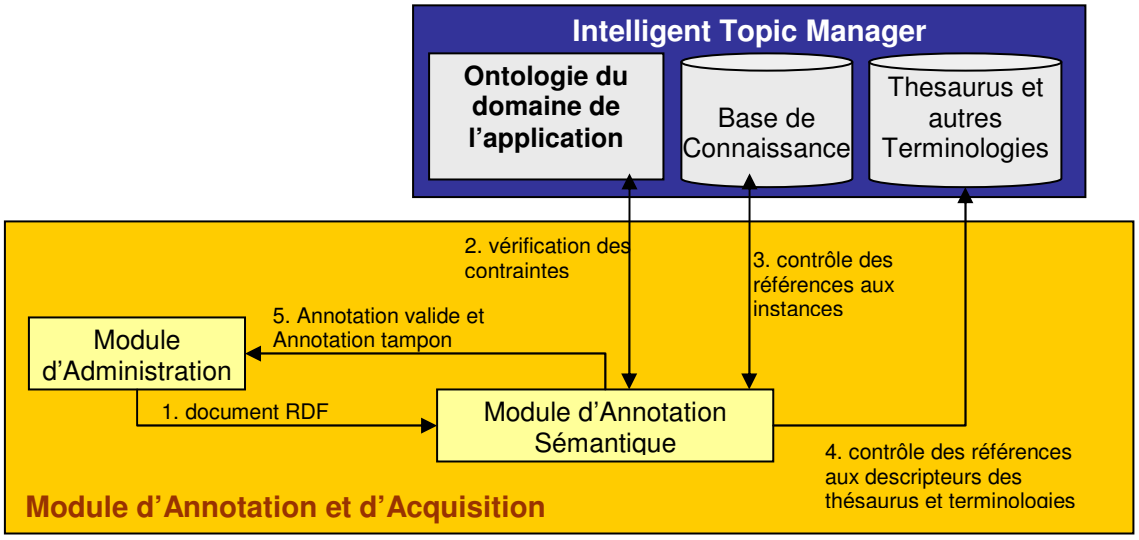


Figure 71. Architecture du Module d'Annotation Sémantique

6.2.3.2 Le processus détaillé

Le Module d'Administration général transmet au Module d'Annotation Sémantique les annotations extraites du document analysé, i.e. le Document RDF qui a été construit à l'issue du Module d'Extraction. Le Module d'Annotation applique les algorithmes dédiés à la consolidation des annotations sémantiques, comme vu à la section 4.2.2, et dont la plupart reprennent les opérations décrites au module précédent, comme le contrôle de l'existence des références aux instances ou aux descripteurs dans le référentiel de l'application ou la vérification des contraintes de domaine et de portée.

A l'issue de la consolidation des annotations sémantiques, deux Document RDF sont créés. L'un contient les annotations contrôlées et valides vis-à-vis de l'ontologie, de la base de connaissance et des thésaurus et l'autre représente les annotations rejetées par les différents contrôles mais qui sont conservées pour validation par un utilisateur humain.

6.2.3.3 L'implémentation technique

Ce module correspond au package Java, `com.mondeca.indexation.metadata`, qui contient 9 classes, réparties en 3 sous-packages (cf. Figure 72) :

a) Le package « Manage » récupère les paramètres enregistrés dans le fichier de configuration « metadata-config.xml » et coordonne les différentes actions.

b) Le package « Merge » vérifie chaque énoncé RDF composé d'un prédicat pointant vers une entité du référentiel de l'application afin de récupérer sa référence. Cette référence est constituée de l'identifiant, du libellé, de la classe et de l'URI de l'entité.

c) Le package « Control » définit les contrôleurs de types de données destinés aux énoncés composés d'un prédicat ayant une valeur Littérale.

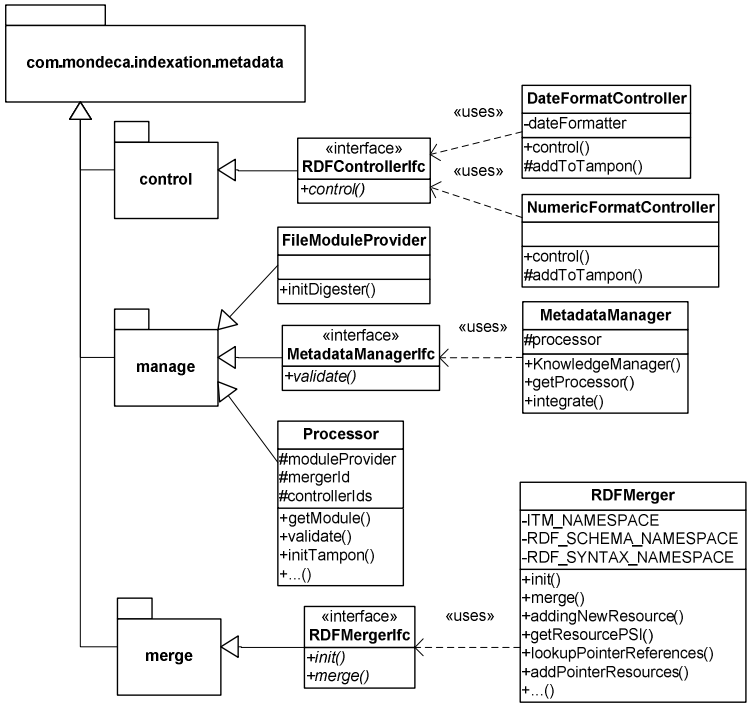


Figure 72. Architecture des classes Java du package « com.mondeca.indexation.metadata »

6.2.4 Le Module de Stockage

Ce module consiste à stocker les annotations sémantiques et les réseaux sémantiques, qu'ils soient valides ou tampon. Il possède différents modes de stockage et offre un accès permanent à ces résultats pour permettre leur exploitation par l'application cliente.

6.2.4.1 L'architecture

Ce module intervient une première fois après le Module de Peuplement d'Ontologie pour enregistrer les réseaux sémantiques valide et tampon. Puis il intervient à nouveau à la suite du Module

d'Annotation Sémantique pour enregistrer les deux modèles RDF. Il peut également être directement appelé par l'application ITM, et notamment par l'interface de validation qui a besoin de récupérer ces quatre résultats pour les présenter à l'utilisateur humain. Ce module s'interface avec les composants suivants (cf. Figure 73) :

- le Module d'Administration qui lui fournit les objets à stocker ainsi que la référence au document source et auquel il renvoie les objets stockés si demandés ;
- le serveur ITM sur lequel il va enregistrer, récupérer ou supprimer le réseau sémantique valide issu du document analysé ;
- un gestionnaire de contenus documentaires (CMS) externe où il peut transférer les annotations sémantiques valides.
- un système de fichiers, local ou distribué, où il va (selon la configuration choisie) enregistrer, récupérer ou supprimer les réseaux sémantiques ou les annotations, qu'ils soient valides ou tampon.

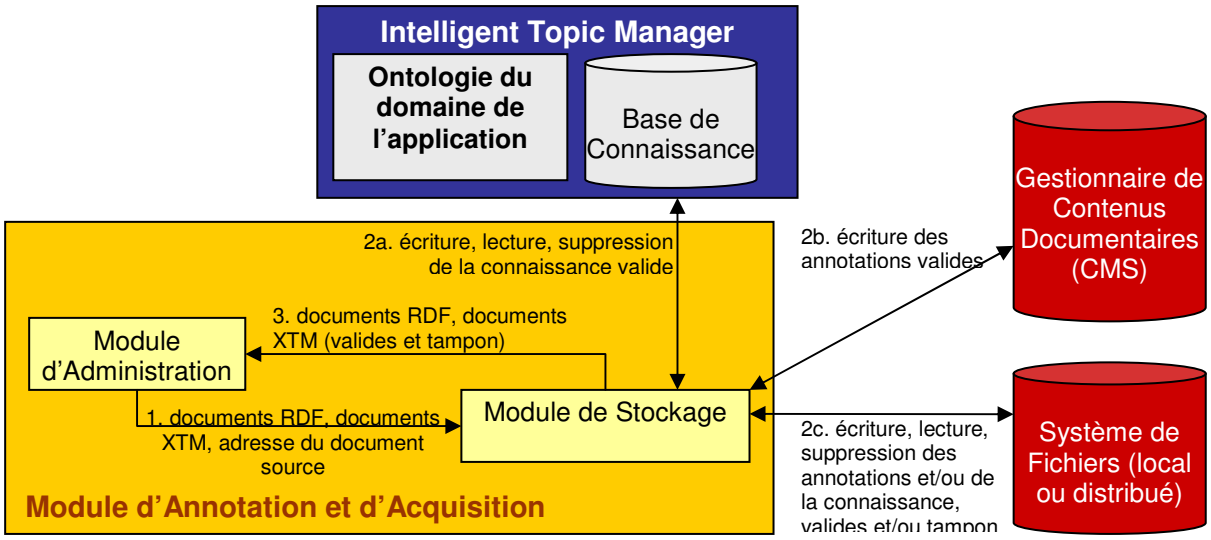


Figure 73. Architecture du Module de Stockage

6.2.4.2 Le processus détaillé

Le Module de Stockage est appelé par le Module d'Administration général à diverses étapes du processus d'Acquisition et d'Annotation. Premièrement, il est appelé pour stocker les résultats du Module de Peuplement d'Ontologie. Les deux réseaux sémantiques, valide et tampon, sont stockés séparément. Généralement, le réseau valide est directement importé dans la base de connaissance d'ITM. Lors de cet import, les trois métadonnées suivantes sont ajoutées à chaque instance :

- la langue : cette métadonnée précise la langue correspondant au libellé de l'instance (Français, Anglais, etc.). Cette information provient du moteur d'extraction linguistique lorsqu'il a pu détecter la langue dans laquelle le document était rédigé. Lorsqu'il n'est pas possible de connaître la langue d'origine, alors l'instance est importée dans ITM avec la valeur « pas de langue ».

- la source de l'information : cette métadonnée précise l'origine de l'instance. En fait cette métadonnée est double : il y a une métadonnée stockant l'adresse du document afin que toutes les instances liées à ce document source puissent être retrouvées ultérieurement par l'application ITM, et il y a une métadonnée stockant l'origine du processus. En effet, deux manières sont aujourd'hui possibles pour créer une instance dans ITM : soit manuellement depuis les interfaces utilisateurs, soit automatiquement par l'utilisation d'OntoPop. Il est important aussi bien pour les différentes interfaces utilisateurs que pour les utilisateurs eux-mêmes de connaître l'origine d'une nouvelle instance dans ITM. En effet, le degré de confiance accordé à cette nouvelle instance n'est pas la même si cette dernière provient d'un utilisateur humain ou d'un processus automatisé. Cette métadonnée précise donc que ces instances ont été créées ou mises à jour via un processus d'extraction linguistique.
- Le statut de l'information : cette métadonnée précise le statut de l'instance créée ou mise à jour dans la base de connaissance ITM. Il y a deux statuts possibles : « à valider » ou « validée ». Lorsqu'une instance est importée dans ITM, son statut est automatiquement « à valider ». Ainsi, dans le cadre d'un processus semi-automatisé où un utilisateur humain doit valider ces instances importées automatiquement, il devient aisé de retrouver toutes les instances ayant ce statut « à valider ». Une fois validées par les utilisateurs humains, leur statut change pour « validée ».

Comme le réseau sémantique tampon ne peut être importé dans la base de connaissance, il est donc enregistré dans un fichier situé sur le serveur ITM. Mais on peut tout aussi bien imaginer ne pas vouloir le stocker du tout, notamment lors d'un processus entièrement automatisé où il n'y aurait pas de validation humaine a posteriori.

Deuxièmement, le Module de Stockage enregistre les résultats du Module d'Annotation Documentaire, i.e. les modèles RDF valide et tampon. Dans la plupart des applications, ils sont stockés dans des fichiers RDF sur le serveur ITM. Mais le modèle valide peut aussi être transmis à un gestionnaire de contenus externe où il est alors rattaché au document source.

Troisièmement, dans le cadre d'un processus semi-automatisé, l'interface de validation récupère, pour un document donné, les quatre résultats produits par le Module d'Annotation et d'Acquisition afin de les afficher dans l'interface utilisateur dédiée à la validation (cf. prochaine section).

Quatrièmement, dans le cadre d'une maintenance de l'application, ITM peut se connecter au Module de Stockage afin de supprimer tout ou partie des résultats. Cela permet notamment de nettoyer le serveur ITM des différents fichiers temporaires créés.

6.2.4.3 L'implémentation technique

Le package Java de ce module, `com.mondeca.indexation.storage`, contient 13 classes réparties en 3 sous-packages (cf. Figure 74) :

a) Le package « Manage » récupère les paramètres enregistrés dans le fichier de configuration « storage-config.xml » et coordonne les différentes actions.

b) Le package « ReadWrite » contient les différentes actions de stockage dépendantes de la configuration de l'application cliente.

c) Le package « Util » contient un utilitaire pour reconstruire un document XTM.

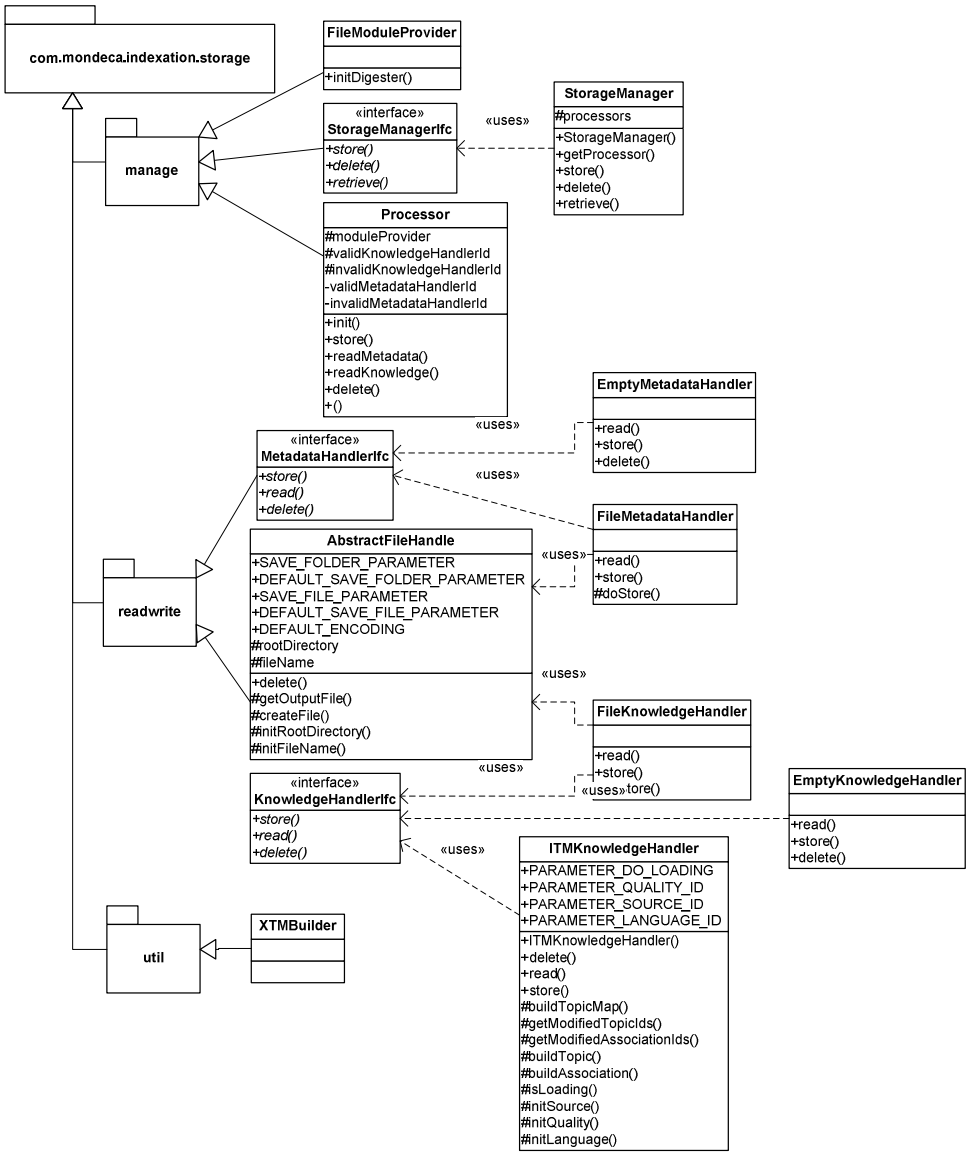


Figure 74. Architecture des classes Java du package « com.mondeca.indexation.storage »

6.2.5 L'Interface de validation

Cette interface a pour objectif de permettre à l'utilisateur humain de valider d'une manière conviviale et efficace les résultats produits par le Module d'Acquisition et d'Annotation. Cette validation porte aussi bien sur les instances créées dans la base de connaissance, sur le réseau sémantique tampon et sur les annotations sémantiques, valides ou tampon.

6.2.5.1 L'architecture

Cette interface est destinée à être utilisée dans le cadre d'un processus semi-automatisé du Module d'Acquisition et d'Annotation où elle présente les résultats obtenus par ce module pour un document donné. On peut aussi envisager de l'utiliser pour créer des annotations sémantiques à partir d'un document présenté dans l'interface dans le cadre d'un processus manuel. A l'issue de la validation

humaine, elle répercute les mises à jour dans la base de connaissance ITM et dans le fichier des annotations sémantiques. Cette interface s'articule avec (Cf. Figure 75) :

- L'utilisateur humain qui valide les résultats du Module d'Acquisition et d'Annotation ;
- Le serveur ITM qui contient la connaissance valide et avec laquelle elle va interagir tout au long du processus de validation des instances de la base de connaissance ;
- Un gestionnaire de contenus documentaires (CMS) qui peut contenir les annotations sémantiques valides selon la configuration de l'application ;
- Le système de fichiers, local ou distribué, où sont stockés les tampons.

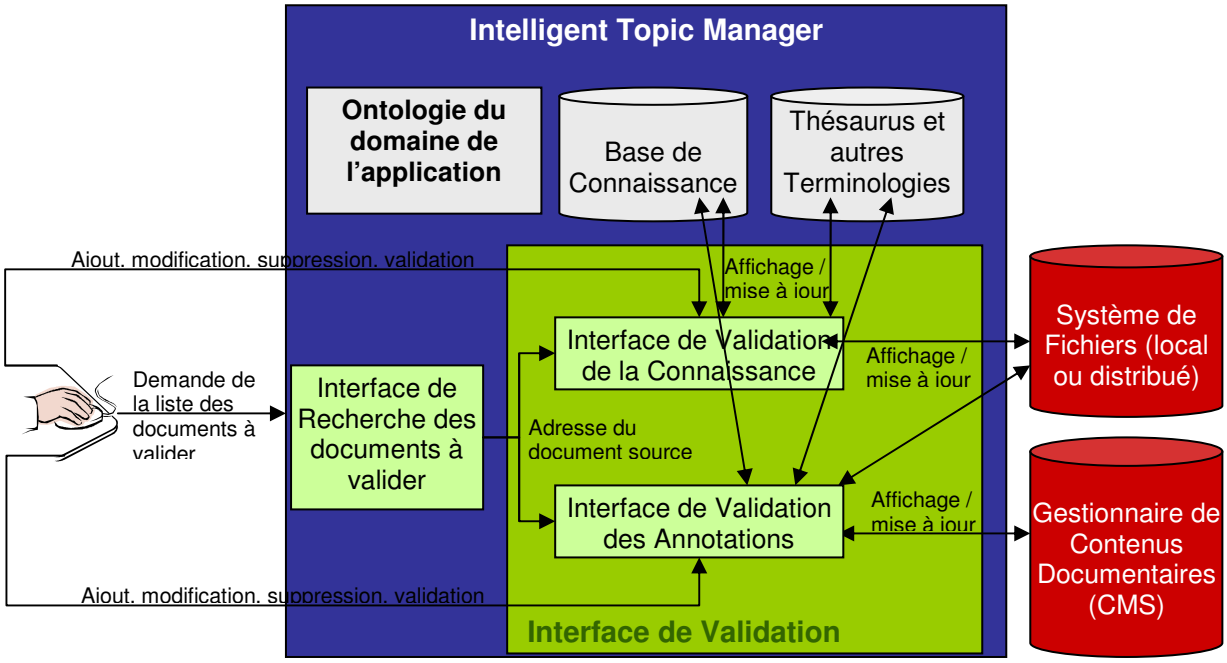


Figure 75. Architecture des Interfaces de Validation

6.2.5.2 Le processus détaillé

L'interface de validation est accessible à partir de deux flux :

1) Soit le processus d'Annotation et d'Acquisition est lancé à partir d'un programme en mode différé sur un corpus documentaire. L'utilisateur doit alors valider les résultats fournis pour chacun des documents traités. Dans ce cas, il recherche tous les documents ayant un statut « à valider » dans la base de connaissance et les sélectionne l'un après l'autre pour validation dans l'interface de validation.

2) Soit ce processus est initié par un utilisateur à partir de l'interface standard d'ITM. Dans ce cas, l'utilisateur lance l'interface de validation à partir d'un document donné, ce qui déclenche automatiquement les traitements effectués par les modules précédents sur ce document et l'affichage des résultats produits dans l'interface.

Que ce soit dans l'un ou l'autre des cas précédents, l'interface de validation récupère les résultats du Module d'Annotation et d'Acquisition, i.e. les réseaux sémantiques (valide et tampon) et les

annotations sémantiques (valide et tampon). Ces objets sont simultanément chargés dans l'interface qui possède un onglet « Annotations » représentant l'espace de travail dédié aux annotations sémantiques et un onglet « Connaissances » pour celui des réseaux sémantiques.

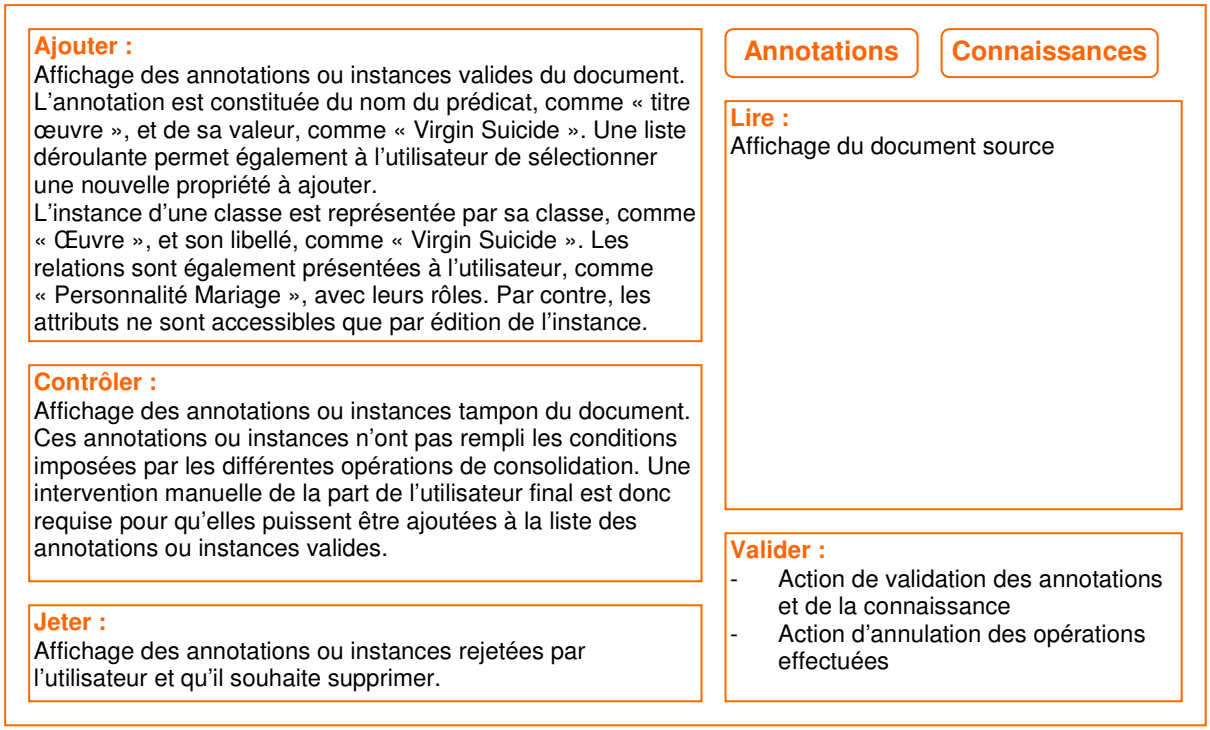


Figure 76. Maquette de l'interface de validation

La partie droite de l'interface est commune aux deux onglets. Elle affiche le contenu du document source (cf. la partie « Lire » de la Figure 76), ce qui permet à l'utilisateur de remettre les annotations et les instances dans leur contexte initial et ainsi de mieux contrôler les erreurs et oublis dûs au Module d'Annotation et d'Acquisition. Cette partie commune comprend aussi les actions « Validate » et « Cancel » (cf. la partie « Valider » de la Figure 76) qui déclenchent respectivement la validation de toutes les informations (annotations et/ou instances) ou l'annulation des actions effectuées par l'utilisateur. Quelle que soit l'action sélectionnée, elle entraîne la fermeture de l'interface de validation.

La partie située à gauche est dépendante de l'onglet sur lequel est positionné l'utilisateur. Ce dernier peut à tout moment changer d'onglet pour visualiser les résultats correspondants. Dans chaque onglet, l'utilisateur peut modifier, supprimer et valider les informations présentées à l'aide de menus contextuels. Il peut également ajouter de nouvelles annotations sémantiques à partir de l'onglet Annotations. Par contre, l'ajout d'instances dans la base de connaissance s'effectue toujours par les écrans standards d'édition d'ITM. La Figure 77 montre l'interface de validation telle qu'implémentée dans ITM du point de vue de l'onglet Annotations.

Figure 77. Onglet « Annotations » de l'Interface de Validation

Une fois la validation de la connaissance terminée, le système enregistre les mises à jour (création, modification, suppression) dans la base de connaissance d'ITM. Le statut de chaque instance enregistrée dans la base de connaissance change de statut pour « validé ». Cette connaissance devient accessible aux utilisateurs de l'application comme dans la Figure 78. La validation des annotations sémantiques est sauvegardée sur le serveur ITM dans un nouveau fichier RDF remplaçant le précédent. Ce fichier pourra être de nouveau livré ou mis à disposition du gestionnaire de contenu externe de l'application cliente si nécessaire.

Figure 78. Exemple de l'interface d'affichage d'une instance dans ITM, ici « Spike Jonze »

6.2.5.3 L'implémentation technique

Toutes les interfaces utilisateurs d'ITM sont développées dans le package `com.mondeca.web`. Ce package possède un sous-package `annotation` regroupant toutes les classes Java relatives à l'interface de validation. Il contient notamment les actions accessibles à l'utilisateur (comme lister les documents à valider, charger l'interface de validation, éditer une entité du référentiel, valider annotations et instances, etc.) et les « beans » manipulés ces actions. Ces classes Java sont utilisées et appelées à partir d'un ensemble de pages JSP, utilisant des scripts ainsi que des feuilles de style CSS pour l'affichage. L'interface de validation est réalisée en complète adéquation avec les interfaces existantes d'ITM, notamment par en respectant l'architecture mise en place par l'équipe de développement de Mondeca. Elle fait partie intégrante de l'offre de Mondeca lors de la mise en place de la solution OntoPop chez les clients. Néanmoins, certains d'entre eux, pour des raisons d'harmonisation avec leurs propres outils, préfèrent développer leurs propres interfaces à partir des APIs mises à leur disposition.

6.3 Le Module de Maintenance des Lexiques

L'objectif de ce module consiste à envoyer au moteur d'extraction les libellés des nouvelles instances ou des nouveaux descripteurs du référentiel afin qu'il puisse compléter ses propres lexiques et ainsi améliorer ses résultats. Ces entités proviennent soit de l'étape de validation, soit d'une édition manuelle via les écrans standards d'ITM.

6.3.1 L'architecture

Comme montré dans la Figure 79, ce module intègre les composants suivants :

- les interfaces de validation et d'édition d'ITM ainsi que le Module d'Annotation et d'Acquisition qui envoient une ou plusieurs entités au moteur d'extraction ;
- le module d'Alerte d'ITM qui déclenche le transfert des entités en mode temps réel ;
- le module d'Export d'ITM qui récupère directement dans la base de connaissance et/ou les thésaurus l'ensemble des entités à transmettre en mode différé.

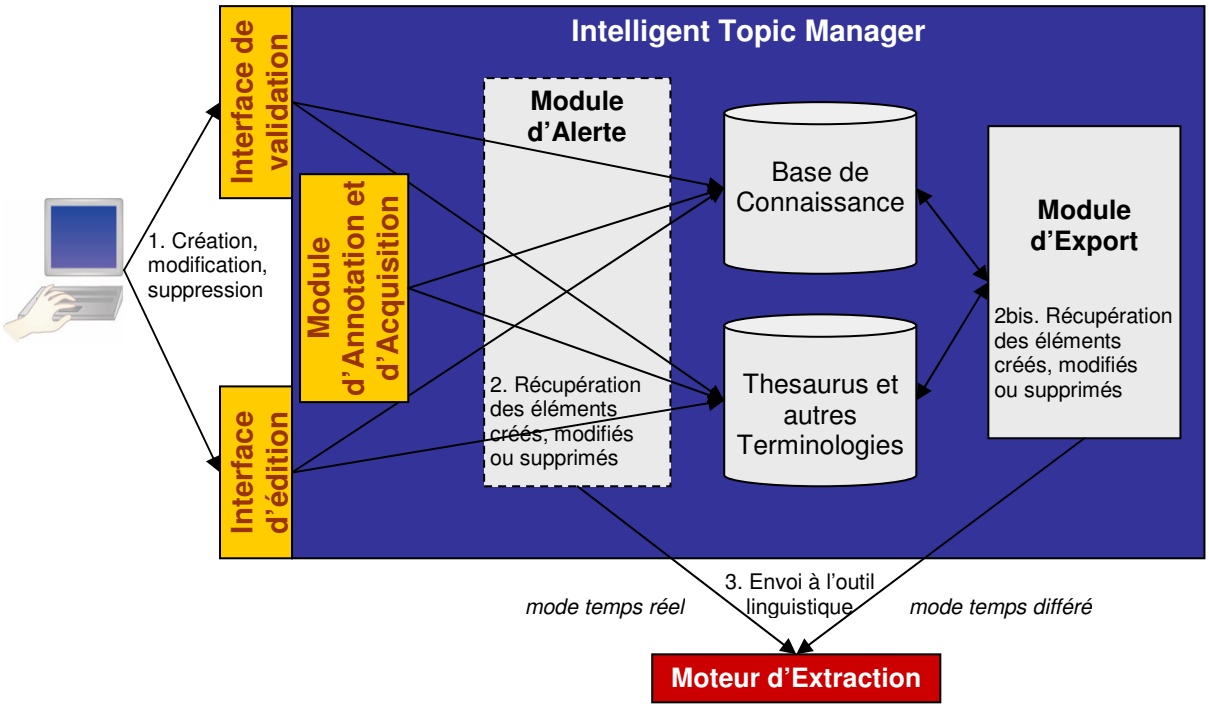


Figure 79. Architecture du Module de Maintenance des Lexiques

6.3.2 Le processus détaillé

Ce module fonctionne de deux manières : soit en temps réel, soit en temps différé. Dans le mode en temps réel, ce module intervient dès qu'une opération de création, modification ou suppression a eu lieu sur une entité du référentiel, à savoir une instance ou un descripteur. Ces opérations sont effectuées manuellement par le biais de l'interface d'édition standard d'ITM ou de l'interface de validation, ou bien automatiquement par le Module d'Annotation et d'Acquisition. Le Module d'Alerte d'ITM est paramétré pour « écouter » les opérations effectuées dans la base de connaissance ou dans les thésaurus et déclencher les actions correspondantes. Ainsi, à chaque création, modification ou suppression d'une entité du référentiel paramétrée, il se connecte au moteur d'extraction d'information utilisé par l'application pour lui transmettre cette entité.

Dans le mode en temps différé, ce module est programmé pour se déclencher à intervalles réguliers selon les besoins de l'application cliente. Il appelle le module d'Export d'ITM afin de récupérer non plus une seule entité mais l'ensemble des informations qui ont été créées, modifiées ou supprimées soit depuis le dernier export réalisé par le module, soit depuis la création de l'application. Ce mode sert de filet de sécurité en cas de panne du mode temps réel et permet aussi de réinitialiser tous les lexiques.

Dans les deux cas, les entités exportées par ce module sont constituées des instances ou des descripteurs des classes spécifiées dans la configuration de l'application cliente, de certains de leurs attributs (comme alias, synonyme, traduction) et des relations de type hiérarchique ou de type associative issues des thésaurus. Les identifiants de ces entités dans ITM ainsi que les informations liées à leur statut (nouveau, modifié, supprimé) sont préservés dans l'outil linguistique. Le statut de

l'instance ou du descripteur permet de déterminer le type de traitement que le moteur d'extraction doit effectuer :

- S'il s'agit d'une création, il faudra insérer une nouvelle entrée dans le lexique correspondant ;
- S'il s'agit d'une modification, il faudra trouver la bonne entrée dans le lexique correspondant et répercuter les modifications ;
- S'il s'agit d'une suppression, il faudra trouver la bonne entrée dans le lexique correspondant et la supprimer.

Une fois l'information transmise à l'outil linguistique, celui-ci doit appliquer toute une série d'heuristiques afin de savoir quel est le lexique impacté, quelle est la manière utilisée pour ajouter, modifier ou supprimer chaque entrée, et enfin quel est son format d'écriture. Sur ce dernier point, les libellés extraits de la base de connaissance ou du thésaurus ne peuvent parfois être utilisés comme tels. Une normalisation est donc nécessaire pour traiter les différents cas : les minuscules, les accents, les marqueurs de ponctuation, etc.

Lorsque chaque entrée a été ajoutée au lexique correspondant, la cartouche linguistique est automatiquement recompilée et réinstallée sur le serveur linguistique en un minimum de temps, surtout dans le cas du mode en temps réel. Ainsi, à la prochaine extraction d'information, les nouvelles entrées du lexique sont automatiquement reconnues et étiquetées dans l'arbre conceptuel généré. Afin de gagner en rapidité, les heuristiques précédemment appliquées sont beaucoup plus souples dans le cadre du mode en temps réel : moins de vérifications sont effectuées, autorisant plus de bruit lors de l'extraction de nouvelles informations. C'est pourquoi à intervalles réguliers, une nouvelle maintenance est effectuée par le mode en temps différé qui sera beaucoup plus précise dans l'application des heuristiques et augmentera ainsi la qualité des lexiques.

6.3.3 L'implémentation technique

Si j'ai effectué l'analyse technique et rédigé les spécifications fonctionnelles de ce module, l'implémentation technique a été réalisée par l'équipe de développement de Mondeca et notamment par la personne responsable des modules d'Alerte et d'Export d'ITM. En effet, l'implémentation technique repose sur deux plug-ins, chacun situés dans un de ces modules. Jusqu'à présent, seuls les plug-ins de connexion à l'IDE ont été implémentés. Mais d'autres plug-ins peuvent être développés en fonction des outils linguistiques utilisés par l'application cliente. La configuration des éléments à exporter vers ces outils est décrite dans les fichiers de configuration propres à chacun des modules d'Alerte et d'Export.

6.4 Conclusion

Nous avons décrit, dans ce chapitre, l'implémentation technique de la plateforme logicielle réalisée tout au long de cette thèse. Cette plateforme s'organise autour de trois principaux modules, dérivés de la démarche proposée par OntoPop, à savoir l'Editeur de Règles d'Acquisition de connaissance, le Module d'Annotation et d'Acquisition et le Module de Maintenance des Lexiques.

Parmi les exigences citées au début de ce chapitre comme nécessaires et indispensables au bon fonctionnement et à une bonne utilisation d'une plateforme logicielle comme OntoPop, je pense avoir répondu à chacune d'entre elles :

- **L'indépendance entre la structure de l'ontologie et la structure des extractions linguistiques**. L'indépendance est préservée grâce à l'implémentation des Règles d'Acquisition de Connaissance. Ces dernières représentent autant de passerelles permettant de passer d'un modèle de représentation à un autre sans pour autant intervenir sur le format de l'un ou de l'autre.
- **La complétude.** Toute information donnée par les outils d'extraction peut être récupérée grâce à la conception de nouvelles Règles d'Acquisition de Connaissance. Celles-ci sont suffisamment flexibles pour s'adapter à toute configuration en arbre conceptuel. D'autre part, elles peuvent s'appliquer plus généralement à toute forme de document XML, et pas seulement ceux fournis par les outils linguistiques. Ainsi, par exemple, le Module d'Annotation et d'Acquisition est capable de se connecter à un serveur documentaire ou à un serveur d'annotation afin de récupérer et de traiter les annotations déjà attachées à un document donné.
- **La standardisation.** OntoPop est complètement indépendant du moteur d'extraction utilisé, tant que ce dernier lui fournit un arbre conceptuel en sortie, i.e. un document XML. Et afin d'accentuer l'interopérabilité de la plateforme logicielle avec d'autres outils, OntoPop repose sur les langages standards tant informatiques, comme Java ou XML, que de représentation des connaissances, comme RDF, XTM, voire OWL.
- **La cohérence.** La cohérence est une priorité dans OntoPop. Elle est traitée à un premier niveau par les règles d'Acquisition qui, grâce à l'introduction de règles contextuelles, permettent de contrôler la pertinence des informations extraites vis-à-vis des annotations ou des instances à créer. Ensuite, elle est traitée à un second niveau de manière plus fine grâce aux différentes opérations de consolidation effectuées par le Module d'Annotation et d'Acquisition soit sur l'ontologie du domaine, soit sur les instances de la base de connaissance ou encore les descripteurs des divers thésaurus utilisés.
- **La facilité d'utilisation**. L'ensemble du processus d'Annotation et d'Acquisition ainsi que celui de maintenance des lexiques sont entièrement transparents aux utilisateurs. L'intégrateur dispose d'une interface utilisateur très simple lui permettant de créer de nouvelles règles d'acquisition qui seront automatiquement enregistrées sur le serveur ITM de l'application. Quant à l'utilisateur final, il visualise les résultats fournis par le module d'Annotation et d'Acquisition dans une seule et même interface regroupant les propositions provenant de l'annotation sémantique et du peuplement d'ontologie. Par ailleurs, le processus méthodologique entre les différents intervenants est basé sur une forte communication ainsi que sur un processus itératif leur permettant de connaitre les enjeux, les besoins et les contraintes de chacun afin de trouver les solutions adéquates le plus en amont possible.
- **La capacité à évoluer.** Par un fonctionnement basé sur l'utilisation de plug-ins, OntoPop est capable d'évoluer et de s'adapter à de nouveaux moteurs d'extraction sans pour autant devoir

redévelopper le noyau de son implémentation. Il est également possible, grâce à son système de configuration très fin des différents sous-modules du Module d'Annotation et d'Acquisition, de le faire évoluer en fonction des besoins des futures applications cibles.

Pour conclure, OntoPop n'a pas seulement donné lieu à une méthodologie et à une plate-forme logicielle théoriques mais a été testé dans divers projets clients de Mondeca auxquels j'ai pu participer tout au long de ma thèse. Dans certains projets, notre solution a été validée par le client et mise en production auprès d'utilisateurs finaux, c'est-à-dire qu'il a atteint la phase de Mise en Service de la méthodologie. Nous allons dans la prochaine partie de ce mémoire décrire les résultats produits par OntoPop dans deux projets forts différents de par leurs problématiques, leurs domaines et leurs contraintes.

Quatrième Partie.

Expérimentations et Bilan de la démarche proposée

Chapitre 7. Expérimentations et évaluation d'OntoPop

La démarche OntoPop, sa méthodologie et sa plateforme sont actuellement utilisées dans plusieurs projets menés conjointement par Mondeca et Temis auprès de clients dont les domaines d'application et les problématiques sont forts différents les uns des autres.

Domaine de l'application	**Stade actuel du projet**	**Peuplement et/ou annotation**	**Nombre d'éléments ontologiques concernés**	**Nombre d'étiquettes linguistiques différentes**	**Nombre de RAC**	**Nombre de documents traités**
Presse "People"	Mise en Service	Les deux	4 classes, 6 relations, 14 rôles, 13 attributs	64 tags	62 RAC	Centaines d'articles par jour
Edition Juridique	Validation	Les deux	9 classes, 3 relations 6 rôles, 27 attributs	66 tags	77 RAC	+ 100000 décisions de jurisprudence
Presse Evénementielle	Couplage	Les deux	4 classes, 12 relations 26 rôles, 7 attributs	27 tags	34 RAC	Milliers de dépêches par jour
Veille Economique	Couplage	Les deux	5 classes, 24 relations 39 rôles, 76 attributs	68 tags	70 RAC	Centaines d'articles par jour
Veille Scientifique	Couplage	Peuplement	5 classes, 15 attributs	18 tags	24 RAC	50 millions de résumés

Tableau 6. Réalisations menées sur divers domaines d'application

Comme résumé dans le Tableau 6, ces projets concernent aussi bien les domaines de la presse people [AMA 06b], de l'édition juridique [AMA 05b] [AMA 06a], de la veille économique ou scientifique [AMA 04] [AMA 06c], etc. Tous ont pour objectif de peupler semi-automatiquement leur ontologie de domaine et d'annoter sémantiquement leurs ressources documentaires, mis à part le projet la Veille Scientifique qui s'intéresse uniquement à l'acquisition de nouvelles connaissances. Tous les éléments d'une ontologie ne sont pas forcément couplés par les Règles d'Acquisition de Connaissance. Cela dépend des prérequis de la couverture du domaine nécessaires à la réalisation des objectifs de l'application concernée. On voit aussi qu'en général, il y a à peu près autant de Règles d'Acquisition de Connaissance (RAC) que d'étiquettes linguistiques générées pour une cartouche linguistique donnée. Néanmoins l'intégrateur doit maintenir manuellement l'ensemble de ces Règles d'Acquisition

de Connaissance pour chaque projet, ce qui peut vite devenir une tâche fastidieuse. Enfin, le Tableau 6 montre aussi que les différents projets en cours se trouvent à différentes étapes de la méthodologie OntoPop : les projets les plus aboutis sont celui de la Presse People, qui a fourni les exemples utilisés tout au long de ce mémoire, et celui de l'Edition Juridique. Les autres projets se situent encore dans la phase de Couplage.

Dans ce chapitre, nous allons présenter le résultat d'évaluations conduites sur les deux projets Presse People et Edition Juridique lors de leurs phases de validation. Nous en dégagerons un ensemble de problèmes soulevés lors de ces expérimentations. Mais auparavant, nous allons expliciter les critères et les mesures utilisées pour les évaluations.

7.1 Mesures pour l'évaluation

Nous avons vu à la section 0 que lors de la phase de validation, l'intégrateur doit rendre compte de la performance de l'application mise en place par rapport au domaine concerné, c'est-à-dire par rapport à l'ontologie modélisée pour ce domaine et au corpus documentaire représentatif de ce domaine. L'intégrateur doit notamment s'assurer des points suivants :

- à partir des étiquettes sémantiques composant les arbres de concepts de chacun des documents du corpus de test, chacune des règles d'acquisition faisant usage de ces étiquettes se déclenche correctement ;
- chaque règle d'acquisition instancie ou annote le bon élément de l'ontologie du domaine ;
- la valeur textuelle de l'instance ou de l'annotation générée est pertinente et cohérente avec l'élément de l'ontologie ;
- les différents contrôles opérés par la chaîne de traitement sont correctement effectués et les éléments erronés ou dupliqués sont rejetés dans les tampons respectifs.

7.1.1 Mesures de la performance des RACs

Cette mesure de la performance est le plus souvent réalisée à partir des mesures de précision et de rappel que nous avons déjà évoquées au Chapitre 2. Rappelons brièvement que ces mesures ont été initialement définies dans le domaine de la Recherche d'Information avant d'être utilisées par le domaine de l'Extraction d'Information pour l'évaluation des moteurs d'extraction dans le cadre des conférences MUC [LAV 04]. La précision y est alors définie comme le nombre de champs correctement remplis dans un formulaire divisé par le nombre de champs remplis par le système. Le rappel est lui défini comme le nombre de champs correctement remplis divisé par le nombre de champs corrects possibles d'après le corpus de référence annoté par un humain [GRI 96]. Tous les champs ont le même poids et leur valeur est comprise entre 0 et 1, l'objectif étant d'atteindre la valeur 1 aussi bien au niveau du rappel que de la précision. Dans la pratique, cet objectif est impossible car si les systèmes produisent une très bonne précision (peu de bruit) sur des domaines bien délimités

alors le rappel est moindre (beaucoup de silence) laissant des informations non extraites ou non trouvées ; et inversement. C'est pourquoi la F-mesure combine les résultats fournis par la précision et le rappel pour fournir un seul chiffre d'évaluation d'un système. La F-mesure permet ainsi de comparer la performance obtenue entre différents systèmes.

J'ai donc décidé d'emprunter à mon tour ces mesures pour évaluer la qualité et la pertinence des résultats fournis par OntoPop et ses Règles d'Acquisition de Connaissance, tant pour le peuplement d'ontologie que pour l'annotation sémantique. Voici donc les nouvelles définitions des mesures de précision et de rappel pour chacun de ces cas :

1) pour le peuplement d'ontologie :
   - **Précision** = le nombre d'instances correctement acquises divisé par le nombre d'instances acquises
   - **Rappel** = le nombre d'instances correctement acquises divisé par le nombre d'instances existantes dans les arbres conceptuels du corpus
2) pour l'annotation sémantique :
   - **Précision** = le nombre d'annotations sémantiques correctement créées divisé par le nombre d'annotations sémantiques créées
   - **Rappel** = le nombre d'annotations sémantiques correctement créées divisé par le nombre d'annotations sémantiques existantes dans les arbres conceptuels du corpus

Dans les deux cas, la F-mesure se définit par la formule suivante :

$$\mathbf{F_\alpha} = (1+\alpha)\ (\text{precision} * \text{rappel}) / (\alpha * \text{précision} + \text{rappel})$$

dans laquelle je considère α=1 pour accorder autant de poids à la précision qu'au rappel. D'où la F-mesure que nous allons appliquer pour évaluer OntoPop dans nos projets :

$$\mathbf{F_1} = 2 * (\text{precision} * \text{rappel}) / (\text{précision} + \text{rappel})$$

7.1.2 Mesure de la complexité des RACs

Pour l'évaluation que nous devons réaliser, un autre aspect important concerne la complexité des Règles d'Acquisition de Connaissance implémentées pour chaque projet. Pour nous aider à évaluer cette complexité, nous avons regardé du côté des expérimentations menées sur les systèmes implémentant la méthode d'exploration contextuelle. Nous avons retenu la mesure de la complexité du corpus fournie par Le Priol dans sa thèse [LEP 00]. Elle l'emploie pour évaluer la complexité de divers corpus documentaires devant être analysés et étiquetés par le système SEEK-Java afin de les comparer. Cette mesure s'intéresse au nombre moyen de règles d'exploration contextuelle qui peuvent être déclenchées par un indicateur linguistique[38] et se définit comme suit :

[38] Rappelons ici que, dans la méthode d'exploration contextuelle, un indicateur linguistique sert à expliciter la valeur sémantique à repérer dans un texte donné (indicateurs discursifs d'annonces thématiques, etc.).

$$Complexité_{corpus} = \frac{\sum_{i=1}^{I} A(indicateur_i) \times N(indicateur_i)}{\sum_{i=1}^{I} N(indicateur_i)}$$

où :

I est le nombre de classes d'indicateurs différentes relevées dans le corpus

A(indicateur$_i$) : ambiguïté de la classe d'indicateurs, c'est-à-dire nombre de règles qui peuvent être déclenchées pour chaque classe d'indicateurs (indicateur$_i$) relevée dans le corpus

N(indicateur$_i$) : nombre d'occurrences de chaque classe d'indicateurs (indicateur$_i$) relevée dans le corpus

Figure 80. Définition de la mesure de complexité d'un corpus d'après Le Priol dans [LEP 00]

J'ai adapté cette mesure de la complexité du corpus en fonction du nombre de RACs qui peuvent être déclenchées pour instancier un même élément de l'ontologie ou pour utiliser cet élément pour annoter sémantiquement une unité textuelle du document source. Par conséquent, il s'agit de calculer la complexité d'un domaine dans son ensemble comme suit :

$$\text{Complexité}_{\text{domaine}} = \frac{\sum_{i=1}^{I} A(\text{élément}_i) \times N(\text{élément}_i)}{\sum_{i=1}^{I} N(\text{élément}_i)}$$

où :

I est le nombre d'éléments de l'ontologie du domaine utilisés par OntoPop

A(élément$_i$) : ambiguïté de l'élément, c'est-à-dire le nombre de règles pouvant être déclenchées pour chaque élément identifié dans l'ontologie du domaine

N(élément$_i$) : nombre d'instances de chaque élément relevé dans le corpus

Figure 81. Définition de la mesure de complexité d'un domaine

Nous allons à présent passer à la présentation des expérimentations menées sur deux projets forts différents, tant en terme de corpus documentaire qu'en terme de besoins, ayant tous deux au moins atteint la phase de validation de la méthodologie OntoPop.

7.2 Les expérimentations

Pour chacune des applications d'OntoPop évaluées, nous allons tout d'abord brièvement rappeler le contexte et les objectifs du projet. Puis nous caractériserons aussi bien le corpus documentaire que l'ontologie avec ses différents éléments (classes, relations, attributs et rôles) du domaine concerné. Enfin, nous fournirons les résultats obtenus grâce à la présentation et à l'analyse des résultats des

mesures décrites ci-dessus pour l'évaluation (cf. annexe III pour le détail des résultats sur chaque document composant les corpus de validation dans l'un ou l'autre des projets).

7.2.1 Le projet « Presse People »

7.2.1.1 Description du projet

Ce projet a été réalisé pour le service de documentation d'un grand groupe de presse. Lorsque les journalistes écrivent un nouvel article pour une publication de ce groupe, dans quel que domaine que ce soit, ils appellent tout d'abord le service de documentation pour que les documentalistes leur fournissent de la matière soit sous forme de fiches signalétiques, soit sous forme d'articles déjà publiés et contenant l'information désirée. Le système existant reposait sur :

1) une lecture de presse quotidienne par les documentalistes de tous les articles publiés par le groupe mais aussi par les autres groupes de presse ;
2) une indexation et un classement manuels des articles potentiellement intéressants à l'aide de divers thésaurus et autres vocabulaires contrôlés ;
3) une mise à jour manuelle de la base de données recueillant les fiches signalétiques.

L'objectif du projet consiste donc à apporter une aide aux documentalistes en automatisant tout ou partie de ces trois étapes afin de les soulager dans ces tâches fastidieuses au quotidien. Ils peuvent alors se recentrer sur leur cœur de métier et répondre de manière précise et efficace aux demandes des journalistes. Le domaine plus particulièrement concerné par le projet est celui de la presse dite « People », c'est-à-dire concernant les personnalités qui font l'actualité. D'après notre client, ce marché est en plein essor dans la presse aujourd'hui et les demandes des journalistes y sont les plus fortes.

Dans la solution mise en place dans ce projet, les articles sont scannés ou bien récupérés à partir de flux numériques disponibles en partenariat avec les autres groupes de presse. Puis ils sont annotés sémantiquement par OntoPop qui enregistre aussi les nouvelles instances créées dans l'outil de représentation des connaissances ITM. Cet outil de représentation des connaissances a également pour objectif d'assister les documentalistes pour la gestion et la maintenance de leur base de connaissance ainsi que des différents thésaurus et vocabulaires contrôlés qui composent le référentiel du service. Cette maintenance doit permettre une meilleure évolutivité de la nouvelle solution par rapport à l'application existante qui était très rigide et très lourde. En guise d'illustration de la taille de la base de connaissance, rien que dans le domaine « People », elle contient 640.000 instances sur les personnalités, les événements, les sociétés, les œuvres (films, pièces de théâtre, livres, etc.) reliées entre elles par 600.000 instances de relations.

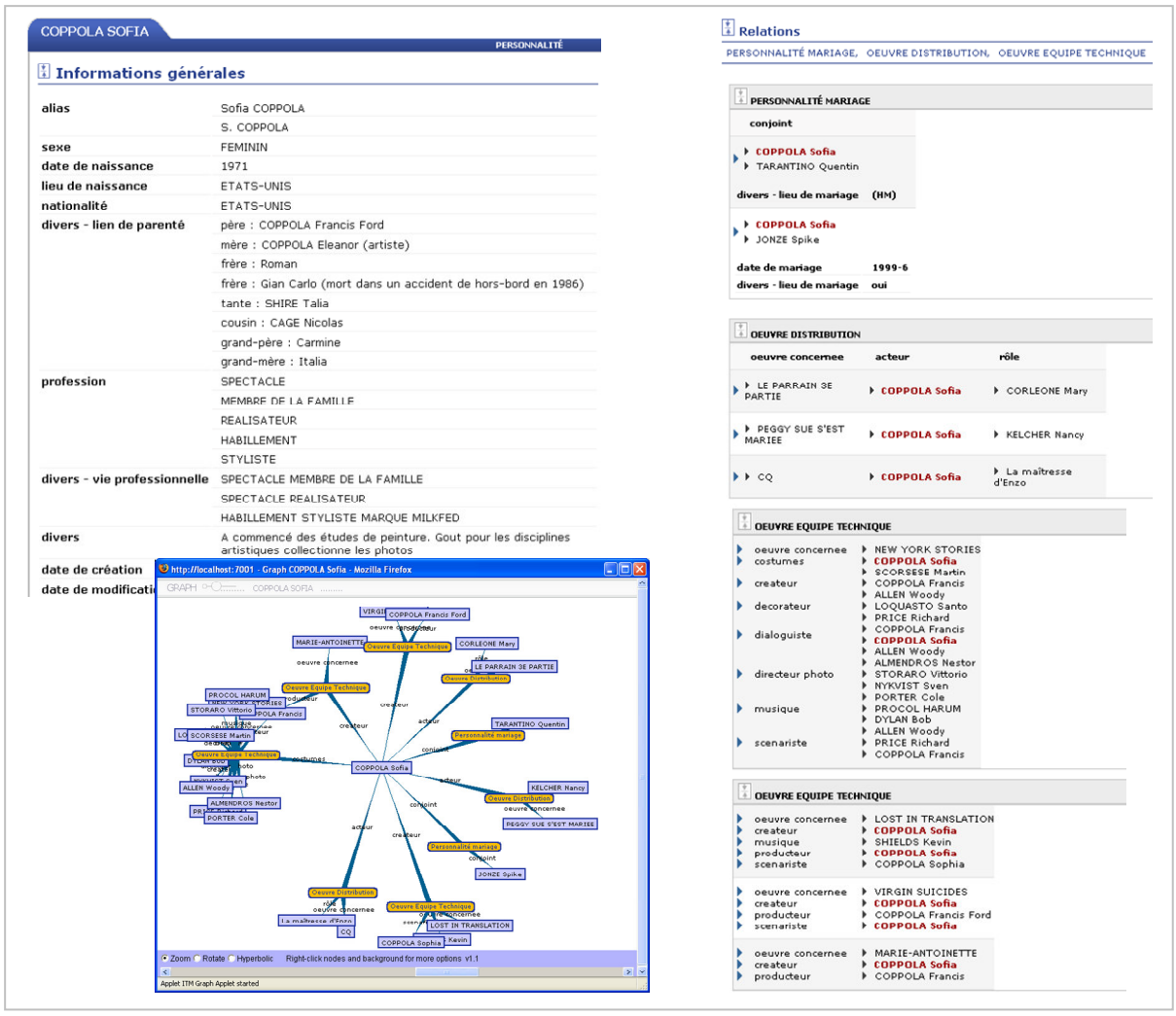


Figure 82. Exemple de fiche signalétique d'une instance, ici « Sofia Coppola », dans la base de connaissance d'ITM

La Figure 82 montre un exemple de fiche signalétique d'une personnalité, en l'occurrence « Sofia Coppola », dans la solution développée. Les documentalistes peuvent naviguer à travers la base de connaissance soit par cette interface, soit par l'interface graphique qui représente le réseau sémantique lié à cette instance (en bas à gauche de la Figure 82). Les documentalistes ont également à leur disposition dans cette nouvelle application plusieurs axes de recherches :

- sémantique (c'est-à-dire en fonction des attributs et relations modélisées dans l'ontologie sur une classe donnée),
- par extension (utilisation des thésaurus pour étendre la recherche aux synonymes et termes associés par exemple),
- par multicritères en fonction des annotations sémantiques modélisées dans l'ontologie du référentiel et contenues dans les ressources documentaires et, enfin,
- par mots-clefs sur le contenu du document.

Ces différents axes de recherche leur permettent de répondre au plus près de la demande du journaliste car ils peuvent jouer sur ces différents angles de vue pour obtenir l'information désirée soit dans la base de connaissance, soit dans la base documentaire. Enfin, les documentalistes peuvent

sauvegarder dans la base de connaissance l'historique de leurs recherches et les utiliser ou les réutiliser pour publier le dossier de recherche qu'ils enverront aux journalistes demandeurs.

7.2.1.2 Caractéristiques du domaine

Comme évoqué à la section 5.3, le corpus documentaire est constitué d'articles parus dans les magasines de la presse people. Ces articles ont pour caractéristiques communes d'être composés de phrases simples et courtes, dans un langage plutôt familier et courant. Il n'y a pas vraiment de jargon à proprement parler qui permettrait de désambiguïser certaines expressions et tournures langagières. Ce genre de vocabulaire familier associé à un manque de régularité dans les expressions constitue généralement un véritable challenge pour les outils d'extraction d'information. Enfin, les articles sont uniquement structurés selon leur titre, parfois un chapô, et quelques sous-titres.

Du côté de la modélisation, tout l'ancien système de documentation a dû être repris, et ceci a notamment entraîné certains choix de modélisation au niveau de l'ontologie de ce domaine comme de modéliser le « divorce» comme un attribut de la relation « Mariage ». Au final, l'ontologie du domaine comprend 16 classes dont 4 sont exploitées par OntoPop (Œuvre, Personnalité, Personnage et Article), 17 relations dont 6 exploitées par OntoPop (Mariage, Famille, Œuvre Equipe Technique, Œuvre distribution, Agent, Inspiration), 38 rôles dont 14 rôles exploités par OntoPop (parent, enfant, conjoint, agent, personne concernée, œuvre concernée, etc.) et 57 attributs dont 13 attributs exploités par OntoPop (date naissance, lieu naissance, lien parenté, date mariage, lieu mariage, indexation personnalité, indexation plan de classement, etc.). Il a fallu également retravailler et restructurer les différents thésaurus afin de les harmoniser, éviter les redondances et ôter les termes inutilisés ou inutilisables. L'application repose aussi sur deux thésaurus (un thésaurus géographique et un thésaurus de termes généraux), sur 20 tables de référence (comme les signes zodiacaux ou la nature d'une œuvre) et enfin sur un plan de classement des documents sur les personnalités comportant une vingtaine de rubriques comme interview, morphologie, goût, couple, etc. Ce plan de classement permet de ranger les articles dans des pochettes nommées en fonction de la personnalité et de la rubrique concernées.

7.2.1.3 Présentation des résultats d'OntoPop et discussion

Afin d'évaluer la performance d'OntoPop dans ce projet, j'ai constitué deux corpus :

- un premier corpus de tests unitaires composé de 18 documents dont 14 organisés autour de différents thèmes (Acteurs, Agents, Vie sentimentale, Enfants et Famille, etc.) et 4 autres représentant de vrais articles issus de la presse people.
- Un second corpus de validation composé d'un panel de 41 articles parus dans les magasines Elle en 2003, Paris-Match en 2002 et Journal du Dimanche en 2003 dont celui sur le clan Coppola qui nous a servi d'exemple tout au long de ce mémoire.

A partir du premier corpus, j'ai créé 100 Règles d'Acquisition de Connaissance pour les 37 éléments concernés par le peuplement d'ontologie ou l'annotation sémantique d'OntoPop. Ainsi, 2,7 règles en moyenne ont été créées pour chaque élément concerné de l'ontologie. Mais par exemple, 23 RACs

ont été définies pour évaluer le seul attribut d'indexation sur le plan de classement car ce dernier peut porter sur une vingtaine de rubriques différentes.

J'ai ensuite analysé le second corpus en fonction de ces Règles d'Acquisition de Connaissance. La complexité du domaine est évaluée à 4,38 règles déclenchées par élément concerné de l'ontologie. La précision du système mis en place est de 0,91 et le rappel de 0,95, ce qui donne un score de 0,93 pour la F-mesure. Ces résultats très bons doivent être mis en perspectives des deux tâches accomplies par le système, à savoir d'une part le peuplement d'ontologie et d'autre part l'annotation sémantique.

Type d'élément dans l'ontologie	Nombre d'éléments existants (A)	Nombre d'instances correctes (B)	Nombre d'instances acquises (C)	Rappel (B/A)	Précision (B/C)	F-mesure (2*R*P)/(R+P)
Classes	**7673**	**7527**	**7548**	**0,98**	**0,997**	**0,99**
Attributs	**117**	**115**	**151**	**0,98**	**0,76**	**0,86**
Relations	**108**	**97**	**125**	**0,90**	**0,78**	**0,83**
Rôles	**184**	**162**	**216**	**0,88**	**0,75**	**0,81**
Total				**0,94**	**0,82**	**0,87**

Tableau 7. Résultats de l'évaluation pour la tâche de peuplement d'ontologie

Le Tableau 7 présente l'ensemble des résultats obtenus pour le peuplement d'ontologie en fonction des 4 types d'éléments modélisés dans l'ontologie : les classes, les attributs, les relations et les rôles. Nous voyons très nettement que les éléments les plus instanciés dans la base de connaissance sont les instances de classes, correspondant aux Entités Nommées. Ce résultat n'est guère surprenant puisque, comme vu au Chapitre 2, la tâche de reconnaissance des Entités Nommées en Extraction d'Information est celle qui donne les meilleurs résultats, notamment car elle s'appuie très fortement sur les lexiques et autres vocabulaires existants. Le rappel et la précision d'acquisition de ces instances par les RAC sont très proches de 1 puisqu'elles sont de 0,98 et 0,997 respectivement. Concernant les instances d'attributs, ceux-ci obtiennent également un très bon rappel avec 0,98 mais la précision est nettement moins bonne à 0,76. En effet, l'analyse faire apparaître que certains attributs étaient acquis plusieurs fois par deux règles différentes. Pour les relations, le rappel descend à 0,90, ce qui reste un score tout à fait honorable, mais la précision atteint seulement 0,78, scores assez similaires de ceux des rôles avec un rappel à 0,88 et une précision à 0,75. Ces résultats s'expliquent notamment par la forte dépendance qui existe entre les relations et leurs rôles. De même que pour les attributs, certaines règles sont entrées en conflit et ont acquis la même information plusieurs fois, répercutant le phénomène à la fois sur la relation concernée et ses rôles. Mais au final, nous pouvons dire que la tâche de peuplement d'ontologie atteint de très bons scores : 0,94 de rappel et 0,82 de précision, surtout pour un domaine tel que celui de la presse people.

Quant aux résultats de l'annotation sémantique, dont les résultats figurent dans le tableau ci-dessous, ils sont encore meilleurs puisque le rappel est de 0,97 et la précision de 1. Cet excellent score est dû

au fait que toutes les informations porteuses d'annotations ont été exploitées à partir des arbres conceptuels des documents sources et que les annotations créées correspondent bien au type d'attribut modélisé dans l'ontologie. Le rappel est dû au fait que certaines règles n'étaient pas bien ajustées et qu'il y a eu de légères pertes d'information, notamment en ce qui concerne les rubriques du plan de classement des personnalités.

Nombre d'annotations existantes (A)	**Nombre d'annotations correctes (B)**	**Nombre d'annotations créées (C)**	**Rappel (B/A)**	**Précision (B/C)**	**F-mesure (2*R*P)/(R+P)**
5359	**5201**	**5201**	**0,97**	**1**	**0,98**

Tableau 8. Résultats de l'évaluation pour la tâche d'annotation sémantique

Concernant l'étape de consolidation du processus d'Annotation et d'Acquisition d'OntoPop, j'obtiens les résultats présentés aux Tableau 9 et Tableau 10. Les résultats d'ensemble pour les deux tâches sont assez similaires : 58% pour les instances et 56% pour les annotations consolidées, c'est-à-dire validées par le Module et créées respectivement dans la base de connaissance ou dans un document RDF annexe ; 39% d'instances et 34% d'annotations supprimées définitivement, généralement parce qu'il s'agit de doublons ; et enfin 3,25% d'instances et 9,63% d'annotations rejetées par le Module et mises de côté dans les tampons respectifs.

	nombre acquis (RAC)	nombre consolidés	% consolidés	nombre tampon	% tampon	nombre supprimés	% supprimés
classe	7548	4269	56,56	159	2,11	3120	41,33
attribut	151	131	86,75	20	13,24	0	0
relation	125	**85**	68	**40**	32	0	0
rôle	216	**171**	79,17	**42**	19,44	3	1,39
Ensemble	8040	4656	57,91	261	3,25	3123	38,84

Tableau 9. Résultats de la consolidation pour la tâche de peuplement d'ontologie

Par contre, on voit bien que dans le cas du peuplement d'ontologie, la situation diffère très largement selon le type d'élément dont il s'agit dans l'ontologie. En effet, les instances de classes sont pour la moitié supprimées et très peu sont mises dans le tampon. Ceci est expliqué par le fait que les instances de classes concernent majoritairement les entités nommées, notamment des personnalités, et que celles-ci étant citées plus d'une fois dans un même article sont acquises autant de fois par les RAC et il faut donc éliminer tous les duplicatas. D'autre part, celles qui sont rejetées dans le tampon sont celles qui peuvent être rapprochées de plusieurs instances dans la base de connaissance, notamment par le jeu des homonymes, et le système ne peut décider seul auquel des homonymes l'instance fait référence. Les instances d'attributs, de relations et de rôles sont le plus souvent consolidées, très peu sont supprimées. Cela est un peu moins vrai pour les relations, qui sont plus fréquemment rejetées par le système dans le tampon. Ceci est dû au fait que les relations constituées d'un seul rôle sont rejetées par le système : les relations doivent au moins faire intervenir deux instances de classes dans deux rôles. Par exemple, une relation de Mariage ne peut exister que si deux rôles de conjoints joués par des Personnalités différentes ont été également acquis par les RAC au cours du processus d'Acquisition.

	nombre acquis (RAC)	nombre consolidés	% consolidés	nombre tampon	% tampon	nombre supprimés	% supprimés
Annotations	5201	2941	56,55	501	9,63	1759	33,82

Tableau 10. Résultats de la consolidation pour la tâche d'annotation sémantique

- **Comparaison des résultats produits par OntoPop vis-à-vis des annotations produites par des annotateurs humains**

L'intégration de la solution au sein du service documentation du groupe a soulevé des problèmes et a empêché toute évaluation des interfaces de validation auprès des utilisateurs finaux de l'application, c'est-à-dire des documentalistes. Par contre, j'ai pu mener une étude intéressante avec une classe de 20 étudiants en Master 2ème année en Information et Communication. Je les ai formés durant 4h à la représentation des connaissances par des ontologies et à l'annotation sémantique de ressources documentaires. A la suite de cette formation, je leur ai demandé d'annoter l'article « Le Clan Coppola » en fonction d'un ensemble d'éléments issus de l'ontologie du projet de la presse People. Les consignes étaient les suivantes : « Voici ci-dessous un extrait d'une ontologie relative au domaine de la presse People :

Classes	**Relations**	**Attributs**
Personnalité (**P**)	Mariage (**RM**)	Parenté (**AP**)
Personnage de Film (**PF**)	Naissance (**RN**)	Lieu (**AL**)
Œuvre (non film) (**O**)	Filmographie (**RF**)	Date (**AD**)
Film (**F**)	Bibliographie (**RB**)	
Société (**S**)	Inspiration (**RI**)	
	Goût (**RG**)	
	Nécrologie (**RNé**)	
	Santé (**RS**)	

Vous devez annoter l'article ci-dessous avec les classes, les relations et les attributs correspondant à l'ontologie décrite ci-dessus. Vous pourrez soit baliser le texte soit le souligner en prenant bien soin d'indiquer l'élément de l'ontologie utilisé. Un exemple vous est donné : <RN><P>Francis Coppola</P> naît le <AD>7 avril 1939</AD> à <AL>Detroit</AL>, dans le <AL>Michigan</AL></RN>

N'oubliez pas de noter le temps que vous avez mis pour annoter l'ensemble du document. »

	Ontologie	**Moyenne des étudiants**	**OntoPop**
Classes	Personnalité	119,75	135
	Personnage	3,53	
	Œuvre	4,25	7
	Film	24,6	
	Société	4,41	
Relations	Mariage	9	3
	Naissance	2	1
	Filmographie	8,53	1
	Bibliographie	2,7	
	Inspiration	2,54	
	Goût	3,63	
	Nécrologie	2,58	
	Santé	1,8	

Attributs	Parenté	35,55	7
	Lieu	17,7	1
	Date	10,6	
Temps mis (en minutes)		**33**	**0,32**

Tableau 11. Résultats de la comparaison de l'étude menée avec les étudiants

Le Tableau 11 nous présente une synthèse de cette étude (cf. Annexe 3). En moyenne, les étudiants ont passé 33 minutes à annoter contre 32 secondes pour OntoPop. L'annotation des étudiants est bien plus riche que celle fournie par OntoPop, même si l'acquisition des personnalités, des œuvres, des mariages et des naissances est assez similaire. Ceci est dû au fait que le moteur d'extraction est capable de bien identifier ces concepts dans les articles du domaine. Pour les autres relations et attributs, l'extraction est plus difficile car les expressions langagières utilisées par les auteurs de ces articles sont moins régulières et donc moins bien implémentées dans les patrons d'extraction.

Néanmoins, l'étude nous montre également que les étudiants entre eux ne sont pas forcément d'accord sur l'annotation du document à partir d'un même élément de l'ontologie. Par exemple, certains ont annoté 162 personnalités alors que d'autres seulement 21. Ou encore, les étudiants ont annoté des termes par plusieurs annotations différentes, comme le cas du syntagme « le parrain », coréférence nominale désignant « Francis Ford Coppola » en référence à son œuvre, qui a été annoté par les classes « Personnalité », « Personnage » et « Film », par la relation « Filmographie » et encore par l'attribut « Parenté ». D'ailleurs la capacité des annotateurs humains à résoudre les anaphores expliquent aussi en partie le nombre plus important de leurs annotations comparé au système d'extraction d'information qui n'est pas capable de récupérer ces informations implicites. Et enfin, certains étudiants ont annoté des concepts, ou plutôt des sous-concepts, comme des instances d'une classe ce qui gonfle également les résultats obtenus par les annotateurs humains. C'est par exemple le cas des termes « court-métrage » ou « making-of » qui, pour certains, représentaient des instances de la classe « Film » comme les vraies instances issues d'entités nommées telles « Le Parrain » ou « Virgin Suicides ».

En conclusion, OntoPop, et notamment le système de Règles d'Acquisition de Connaissance mis en place dans ce projet, est très performant même si un annotateur humain annoterait bien d'autres informations que celles disponibles dans les arbres conceptuels fournis par le moteur d'extraction utilisé. D'ailleurs, je n'ai pas pris en compte la pertinence des résultats des extractions de ce moteur qui ne fournit pas toujours des informations correctes dans l'arbre conceptuel. La performance du moteur d'extraction a plutôt une F-mesure de 0,50 que de 0,80 comme cela devrait être le cas dans une application en entreprise. Ceci est dû à la forte ambiguïté présente dans le contenu des articles de la Presse People. Mais l'utilisation des RAC et leur flexibilité permet de s'adapter à ces résultats et d'éliminer un certain nombre d'incohérences. Par exemple, elles ne permettent pas d'acquérir une relation « Famille », constituée d'un rôle « parent » et d'un rôle « enfant », s'il manque un seul de ces deux rôles. De même, un attribut ne peut être assigné à une personne que si cette dernière est identifiée soit par un lexique soit par un patron d'extraction qui se base sur la reconnaissance d'un

prénom accompagné ou non d'un nom de famille. Enfin, pour conclure et donner un ordre d'idée sur l'implémentation des RAC, leur création dans la phase de couplage de ce projet a nécessité moins d'une semaine de travail.

7.2.2 Le projet « Edition juridique »

7.2.2.1 Description du projet

Ce projet a été initié à la demande d'un grand groupe français de l'édition juridique qui publie régulièrement une vaste gamme de produits d'information juridique de référence (des encyclopédies mises à jour, des revues et des CD-Roms) et qui propose également un service d'information en ligne sur Internet. Il met notamment à disposition des professionnels du droit toutes les décisions de jurisprudence depuis 1960. Ce projet a été mis en place avec les objectifs suivants :

- faciliter l'écriture de nouveaux articles dans les revues du groupe, et plus précisément la rédaction des références aux textes de lois, codés ou non codés, aux décisions, aux autres publications officielles car ces références sont extrêmement codifiées et doivent donc respecter une certaine norme. Ces références sont balisées et annotées dans l'article (cf. Figure 83) en fonction de leurs propriétés comme le nom du code ou du texte de loi, sa date de publication, son émetteur, etc.
- aider à l'acquisition des nouvelles décisions de jurisprudence dans leur base de connaissance en étant capable d'identifier automatiquement chaque propriété identificatoire de cette décision comme sa juridiction, son siège, sa formation, la date de la décision, la décision elle-même, les parties respectives, etc.

L'idée derrière ce projet est de fournir une unique source d'information, le référentiel ITM, qui permette de contrôler la qualité des nouvelles décisions acquises ainsi que des renvois, c'est-à-dire les références aux différentes informations juridiques, d'enrichir les nouveaux articles avec ces informations de manière standardisée et enfin de créer du nouveau contenu à partir de ces dernières. Le projet est divisé selon ces deux axes : l'identification des propriétés décrivant les nouvelles décisions de jurisprudence et le balisage des renvois aux textes juridiques en fonction de leurs propriétés particulières.

Pour ce faire, nous avons modélisé une ontologie pour le domaine de l'Edition Juridique avec l'aide du client, développé un module d'assistance au balisage des renvois juridiques et implémenté OntoPop pour répondre aux besoins des deux axes pré-cités avec un contrôle très fin de la structure interne de la référence, aussi bien pour un renvoi dans un article que pour une nouvelle décision de jurisprudence. Pour le client, ce projet permet une meilleure productivité, une meilleure qualité de ses références juridiques, et donc de ses articles, et enfin une valeur ajoutée certaine sur ses produits finaux comme la recherche de décisions avec son service d'information en ligne.

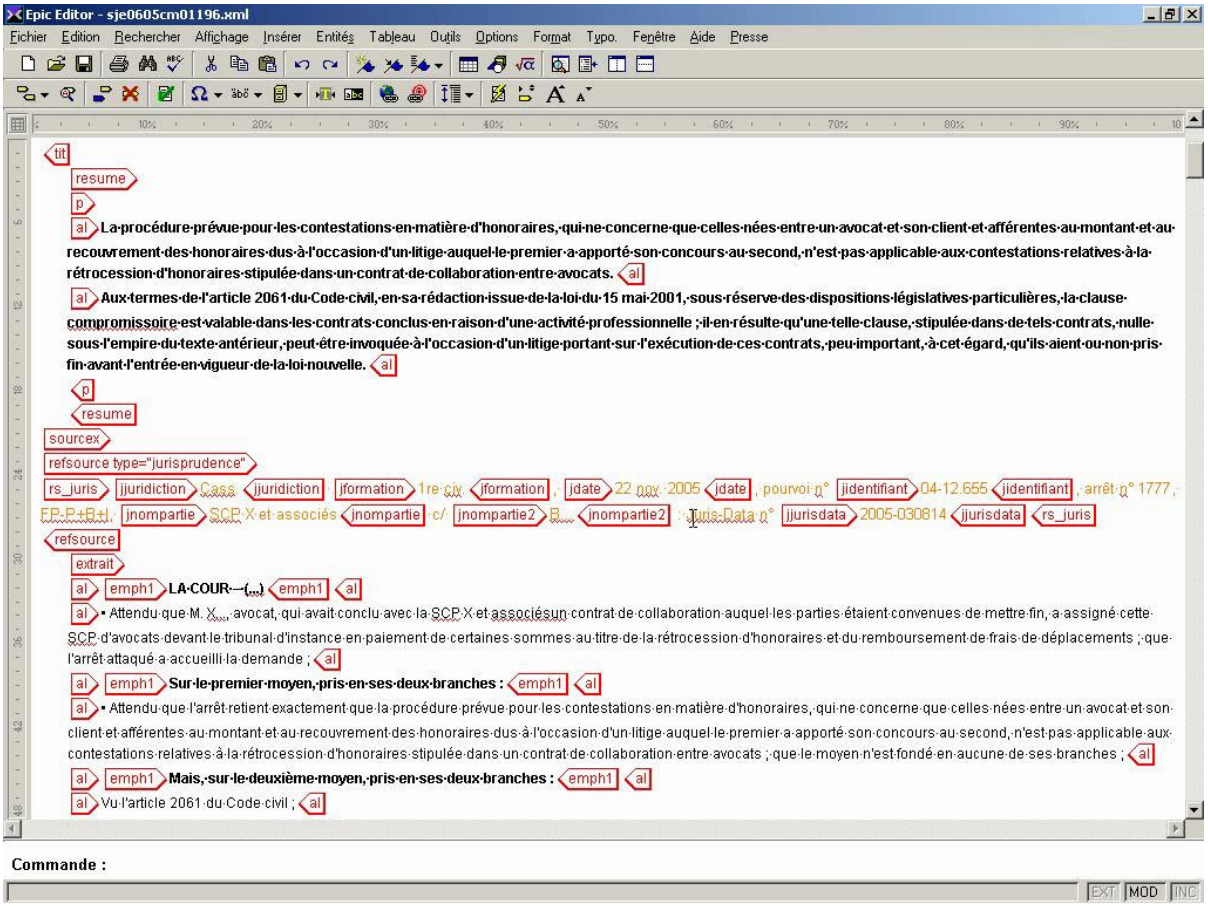


Figure 83. Exemple d'un balisage d'un renvoi juridique dans une décision de jurisprudence à l'aide d'OntoPop

7.2.2.2 Caractéristiques du domaine

Le corpus documentaire de ce projet est constitué d'un côté des renvois aux textes juridiques et de l'autre des décisions de jurisprudence. Les renvois sont constitués de libellés plus ou moins codifiés qui correspondent généralement à des termes juridiques très précis, même si ces derniers peuvent parfois être ambigus car un même terme, une même abréviation peut renvoyer à plusieurs significations. En général, ces renvois sont relativement courts (entre une à deux lignes) comme par exemple : « *Déc. n° 2241/2004/CE, 15 déc. 2004* » ou « *L. fin. pour 2005, n° 2004-1484, 30 déc. 2004, art. 9, I, 2° et III* ».

Les décisions de jurisprudence, comme évoqué à la section 5.3, sont des textes de plusieurs pages structurés en deux parties bien distinctes : un entête composé de toutes les propriétés identificatoires de la décision (la juridiction, la date, la formation, etc.) suivi du corps du document narrant, en une seule phrase, la présentation et les argumentaires des différentes parties jusqu'à la prise de la décision finale. Par conséquent, mis à part la décision qui se trouve en fin de document, toutes les propriétés identificatoires qui nous intéressent pour l'acquisition et l'enrichissement de la base de connaissance, se trouvent dans l'en-tête structuré, ce qui facilite grandement la tâche du moteur d'extraction. Néanmoins, là encore, beaucoup de précautions sont à prendre car cela ne veut pas dire qu'il n'existe pas d'ambiguïtés et, surtout, de cas particuliers possibles à prendre en compte pour l'extraction des informations.

L'ontologie a été réalisée en fonction de l'expertise du client dans ce domaine et de ses besoins propres. Outre la classe correspondant aux articles, nommée « Content Unit », il existe deux ensembles de classes : les « Références A » qui correspondent aux renvois à un texte juridique existant et les « Références DE » qui correspondent à la description des différents textes juridiques. La distinction repose essentiellement aussi sur les attributs qui seront plus précis pour les « Références DE » que pour les « Références A ». Chaque ensemble possède :

- des références législatives, i.e. « Référence A Législation » et « Référence DE Législation », qui se subdivisent en textes codés, ou « TC », et en textes non codés, ou « TNC » ;
- des références de publication, i.e. « Référence A Publication » et « Référence DE Publication » ;
- des références à la jurisprudence, i.e. « Référence A Jurisprudence » et « Référence DE Jurisprudence ».

Les classes utilisées par OntoPop pour le balisage des renvois correspondent à toutes les sous-classes « Référence A » et la classe utilisée pour l'identification des décisions de jurisprudence est bien sur « Référence DE Jurisprudence ». Chacune de ces classes possède des attributs particuliers décrivant chacun des propriétés identificatoires de la référence en question : son nom, son texte, sa date de publication, son émetteur, la décision rendue, etc. Enfin, trois relations sont utilisées dans OntoPop qui permet de relier la référence en question au document d'origine, que ce soit l'actuel document de la décision de jurisprudence dans un cas ou l'article en cours de rédaction dans l'autre cas.

Enfin, il est important de préciser que le référentiel ITM de ce projet contient également un thésaurus de termes juridiques utilisés pour l'indexation des articles publiés par le groupe, mais surtout un ensemble de tables de référence comprenant les entités autorisées pour la plus grande partie des propriétés identificatoires. Par exemple, il existe une table de référence des types de juridiction pour les décisions de jurisprudence recensant l'ensemble des libellés autorisés pour décrire une juridiction. Seules les « formations juridiques » peuvent être créées automatiquement à la volée tout en respectant une norme bien précise qu'il est possible de préciser grâce au paramétrage de la sortie des Règles d'Acquisition de Connaissance.

7.2.2.3 Présentation des résultats d'OntoPop et discussion

Les corpus d'évaluation d'OntoPop ont été constitués en fonction des deux axes du projet. Ainsi, j'ai formé deux corpus de tests, l'un pour le balisage des renvois et l'autre pour l'identification des jurisprudences. Le corpus de test pour le balisage est constitué de 3 documents : l'un pour le balisage des renvois de jurisprudence, le second pour le balisage des renvois législatifs et de publication et le dernier contenant des faux exemples ne devant pas générer de balisage. Les deux premiers documents contenaient autant de références que nécessaire pour couvrir tous les cas possibles pouvant être rencontrés pour le balisage des renvois. Le corpus de test pour l'identification est composé de 57 décisions de jurisprudence délivrées par des juridictions différentes, là aussi pour couvrir un maximum de cas possibles dans l'identification de ses propriétés identificatoires.

De même pour les corpus de validation, j'ai formé deux groupes. Le corpus de validation du balisage est formé de 4 documents regroupant plus de 150 renvois à de la jurisprudence, à de la législation et à des publications officielles. Le corpus de validation de l'identification est constitué de 89 décisions de jurisprudence délivrées également par diverses juridictions.

L'ensemble des documents composant ces différents corpus de test et de validation a été donné par le client qui a aussi fourni les attendus pour chacun d'entre eux. J'ai donc eu à ma disposition un étalon (ou « gold standard ») [HAB 05] qui m'a permis d'évaluer objectivement chacun des résultats générés par OntoPop sur ces différents corpus. Cela a beaucoup aidé à la définition des RACs mais aussi lors de l'évaluation finale.

A partir des deux corpus de test, j'ai créé 148 Règles d'Acquisition de Connaissance : 15 pour les 9 classes utilisées par OntoPop, 116 pour les 27 attributs utilisés, 8 pour les 3 relations et enfin 9 pour les 6 classes. Ainsi, même si en moyenne, 4 RAC ont été créées par élément de l'ontologie concerné par le projet, on voit bien que l'effort a avant tout été mis sur l'acquisition des attributs correspondant aux différentes propriétés identificatoires des renvois comme des décisions de jurisprudence. Comparé au projet sur la presse People où 2,7 RAC ont été créées en moyenne par élément, nous nous rendons compte ici de la complexité imposée par ce domaine de l'Edition Juridique. Et pourtant, la complexité du domaine pour l'ensemble des corpus de validation est évaluée à 4,48 règles déclenchées par élément concerné, ce qui est quasi-similaire à la complexité du domaine de la presse people qui, nous le rappelons, est de 4,38 Règles déclenchées par élément. Par ailleurs, j'ai évalué l'ensemble de la précision d'OntoPop pour ce projet à 0,99 et le rappel à 0,99 également. Ces résultats, encore meilleurs que ceux obtenus pour la presse people, sont notamment dus à la forte normalisation du domaine ainsi qu'à son jargon spécifique que les outils d'extraction d'information maîtrisent mieux. Mais présentons à présent les résultats obtenus pour le balisage et ceux de l'identification.

Type d'élément dans l'ontologie	Nombre d'éléments existants (A)	Nombre d'instances correctes (B)	Nombre d'instances acquises (C)	Rappel (B/A)	Précision (B/C)	F-mesure (2*R*P)/(R+P)
Classes	**157**	**150**	**157**	**0,955**	**0,955**	**0,955**
Attributs	**585**	**585**	**585**	**1**	**1**	**1**
Relations	**150**	**150**	**150**	**1**	**1**	**1**
Rôles	**300**	**300**	**300**	**1**	**1**	**1**
Total				0,988	0,988	0,988

Tableau 12. Résultats de l'évaluation pour le balisage des renvois juridiques

Contrairement au domaine de la presse people, ce qui frappe dans le Tableau 7 est le nombre important d'acquisitions d'instances d'attributs comparé au nombre d'instances de classes. C'est que dans ce domaine, il n'y a pas véritablement d'Entités Nommées à récupérer. Chaque instance de classe correspond à un renvoi particulier, soit de législation, soit de jurisprudence, soit encore de

publication officielle. A chaque renvoi reconnu, une relation est instanciée entre celui-ci et l'instance de la classe « Content Unit » représentant l'article en cours de rédaction. Et comme cet article et le renvoi jouent chacun un rôle, ceci explique pourquoi il y a autant de relations que d'instances de classes acquises et leur double de rôles. Par contre, bien évidemment, le plus intéressant à regarder sont les résultats des attributs qui rendent compte à la fois d'une grande précision et d'un rappel conséquent puisque chacun de ces chiffres atteignent un score de 1. En effet, après avoir attentivement analysé le corpus de test présentant tous les cas possibles pour la création des RACs, on peut à présent constater que celles-ci rendent compte parfaitement de tous ces cas en situation réelle. Seules les instances de classes ont généré quelques erreurs dues au fait que ces instances n'ont pas été acquises dans la bonne classe, mais dans la super-classe. En effet, ces instances sont des articles législatifs et devraient donc faire partie de la classe « Référence A Législation TNC article » alors qu'elles ont été instanciées au niveau de la classe « Référence A Législation TNC ». Cette affectation n'est pas foncièrement fausse puisqu'il s'agit tout de même de références législatives, mais l'objectif est que le système soit le plus précis possible pour répondre au plus près des demandes des utilisateurs finaux lorsque qu'ils vont rechercher des articles contenant ces renvois. La tâche de balisage des renvois obtient tout de même de très bons scores : 0,988 de rappel et autant pour la précision.

Type d'élément dans l'ontologie	Nombre d'éléments existants (A)	Nombre d'instances correctes (B)	Nombre d'instances acquises (C)	Rappel (B/A)	Précision (B/C)	F-mesure (2*R*P)/(R+P)
Classes	**178**	**178**	**178**	**1**	**1**	**1**
Attributs	**862**	**847**	**847**	**0,9825986**	**1**	**0,9912229**
Relations	**89**	**89**	**89**	**1**	**1**	**1**
Rôles	**178**	**178**	**178**	**1**	**1**	**1**
Total				0,996	1	0,998

Tableau 13. Résultats de l'évaluation pour l'identification des décisions de jurisprudence

Pour la tâche d'identification, les résultats sont en général assez similaires puisque l'on observe les mêmes phénomènes, notamment au sujet de l'acquisition des attributs. Mais les erreurs générées par les RAC d'OntoPop se situent non plus au niveau des instances de classes qui sont correctement identifiées mais au niveau de certains attributs. En fait, après analyse, ils concernent un attribut qui est la date de décision qui n'est pas rédigée de la même manière selon la juridiction de provenance de la décision. Ceci entraîne des changements dans l'arbre conceptuel généré qui ne sont pas pris en compte par les RAC. Mais les résultats d'ensemble de cette tâche sont encore meilleurs que ceux de la tâche de balisage avec un rappel à 0, 996 et une précision à 1. L'exercice d'identification des décisions est donc quasi-parfait et ceci notamment grâce à la forte structuration des en-têtes de ces documents.

Par contre, ces deux tâches étant entièrement automatisées, je n'ai pas évalué l'interface de validation puisque cette dernière n'est pas utilisée dans le projet. Et concernant l'évaluation de la consolidation des résultats fournis par les RAC, ces résultats sont déjà si précis que l'étape de consolidation ne rejette rien dans le tampon. Pourtant, l'algorithme vérifie en profondeur que tout est bien correct et valide vis-à-vis du référentiel ITM et, plus particulièrement, auprès des tables de référence. Puis il envoie les annotations pour le balisage directement à l'interface d'édition des nouveaux articles ou crée la référence à la nouvelle décision avec toutes ses propriétés identificatoires dans la base de connaissance.

Pour conclure, sur ce domaine bien particulier d'Edition Juridique qui est pourtant très complexe à manipuler et très rigide, les résultats d'OntoPop sont extrêmement encourageants. Ces résultats sont aussi dus à la précision de chaque corpus de tests vis-à-vis de la modélisation de tous les cas possibles et ils sont fort nombreux. Mais la capacité d'adaptation des RAC permet de prendre en compte chacun de ces cas et de les implémenter dans OntoPop sans pour autant générer beaucoup de conflits et d'erreurs entre toutes ces règles. Par ailleurs, ce qui prend du temps à l'intégrateur, c'est l'analyse des arbres conceptuels de tous ces tests unitaires, mais la création des RAC se fait assez rapidement. En tout, il m'a fallu environ deux semaines pour réaliser la phase de couplage pour ce projet.

7.3 Réflexions sur l'évaluation des systèmes d'annotation sémantique ou de peuplement d'ontologie

Les résultats que nous venons de présenter pour ces deux projets sont excellents. Mais du coup, ils nous poussent à nous interroger sur la validité de ces mesures pour évaluer un système d'annotation sémantique ou de peuplement d'ontologie tel qu'OntoPop. Les mesures de précision et de rappel que nous avons utilisées ont récemment fait l'objet de remises en question, aussi bien au sein du domaine de l'Extraction d'Information [LAV 04] que dans celui de l'annotation sémantique [MAY 05b]. En effet, l'un des principaux reproches mentionnés par rapport à ces mesures concerne leur rigidité puisqu'un résultat est considéré comme correct ou incorrect. Il n'y a aucune place pour des résultats partiels, c'est-à-dire qui peuvent être en partie corrects. Par exemple, Lavelli et al. [LAV 04] soulèvent la question suivante : une information extraite doit-elle être considérée comme fausse si elle contient un fragment non pertinent comme une virgule ? Pourtant en 1998, Freitag [FRE 98] avait proposé trois critères concernant la qualité des informations extraites : **exacte**, i.e. l'information extraite correspond parfaitement à l'information attendue ; **contenue**, i.e. l'information attendue est contenue dans l'information extraite ; **imbriquée**, i.e. l'information attendue dépasse de l'information extraite. Chacun de ces critères peut être utilisé pour observer la performance, mais aucune mesure standardisée et reconnue par la communauté, comme le rappel ou la précision utilisés dans ce mémoire, ne les exploite vraiment.

Dans le domaine de l'annotation sémantique, nous avons pourtant affaire au même cas de figure et cela devient encore plus crucial que pour l'extraction d'information. En effet, comme le souligne Maynard dans [MAY 05b], les besoins pour l'annotation sémantique et le peuplement d'ontologie sont plutôt différents : les mesures doivent pouvoir évaluer sur une échelle les différents degrés de pertinence des annotations ou des instances en fonction de leur affectation de la bonne classe ou de la bonne propriété dans l'ontologie de référence. En effet, comme nous l'avons vu dans le domaine de l'Edition Juridique, annoter un article de renvoi législatif comme une « Référence A Législation TNC » plutôt que comme sa sous-classe « Référence A Législation TNC article » n'est pas incorrect mais plutôt partiellement correct. De plus, cette annotation est même plus correcte que si ce renvoi avait été annoté comme une « Référence A Publication » qui n'a rien en commun avec une référence législative. Par conséquent, une adaptation des mesures de précision et de rappel est absolument nécessaire pour tenir compte de la position de l'annotation ou de l'instance dans l'ontologie ainsi que sa proximité avec la bonne position dans l'ontologie. Malheureusement, jusqu'à présent, les quelques tentatives, comme celle de Maynard, de proposer une nouvelle mesure tenant compte de ces nouveaux critères n'ont pas trouvé de consensus qui permettrait de remplacer les mesures utilisées actuellement et de pouvoir comparer un système avec un autre. La communauté doit donc absolument se mobiliser pour fournir une méthodologie ainsi qu'un ensemble de mesures pour une évaluation plus adéquate des outils d'annotation sémantique et de peuplement d'ontologie dans le cadre du Web Sémantique.

Une première tentative pour rassembler les chercheurs autour de cette problématique a pourtant eu lieu en juin 2006 lors du Worshop "Mastering the Gap from IE to Semantic Representation »[39] qui s'est tenu en parallèle de la conférence européenne pour le Semantic Web (ESWC'06). Il a été convenu qu'il était important de fournir des mesures standards pour l'évaluation ainsi qu'un ensemble de corpus étalons qui permettrait de faire progresser la recherche, en s'inspirant de ce qui a été fait dans les conférences MUC ou TREC. Mais aucune suite n'a encore été donnée à cette réflexion. Toutefois, nous espérons pouvoir participer à une initiative dans les prochains mois.

Enfin, s'il est important de pouvoir évaluer les performances de son propre système, il est tout aussi intéressant de le comparer à d'autres outils. Or, là encore il n'existe pas vraiment de critères standards avec un système de notation. Pour autant, Maynard [MAY 05b] et Sazedj [SAZ 05] ont proposé un ensemble de critères se recoupant ainsi :

- les fonctionnalités attendues (annotation et/ou peuplement, niveau d'automatisation, le temps de traitement, la consolidation des annotations et des instances, etc.),
- l'interopérabilité (avec d'autres outils ou avec des utilisateurs différents, le moyen de stockage, les langages utilisés, etc.),
- la convivialité (facilité d'installation, de prise en main des interfaces, mise à disposition de manuels pour les utilisateurs et les développeurs, etc.), l'évolutivité (adaptation à de nouveaux besoins ou utilisateurs, gestion des référentiels, etc.) et enfin,

[39] Site web du Workshop : http://tev.itc.it/mtg.html

- la réutilisation (dans de nouveaux domaines, avec de nouveaux corpus documentaires, avec de nouvelles ontologies).

Sazedj a notamment essayé de quantifier chacun de ses critères pour donner une note globale aux systèmes évalués. Mais cette quantification des critères n'est pas toujours bien définie ou bien se base sur une opinion subjective. Par conséquent, elle ne nous semble pas forcément évidente à mettre en pratique pour comparer différents outils. Néanmoins elle a le mérite de fournir une base de réflexion quant à ce qui pourrait être amélioré pour obtenir une vraie grille d'évaluation.

Nous pouvons tenter d'évaluer rapidement OntoPop en fonction des quelques facettes énumérées ci-dessus. OntoPop fournit l'ensemble des fonctionnalités attendues pour un tel système : il propose au choix l'annotation sémantique des ressources documentaires et/ou le peuplement d'ontologie, de manière automatisée ou semi-automatisée pour correspondre au plus près des besoins de l'application. Il fournit aussi un algorithme de consolidation très sophistiqué qui vérifie les contraintes aussi bien au niveau du modèle qu'au niveau du contenu du référentiel pour les futures annotations et instances. Grâce à ses APIs, il est interopérable avec toute application cliente. Il est d'ailleurs utilisé dans des applications très diverses. Il s'intègre parfaitement à l'outil de représentation des connaissances ITM et permet un couplage avec tout outil d'extraction d'information par l'intermédiaire de plug-ins. Il est basé sur l'utilisation des langages recommandés par le W3C comme XML, RDF et OWL ainsi que sur le standard des Topics Maps. Son intégration à ITM lui permet également de s'adapter aux besoins changeants des applications clientes comme des utilisateurs et de faciliter la gestion et la maintenance des différents référentiels. Enfin, OntoPop est utilisé dans divers projets, de divers domaines, chacun ayant leurs besoins spécifiques. Sa méthodologie, présentée au chapitre 5, permet de l'adapter rapidement à ces nouveaux besoins. Le temps d'implémentation de la phase de couplage est ainsi réduit : entre quelques jours seulement pour les projets les plus simples à deux semaines pour les plus complexes. D'ailleurs la flexibilité procurée par les Règles d'Acquisition de Connaissance permet de l'adapter à n'importe quel domaine, à n'importe quel arbre conceptuel et à n'importe quelle ontologie, pourvu qu'ils respectent les langages et formats standards.

7.4 Conclusion

D'après les premiers projets validés par les clients, OntoPop fournit un excellent système de couplage entre un moteur d'extraction et un outil de représentation des connaissances. En effet, quel que soit le domaine, plus ou moins complexe, plus ou moins ambigu, sa précision et son rappel dépassent les 90% de performance. Malgré tout, nous jugeons ces mesures insuffisantes ou dépassées pour réellement évaluer un outil d'annotation sémantique et de peuplement d'ontologie. Elles doivent être adaptées à la réalité pour mieux saisir la complexité des tâches réalisées par ce genre d'outil. A ce sujet, il reste encore beaucoup à faire et la communauté toute entière doit se mobiliser pour fournir des critères et des mesures d'évaluation qui puissent devenir des standards. L'obtention de ces standards facilitera alors la comparaison entre les différents outils, leurs ressemblances et leurs traits caractéristiques. Ceci peut être important selon l'application devant être réalisée, notamment dans le

milieu des entreprises qui demande beaucoup de rigueur et de précision. Nous allons, au chapitre suivant, aborder le bilan d'OntoPop et ses perspectives futures.

Chapitre 8. Bilan et perspectives d'évolution pour OntoPop

Nous avons présenté tout au long de ce mémoire une solution pour l'annotation sémantique et le peuplement d'ontologie qui repose sur la création de Règles d'Acquisition de Connaissance. Les résultats des évaluations, menées sur deux projets différant sur leurs besoins et les caractéristiques de leurs domaines respectifs, sont particulièrement bons. Les RAC fournissent une solution qui s'adapte facilement aux besoins et spécificités de chacun des projets rencontrés jusqu'à présent auprès de nos clients. Le projet concernant la presse People est actuellement utilisé quotidiennement par les documentalistes du groupe et le projet de l'Edition Juridique va être mis en service début juillet 2007. Nous avons réalisé plusieurs démonstrateurs pour différents appels d'offres qui ont remporté un certain succès et donné suite à la réalisation du projet. Nous sommes donc très satisfaits de la réussite d'OntoPop auprès des entreprises. Mais contrairement aux outils d'annotations sémantiques de l'état de l'art, notre solution s'adresse avant tout à ces entreprises qui ont un fort besoin en gestion documentaire et en gestion de la connaissance. OntoPop ne cherche pas à apporter une réponse générique à l'annotation du Web Sémantique comme celle proposée par SemTag, KIM ou OntoMat-Pankow. Par ailleurs, comme nous allons le voir dans la suite de ce chapitre, OntoPop est limité par certains aspects des Règles d'Acquisition de Connaissance. Mais nous entrevoyons une nouvelle perspective pour OntoPop grâce à l'alignement d'ontologies. Nous pensons que l'utilisation de ses méthodes et outils pourraient nous aider à résoudre en partie les points faibles que nous allons à présent décrire.

8.1 Les limites d'OntoPop

A travers les expérimentations menées sur les deux projets présentés lors du chapitre précédent, mais également sur les autres projets en cours, nous avons dégagé un certain nombre de limites que l'on peut séparer en deux catégories : les problèmes liés à la définition des Règles d'Acquisition de Connaissance et les problèmes liés à leur déclenchement dans le Module d'Annotation et d'Acquisition. Nous présentons ces différentes problématiques en les illustrant à partir d'exemples issus de l'analyse de l'article « Le Clan Coppola ». Chaque problème soulevé est accompagné d'une proposition de solution. Quelques unes de ces solutions nécessiteraient notamment une recherche linguistique plus approfondie.

8.1.1 Problèmes liés à la définition des Règles d'Acquisition de Connaissance

Les problèmes liés à la définition des Règles d'Acquisition de Connaissance concernent le format des informations extraites par les outils linguistiques, ou la précision de l'information ou encore la proximité de cette information. Nous détaillons dans la suite ces trois sortes de problèmes.

8.1.1.1 Le format des informations extraites

Les informations extraites par les outils linguistiques ne peuvent pas être stockées dans la structure de données modélisée dans l'ontologie du domaine. Par exemple, l'outil d'extraction d'information repère des dates comme « le 7 avril 1939 » ou encore « hier ». Cette dernière nécessiterait d'ailleurs un calcul de la date effective mentionnée (est-ce hier par rapport à une date mentionnée précédemment dans le texte d'origine ? ou est-ce hier par rapport à la date de publication du document ?).

Dans les deux cas, admettons qu'un attribut « date » ait été modélisé dans l'ontologie du domaine, avec comme structure de données, non pas une chaîne de caractère, mais un format calendaire du genre « jj/mm/aaaa ». Ce format de données concernant les dates permet de stocker des valeurs comme « 07/04/1939 » qui peuvent être exploitées par les futures applications, notamment à des fins de recherche sur des intervalles de dates expressément connues et interprétables par une machine. Or, il ne sera pas possible d'enregistrer les deux dates mentionnées plus haut dans la base de connaissance avec une telle structure de données.

Parmi les solutions, il faut soit faire intervenir un module d'interprétation des dates afin de transformer les expressions linguistiques de ces dates en une structure formelle, mais ceci est extrêmement complexe car comme dans le cas de « hier », il n'est pas toujours possible de savoir quelle est la date de référence permettant de calculer la valeur de « hier ». D'un autre côté, si cette information est absolument requise pour l'annotation ou pour le peuplement de l'ontologie, alors il faut modifier la structure de données de l'attribut date pour que celle-ci ne corresponde plus qu'à une simple chaîne de caractère. Ces choix de modélisations des formats de données adéquats dans l'ontologie du domaine sont véritablement dépendants de l'application et des utilisations qui en seront faites ultérieurement.

8.1.1.2 La précision de l'information

Le degré de précision entre les étiquettes sémantiques et les concepts de l'ontologie du domaine n'est pas toujours identique. En effet, il arrive parfois que la sémantique modélisée dans l'ontologie soit plus fine que celle des étiquettes sémantiques ou inversement. Ceci est dû principalement au fait que la modélisation des règles d'extraction d'information et celle de l'ontologie du domaine sont réalisées de manière indépendante. C'est le rôle de l'intégrateur et du client que de faire en sorte qu'il y ait le moins de discordance possible au niveau de la modélisation des deux structures mais il faut également prendre en compte les contraintes liées à chacun des outils et méthodes.

Pour illustrer le premier cas, prenons une ontologie dans laquelle une classe « Famille » possède trois propriétés « aPère », « aMère », « aEnfant » chacune pointant sur une instance de la classe « Personne ». L'extraction d'information sur l'article « La Tribu Coppola » produit l'arbre conceptuel de la Figure 84 représentant une relation de parenté entre un parent « Francis Coppola » et un enfant « Sofia ». Or, dans ce cas, nous n'avons aucun moyen de savoir si le parent « Francis Coppola » est actuellement le père ou la mère de l'enfant comme modélisé dans l'ontologie. L'ontologie du domaine est plus précise que l'extraction et il ne sera alors pas possible d'instancier cette relation de parenté dans la base de connaissance.

```
/QualificationPersonne (Francis Coppola avec sa fille Sofia)
   /ActorParent (Francis Coppola)
   /Parenthood (avec sa fille Sofia)
      /Child (Sofia)
         /Prenom (Sofia)
```

Figure 84. Exemple d'arbre conceptuel modélisant une relation de parenté

L'ajout d'un module de raisonnement permettrait de résoudre certains manques de précision. Ainsi, dans le cas présenté ci-dessus, si l'ontologie du domaine modélise :

1) la couverture de la propriété « aPère » comme une instance de la classe « Personne » qui possède un attribut « sexe » ayant la valeur « masculin »,
2) la couverture de la propriété « aMère » comme une instance de la classe « Personne » qui possède un attribut « sexe » ayant la valeur « féminin »,

et si l'instance « Francis Coppola » possède cette valeur « masculin » pour l'attribut « sexe », alors le module de raisonnement pourra en déduire que l'extraction d'information signale bien une relation de parenté entre le père « Francis Coppola » et l'enfant « Sofia ».

Inversement, les étiquettes sémantiques produites par l'analyse linguistique peuvent être plus précises que les concepts modélisés dans l'ontologie du domaine. Ainsi, l'ontologie du domaine « People » possède une seule classe concernant les événements liés au couple : la classe « Personnalités Mariage ». Cette dernière a un attribut booléen nommé « divorce » qui mentionne si le mariage en question a conduit ou non à un divorce entre les personnalités (conjoints) concernées. Or, comme montré dans la Figure 85, l'outil d'extraction est capable d'extraire un événement de rupture entre deux personnalités par la présence de l'étiquette sémantique « /Rupture » comme nœud fils de « /COUPLE » et nœud frère des deux personnalités « /ActorNamed ». Or, doit-on en conclure qu'il s'agit d'une rupture dans un couple non marié ? Dans un couple marié ? Dans ce cas, est-ce l'annonce d'un probable divorce ? Comment instancier cet événement dans la base de connaissance puisqu'il n'y a pas de classe qui corresponde, ni d'attribut permettant de stocker d'une manière ou d'une autre cette information pourtant intéressante dans le domaine des « People » ? …

```
/COUPLE (Spike Jonze et Sofia Coppola ont rompu en 2001)
   /ActorNamed (Spike Jonze)
      /Personality (Spike Jonze)
   /ActorNamed (Sofia Coppola)
      /Personality (Sofia Coppola)
   /Break (ont rompu)
   /DATE (2001)
```

Figure 85. Exemple d'arbre conceptuel modélisant un événement « rupture » entre deux personnalités

Dans ce cas précis, il serait intéressant de pouvoir gérer automatiquement l'évolution de l'ontologie en lui ajoutant de nouveaux éléments (classes, attributs, relations) à partir d'étiquettes sémantiques pouvant correspondre à un certain schéma dans l'arbre conceptuel. En effet, toutes les étiquettes sémantiques non utilisées par les règles d'acquisition ne donnent pas nécessairement lieu à de nouveaux concepts dans l'ontologie. Néanmoins certains schémas utilisés dans quelques Règles d'Acquisition de Connaissance pourraient être étendus à d'autres étiquettes sémantiques et ces dernières utilisées pour labelliser le nouvel élément introduit dans l'ontologie. Bien évidemment, une étape de validation par un utilisateur humain serait absolument nécessaire pour vérifier la qualité de ces nouveaux éléments.

8.1.1.3 La proximité de l'information dans l'arbre

Cette limite est liée au positionnement des étiquettes sémantiques dans l'arbre conceptuel. En effet, ce positionnement peut conduire à certaines ambiguïtés entre plusieurs étiquettes sémantiques qui pourraient être utilisées pour instancier ou annoter le même concept alors qu'il ne devrait y avoir qu'un seul choix possible.

Par exemple, la classe « Personnalité » dans l'ontologie « People » possède un attribut « lien de parenté ». Cet attribut prend habituellement pour valeur le contenu de l'étiquette sémantique « /LienParente » qui se situe au même niveau que l'étiquette sémantique « /ActorNamed » qui elle-même contient la valeur de l'instance de référence de la classe « Personnalité ». Or, dans l'exemple donné par la Figure 86, deux étiquettes sémantiques « /ActorNamed » sont situées au même niveau que celle « /LienParente ». Comment alors savoir si cet attribut est relatif à la première personnalité ou à la seconde ?

```
/QualificationPersonne (Anton Coppola, l'oncle de Francis, …)
     /ActorNamed (Anton Coppola)
        /Personality (Anton Coppola)
     /LienParente (oncle de Francis)
     /ActorNamed (Francis)
        /FirstName (Francis)
```

Figure 86. Exemple d'arbre conceptuel représentant une relation de parenté entre personnalités

En fait, la Règle d'Acquisition de Connaissance se déclenchera sur les deux références de personnalités et ces deux instances de personnalités auront chacune un nouvel attribut « lien de parenté » ayant pour valeur « oncle de Francis », même si dans le deuxième cas ceci est faux. Pour

éviter de telles erreurs, la Règle d'Acquisition de Connaissance ne doit être définie que s'il n'existe aucun risque potentiel d'ambiguïté par rapport à la structure de l'arbre conceptuel.

8.1.2 Problèmes liés au déclenchement des Règles d'Acquisition de Connaissance

Ces autres problèmes soulevés par OntoPop sont liés à l'intégration de l'outil d'extraction d'information et de représentation des connaissances ainsi qu'au déclenchement des règles en général. Nous voulons insister sur le fait que ces problémes sont directement influencés par la notion de contexte déjà évoquée précédemment. En effet, l'écriture des Règles d'Acquisition de Connaissance peut vite devenir une tâche complexe car l'instance d'un concept de l'ontologie ne correspond pas forcément à la valeur textuelle d'une seule étiquette sémantique. Il faut donc bien prendre en compte l'ensemble des étiquettes sémantiques pouvant être utilisées pour instancier un même concept, outre celles caractérisant en plus les indices contextuels. Cela génère donc des problèmes de consistance de la connaissance, de conflits entre règles et de maintenance de l'application cible.

8.1.2.1 La consistance de l'information

Le respect de la sémantique originelle du texte doit être une priorité. Cela signifie que la sémantique des divers éléments de l'ontologie du domaine doit être consistante avec celle des étiquettes sémantiques des arbres conceptuels. Ainsi, l'étiquette sémantique « Personnalite » sert à créer les instances de la classe « Personnalité », classe décrivant toutes les personnes dites « People » dans l'ontologie de ce domaine. Cette étiquette ne peut convenir à l'instanciation de la classe « Personnage de film », même si celles-ci peuvent devenir des personnalités dans l'imaginaire collectif, comme « Rocky », « Zorro » ou « JFK ».

```
/OeuvreFilm (l'innoubliable Adrienne de "Rocky")
     /ActorNamed (l'innoubliable Adrienne)
        /Prenom (Adrienne)
     /Oeuvre (Rocky)
        /ActorNamed (Rocky)
         /Personnalite (Rocky)
```

Figure 87. Exemple d'un arbre conceptuel représentant un événement « Œuvre Casting »

Ainsi, dans l'arbre conceptuel présenté ci-dessus, l'extracteur d'information a repéré une relation entre une œuvre de cinéma, « Rocky », et un personnage, « Adrienne ». Or, il a aussi étiqueté le nom du film comme étant une « personnalité », car « Rocky » est également le personnage principal de ce film et il est devenu une personnalité à laquelle on fait référence. Pourtant, ce n'est pas une personne réelle et pour cette raison, une instance « Personnalité » ayant pour valeur « Rocky » ne doit pas être créée dans la base de connaissance afin de préserver sa consistance de cette dernière.

8.1.2.2 Les conflits entre Règles d'Acquisition de Connaissance

Une Règle d'Acquisition de Connaissance ayant des conditions mal définies sur le contexte de l'étiquette sémantique « déclencheur » peut instancier un autre concept que celui pour lequel elle a été définie, bien que cet autre concept possède également ses propres règles d'acquisition. Ce problème concerne des cas complexes, notamment lorsqu'une classe et ses sous-classes possèdent des règles d'acquisition très fortement similaires. Généralement l'étiquette sémantique « déclencheur » est identique pour chacune des classes et la distinction est réalisée uniquement à partir de la résolution des indices contextuels.

```
/COUPLE (son cousin, Nicolas, est en train de divorcer de Patricia)
   /ActorNamed (son cousin, Nicolas)
      /Famille (son cousin, Nicolas)
      /Prenom (Nicolas)
   /EvenementImminent (est en train de)
   /Divorce (divorcer)
   /ActorNamed (Patricia)
      /Prenom (Patricia)
```

Figure 88. Exemple d'arbre conceptuel représentant un événement de divorce entre personnalités

Ainsi, admettons qu'une classe « Evénement Privé » possède deux sous-classes « Mariage » et « Divorce ». L'étiquette sémantique « /COUPLE » est le même nœud indicateur déclencheur pour les trois classes présentées. Mais si cette étiquette sémantique possède un nœud fils « Mariage », alors la classe « Mariage » sera instanciée alors que s'il s'agit du nœud fils « /Divorce », comme cela est le cas dans la Figure 88, ce sera la classe « Divorce ». Si aucun de ces deux nœuds « /Mariage » ou « /Divorce » ne sont précisés, alors la classe mère « Evénement Privé » sera instanciée. Ce peut également être un moyen de résoudre le problème de précision de l'information évoqué plus haut dans la Figure 85, au sujet de l'événement de rupture dans un couple de personnalités.

8.1.2.3 La maintenance des Règles d'Acquisition de Connaissance

A chaque modification de l'ontologie ou de la structure de l'arbre conceptuel produit en sortie des outils d'extraction d'information, toutes les règles impactées par ces modifications doivent être ajustées pour répercuter les changements opérés d'un côté comme de l'autre. Même si le traitement d'un document est ensuite entièrement automatisé, la maintenance de ces règles doit être réalisée par un utilisateur humain dont la tâche peut vite devenir complexe en fonction de la taille des arbres conceptuels, de la couverture du domaine et du modèle de l'ontologie.

8.2 Vers l'alignement d'ontologies ?

Tous ces problèmes soulevés par OntoPop nous ont amenés à réfléchir à une solution pour pouvoir créer plus facilement les Règles d'Acquisition de Connaissance tout en respectant les exigences de cohérence, de résolution des conflits et de leur maintenance pour un domaine donné. Nous avons dit au début de ce mémoire que la problématique de l'annotation sémantique ou du peuplement d'ontologie consistait à passer d'un format de représentation d'une ressource textuelle, i.e. les arbres

conceptuels, à un format de représentation des connaissances formel. Nous avons aussi vu que ces deux formats s'appuient sur différents niveaux de l'architecture des langages recommandés dans le cadre du Web Sémantique : XML pour les arbres conceptuels, RDF pour les annotations, XTM ou OWL pour les instances du domaine. Il nous faut donc trouver un moyen d'aligner ces niveaux pour faciliter le couplage entre les deux formats de représentation.

Au chapitre 2, nous avons décrit les arbres conceptuels comme un format intermédiaire entre une représentation purement syntaxique du document et une représentation de sa sémantique par des graphes conceptuels ou des prédicats logiques. Autrement dit, l'arbre conceptuel se trouve à mi-chemin d'une représentation textuelle du document et d'une représentation sémantique de la connaissance contenue dans le document. En fait, je pense que lorsque les ingénieurs linguistes créent les patrons d'extractions relatifs à un domaine, ils modélisent plus ou moins implicitement les concepts de ce domaine. Et ainsi, à chaque fois qu'un document est analysé par le moteur d'extraction d'information, l'arbre conceptuel produit est une instanciation de ces concepts, sans pour autant qu'ils soient interconnectés entre eux comme dans une ontologie.

Par conséquent, je suis convaincue qu'il serait particulièrement intéressant de pouvoir expliciter ces concepts modélisés dans les patrons afin de les modéliser dans une ontologie des arbres conceptuels. Il faudrait aussi y ajouter les relations et les attributs à ces concepts afin de pouvoir pallier par la suite au manque d'interconnection entre les instances dans les arbres. Si nous pouvons modéliser cette ontologie des arbres conceptuels, cela voudra dire que nous pourrons aborder le problème du couplage et de la création des règles sous l'angle de **l'alignement d'ontologies** [EUZ 04]. En effet, les méthodes et les techniques de ce champ de recherche très actif du Web Sémantique pourraient être exploitées pour aligner l'ontologie des arbres conceptuels avec celle du domaine de l'application. Et nous espérons ainsi faciliter la création et la maintenance des Règles d'Acquisition de Connaissance.

Il est important de préciser ici que durant ces dernières décennies, et notamment depuis la création des bases de données relationnelles, des chercheurs ont travaillé sur l'intégration de schémas de bases de données [DOA 05], puis sur l'intégration de schémas XML [SHV 05]. Plus récemment, ils se sont aussi intéressés à l'intégration entre ontologies, notamment en OWL [NOY 04]. Ils recherchent les méthodes et outils permettant d'aligner différentes représentations d'un même format afin de pouvoir les fusionner ou bien les traduire l'une dans l'autre. Cela permet, entre autres, de pouvoir interroger ces bases de données ou ontologies à partir d'un même point d'entrée.

Euzenat dans [EUZ 04] définit l'alignement d'ontologies comme un ensemble de correspondances entre les entités e' d'une Ontologie O' et les entités e'' d'une Ontologie O''. Autrement dit, étant données deux ontologies O' et O'', alignemer une ontologie vers une autre signifie que pour chaque entité (classe C, relation R, attribut A ou instance I), est recherchée une entité ayant la même sémantique dans l'ontologie O''. Cet alignement est réalisé par le biais de règles de couplage de la forme : aligner(e')=e''.

Les méthodes permettant de déduire les règles de couplage s'inspirent de recherches issues de l'analyse de données, de l'apprentissage, du traitement du langage naturel, des statistiques ou encore de la représentation des connaissances selon les entités manipulées. Il existe quatre grandes catégories de méthodes : celles utilisant la terminologie des entités, celles reposant sur l'analyse de la structure du modèle (position d'une entité dans une taxonomie par exemple), celles qui exploitent les instances des entités comparées et enfin celles qui s'appuie sur la sémantique des modèles (déduisant l'alignement grâce à des raisonnements basés sur les logiques de description par exemple). Il existe aussi d'autres méthodes situées au niveau plus global des modèles et plus seulement au niveau des entités de ces modèles [EUZ 04].

Dans la plupart de ces méthodes, les règles correspondent en fait à l'application d'un algorithme qui permet de déduire e'' à partir de e' dans des ontologies O' et O'' données [KAL 03a]. Il n'existe donc pas de représentation standardisée de ces règles de couplage bien qu'Euzenat [EUZ 06], Bouquet [BOU 04] et d'autres aient tenté de fournir un cadre pour la définition des couplages d'un point de vue plus abstrait. Nous pensons d'ailleurs que l'adaptation des Règles d'Acquisition de Connaissances à la problématique de l'alignement d'ontologies pourrait aider à la formalisation de ces règles de couplage.

Loin de toute certitude concernant l'apport réel des méthodes d'alignement d'ontologies aux limites vues précédemment, j'ai commencé à réfléchir à la possibilité de cette perspective d'évolution pour OntoPop. La première étape consiste à modéliser l'ontologie des arbres conceptuels à partir des patrons d'extraction. Or, s'il n'est pas toujours possible d'accéder directement aux patrons d'extraction, car le plus souvent ils sont compilés à l'intérieur de la cartouche linguistique qui constitue alors une « boîte noire », la structure de ces patrons se reflète dans la structure hiérarchique des arbres conceptuels comme nous l'avons vu à la section 2.3. Ainsi, à partir du corpus de tests unitaires, constitué durant la phase de couplage de la méthodologie, nous devons disposer de l'ensemble des structures des arbres conceptuels pouvant être générés par la cartouche linguistique du domaine, à quelques exceptions près. Chacun de ces arbres pouvant être représentés au format XML, il devient alors possible de définir le XML Schéma de la cartouche et par conséquent des patrons d'extractions. Ensuite, une fois ce Schéma XML obtenu, il faut pouvoir en déduire l'ontologie correspondante en OWL. Pour ce faire, des recherches sur le sujet ont très récemment abouti à la proposition de méthodes et d'outils qu'il nous faudra étudier [BOH 05] [ROD 06].

Mais avant cela, il faut pouvoir savoir si notre première hypothèse concernant la transposition des arbres conceptuels en Schéma XML fait sens. J'ai donc repris l'analyse des arbres conceptuels du domaine de la Presse People afin de voir s'il était effectivement possible d'en définir un Schéma XML associé qui soit cohérent. D'après cette analyse, j'ai repéré trois sortes de motifs : les entités nommées, les attributs et les scénarios. Par motif, j'entends une structure redondante et générique dans les arbres conceptuels générés. Chaque motif est défini par un ensemble d'étiquettes provenant soit des lexiques, soit des règles d'extractions utilisées. Le caractère « ? » précédant certaines étiquettes indique que ces étiquettes sont optionnelles, c'est-à-dire qu'elles peuvent ou non apparaître

dans le motif. Le Tableau 14 présente un ensemble de motifs pour la reconnaissance des Entités Nommées de type « Personne » ? Chaque motif est illustré par un exemple issu du corpus documentaire et accompagné par des remarques qui précisent la spécificité du motif. Ainsi, ce tableau comporte six motifs : les quatre premiers représentent des personnes connues, i.e. provenant d'un lexique ou déduites par un patron d'extraction, alors que les deux derniers reconnaissent des personnes inconnues, i.e. repérées par des pronoms ou des groupes nominaux. Ces deux motifs sont inexploitables pour le peuplement d'ontologie ou pour l'annotation car elles ne font pas explicitement référence à des personnalités connues. Par contre, si le moteur d'extraction est capable de résoudre les coréférences pronominales ou nominales, alors ces motifs peuvent devenir intéressants par la suite.

Motifs des arbres	**Exemples phrase**	**Remarques**
/ActorNamed /Personnalite	« Francis Ford Coppola »	Personnalités connues, issues d'un lexique
/ActorNamed /Prenom (?/SEXE_MASCULIN ?/SEXE_FEMININ ?/SEXE_INCONNU)	« Sofia »	Prénoms issus d'un lexique avec info du sexe, peut être masculin, féminin ou inconnu
/ActorNamed /ProperName	« Coppola »	Personnalités devinées à partir d'un nom de famille issu d'un lexique
/ActorNamed ?/NomDePersonnePotentiel /ProperName /Prenom (?/SEXE_…)	« Coppola Sofia »	Personnalités devinées à partir d'un prénom et d'un nom de famille
/ActorUnknown (?/MascSing ?/FemSing ?/NeutSing ?)	je, tu, il, elle, on	Pronoms
/CoupleActeur (?/MascPlu ?/FemPlu ?/NeutPlu)	nous, vous, ils, elles, « le couple »	Pronoms ou groupe nominal issu d'un lexique

Tableau 14. Exemple des motifs obtenus pour les Entités Nommées « Personne »

A partir de ces six motifs, j'ai construit la DTD des motifs relatifs à l'identification des personnes, cf. Figure 89. Cette DTD, traduisible en Schéma XML, distingue les deux sortes de personnes : les personnes connues, donc identifiées, et les personnes inconnues, i.e. non identifiées. Ces deux catégories de personnes peuvent apparaître dans un motif au singulier, elle est la seule citée, ou dans un motif au pluriel, elle est citée avec d'autres personnes. Par exemple, l'élément de la DTD « PersonneNonIdentifiée » fait référence au motif sur les pronoms singuliers « je, tu il, elle, on » et l'élément « PersonnesNonIdentifiées » au motif sur les pronoms pluriels « nous, vous, ils, elles ». J'ai

également repéré que seules les personnes identifiées (singulier ou pluriel) apparaissent aussi bien dans le contexte d'un scénario qu'en dehors de tout contexte, c'est-à-dire à la racine des arbres conceptuels. Les personnes non identifiées (singulier ou pluriel) n'apparaissent que dans les scénarios. Cela renforce notre idée qu'elles sont extraites pour être potentiellement utilisées par un algorithme de résolution des coréférences qui permettrait de les identifier et ainsi d'acquérir plus de scénarios pertinents.

```
< !ELEMENT PersonneHorsContexte (REFERENCE-ACTEUR*)>
< !ELEMENT REFERENCE-ACTEUR (PersonneIdentifiée ? | PersonnesIdentifiées ?)>
< !ELEMENT PersonneDansScenario (PersonneIdentifiée* | PersonnesIdentifiées* | PersonneNonIdentifiée*
| PersonnesNonIdentifiées*)>
< !ELEMENT PersonneIdentifiée (ActorNamed ? | Personnalite ? | ProperName ?)>
< !ELEMENT ActorNamed (Personnalite ? | Prenom ? | ProperName ? | NomDePersonnePotentiel ?)>
< !ELEMENT Personnalite (#PCDATA)>
< !ELEMENT NomDePersonnePotentiel (Prenom ? | ProperName ?)>
< !ELEMENT Prenom (SEXE_MASCULIN ? | SEXE_FEMININ ? | SEXE_INCONNU ?)>
< !ELEMENT SEXE_MASCULIN (#PCDATA)>
< !ELEMENT SEXE_FEMININ (#PCDATA)>
< !ELEMENT SEXE_INCONNU (#PCDATA)>
< !ELEMENT ProperName (#PCDATA)>
< !ELEMENT PersonnesIdentifiées (ActorPlural)>
< !ELEMENT ActorPlural (ActorNamed * | ProperName * | NomDePersonnePotentiel *)>
< !ELEMENT PersonneNonIdentifiée (ActorUnknown)>
< !ELEMENT ActorUnknown (MascSing ? | FemSing ? | NeutSing ? | #PCDATA)>
< !ELEMENT MascSing (#PCDATA)>
< !ELEMENT FemSing  (#PCDATA)>
< !ELEMENT NeutSing  (#PCDATA)>
< !ELEMENT PersonnesNonIdentifiées (CoupleActeur)>
< !ELEMENT CoupleActeur (MascPlu ? | FemPlu ? | NeutPlu ? | #PCDATA)>
< !ELEMENT MascPlu (#PCDATA)>
< !ELEMENT FemPlu  (#PCDATA)>
< !ELEMENT NeutPlu  (#PCDATA)>
```

Figure 89. DTD associée aux motifs des Entités Nommées « Personne »

J'ai procédé de même pour les autres entités nommées, les attributs et les scénarios. Les entités nommées dans ce domaine correspondent clairement aux classes de l'ontologie du domaine : Personnalité, Personnage et Œuvre principalement. En ce qui concerne les attributs et les scénarios, les motifs sont issus de la structure syntaxique de la phrase. Ceci concorde avec l'analyse des arbres conceptuels que nous avons faite au chapitre 2. Les motifs des attributs sont généralement placés sous l'étiquette représentant l'entité décrite par ces attributs alors que les motifs des scénarios sont chapeautés par l'étiquette symbolisant la sémantique de ce scénario. Ces attributs et scénarios peuvent correspondre soit à des attributs, soit à des relations dans l'ontologie du domaine. Mais attention, un motif d'attribut peut donner lieu à une relation dans l'ontologie et inversement, un motif de scénario peut instancier un attribut dans l'ontologie. Par exemple, il existe un scénario « Date-Naissance » dans l'arbre qui va permettre d'instancier les attributs de « date de naissance » et de

« lieu de naissance » d'une personnalité donnée. La distinction se fait plutôt grâce au nombre d'entités nommées participant au motif. S'il peut y en avoir plusieurs dans un même motif, alors celui-ci peut être traduit en une relation du domaine. Par contre, si le motif possède une et une seule entité nommée, alors il peut être traduit en attribut.

A partir de cette analyse, j'ai pu dégager le Schéma XML global de la cartouche linguistique du domaine de la Presse People et ébaucher manuellement une première version d'une ontologie des arbres conceptuels, cf. Figure 90. En fait, cette ontologie est constituée d'une méta-ontologie qui décrit la structure des arbres conceptuels et de l'ontologie des arbres conceptuels du domaine concerné. La méta-ontologie modélise un arbre conceptuel comme constitué de plusieurs « Motif » de trois sortes, « Motif Attribut », « Motif Entité Nommée » et « Motif Scénario », qui sont eux-mêmes composés d'une ou plusieurs « Etiquette », également de trois sortes. Cette méta-ontologie permet de modéliser l'ontologie des arbres conceptuels quel que soit le domaine concerné. Puis, l'ontologie des arbres conceptuels du domaine reprend plus particulièrement les éléments du schéma XML défini précédemment pour les organiser en fonction des contraintes imposées par la méta-ontologie. Par exemple, dans l'ontologie des arbres du domaine de la Presse People, il existe la classe « PersonnesNonIdentifiées », sous-classe de « Etiquette Personnalité », elle-même sous-classe de « Etiquette Entité Nommée » qui appartient à la méta-ontologie. Les instances de cette classe « PersonnesNonIdentifiées » correspondent aux valeurs des étiquettes « /MascPlu », « /FemPlu » et « /NeutPlu » qui apparaissent dans les arbres conceptuels.

Nous pensons que pour déduire les RAC entre cette ontologie des arbres conceptuels et l'ontologie de référence du domaine, il devient alors tout à fait envisageable d'utiliser certains des algorithmes d'alignement basés sur :

- l'analyse de la terminologie des entités (classes, attributs, relations) des deux ontologies et notamment des noms des concepts, de leurs définitions, de leur structure lexicale, de la distance entre deux chaînes de caractères, etc. [HOV 98],
- la comparaison de la structure de ces ontologies (taxonomie, contraintes, restrictions, etc.) [EHR 04], et notamment à l'aide d'une représentation sous la forme de graphes [NOY 03] [MEL 02],
- la comparaison des instances des entités de chaque ontologie par des méthodes d'apprentissage et de calculs statistiques [DOA 04] [LI 00].

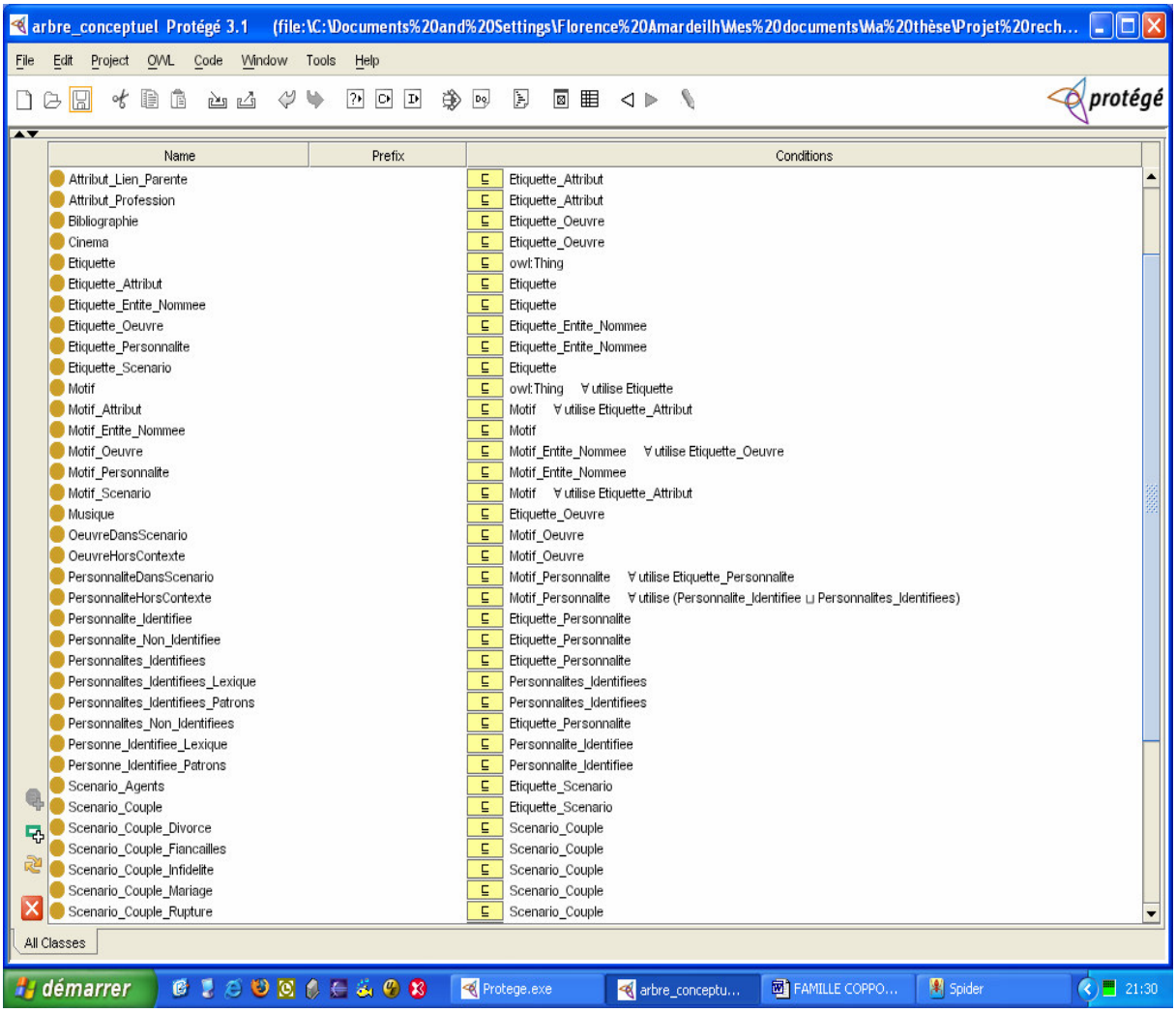


Figure 90. Ebauche de l'ontologie des arbres conceptuels

Il me reste donc encore beaucoup à faire pour évaluer la potentialité de cette perspective :

- appliquer les méthodes et outils de transformation des Schémas XML pour la création d'une ontologie en OWL afin d'en étudier les résultats et les points communs avec l'ontologie que nous avons élaborée manuellement,
- étudier les outils et méthodes d'alignement d'ontologie pour vérifier notre deuxième hypothèse quant à la création semi-automatisée des RAC, et enfin,
- envisager dans quelle mesure les RACs pourraient être utilisées ou adaptées pour l'alignement d'ontologies.

Il faudra ensuite dégager une méthodologie et tester son adaptabilité à des cartouches linguistiques de différents domaines et corpus, ainsi qu'à des arbres conceptuels générés par différents moteurs linguistiques. C'est donc une perspective d'évolution très intéressante et très riche pour OntoPop, à laquelle une nouvelle thèse pourrait être consacrée.

8.3 Conclusion

Dans OntoPop, les Règles d'Acquisition de Connaissance agissent comme un médiateur entre les arbres conceptuels et l'ontologie du domaine. Elles représentent le cœur de la démarche, de la méthodologie et de la plateforme d'OntoPop. Mais malgré leur grande flexibilité et leur capacité d'adaptation aux différents corpus et domaines, leur implémentation comme leur utilisation soulèvent un certain nombre de problèmes abordés dans ce chapitre. Notamment la question de leur maintenance, aujourd'hui réalisée manuellement par l'intégrateur du projet concerné, devient cruciale lorsque l'ontologie ou la cartouche linguistique du domaine sont modifiées. Plus l'ontologie est importante et plus la cartouche linguistique est complexe, plus la création et la maintenance de ces règles devient une tâche fastidieuse pour l'humain. Notre priorité vis-à-vis de la solution apportée par d'OntoPop consiste donc à faire évoluer cette tâche manuelle vers une tâche semi-automatisée. Pour cela nous comptons nous appuyer sur les méthodes et outils issus des recherches sur l'alignement d'ontologie. Nous avons démarré une étude de faisabilité qui nous semble prometteuse, mais il nous reste à la mettre en pratique afin de vérifier sa réelle pertinence.

Conclusion Générale

Nous venons de présenter dans cette thèse le fruit de nos travaux de recherche : la démarche OntoPop. Cette démarche constitue notre contribution à la conception et à la réalisation d'applications pour l'annotation sémantique et le peuplement d'ontologie à partir de documents textuels. Elle s'appuie sur les théories, les méthodes et les techniques développées tant en Informatique Linguistique (IL) que dans le domaine plus récent du Web Sémantique (WS). La thèse commence par présenter et définir l'annotation sémantique dans le contexte du Web Sémantique. En effet, le Web Sémantique a besoin pour son développement futur de pouvoir créer des annotations à partir de représentations formelles de la connaissance d'un domaine, telles que les ontologies. Nous avons vu que pour cela la plupart des outils existants utilisent des moteurs d'extraction d'information pour aider les utilisateurs dans cette tâche extrêmement fastidieuse. Par contre, ces outils ne se sont presque pas intéressés à un point qui nous semble pourtant fondamental quant à l'optimisation de la qualité des annotations créées. Il s'agit de la manière de combler le fossé existant entre d'une part les représentations du contenu du texte par les outils d'extractions d'information, à savoir les arbres conceptuels, et d'autre part la représentation de ce même contenu par l'utilisation d'une ontologie.

Je me suis donc intéressée à cette problématique tout au long de mes travaux de recherche. Pour ce faire, j'ai tout d'abord analysé la structure des arbres conceptuels afin d'en déduire leurs caractéristiques et ce qui pouvait leur manquer pour devenir une réelle représentation sémantique formelle. Ces arbres se situent en fait à un niveau intermédiaire entre des graphes conceptuels et une représentation purement syntaxique du contenu qui reste basée sur un ordonnancement séquentiel. Puis, je me suis penchée sur la transposition de ces arbres conceptuels vers la représentation sémantique nécessaires aux annotations, mais aussi aux futures instances de l'ontologie qui peuvent également être produites à partir de ces mêmes arbres.

Mon travail de thèse a su apporter des réponses à la médiation nécessaire entre ces deux niveaux de représentation grâce à la conceptualisation des Règles d'Acquisition de Connaissance (RAC). Non seulement avons-nous identifié les différents constituants de ces règles qui permettent de coupler un arbre conceptuel avec les éléments d'une ontologie mais plus encore, nous avons défini un langage, OPAL, qui permet de les implémenter concrètement. Grâce à ce langage, les RAC présentent l'avantage de rendre la tâche du couplage indépendante de toute réalisation informatique et d'articuler efficacement analyse linguistique et traitement informatique dans une même architecture logicielle. Cette implémentation des règles n'est donc pas limitée à une infrastructure spécifique mais offre un cadre de travail adaptable à tout besoin des applications du monde réel.

D'ailleurs, j'ai pu me confronter à cette réalité métier tout au long de ma thèse grâce à mon implication dans divers projets industriels en entreprise. En effet, le contexte et le déroulement de ma thèse m'ont encouragée à considérer une problématique plus importante que la simple capture de la sémantique à partir d'arbres conceptuels. Du fait de l'ancrage de mes travaux dans des projets concrets d'utilisation de ces RAC, la question de leur application aux diverses problématiques soulevées par chacun de ces projets a pris une part de plus en plus importante au fur et à mesure de l'avancée de mes travaux. Ceci m'a guidé par exemple vers des choix de conceptualisation, comme l'ajout de la partie « Options » aux règles. Mais plus encore, j'ai également dû prendre en considération les tâches d'annotation sémantique et de peuplement d'ontologie dans leur globalité.

Cette nouvelle réflexion m'a conduite à étudier attentivement le cycle de vie des ressources terminologiques et ontologiques dans ces deux tâches afin de proposer des solutions adéquates aux problèmes soulevés par :

- l'application des RAC, et notamment la nécessité de prise en compte du contexte des étiquettes linguistiques dans les arbres conceptuels pour lever les ambiguïtés sémantiques dans l'ontologie du domaine ;
- la consolidation des annotations sémantiques et du réseau sémantique produits par ces règles en fonction du modèle de cette ontologie et des ressources terminologiques ou ontologiques existantes dans le référentiel ;
- la mise à jour des lexiques des outils d'extraction utilisés afin que ce cycle de vie des RTO devienne un cercle vertueux, profitant ainsi de la synergie entre les domaines de l'IL et du WS évoquée en introduction de ce mémoire.

J'ai ensuite montré dans cette thèse que la démarche proposée n'est pas seulement théorique mais peut être opérationnalisée par la mise en œuvre d'une méthodologie de projet et par l'implémentation d'une plateforme logicielle dédiée qui répondent à ses critères. L'architecture de la plateforme, que j'ai conçue avec l'aide de l'équipe de développement de Mondeca, privilégie le concept de composants logiciels indépendants qui favorisent ainsi la flexibilité de l'ensemble du système et sa modularité. Cette modularité a comme avantage de modifier un aspect du processus général sans être obligé de reprendre intégralement le développement informatique de la solution pour une application donnée. Une interface utilisateur est dédiée à la validation des annotations sémantiques et des ressources terminologiques ou ontologiques du référentiel à travers une même IHM intuitive et conviviale, bien que nous n'ayons pas eu l'opportunité de l'évaluer auprès d'utilisateurs finaux.

Cette plateforme a pourtant été éprouvée lors de plusieurs projets, aux objectifs, besoins et domaines variés, tout comme la méthodologie que j'ai définie. Celle-ci est d'ailleurs opérationnelle et utilisée dans toutes les collaborations entre les équipes de Mondeca et de Témis sur nos projets communs.

Mondeca a également de nouveaux projets en route avec d'autres partenaires possédant leurs propres outils d'extraction d'information :

- le laboratoire de recherche « Natural Language Processing Group »[40], de l'Université de Sheffield, qui est à l'origine de la plate-forme d'ingénierie linguistique GATE. Nous collaborons dans le cadre du projet « Transitioning Applications to Ontologies »[41] (TAO), financé par le programme IST-FP6 de l'Union Européenne.

- le laboratoire de recherche « Modèles, Dynamiques, Corpus »[42] (MoDyCo), de l'Université Paris X – Nanterre, avec lequel nous travaillons sur le projet EIFFEL[43], financé par le programme RNTL du Ministère de l'Industrie.

En plus des objectifs propres à chacun de ces projets de recherche, notre but est de pouvoir tester à la fois l'applicabilité de notre méthodologie OntoPop avec ces nouveaux partenaires mais aussi la facilité d'utilisation de notre plateforme OntoPop dans de nouveaux environnements. Nous allons d'ailleurs présenter cette nouvelle plateforme dans le chapitre suivant.

Néanmoins, il est important de noter ici que l'opérationnalisation des processus d'annotation sémantique et de peuplement d'ontologie est évidemment indissociable d'une réflexion complète sur l'utilisation de ces processus dans un contexte particulier. L'annotation sémantique des ressources documentaires et le fait que ces annotations dépendent de ressources terminologiques ou ontologiques permet :

- d'améliorer la pertinence des résultats de recherche dans les bases documentaires ;
- de proposer plusieurs angles d'entrée à ces bases comme la recherche sémantique (utilisation des différents éléments d'une ontologie comme critères de recherche), l'extension sémantique (utilisation des synonymes, des alias, des termes reliés dans les thésaurus ou les bases de connaissances pour obtenir plus de résultats pertinents), etc. complétant ainsi le mode de recherche par simples mots-clefs ;
- la découverte de nouvelles connaissances grâce aux mécanismes d'inférence et de raisonnement logique mis en place sur la base des ontologies ;
- l'exploitation des annotations et des connaissances interconnectées pour créer de nouveaux documents, de nouvelles publications ;
- l'interopérabilité des annotations et des ressources terminologiques ou ontologiques entre plusieurs applications grâce à l'utilisation des langages standards du WS ;
- etc.

[40] http://nlp.shef.ac.uk/
[41] http://www.tao-project.eu/
[42] http://infolang.u-paris10.fr/modyco/
[43] http://www.rntl.org/projet/resume2005/eiffel.htm

Pour finir, nous avons discuté au chapitre précédent des limites soulevées par ces RAC et les perspectives d'évolution envisagées pour y remédier. Il est tout aussi légitime de se demander si nos réflexions et solutions peuvent sortir du cadre de cette thèse afin d'être transposées dans d'autres problématiques de recherche. Par exemple, nous avons vu parmi les limites que certaines étiquettes linguistiques étaient sémantiquement plus précises que les éléments ontologiques ou bien que les arbres contenaient des étiquettes linguistiques pertinentes vis-à-vis du domaine mais non prises en charge dans l'ontologie du domaine. Une des adaptations possibles des RAC pourrait donc être la maintenance et l'évolution des ontologies en les enrichissants par des nouveaux concepts identifiés à partir des arbres conceptuels. La proposition d'utiliser les annotations sémantiques pour l'évolution des ontologies est une question ouverte [STO 02]. Néanmoins, la modification des éléments ontologiques doit également se répercuter sur les annotations sémantiques qui en dépendent et ceci est un autre sujet de recherche important.

Nous voulons aussi étendre dans le futur le travail réalisé dans cette thèse avec de nouvelles fonctionnalités comme :

- l'annotation sémantique de contenus multimédias, où le couplage entre les outils d'extraction d'information pour ces contenus et l'outil de représentation des connaissances doit pouvoir gérer les nouveaux standards multimédias comme MPEG-7, MPEG-21ou SMIL pour la description des annotations sémantiques ;
- la consolidation des annotations et des réseaux sémantiques par l'utilisation de raisonnements logiques et de mécanismes d'inférence ; d'ailleurs nous avons mené une première expérimentation en couplant le Module d'Annotation et d'Acquisition d'ITM avec l'outil de raisonnement à base de graphes conceptuels Cogitant, ce qui a donné des résultats encourageants [AMA 06c] ;
- l'amélioration des interfaces hommes machines, et plus précisément de l'interface de validation, afin de les rendre plus interactives avec les différents objets manipulés et plus dynamiques.

Un dernier point que nous souhaitons soulever ici est la comparaison qui peut être faite de notre travail avec la plateforme UIMA (Unstructured Information Management Architecture) développée par le laboratoire de recherche Alphaworks d'IBM[44]. Comme le montre la figure ci-dessous, cette plateforme se compose de trois sortes de composants : le « Collection Reader » qui prend des contenus documentaires en entrée pour les convertir au format manipulé par UIMA, à savoir le « CAS » ou « Common Analysis Structure ». Puis ces contenus sont analysés par un ou plusieurs « Analysis Engine » qui représentent les ressources linguistiques implémentées pour l'application cible. Ces ressources linguistiques, à l'instar des ressources CREOLE dans GATE (cf. section 2.2), correspondent à autant de traitements linguistiques comme la segmentation ou l'analyse morpho-

[44] http://www.research.ibm.com/UIMA/

syntaxique. Le résultat de ces « Analysis Engine » vient enrichir le CAS de départ qui est finalement mis à disposition des « CAS Consumer », c'est-à-dire des outils qui vont exploiter cette structure pour annoter sémantiquement les documents, peupler une base de connaissance, exécuter des traitements pour la recherche d'information, etc.

La différence majeure entre OntoPop et UIMA réside dans l'utilisation de ce format « CAS ». Pour UIMA, il est dès le départ la structure commune à toutes les manipulations et traitements effectués dans la plateforme alors qu'OntoPop base sa démarche sur l'interprétation des différents formats générés tant au niveau de l'analyse linguistique que de la représentation de connaissances et des annotations sémantiques. On pourrait aussi considérer OntoPop comme un « CAS Consumer » au sein de l'architecture préconisée par UIMA qui utiliserait des arbres conceptuels au format « CAS ». En effet, GATE et IDE ont récemment adapté leurs APIs pour devenir de véritables « Analysis Engine » et ainsi produire des arbres conceptuels au format « CAS ». On constate une fois de plus qu'OntoPop s'adapte assez aisément à tous les cas d'architectures et nous allons d'ailleurs dans les nouveaux projets mentionnés précédemment travailler à cette intégration avec UIMA et son format « CAS ».

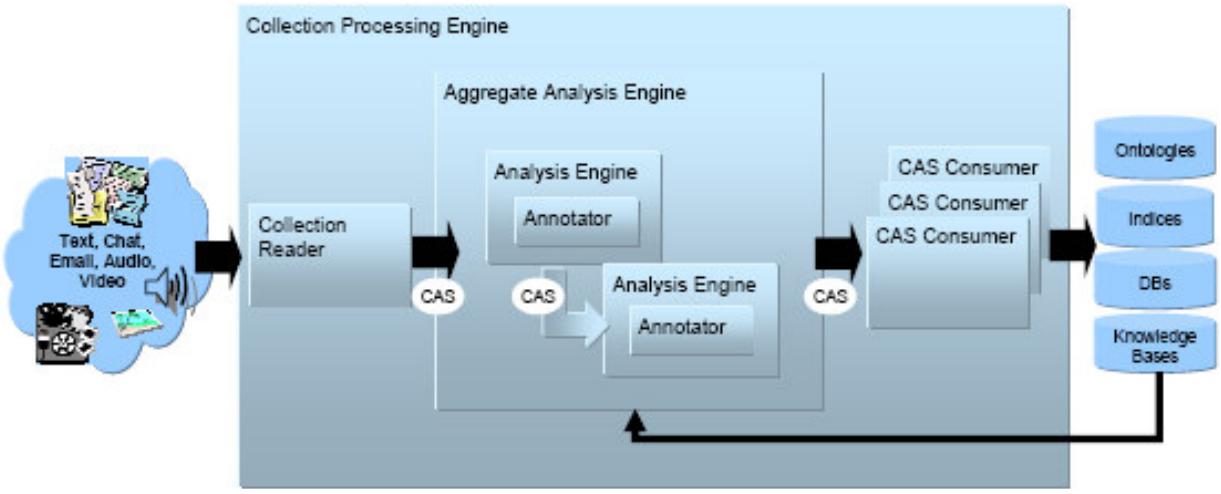


Figure 91. Schéma d'architecture de la plateforme UIMA [IBM 06]

En résumé, on peut constater que cette thèse apporte un ensemble de propositions pour la réalisation des activités d'annotation et de peuplement mais aussi pour leur utilisation dans la maintenance des bases de connaissance, des thésaurus et des autres ressources terminologiques d'un référentiel métier. Ces points ne constituent pas des idées indépendantes mais un cadre global auquel je me suis efforcée de donner une réelle cohérence. Ces propositions remplissent également le contrat CIFRE passé dans cette thèse car si les préoccupations scientifiques sont reliées à des applications pratiques, nous avons aussi fourni les moyens pour une entreprise de bénéficier directement de nos réflexions les plus théoriques – comme les Règles d'Acquisition de Connaissance – pour améliorer la qualité de ses applications de gestion documentaire et de gestion de la connaissance.

Annexes.

Annexe I. Etude des outils d'annotation sémantique

I.1 La Grille de lecture

Nous avons défini notre grille de lecture à partir des études réalisées dans le cadre des projets Advanced Knowledge Technologies[45] (AKT) [réf 2.], Dot.Kom[46] [réf 2.] et Esperonto[47] [CON 03], ainsi que de celle réalisée par Handschuh et al. [HAN 03a]. Nous avons complété ou précisé les critères avancés par ces études à partir de notre propre expérience dans ce domaine et de notre interprétation de la littérature existante au sujet de chacun de ces outils. Nous nous sommes intéressés aux différentes fonctions mises en avant dans leurs descriptions et à leurs spécificités les uns par rapport aux autres. De cette analyse, nous avons pu dégager 5 grandes familles de critères qui constituent les 5 points d'entrée de notre grille de lecture :

- Les ressources documentaires traitées
- Les traitements pour l'annotation sémantique
- Les ontologies utilisées
- Le stockage des annotations
- L'interfaçage avec les utilisateurs, tant humains que machines

Chaque point d'entrée possède un ensemble de critères, synthétisés dans le Tableau 15 et détaillés par la suite. Un tableau de comparaison des outils en fonction de ces points d'entrée et de leurs critères est présenté à la fin de cette annexe.

Ressources	Traitement	Ontologies	Stockage	Interfaçage
Provenance de la ressource	Méthode d'annotation utilisée	Langage de modélisation des ontologies	Annotations embarquées ou débarquées	Interfaces homme-machine
Format d'entrée	Système d'extraction d'information utilisé	Ontologie utilisée	Support de stockage	Fonctionnalités supplémentaires pour l'utilisateur
Format de sortie	Apprentissage ou non	Ontologie locale ou distribuée	Format utilisé	Interopérabilité
Niveau de structuration du	Quantité de travail manuel requis	Annotation par une ou plusieurs	Base de connaissance,	Disponibilité, Open-Source

[45] http://www.aktors.org/akt/
[46] http://nlp.shef.ac.uk/dot.kom/
[47] http://www.esperonto.net/semanticportal/jsp/frames.jsp

texte Type d'information intéressante	 Niveau d'automatisation	ontologies Eléments de l'ontologie concernés	serveur d'annotations utilisé(e)	

Tableau 15. Critères pour chaque point d'entrée de la grille de lecture des outils d'annotation

- **Les Ressources Documentaires**

Les ressources documentaires à analyser proviennent soit du Web, via une adresse URL, soit d'un système de fichier local ou distribué.

Les formats d'entrée sont variés, même si dans cette étude nous ne nous attacherons qu'aux formats des documents textuels. Ces derniers sont le plus souvent aux formats HTML ou XML, surtout lorsqu'ils proviennent du Web. Mais les ressources documentaires en entreprise comprennent aussi des formats propriétaires tels que MS Word ou Adobe PDF qu'il faut également savoir traiter. Ils constituent un corpus non négligeable de ressources contenant de la connaissance cruciale pour les applications [URE 06].

Les ressources textuelles peuvent être divisées en trois ensembles selon le niveau de structuration du texte : le structuré (tableaux, bases de données), le semi-structuré (pages Web, XML) et le non structuré (texte libre). Mis à part les outils d'annotation manuelle, il est assez rare de trouver des outils sachant traiter l'ensemble de ces structures documentaires. En effet, les techniques mises en œuvre pour automatiser les processus d'annotation sont fortement dépendantes de la nature du texte analysé comme nous le verrons par la suite.

Ces techniques ne s'intéressent pas non plus aux mêmes informations contenues dans les documents analysés. La plupart d'entre elles sont capables de repérer les noms propres ou plus généralement les Entités Nommées d'un document. Ce terme est issu du domaine de l'Extraction d'Information. Il a été défini [GRI 96] dans la série des Message Understanding Conferences (MUC) organisées dans les années 90 comme regroupant à la fois les noms propres (représentant les noms de personnes, les noms d'organisation, les noms de lieux) mais aussi les dates, les montants, et autres informations aisément identifiables dans un document textuel. Plus rares sont les techniques qui permettent de repérer les attributs de ces Entités Nommées ou encore les relations exprimées entre elles, bien que ces tâches aient également fait partie des objectifs des conférences MUC.

Enfin, le format de sortie peut aussi différer d'un outil d'annotation à l'autre. Ce format est généralement dépendant du langage utilisé pour implémenter l'ontologie de référence. Plusieurs langages ont vu le jour simultanément ou successivement afin de formaliser au mieux ces ontologies (cf. chapitre précédent). Les annotations sémantiques reposent sur ces normes et standards définis dans le cadre du Web Sémantique. Leur utilisation est hautement préférable car ils permettent un interfaçage plus aisé avec les différentes applications utilisatrices des annotations ainsi générées. Dans le monde des entreprises, la conformité aux standards permet aussi de les libérer des contraintes des formats propriétaires [URE 06].

- **Les Traitements**

Comme mentionné dans la section 1.1.2, il existe trois niveaux d'automatisation de la procédure d'annotation : manuelle, semi-automatique ou entièrement automatisé. Rappelons brièvement que l'annotation manuelle est entièrement effectuée par un annotateur humain qui place les annotations de son choix dans un document existant ou en phase de rédaction. L'annotation automatisée est entièrement réalisée par un outil d'extraction d'information. L'annotation semi-automatisée utilise également un outil d'extraction d'information pour suggérer des annotations à l'utilisateur qui doit ensuite les valider manuellement. A travers les interfaces homme-machine, plusieurs niveaux d'automatisation peuvent être proposés à l'utilisateur final afin qu'il puisse opter pour une annotation manuelle ou avec l'aide d'un système d'extraction des connaissances.

L'automatisation du traitement opéré par les outils d'annotation sémantique est particulièrement importante dans l'objectif du Web Sémantique. En effet, il est primordial de faciliter l'acquisition de connaissances, surtout lorsqu'il s'agit d'annoter de grandes collections de ressources documentaires comme le Web. Les systèmes d'extraction d'information intégrés aux outils d'annotation sémantique représentent un facteur prépondérant pour la qualité et la performance de ces derniers. Ils conditionnent également les applications qui peuvent réalisées en fonction de la connaissance qu'ils sont capables d'extraire. Le niveau d'automatisation des outils d'annotation sémantique dépend principalement de l'intégration de ces systèmes d'extraction d'information dans le processus d'annotation. A titre d'exemple, les deux systèmes d'extraction d'information les plus fréquemment employés sont :

- GATE [CUN 03] qui implémente la méthode d'extraction par traitement du langage naturel grâce à un système de règles, appelé « Java Annotation Patterns Engine » (JAPE), lesquelles sont ensuite compilées sous la forme d'un automate à états-finis.
- Amilcare [CIR 03b] qui implémente un algorithme d'apprentissage supervisé basé sur la redondance du contenu dans la structure du document (listes, tableaux, etc.). Ce système repose d'ailleurs sur un ensemble de ressources linguistiques, comme des règles de reconnaissance d'entités nommées, développées dans GATE et exprimées en JAPE.

Par conséquent, il est important lorsqu'on étudie un outil d'annotation de connaître le système d'extraction d'information employé, si celui-ci utilise ou non des algorithmes d'apprentissage, supervisés ou non. Ceci détermine aussi la quantité de travail manuel requise de la part de l'annotateur humain pour annoter une nouvelle ressource documentaire mais aussi le temps d'adaptation de cet outil d'annotation à un nouveau domaine d'étude.

- **Les Ontologies**

Un autre aspect important concerne les ontologies qui formalisent les annotations générées. Ces ontologies peuvent diverger sur leur langage d'implémentation et sur le domaine modélisé. D'une part, tout le panel des langages présentés à la section 1.3 est susceptible d'être utilisé. Il est d'ailleurs possible de retracer l'évolution des outils en même temps que celle des standards. D'autre part, il est intéressant de connaître le niveau de modélisation permis par l'outil d'annotation : les annotations

sémantiques reposent-elles sur une ontologie générique et de haut niveau ou au contraire sur une ontologie de domaine, relative à un champ de connaissance bien particulier, quel qu'il soit. Selon l'ontologie sélectionnée pour annoter un document, les annotations sémantiques générées peuvent prendre la forme de simples métadonnées sur le document ou de références vers les instances des classes de cette ontologie. Elles peuvent aussi servir à constituer un véritable réseau de connaissances à partir des relations présentes entre ces instances et de leurs attributs.

Plusieurs ontologies peuvent aussi être utilisées pour annoter un même document. Les annotations doivent alors explicitement déclarer à quelle ontologie elles font référence. Ces ontologies peuvent être locales ou distribuées et accessibles par leur adresse Web. L'utilisateur humain doit pouvoir choisir, modifier et adapter la ou les ontologies utilisées afin de rendre l'outil plus flexible et plus ouvert à tout autre domaine que celui initialement prévu. Dans ce cas, les outils d'annotation doivent également être capables de gérer les modifications faites aux ontologies tant au niveau des classes que des instances. Le problème consiste à assurer la pérennisation des annotations vis-à-vis des ontologies utilisées. La conceptualisation d'un environnement d'annotation sémantique doit déterminer la manière dont les changements doivent être reflétés dans le serveur d'annotation et alerter les utilisateurs humains lorsque ces modifications créent des conflits avec des annotations existantes [URE 06]. Enfin, l'outil d'annotation doit proposer à ses utilisateurs des fonctionnalités facilitant l'exploration et l'édition des ontologies utilisées.

- **Le Stockage**

Nous avons vu section 1.1.2 que les annotations sémantiques peuvent être stockées de deux manières vis-à-vis de son document d'origine : débarquées, i.e. séparément, ou embarquées, i.e. comme partie intégrante de celui-ci comme cela est le cas dans les applications traditionnelles telles MS Word [URE 06]. Le fait de stocker les annotations sémantique séparément du contenu est plus approprié à la vision du Web Sémantique, surtout que dans certains cas il est impossible de modifier le document source car celui-ci est la propriété du site Web d'origine ou du réseau d'entreprise.

Concernant le support, les annotations sémantiques peuvent être stockées soit dans un serveur d'annotation dédié, soit au sein même de la base de connaissance, soit les deux. Alors que le stockage dans le serveur d'annotation peut être représenté sous la forme d'un triplet « décrit(annotation,document) », le stockage dans la base de connaissance est explicitement gouvernée par la modélisation de l'ontologie de référence. Ceci signifie que le document est par exemple modélisé comme une instance de cette base de connaissance et l'annotation comme une de ses propriétés. Dans les deux cas, lorsque l'annotation a pour valeur une instance de la base de connaissance ou un descripteur d'un thesaurus, elle conserve un lien vers cette référence.

Enfin, les résultats du système d'extraction d'information ayant permis la création des annotations sémantiques peuvent également être exploitées pour enrichir une base de connaissance existante. Les extractions sont alors considérées comme des instances de classes, d'attributs ou de relations, indépendamment des documents auxquelles elles réfèrent (même si une métadonnée conservant la trace du document source peut être ajoutée à ces instances). L'ensemble de ces instances dans la

base de connaissance constituent alors un réseau sémantique pouvant être interrogé et réutilisé tel quel.

- **L'Interfaçage**

Il est important de fournir des interfaces simples et conviviales aux annotateurs afin de leur simplifier le processus de l'annotation et de maximiser l'aide apportée dans leurs tâches quotidiennes. Une bonne approche serait de pouvoir leur fournir un unique point d'entrée, i.e. un environnement dans lequel les utilisateurs créent, lisent, annotent et partagent les documents [URE 06]. Malheureusement, il n'est pas toujours possible d'être à la source des documents, surtout lorsque l'utilisateur doit annoter des flux quotidiens de ressources documentaires. A ceci s'ajoute d'autres problématiques telles que la provenance des annotations, la confiance qui peut leur être accordée et leurs droits d'accès.

Afin d'être plus conviviaux et pour ainsi dire plus compétitifs, les outils d'annotation doivent être capables de proposer des fonctionnalités supplémentaires à l'utilisateur humain comme la possibilité de naviguer dans l'ontologie pour choisir les classes ou propriétés pour les annotations, de valider les suggestions des annotations proposées par les systèmes semi-automatisés, ou encore de vérifier la cohérence, l'intégrité et la validation des contraintes ou restrictions imposées par la modélisation de l'ontologie.

La possibilité d'interfacer les outils d'annotation sémantique avec divers systèmes ou applications existantes, comme des serveurs d'annotations ou bases de connaissances externes, des éditeurs d'ontologie ou de documents, etc., permet de compléter efficacement et intelligemment les fonctionnalités disponibles.

La mise à disposition des outils d'annotation en open source ou dans une version d'évaluation en ligne, i.e. téléchargeables librement, permet aux futurs utilisateurs intéressés de tester l'outil avant son utilisation et dans le cas de l'open source de participer plus activement à son amélioration. Ces démarches démontrent une volonté de la part de leurs concepteurs à communiquer leurs résultats pour faire progresser la recherche et le savoir-faire dans ce domaine.

A présent, sur la base de cette grille de lecture que nous venons de définir, nous allons présenter les divers outils d'annotation sémantique que nous avons étudiés.

1.2 Description des outils

Les ontologies sont apparues comme l'épine dorsale de l'annotation documentaire de métadonnées à partir d'applications pré-Web Sémantique comme le projet SHOE, l'initiative (KA)2 et le projet Onto-Planet, parmi d'autres. Avec l'émergence du Web Sémantique, l'annotation sémantique a été le point de mire de beaucoup de projets et d'applications. Depuis la disponibilité de contenu annoté dans le Web d'aujourd'hui est devenu l'un des challenges clefs pour la réalisation du Web Sémantique [BEN 02].

Nous allons passer en revue les outils d'annotation grâce à la grille de lecture définie ci-dessus. Toutefois, nous allons distinguer les outils purement dédiés à l'annotation sémantique dans le cadre du développement d'un Web Sémantique de ceux utilisant les annotations afin de peupler un scénario d'extraction d'information, ou mieux encore une ontologie. Néanmoins, il n'est pas impossible de retrouver certains outils à la croisée de ces deux approches, proposant aux utilisateurs humains une aide pour le peuplement d'une ontologie tout en annotant le document à partir de références vers le contenu de cette ontologie. Enfin, nous terminerons cette section par l'évocation de quelques outils récents qui n'ont pas encore atteint un degré de maturation suffisant ou convaincants pour la communauté jusqu'à présent mais qui proposent néanmoins des pistes de recherche intéressantes.

1.2.1 L'approche Web Sémantique

Tout au long de la réflexion entreprise par les chercheurs sur la vision du Web Sémantique, divers projets ont eu pour but l'annotation des pages Web soit pour extraire la connaissance contenue dans ces documents, soit pour améliorer la recherche et l'accessibilité de ces documents, soit enfin pour les relier les uns aux autres de manière sémantique et explicite. Ces outils sont pour la plupart des interfaces d'annotation manuelle. Ils ont suivi de près l'évolution des langages pour la représentation des annotations sémantiques, depuis SHOE et HTML-A jusqu'à RDF et DAML+OIL.

- **SHOE Knowledge Annotator**

Un groupe de chercheurs à L'Université de Maryland, rassemblés autour du projet Mindswap[48], ont été parmi les premiers à s'intéresser à l'annotation des pages web. Ils ont développé le SHOE Knowledge Annotator[49] (ou SHOE KA) [HEF 01] pour annoter sémantiquement les pages HTML à l'aide d'une ontologie modélisée en SHOE (Cf. section 1.3.1).

Les Ressources :

Cet outil permet d'annoter les pages HTML, qu'elles soient structurées ou non. Le format de sortie de ces annotations est donc des documents SHOE, i.e. des pages HTML dans lesquelles sont ajoutées les étiquettes XML correspondant au langage SHOE.

Les Traitements :

L'annotation s'effectue manuellement par l'utilisateur final. Il n'y a donc pas d'automatisation du processus d'annotation.

Les Ontologies :

Cet outil n'utilise pas une ontologie de référence générique. Au contraire, l'utilisateur peut annoter ses documents avec une ou plusieurs ontologies de domaine. Ces ontologies doivent être implémentées en langage SHOE et accessibles soit localement soit sur le Web via une adresse URL. Elles mettent à disposition un ensemble d'instances de classes et de relations permettant à l'utilisateur d'annoter

[48] http://www.mindswap.org/
[49] http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html

sémantiquement sa page Web, tout en corrigeant automatiquement les erreurs de syntaxe SHOE. Par contre, l'outil ne permet par de gérer les contraintes sur les valeurs autorisées pour les instances de relations, i.e. les contraintes de portée (range).

Le Stockage :

Les annotations sont internes au document annoté, c'est-à-dire que ces annotations sémantiques vont être directement ajoutées au document et celui-ci sauvegardé localement sur l'ordinateur de l'utilisateur. Cet outil n'a pas pour objectif d'enrichir une base de connaissance en créant de nouvelles instances dans l'ontologie du domaine. Il ne permet pas non plus de stocker les annotations dans un serveur d'annotations afin qu'elles puissent être recherchées et réutilisées ultérieurement.

L'interfaçage :

Cet outil est disponible soit via une applet sur le site web (mais celle-ci ne fonctionne plus à la date d'écriture de ce mémoire), soit comme une application pouvant être installée localement sur un ordinateur. Les annotations sauvegardées en format SHOE peuvent être réutilisées par d'autres applications sachant interpréter le langage SHOE comme Exposé, SHOE Search, PIQ et Semantic Search (cf. le site Web dédié à SHOE[50]).

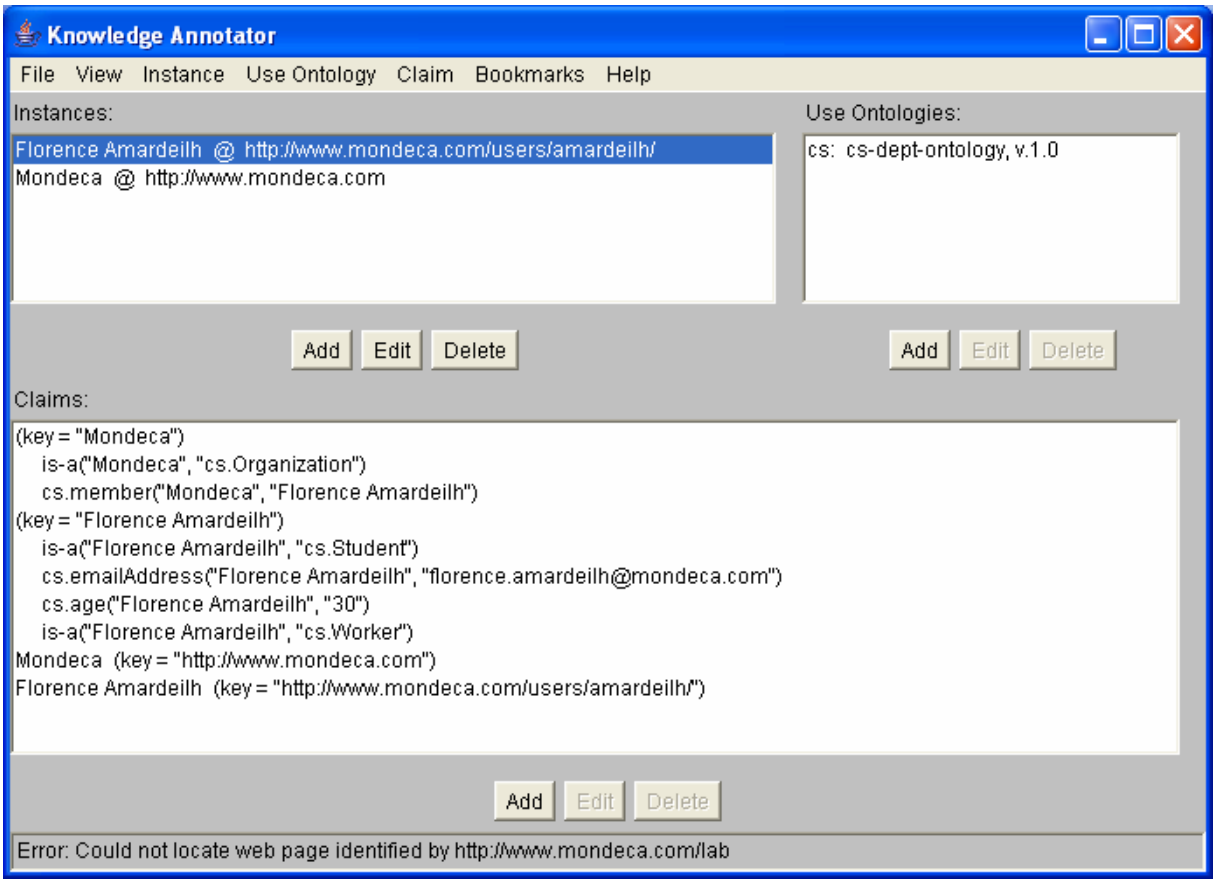


Figure 92. Exemple d'annotations sémantiques créées dans SHOE KA

[50] http://www.cs.umd.edu/projects/plus/SHOE/index.html

SHOE KA est plus un outil de démonstration qu'un outil pouvant être concrètement utilisé dans les applications du monde réel, notamment en entreprise. Néanmoins, le groupe de recherche ayant développé SHOE KA s'est rendu compte de la difficulté à mettre en pratique ce processus d'annotation manuelle. Ils ont donc dans un second temps incorporé un outil d'extraction d'information, nommé Running SHOE [HEF 01]. Running SHOE est un système d'apprentissage supervisé de la famille des wrappers, i.e. permettant l'apprentissage des règles d'extraction dans les textes structurés ou semi-structurés. Il fournit à l'utilisateur les moyens de spécifier comment extraire les instances de classes à partir de listes ou de structures régulières dans les pages annotées.

Enfin, dernièrement RDF et OWL sont devenus les langages standards pour le Web Sémantique. Le groupe de recherche a donc fait évolué SHOE KA vers une implémentation des ontologies et des annotations générées en OWL, donnant naissance à l'outil SMORE[51] [KAL 03b]. Celui-ci conserve néanmoins les caractéristiques mentionnées ci-dessus, si ce n'est que les interfaces utilisateurs sont grandement améliorées. En plus d'annoter sémantiquement des pages Web existantes, SMORE permet également de combiner l'annotation et la création de nouvelles pages HTML grâce à son éditeur HTML incorporé. Il intègre aussi l'éditeur d'ontologie SWOOP[52], également conçu par Mindswap, qui permet de naviguer dans une ontologie et de sélectionner les classes, relations ou attributs afin de créer les triplets qui seront ajoutés aux pages HTML traitées. Enfin, SMORE permet de vérifier les propriétés de domaine et de portée pour détecter les triplets invalides et en alerter l'utilisateur humain pour correction.

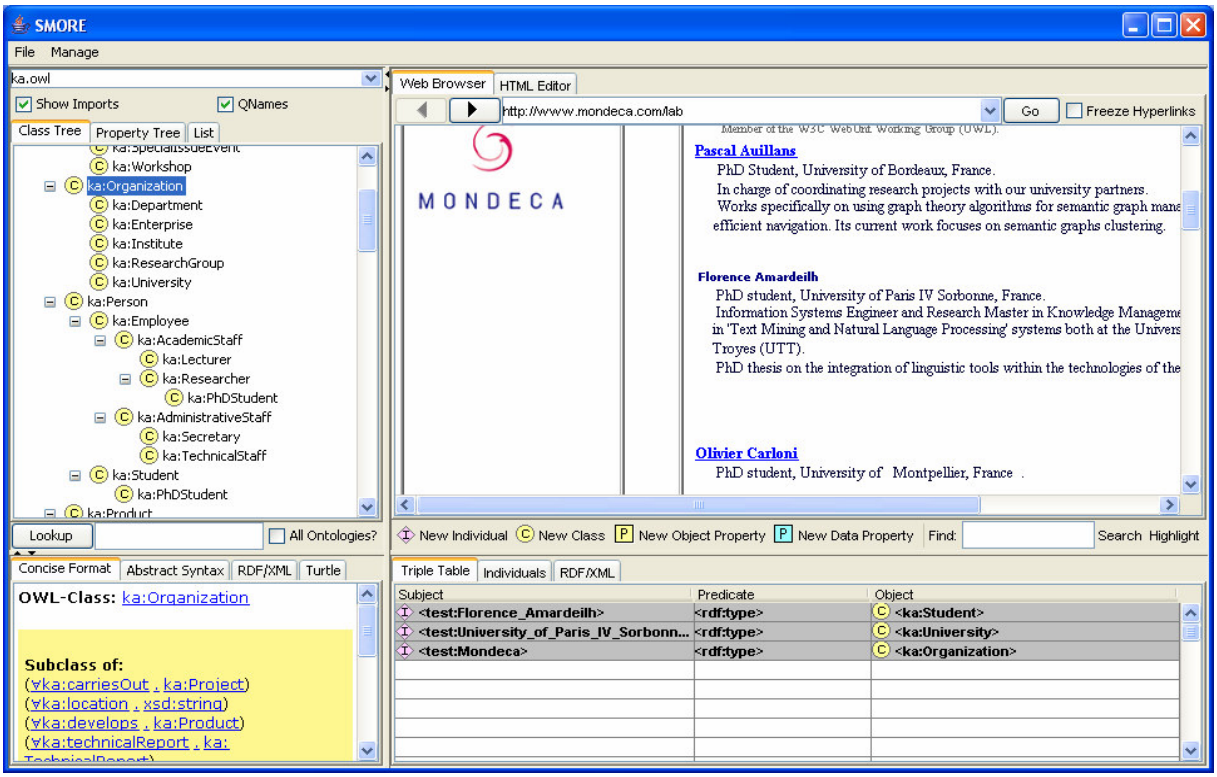


Figure 93. Exemple d'annotations sémantiques créées dans SMORE

[51] http://www.mindswap.org/2005/SMORE/
[52] http://www.mindswap.org/2004/SWOOP/

- **Annotea**

Annotea[53] [HAN 01] est le résultat d'un projet du W3C nommé LEAD (Live Early Adoption and Demonstration). Ce projet a pour objectif de démontrer comment les nouveaux langages définis par le W3C, et notamment dans ce cas RDF, peuvent être utilisés dans le cadre d'applications pour le Web Sémantique. Annotea a donc été le premier outil d'annotation de pages Web à faire usage des normes et langages pour le Web Sémantique tels que recommandés par le W3C [KAH 01].

Les Ressources :

Annotea permet d'annoter les pages Web aux formats HTML et XML et produit des annotations sous la forme de triplets RDF. Comme l'annotation est manuelle, il peut traiter n'importe quel type de contenu de page Web, du structuré au non structuré.

Les Traitements :

L'annotation se fait de manière manuelle. Il n'y a aucun outil permettant d'automatiser, même partiellement, le processus de l'annotation. L'utilisateur peut néanmoins annoter un document existant ou un document qu'il serait en train de créer par la même interface.

Les Ontologies :

Par défaut, Annotea utilise les propriétés du DublinCore, telles que l'auteur, la date, le titre, l'éditeur, etc., pour créer des métadonnées sur les documents traités. Annotea permet aussi d'annoter tout ou partie du document avec du texte libre, i.e. avec des commentaires ou remarques. L'utilisateur peut également fournir une ontologie de domaine modélisée en RDFS.

Le Stockage :

Le stockage des annotations est externe au document annoté. Ces annotations sont alors disponibles soit localement sur l'ordinateur de l'utilisateur, soit sur un serveur d'annotation RDF public. Ces serveurs d'annotation permettent de partager les annotations avec d'autres utilisateurs, encourageant ainsi l'aspect collaboratif du processus d'annotation.

L'Interfaçage :

Lorsqu'un navigateur, capable d'interpréter ces métadonnées, affiche une page Web, il regarde dans sa liste de serveurs Annotea s'il existe des commentaires associés à cette page Web. Si tel est le cas, il place des indicateurs visuels pour signaler l'emplacement de ces annotations dans la page Web affichée. Il existe plusieurs sortes de navigateurs permettant de créer et de visualiser les annotations générées sur la base d'Annotea. Parmi ceux-ci citons Amaya[54], le plus connu car également développé par le W3C [STA 01a], ou encore Annozilla[55], implémenté sur la base du navigateur

[53] http://www.w3.org/2001/Annotea/
[54] http://www.w3.org/Amaya/
[55] http://annozilla.mozdev.org/

Mozilla. D'autres variations ont également vu le jour comme Vannotea[56] qui permet d'annoter des documents multimédias.

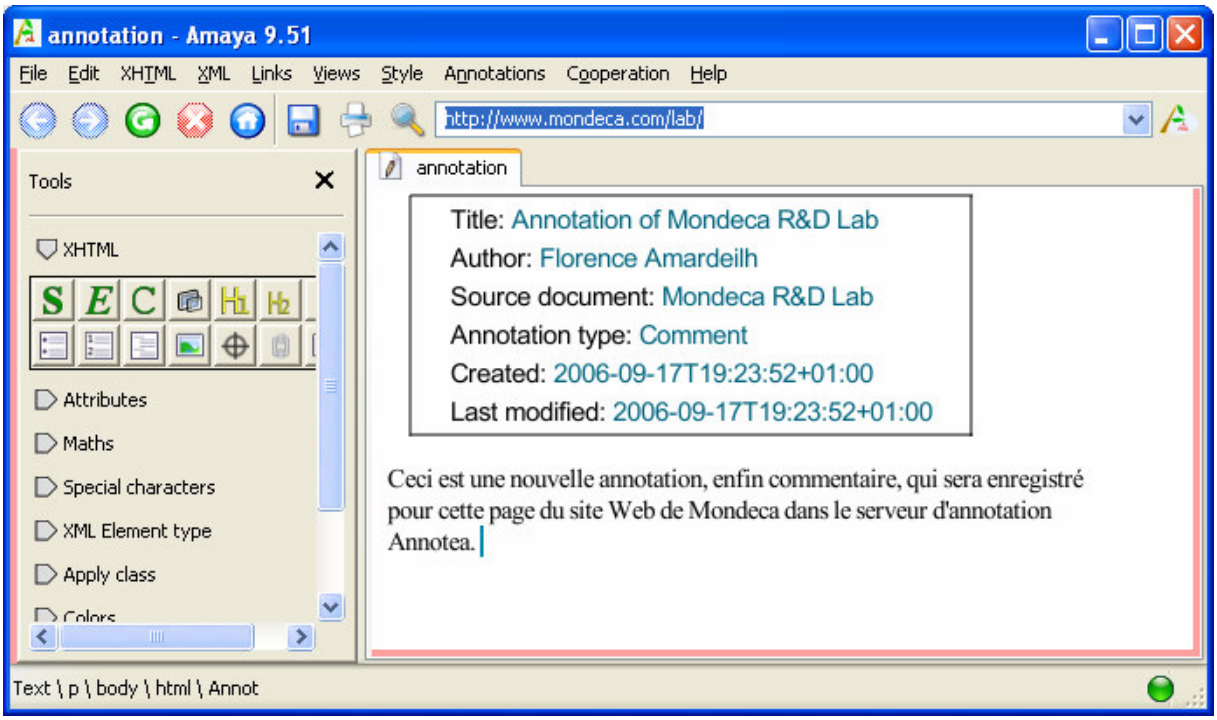


Figure 94. Exemple d'annotation créée dans Amaya, un des navigateurs d'Annotea

En dehors des métadonnées proposées par le DublinCore, Annotea ne permet pas vraiment de créer des annotations sémantiques, telles que nous l'avons défini. Par contre, tout auteur ou lecteur d'un document Web peut ajouter ses remarques ou commentaires en créant des annotations textuelles, comme cela est réalisé sur les documents papier. En fait, les annotations dans Annotea correspondent à de simples énoncés textuels d'un auteur au sujet d'un document. Annotea n'apporte aucune sémantique sur le contenu même du document annoté, ses concepts et leurs propriétés.

- **COHSE Annotator**

COHSE Annotator[57] [BEC 01] [BEC 02] [BEC 03] a été développé conjointement par l'Université de Southampton et l'Université de Manchester dans le cadre d'un projet financé par l'EPSRC (Engineering and Physical Sciences Research Council), un organisme gouvernemental anglais. L'objectif du projet était de trouver un moyen de séparer les liens hypertextes des pages Web et de rendre ces liens conceptuels, i.e. potentiellement générés à partir de l'ontologie du domaine concerné. Ainsi, les annotations sont utilisées pour créer des ancres dans le document annoté qui seront liées à

[56] http://www.itee.uq.edu.au/~eresearch/projects/vannotea/index.html
[57] http://cohse.semanticweb.org

d'autres pages Web externes. En clair, les documents annotés sont liés par des métadonnées qui décrivent leur contenu.

Les Ressources :

COHSE permet d'annoter les pages Web au format HTML et génère des annotations au format DAML+OIL.

Les Traitements :

Le processus d'annotation est manuel ou semi-automatisé par l'utilisation de lexiques. Ces lexiques permettent à l'outil d'ajouter automatiquement des ancres aux documents. Puis l'utilisateur final valide ces ancres ou en ajoute de nouvelles qui serviront de liens avec d'autres documents externes. Cet outil n'utilise pas de système d'apprentissage.

Les Ontologies :

Les ontologies modélisent un domaine de référence et sont implémentées en OIL ou DAML+OIL. Ces ontologies sont accessibles localement ou bien à partir de leur adresse URL sur le Web. Seules des annotations correspondant à des instances de concepts peuvent être créées. L'outil ne permet pas de créer des relations entre concepts ni de déterminer la valeur des attributs. Il n'est évidemment pas possible de contrôler les valeurs possibles pour ces relations ou attributs, ni de contrôler les contraintes.

Le Stockage :

Les annotations peuvent être sauvegardées aussi bien à l'intérieur du document annoté qu'à l'extérieur dans un serveur d'annotation comme Annotea ou tout autre serveur d'annotation sachant stocker des annotations au format DAML+OIL. Par contre, cet outil ne permet pas d'enrichir une base de connaissance existante.

L'Interfaçage :

Cet outil s'intègre avec les navigateurs Mozilla et internet Explorer sous la forme de plugins. Il est interopérable avec les serveurs d'annotations comme Annotea. Son interface permet de naviguer dans la taxonomie des concepts afin de créer les annotations.

Néanmoins, si COHSE permet de lier une page Web externe à une entité, caractérisée par un concept de la taxonomie, il ne permet pas pour autant la désambiguïsation de la nature de ce lien d'un point de vue sémantique.

Nous ne sommes malheureusement pas en mesure de présenter une copie d'écran d'annotation créée dans COHSE car nous n'avons pas réussi à pleinement installer cette application.

- **AeroDAML**

Créé par Ubot, AeroDAML [KOG 01] est un outil d'annotation sémantique qui utilise des techniques de TAL pour extraire l'information contenue dans les pages Web et générer automatiquement des annotations en DAML.

Les Ressources :

Les documents à annoter sont au format HTML, qu'ils soient accessibles localement ou sur le Web. Les annotations sont générées en format DAML en sortie. AeroDAML s'intéresse aux entités nommées comme les lieux, les personnes et les organisations, et à leurs relations.

Les Traitements :

Le processus d'annotation d'AeroDAML est entièrement automatisé grâce à l'utilisation d'AeroText, un outil d'extraction d'information basé sur le traitement du langage naturel. Les patrons d'extraction des entités nommées et de leurs relations doivent avoir été créés manuellement par un expert du domaine. Il n'utilise pas de système d'apprentissage.

Les Ontologies :

Dans sa version de démonstration sur son site Web, AeroDAML utilise une ontologie de référence qui est celle de WordNet, mais il est possible d'utiliser des ontologies de domaine lorsque l'application est installée localement. Les classes et propriétés de ces classes servant à créer les annotations sont modélisées en DAML.

Le Stockage :

Les annotations sont directement insérées dans le document annoté et présenté à l'utilisateur. Celui-ci peut alors stocker le document et ses annotations localement sur son ordinateur. AeroDAML peut aussi se connecter à un serveur d'annotation comme Annotea pour stocker ses annotations.

L'Interfaçage :

AeroDAML existe sous deux formes : une application web de démonstration et une application à installer localement sur un ordinateur. Il est possible de l'interfacer avec Annotea ou Amaya pour stocker les annotations créées automatiquement par AeroText.

Après la publication du langage OWL comme langage de référence pour la définition d'ontologies dans les applications Web Sémantique, Ubot a sorti une nouvelle application appelée AeroSwarm. Cet outil fonctionne exactement comme AeroDAML, avec un traitement automatisé par AeroText. Mais les ontologies ainsi que les annotations générées automatiquement sont créées au format OWL.

Nous ne sommes malheureusement pas en mesure de présenter une copie d'écran d'annotation créée dans AeroDAML ou AeroSwarm car le site Web permettant de tester et de télécharger ces applications a été mis hors service.

- **CREAM**

CREAM, ou Create RElational Annotation-based Metadata, [HAN 01] [HAN 03b] est un cadre de travail défini par l'Institut AIFB[58] de l'université de Karlsruhe. Il spécifie les composants requis par tout système d'annotation en général. Pour ses auteurs, un outil d'annotation se doit d'inclure : une interface d'annotation manuelle, un système de gestion de documents et un serveur d'annotation. Ils ont d'ailleurs par la suite implémenté deux outils à partir de ce cadre de travail : Ont-O-Mat [HAN 01], renommé depuis OntoMat-Annotizer[59] [VOL 04], une version de démonstration des technologies pouvant être utilisées pour l'annotation et OntoAnnotate [STA 01b] [STA 01a] la version commercialisée par la société Ontoprise GmbH[60]. Ces deux outils possèdent les mêmes caractéristiques suivantes.

Les Ressources :

Les documents à annoter sont de deux types : soit ils sont en cours de création, soit ils existent déjà et dans ce cas sont accessibles localement ou via le Web. En entrée, ces documents sont au format HTML ou XML et en sortie, les documents annotés sont au format DAML+OIL.

Les Traitements :

La première version de CREAM est une version entièrement manuelle permettant de rédiger de nouveaux documents et de leur associer des annotations basées sur une ontologie de référence. Nous verrons par la suite que ce cadre de travail a été étendu pour pouvoir aider les utilisateurs à annoter de manière semi-automatique leurs documents, cf. S-CREAM.

Les Ontologies :

Les ontologies de référence ne sont pas des ontologies génériques, mais des ontologies de domaine chargées par l'utilisateur en fonction de ses besoins. Ces ontologies peuvent être accessibles localement ou sur le Web. Elles modélisent au format DAML+OIL les classes, attributs et relations servant à l'annotation. Le composant « Annotation Inference Server », i.e. le serveur d'annotation recommandé par CREAM, permet de contrôler la cohérence des annotations créées, de fournir une liste des valeurs possibles pour les relations et ainsi d'améliorer le contrôle des contraintes modélisées dans l'ontologie de référence.

Le Stockage :

Les annotations sont à la fois stockées dans le document qui sera sauvegardé dans le composant de gestion de documents et dans le serveur d'annotation mentionné ci-dessus. Il n'y a pas d'enrichissement de base de connaissance en tant que telle.

L'Interfaçage :

[58] http://www.aifb.uni-karlsruhe.de/
[59] http://annotation.semanticweb.org/ontomat/index.html
[60] http://www.ontoprise.de/content/index_eng.html

Ce cadre de travail étant conçu pour être totalement ouvert, il est possible d'intégrer les différents composants recommandés entre eux. C'est le cas des deux implémentations mentionnées précédemment qui s'intègre avec des serveurs d'annotations existants comme Annotea ou OntoBroker developpé par Ontoprise GmbH.

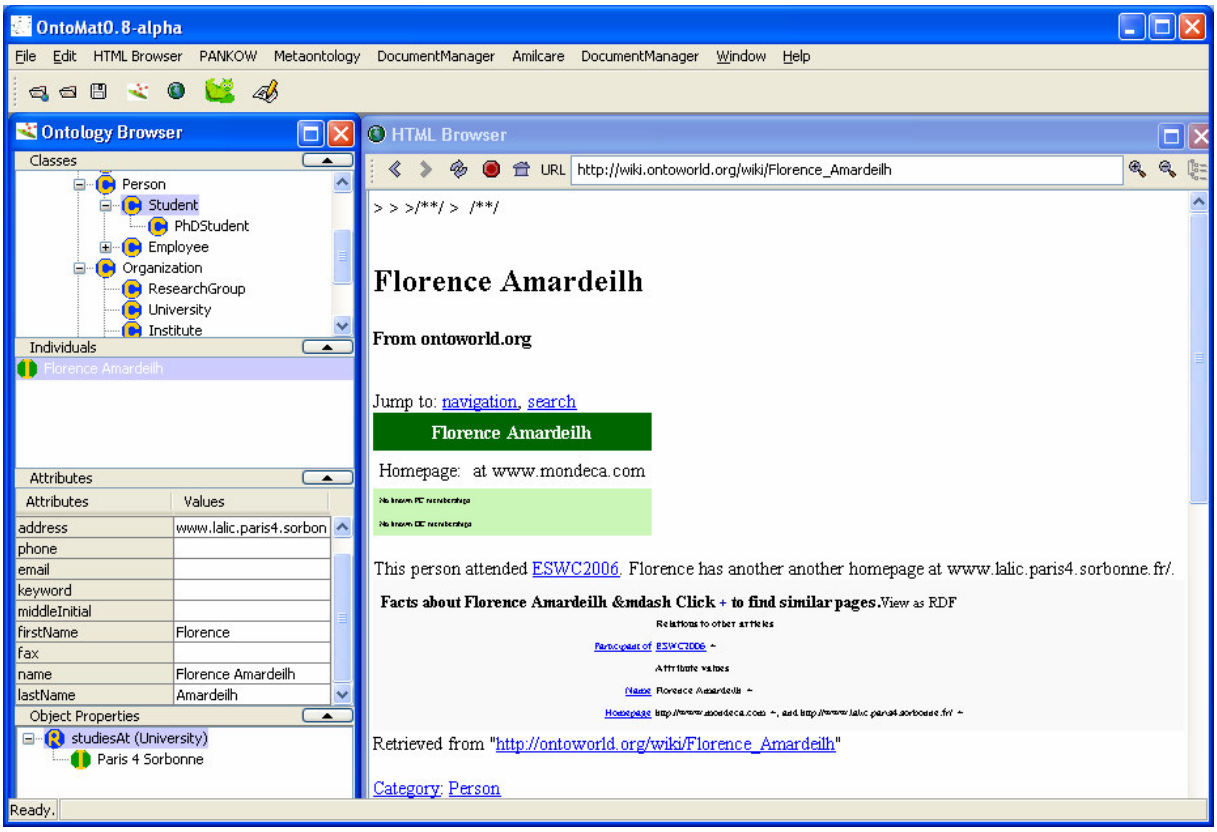


Figure 95. Exemple d'annotations sémantiques créées dans OntoMat-Annotizer

Au cours des dernières années, CREAM et ses implémentations ont su évoluer et s'adapter au monde du Web Sémantique. Par exemple, les ontologies et les annotations générées ne sont plus au format DAML+OIL, mais au format OWL. D'autre part, ils ont développé différentes méthodes pour aider l'utilisateur à créer automatiquement ou semi-automatiquement des annotations plutôt que ce processus soit entièrement manuel. Parmi ces méthodes, citons S-CREAM, OntoMat-DeepAnnotatizer et PANKOW.

Premièrement, S-CREAM, ou Semi-automatic Creation of RElational Annotation-based Metadata, [HAN 02] est en fait une extension du cadre du travail défini par CREAM permettant d'ajouter un composant pour aider l'utilisateur à annoter non plus uniquement de manière manuelle les documents mais aussi de manière semi-automatique. La version d'OntoMat-Annotizer a donc été complétée par l'intégration du système d'apprentissage supervisé Amilcare. L'utilisateur peut entraîner l'algorithme d'Amilcare à travers les interfaces d'annotation manuelle d'OntoMat, corriger ses erreurs, puis valider

les propositions d'annotations fournies par le système avant de les enregistrer dans le serveur d'annotation ou localement.

Deuxièmement, l'idée derrière OntoMat-DeepAnnotatizer [VOL 04] est de pouvoir annoter semi-automatiquement les documents structurés dont le contenu provient directement des bases de données sous-jacentes. Ainsi, au lieu d'annoter les documents eux-mêmes, ils ont mis en place une méthode permettant d'aligner les champs pertinents du schéma de la base de données à analyser avec les classes, attributs et relations de l'ontologie. L'utilisateur intervient dans la construction de cette correspondance et valide ensuite les annotations créés automatiquement par le système.

Troisièmement, le fait d'apporter une solution semi-automatisée à l'annotation sémantique n'était pas suffisante, et le composant PANKOW a été développé pour y remédier [CIM 04]. L'idée est de combiner l'annotation linguistique par patrons d'extraction permettant d'identifier certaines relations entre entités nommées avec le vaste contenu du Web. Les extractions linguistiques trouvées dans le document analysé sont envoyées à un moteur de recherche qui exploite la redondance de l'information disponible sur le Web pour valider automatiquement les nouvelles annotations pertinentes. PANKOW combine donc une méthode d'analyse linguistique à une méthode d'apprentissage non supervisée basée sur les résultats statistiques calculés par le moteur de recherche interrogé.

Nous ferons remarquer que ces trois méthodes sont assez performantes sur des documents structurés ou semi-structurés, mais insuffisantes pour annoter des documents textuels non structurés.

Pour terminer, les équipes travaillant sur ces outils se sont également intéressés à l'annotation de documents multimédias et ont adapté OntoMat-Annotizer pour créer une interface d'annotation manuelle d'images et de vidéos, appelée M-OntoMat-Annotizer [BLO 05].

- **MagPie**

MagPie[61] [DZB 03] [DOM 03] est également un outil d'annotation réalisé dans le cadre d'un projet AKT, par une équipe de l'Open University. Cet outil a pour objectif d'annoter en « temps réel » des pages web directement dans un navigateur comme Internet Explorer. Il surligne automatiquement par différentes couleurs les chaînes de caractères relatives aux entités d'une ontologie prédéfinie par l'utilisateur.

Les Ressources :

Les documents analysés sont des pages HTML, directement saisies dans un navigateur Web. L'outil fournissait des annotations HTML-A dans sa première version, mais à présent les annotations sont des triplets RDF décrivant les entités nommées repérées dans le document analysé.

Les Traitements :

[61] http://www.aktors.org/technologies/magpie/

Le processus d'annotation est entièrement automatisé grâce à l'intégration de MnM, et plus spécifiquement de son système d'apprentissage Amilcare. Les entités nommées contenues dans les documents sont repérées par un système de reconnaissance des entités nommées ESpotter, qui utilise toutes sortes de lexiques. Ces entités nommées sont exploitées par Amilcare pour construire les règles d'extraction utilisées par le wrapper. Puis les extractions sont utilisées pour enrichir la base de connaissance de l'ontologie de référence et insérer les annotations dans le document d'origine. Dans le navigateur Web, ces annotations seront coloriées en fonction des classes instanciées de l'ontologie.

Les Ontologies :

Les ontologies dans MagPie sont des ontologies de domaine, représentées en RDFS, DAML+OIL ou encore OCML. Ces ontologies sont peuplées avec des instances de classes et de relations, provenant de bases de données diverses, avant même l'utilisation de l'outil pour l'annotation. Ces instances existantes seront utilisées par le système d'extraction d'information comme ressources lexicales pour repérer les entités nommées dans les textes.

Le Stockage :

Les annotations sont à la fois insérées dans le document d'origine pour être exploitées par le navigateur Web et utilisée pour l'enrichissement de la base de connaissance de l'ontologie du domaine. Les nouvelles instances servent à enrichir les lexiques utilisés par le module de reconnaissance des entités nommées, ESpotter.

L'Interfaçage :

Nous n'avons pu tester les interfaces utilisateurs. Par ailleurs, MagPie s'intègre avec MnM, ainsi que d'autres outils développés dans le cadre des projets AKT, pour automatiser l'annotation des documents.

Une version de démonstration est accessible sur leur site Web[62], mais ne fonctionnait pas au moment de mon étude. Nous sommes toujours dans la catégorie des démonstrateurs et autres outils expérimentaux issus de la recherche. MagPie a pour vocation d'être plus un navigateur Web Sémantique, comme Internet Explorer pour le Web, qu'un réel outil d'annotation.

- **SemTag**

SemTag[63] [DILL 03a] [DILL 03b] a été développé à la Standford University[64] dans le but de fournir un moyen d'annoter quantités de pages Web à la fois et non pas un fonctionnement document par document, comme cela était le cas jusqu'à présent. L'objectif est ensuite de produire un entrepôt

[62] http://plainmoor.open.ac.uk/magpie/
[63] http://tap.stanford.edu/semtag/index.html
[64] http://www.stanford.edu/

public d'annotations que des agents du Web puissent requêter via des APIs pour demander à voir les annotations d'une page Web donnée.

Les Ressources :

SemTag se donne l'annotation de l'ensemble des pages HTML du Web pour but. Il veut pouvoir offrir une solution à la vision d'un Web Sémantique créé à partir de l'annotation des pages du Web actuel. Les annotations de SemTag sont générées au format RDF à partir d'un ensemble de termes descripteurs d'une taxonomie donnée (et non à partir des instances d'une base de connaissance).

Les Traitements :

Le processus d'annotation est entièrement automatisé et utilise un algorithme d'apprentissage supervisé appelé « Taxonomy-Based Disambiguation ». Cet algorithme se définit en trois phases. Premièrement le système examine les mots du document, grâce à un outil nommé Seeker, pour trouver des concordances avec les descripteurs de la taxonomie utilisée. Deuxièmement, un échantillon du corpus est étudié pour calculer la distribution des termes pour chaque descripteur de la taxonomie et par conséquent leur degré de similarité. Troisièmement, les termes proposés sont désambiguïsés puis stockés dans un serveur d'annotation.

Les Ontologies :

Il ne s'agit pas ici d'une ontologie véritablement, mais plutôt d'une taxonomie nommée TAP qui couvre un ensemble d'information lexicale et taxonomique ayant pour sujet la musique, les films, les auteurs, le sport, la santé, etc. Il n'y a pas d'attributs ni de relations modélisées.

Le Stockage :

Les annotations ne sont pas stockées dans le corps même du document, mais séparément dans un serveur d'annotation, appelé un « Label Bureau ». Avec les annotations sont également enregistrés l'URL du document, le segment textuel d'origine ainsi que d'autres métadonnées du document.

L'Interfaçage :

Nous n'avons pu tester l'outil à cause de l'impossibilité de se connecter à leur site Web. SemTag comporte apparemment un ensemble d'APIs permettant de s'interfacer avec des agents du Web voulant obtenir les annotations de pages Web données.

SemTag est la première tentative d'annotation du Web dans l'optique de la concrétisation de la vision du Web Sémantique où toutes les pages Web seraient annotées sémantiquement. Or s'il s'agit d'un bel effort, l'annotation sémantique n'est que partielle puisqu'elle fait référence à une taxonomie et non à une véritable ontologie.

1.2.2 L'approche Acquisition des Connaissance

- **MnM**

Au Knowledge Media Institute[65] (KMI) de l'Open University, un environnement appelé MnM[66] [VAR 01] [VAR 02a] [VAR 02b] a tout d'abord été créé pour manuellement annoter un corpus d'apprentissage afin de nourrir un système d'apprentissage avec ce corpus d'entraînement. Le système d'apprentissage en question est Amilcare dont nous avons déjà parlé précédemment. MnM permet de créer deux sortes de règles : l'annotation, i.e. l'insertion d'étiquettes sémantiques dans le texte d'origine, et la correction, i.. l'insertion d'information modifiant l'emplacement de ces étiquettes en fonction des retours utilisateur. Suite à l'importance prise par les systèmes d'annotation dans le cadre du Web Sémantique, MnM a été détourné de sa fonction première pour plus largement permettre l'annotation manuelle et semi-automatisée de ressources du Web.

Les Ressources :

Seuls les fichiers HTML sont pris en compte par cet outil, qu'ils soient accessibles localement ou via une URL sur le Web. En sortie, les documents sont enregistrés en HTML ou XML ou comme de simples fichiers textes.

Les Traitements :

Le processus d'annotation dans MnM était premièrement manuel puisque l'objectif consistait à pouvoir annoter un corpus pour entraîner un système d'apprentissage en mode supervisé. Puis, avec les avancements du Web Sémantique, finalement MnM représente à la fois une interface pour aider l'utilisateur à construire son système d'extraction d'information et une interface pour valider les résultats du système d'extraction d'information pour peupler une ontologie existante. Le processus d'annotation est donc à la fois manuel et semi-automatisé dans sa version finale, faisant appel à Amilcare comme outil d'extraction d'information. Le mode d'apprentissage de cet outil est supervisé par l'utilisateur. Le travail manuel de l'utilisateur pour définir les règles d'extraction et leurs validations/corrections reste toutefois important.

Les Ontologies :

Les ontologies chargées dans MnM sont des ontologies de domaine accessibles soit localement soit sur un serveur externe, en l'occurrence celui de WebOnto. Dans la première version de MnM, les ontologies étaient modélisées en OCML. Puis avec l'arrivée des standards du Web Sémantique, elles sont à présent modélisées en DAML+OIL ou en RDF. Il est possible d'annoter les documents en utilisant les classes de l'ontologie mais aussi leurs attributs et relations. Par contre, MnM ne permet pas de lister les valeurs possibles pour les attributs et les relations, ni de contrôler les contraintes modélisées dans l'ontologie de référence. Les valeurs des instances sont contrôlées uniquement lorsqu'elles sont transmises à la base de connaissance pour vérifier que cette instance n'existe pas déjà.

[65] http://kmi.open.ac.uk/
[66] http://kmi.open.ac.uk/projects/akt/MnM/

Le Stockage :

Les documents annotés peuvent être enregistrés localement comme des fichiers HTML, XML ou plein texte mais ils ne sont pas compatibles avec les documents du Web Sémantique, adoptant un format RDF par exemple. Les annotations sont néanmoins utilisées pour enrichir une base de connaissance liée à l'ontologie de domaine de référence, i.e. pour peupler cette ontologie. L'ontologie sert tout simplement de patron d'extraction.

L'Interfaçage :

MnM présente une interface utilisateur dans laquelle le document à annoter est présenté dans un navigateur HTML, mais il n'est pas possible de créer de nouveaux documents. L'interface permet également de naviguer dans l'ontologie de référence. MnM s'intègre avec le serveur d'ontologie WebOnto et possède une API permettant d'intégrer d'autres systèmes d'extraction d'information qu'Amilcare.

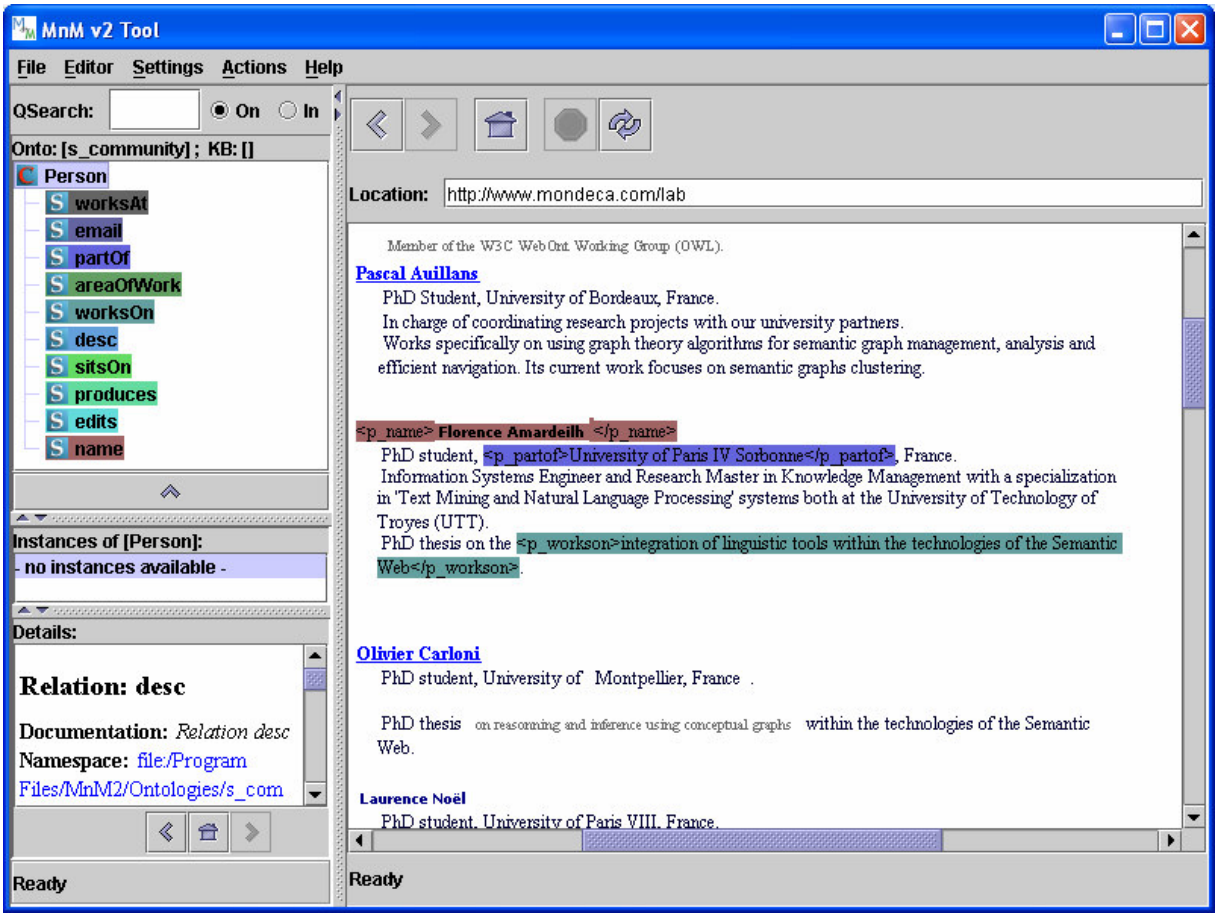


Figure 96. Exemple d'annotations sémantiques créées dans MnM

- **Melita**

Melita[67] [CIR 02] [DIN 03a] est réalisé au sein de l'équipe Natural Language Processing[68] de l'Université de Sheffield, dans le cadre du projet AKT[69] financé par le gouvernement anglais. Melita n'est pas vraiment un outil d'annotation pure, mais a plutôt été developpé dans le but de démontrer comment il était possible d'interagir avec un outil d'extraction d'information, un peu comme MnM.

Les Ressources :

Les documents à annoter doivent être au format HTML ou plein texte. Les documents annotés ne sont pas générés dans un format conforme aux standards du Web Sémantique, mais de simples tags XML sont ajoutés aux documents originaux.

Les Traitements :

Le processus d'annotation est à la fois manuel et semi-automatique. Il utilise un système d'apprentissage supervisé, Amilcare. Mais contrairement aux autres outils utilisant également Amilcare, Melita ne choisit pas son corpus d'entraînement au hasard. Melita possède un algorithme permettant de sélectionner automatiquement les documents semblant être les plus pertinents à l'apprentissage des règles d'extraction concernant le domaine de référence. L'utilisateur doit tout de même fournir les annotations de départ permettant au système d'apprendre à annoter, puis il doit également corriger et valider les propositions d'annotations fournies par le système.

Les Ontologies :

Les ontologies de références sont des ontologies de domaine et non des ontologies génériques. Elles sont modélisées en RDFS ou DAML+OIL. Seules les classes sont exploitées pour annoter les documents. Il n'y a donc pas de contrôle sur les valeurs possibles des attributs et relations puisque ceux-ci ne sont pas pris en compte.

Le Stockage :

Il n'y a pas de stockage des annotations dans un serveur d'annotation externe, ni d'enrichissement de base de connaissance. Les annotations sont utilisées pour créer des étiquettes XML à partir du nom des classes de l'ontologie dans le document d'origine.

L'Interfaçage :

Melita dispose d'une interface utilisateur assez intuitive où sont affichés d'un côté le document à annoter et de l'autre l'ontologie afin de sélectionner les classes utilisées pour l'annotation. Melita ne s'interface pas avec d'autres outils.

[67] http://nlp.shef.ac.uk/melita/
[68] http://nlp.shef.ac.uk/
[69] http://www.aktors.org/akt/

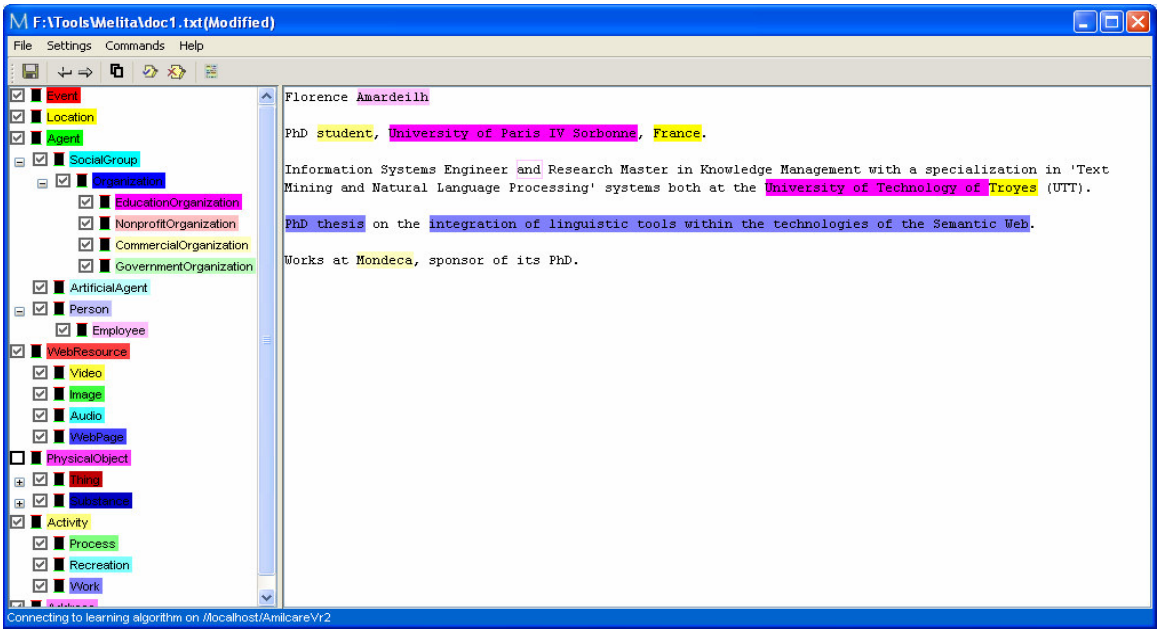


Figure 97. Exemple d'annotations sémantiques créées dans Melita

Melita est réellement un démonstrateur de technologies et méthodes. L'annotation semi-automatisée s'applique surtout à des documents structurés voire semi-structurés et non à des documents de textes libres. Cet outil ne peut donc être utilisé dans des cas réels, surtout dans le monde de l'entreprise, néanmoins son interface utilisateur pour l'annotation est très intéressante au point de vue simplicité et convivialité.

- **Armadillo**

Armadillo[70] [CIR 03a] [DIN 03b] [CIR 04] est aussi développé par l'équipe Natural Language Processing de l'Université de Sheffield, et également dans le cadre du projet AKT. Armadillo a pour but l'enrichissement automatique d'une base de connaissance donnée à partir des informations pertinentes à un domaine extraite de pages Web ayant une structure régulière comme les listes, les tableaux ou encore des textes dont l'information est structurée identiquement (annonces de séminaires par exemple).

Les Ressources :

Les pages Web analysées sont au format HTML et peuvent être accessibles soit localement, soit directement mappées à partir d'une base de données servant à les générer. Cet outil annote donc uniquement les pages Web ayant une structure de document hautement régulière comme les tables ou les listes d'items.

Les Traitements :

Le processus d'annotation d'Armadillo est entièrement automatisé. Il utilise pour cela un système d'apprentissage supervisé, basé sur Amilcare. Mais au lieu de faire intervenir l'utilisateur afin de créer

[70] http://nlp.shef.ac.uk/wig/armadillo_home.html

dans une interface utilisateur le corpus annoté servant à l'entraînement du système, Armadillo utilise un automate à états finis pour initier l'annotation des entités nommées. Ensuite, Amilcare prend le relais pour apprendre à partir de ces premières annotations. Enfin, les résultats d'Amilcare sont envoyés à des moteurs de recherche sur le Web. Les résultats de ces recherches seront analysés statistiquement grâce à la redondance de l'information sur le Web pour déterminer quelles propositions fournies par Amilcare sont les plus pertinentes dans le domaine de référence. L'utilisateur n'intervient que pour fournir l'URL de la page Web à annoter ainsi que pour compléter les suggestions fournies par le système.

Les Ontologies :

Les ontologies de référence ne sont pas génériques, ce sont des ontologies de domaine modélisées en RDFS.

Le Stockage :

Les informations extraites par Armadillo sont automatiquement stockées dans une base de connaissance. Celle-ci peut ensuite être recherchée pour ajouter des annotations sous la forme de triplets RDF au document d'origine.

L'Interfaçage :

Il n'a pas été possible de tester les interfaces utilisateurs, si celles-ci existent. La recherche dans la base de connaissance se fait grâce à une connexion par des web services externes.

Nous n'avons pu télécharger une version d'Armadillo afin de tester cet outil. Mais il nous semble que là encore, il s'agit plus d'un outil de démonstration que d'un véritable outil d'annotation. D'ailleurs, plus qu'un outil d'annotation, il s'agit en fait d'un outil d'extraction d'information non supervisé qui stocke les éléments extraits dans une base de connaissance. Par ailleurs, seules les pages ayant une forte structuration ou issues de bases de données peuvent être analysées par cet outil, un peu à la manière de OntoMat-PANKOW vu précédemment.

- **ArtEquAkt**

Pour continuer avec les projets financés par l'AKT, un autre projet intéressant est ArtEquAkt[71] [KIM 02] [ALA 03], développé par l'Université de Southampton. Il a pour objectif d'implémenter un système capable de rechercher à travers le Web les ressources concernant les biographies d'artistes afin d'en extraire les informations pertinentes et de produire automatiquement des résumés.

Les Ressources :

[71] http://www.aktors.org/technologies/artequakt/

Les documents analysés sont des pages HTML. Les annotations seront générées sous la forme de triplets RDF. Les informations pertinentes à extraire concernent les entités nommées et les relations extraites dans le domaine de la Peinture.

Les Traitements :

ArtEquAkt utilise un système entièrement automatisé basé sur l'approche des automates à états finis. Pour cela, il combine le thesaurus de WordNet à l'outil d'extraction d'information GATE. Le processus d'annotation réalise toute une série d'analyses linguistiques fines à partir des termes du thesaurus trouvés dans les documents analysés. Néanmoins, un expert en linguistique doit tout d'abord définir les règles d'extraction linguistiques concernant le domaine étudié.

Les Ontologies :

Dans ArtEquAkt, il n'y a qu'une ontologie, non générique mais du domaine de l'Art et plus particulièrement de la Peinture. Elle a été développée tout spécialement par les membres du projet. Elle permet de modéliser les classes, attributs et relations de ce domaine.

Le Stockage :

Les annotations extraites ne sont pas stockées dans le document d'origine mais dans une base de connaissance implémentée dans l'outil Protégé. Toutes les annotations sont enregistrées dans la base de connaissance, mais ensuite un outil de consolidation de l'information est employé pour identifier les instances potentiellement identiques et les fusionner. Cette fusion des instances prend en considération les cas où la même instance possède soit les mêmes attributs, soit des attributs différents qu'il sera nécessaire de garder.

L'Interfaçage :

Nous n'avons pas pu tester les interfaces utilisateurs. Par ailleurs, la base de connaissance stockant les annotations est couplée avec un moteur de génération de langage naturel capable. Ce dernier est capable de rendre à l'utilisateur final des biographies d'artistes sur la base des annotations stockées dans la base de connaissance.

Une version de démonstration est accessible sur leur site Web[72], mais ne fonctionnait pas lors de mon étude. Par conséquent, je n'ai pu tester cet outil. L'aspect important de ce projet n'est pas l'annotation en elle-même, trop spécifique au cas particulier du projet, mais c'est surtout sa capacité à consolider l'information une fois les annotations enregistrées dans la base de connaissance (cf. §4.2.2). Ce problème hautement important est rarement traité par les autres outils.

[72] http://www.artequakt.ecs.soton.ac.uk/demo/

- **Knowledge & Information Manager (KIM)**

KIM[73] [POP 03] [KIR 05] a été développé par OntoText[74], le laboratoire R&D de la société SIRMA. Il s'agit d'une plateforme pour à la fois annoter des documents et peupler l'ontologie de référence. Cette plateforme possède plusieurs outils pour l'annotation, notamment un plugin pour le navigateur Internet Explorer et une interface pour lancer le processus semi-automatisé d'annotation.

Les Ressources :

Les documents à annoter sont aussi bien des pages Web au format HTML que de simples documents textuels. Par contre, les documents annotés s'ils conservent leurs formats d'origine contiennent les annotations au format RDF. Dans KIM, les annotations ont pour cible les entités nommées et les relations de haut niveau entre ces entités nommées.

Les Traitements :

Le processus d'annotation est semi-automatique. KIM intègre l'outil d'extraction d'information GATE, mettant en œuvre un ensemble d'analyses linguistiques permettant l'écriture des règles d'extraction liées au domaine de référence. L'écriture de ces règles est effectuée par un expert du domaine avant de pouvoir utiliser cet outil. Puis, le système extrait les informations des documents analysés et propose ses suggestions à l'utilisateur final pour validation et correction manuelle. Dans un récent projet nommé SEKT, l'équipe ayant développé KIM a également travaillé sur une intégration avec Amilcare, l'autre outil d'extraction d'information largement utilisé dans les projets européens, comme nous l'avons vu précédemment [MAN 06].

Les Ontologies :

KIM utilise une ontologie générique, nommée KIMO (KIM Ontology) dans sa première version puis PROTON. Cette ontologie, disponible aux formats RDFS et OWL, modélise des concepts de haut niveau comme « Personne », « Lieu » ou « Organisation » qui correspondent aux entités nommées pouvant être extraites des documents. PROTON modélise aussi un ensemble de relations entre ces concepts comme travaille(Personne, Organisation) ou est_située(Organisation, Lieu), etc. Par contre, il est impossible de charger une ontologie de domaine dans l'outil.

Le Stockage :

Les annotations sont à la fois stockées directement dans le document annoté ainsi que dans un serveur d'annotation RDF construit sur la base des APIs offertes par Sesame. Ces annotations servent également à enrichir la base de connaissance de PROTON. Cette base de connaissance est pré-remplie avec des listes d'instances représentant les entités nommées comme Personne, Lieu, organisation, etc. Ceci permet d'alimenter les lexiques utilisés par GATE. L'ontologie et la base de connaissance sont également stockées dans l'entrepôt RDF.

L'Interfaçage :

[73] http://www.ontotext.com/kim/
[74] http://www.ontotext.com/index.html

KIM possède deux types d'interfaces : la première est un plug-in Internet Explorer qui permet de charger une page Web et de visualiser les annotations générées automatiquement, cf. Figure 98, et la seconde est l'interface permettant d'annoter un ensemble de documents et d'enrichir automatiquement la base de connaissance de PROTON, cf. Figure 99. De plus, outre l'entrepôt de triplets RDF Sesame, KIM s'interface aussi avec le moteur d'indexation et de recherche Lucene. Ces deux outils permettent l'accès aux annotations et à la connaissance extraite par KIM.

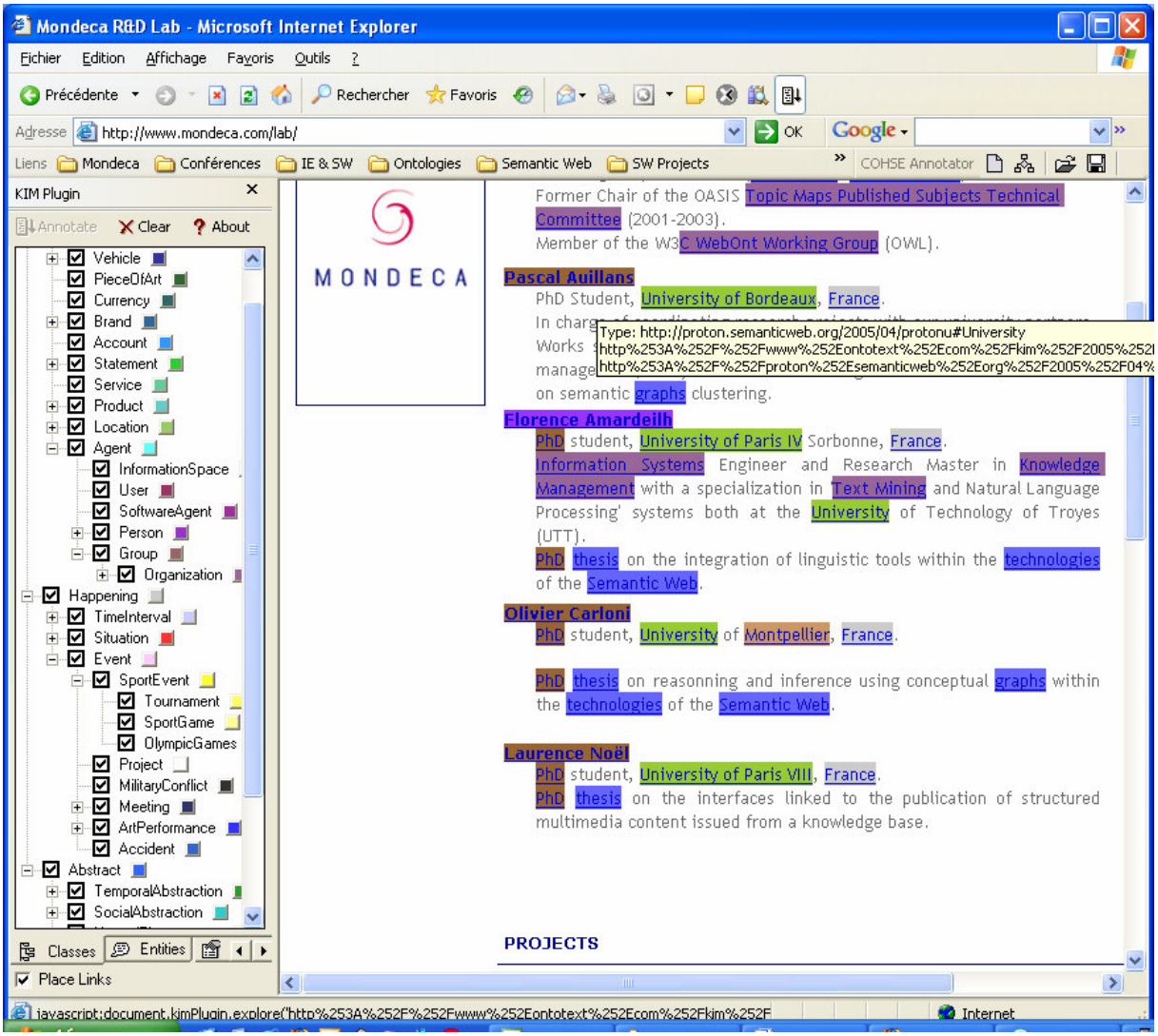


Figure 98. Exemple d'annotations sémantiques créées par KIM

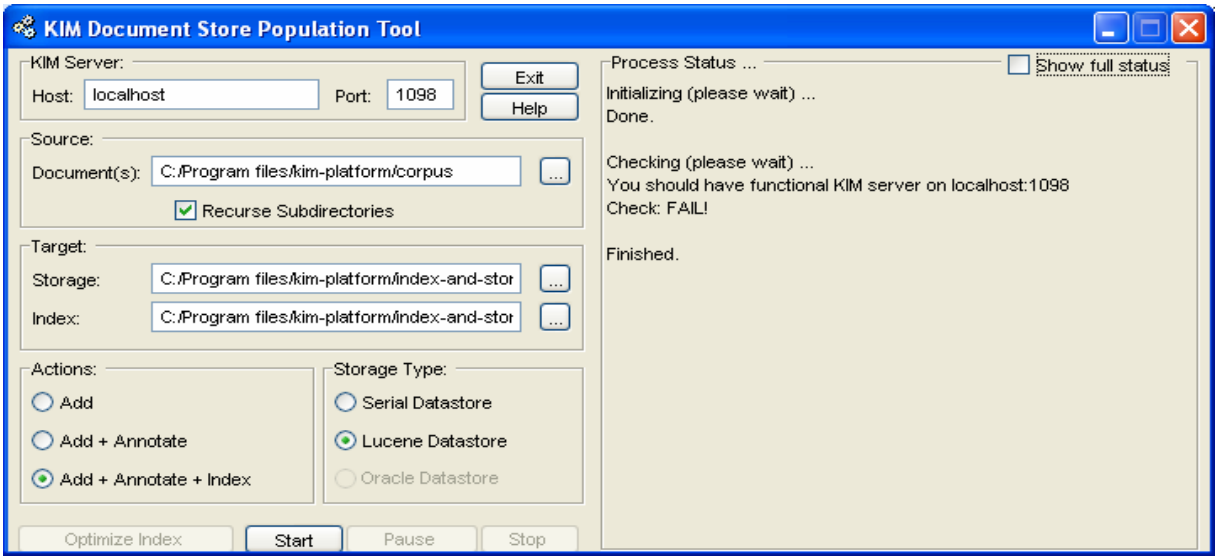


Figure 99. Interface de l'outil de peuplement d'ontologie dans KIM

De tous les outils, KIM est celui dont nous nous sentons le plus proche car il est commercialisé par une société et utilisé dans le cadre d'applications réelles et concrètes dans divers projets européens ou entreprises. De plus, grâce une gestion simple et efficace, il permet d'annoter les documents et de peupler une ontologie avec les nouvelles instances dans un même processus. Néanmoins, son défaut principal à nos yeux est d'utiliser une ontologie générique et non une ontologie de domaine. Cette ontologie, et par conséquent les annotations qui en découlent, sont de trop haut-niveau et ne modélisent pas assez sémantiquement le contenu des documents.

- **CROSSMARC**

« Cross-lingual Multi Agent Retail Comparison », ou CROSSMARC[75] [VAL 03] [VAL 04], est un projet financé dans le cadre du programme de l'Information Society Technologies[76] (IST) de l'Union Européenne. Il réunit différents acteurs européens, notamment universitaires, afin de mettre en place un système afin d'annoter sémantiquement les pages web avec des entités spécifiques au domaine du commerce et de la vente d'ordinateurs sur Internet.

Les Ressources :

Les documents à annoter sont des pages Web hautement structurées au format HTML, i.e. contenant des listes et des tableaux et dont les caractéristiques sont assez régulières pour leur extraction. Les annotations sont ensuite générées dans un langage utilisant une syntaxe XML mais non standardisée.

Les Traitements :

Le processus d'annotation est semi-automatisé bien que le système de reconnaissance des entités nommées soit un système d'apprentissage supervisé utilisant un modèle statistique : les Hidden

[75] http://www.iit.demokritos.gr/skel/crossmarc/
[76] http://cordis.europa.eu/ist//home.html

Markov Models (HMM). En plus d'identifier les instances existantes de l'ontologie dans le corpus à annoter, le modèle HMM est aussi capable d'extraire de nouvelles entités nommées. L'ensemble des extractions comprenant les entités nouvelles et existantes sont utilisées pour annoter le document. Enfin, un utilisateur final doit vérifier la cohérence de ces extractions et les valider manuellement.

Les Ontologies :

CROSSMARC utilise une ontologie de domaine concernant les Ordinateurs, modélisée en XML et non dans un des langages standards pour le Web Sémantique. Les classes sont organisées dans une taxonomie de type « partie de » entre le concept représentant un ordinateur et ses constituants. Aucune autre relation n'est modélisée, seuls les attributs de classes sont pris en compte. Les instances sont multilingues et utilisées pour générer les lexiques utilisés par le système d'extraction d'information.

Le Stockage :

Les annotations ne sont pas directement stockées dans le corps du document annoté. En fait, elles sont utilisées pour enrichir la base de connaissance de l'ontologie de référence.

L'Interfaçage :

Nous n'avons pu tester l'application car non téléchargeable. De plus, nous n'avons pas plus d'information concernant l'interfaçage de cet outil avec d'autres outils, notamment avec les agents du Web qui sont censés pouvoir interroger la base de connaissance afin de trouver les modèles d'ordinateur répondant aux critères des utilisateurs.

Ce projet est très spécifique à un domaine en particulier et à un type de document semi-structuré voir structuré. Il s'intéresse notamment à la correspondance entre les résultats de l'outil de reconnaissance des entités nommées et les étiquettes XML utilisées pour modéliser l'ontologie de référence et la base de connaissance sous-jacente. Néanmoins, il ne nous semble pas possible d'utiliser concrètement les résultats de cette expérience.

- **OntoSophie**

OntoSophie [VAR 04] [CEL 04] est un outil réalisé à l'institut KMI de l'Open University, tout comme MnM et MagPie précédemment présentés. Il a pour but d'identifier les événements dans les articles de presse et d'en peupler semi-automatiquement une ontologie de référence.

Les Ressources :

Les documents à annoter sont des pages Web aux formats HTML ou plein texte. Les annotations sont ensuite générées en XML sur la base des entités nommées et de leurs relations extraites. Cet outil est plutôt destiné aux documents ayant un contenu semi-structuré ou structuré.

Les Traitements :

Le processus d'annotation est semi-automatique et s'appuie à la fois sur une approche purement linguistique afin de repérer les entités nommées dans le document source (Marmot) et sur une approche d'apprentissage supervisé afin de classifier les entités extraites en fonction des événements modélisés dans l'ontologie de référence (Crystal). Puis, l'utilisateur doit contrôle et valider les annotations produites avant qu'elles ne soient enregistrées dans la base de connaissance.

Les Ontologies :

Cet outil utilise des ontologies de domaine, mais il n'est pas précisé dans quel langage ces ontologies sont modélisées. Un ensemble de classes relatives aux événements étudiés ainsi que leurs propriétés sont pourtant définis dans ces ontologies.

Le Stockage :

Les annotations ne sont pas stockées dans le document d'origine ni sur un serveur d'annotation externe. Elles sont utilisées pour peupler l'ontologie de référence, i.e. enrichir avec les instances extraites la base de connaissance sous-jacente.

L'Interfaçage :

Nous n'avons pu tester l'application et donc apprécier la qualité de ses interfaces. D'autre part, il n'est pas spécifié non plus si cet outil est capable de s'interfacer avec d'autres applications.

Outre le fait que ce projet n'a pas vraiment abouti à un réel outil puisqu'une version de démonstration n'est même pas téléchargeable, les résultats fournis dans les articles ne sont pas très probants quant à la performance de ce système. Par contre, ces mêmes articles font entrevoir toute une série de questions concernant l'intégration des annotations dans la base de connaissance en tant que nouvelles instances. Notamment, ils soulèvent les problèmes liés à l'extraction de plusieurs valeurs pour un unique attribut ou l'appartenance d'une instance à plusieurs classes. Ces deux points nous paraissent très importants lorsqu'il s'agit de maintenir l'intégrité et la qualité de la base de connaissance.

1.2.3 Les développements récents

Dans la suite de cette section, nous allons présenter plus brièvement les dernières applications apparues courant 2005 mais qui n'ont pas encore atteint un degré de maturation suffisamment important pour être décrit aussi précisément que les outils mentionnés ci-dessus. Nous allons donc passer chacun d'entre eux en revue en nous attachant à leurs objectifs et leurs caractéristiques particulières.

- **OntoText**

OntoText[77] [MAG 05] est un projet développé au sein du centre pour la recherche scientifique et technique[78] (ITC-irst), appartenant à « l'Instituto Trentino di Culture ». L'objectif de ce projet est d'étudier et de développer des techniques d'extraction d'information innovantes pour produire des annotations plus efficientes dans le cadre du Web Sémantique. Pour cela, les documents sont annotés avec des informations linguistiques et les annotations générées dans un format XML. Ces annotations XML sont ensuite associées avec une instance d'un concept modélisé dans l'ontologie de référence. Il n'est pas précisé comment ces ontologies sont modélisées. Enfin, un processus de normalisation des nouvelles instances créées permet de désambiguïser les différentes entités et de les intégrer dans la base de connaissance liée à l'ontologie. Par rapport aux autres projets vus précédemment, il n'y a pas vraiment à notre sens de progrès réalisé mais ce projet est toujours en cours d'élaboration et il faudra attendre les résultats avant de pouvoir se faire une opinion objective de ses apports à la communauté.

- **SmartWeb**

SmartWeb[79] [BUI 05] est un projet financé par le Ministère Fédéral Allemand de l'Education et de la Recherche (BMBF) et regroupant divers acteurs, aussi bien universitaires que commerciaux. Ils ont pour objectif d'étudier les approches d'apprentissage non supervisé pour peupler une base de connaissance composée de triplets RDF. L'ontologie de domaine, modélisant en RDFS ou OWL le monde du Football pour la coupe du monde 2006, est essentiellement constituée de classes. Le processus d'annotation utilise tout d'abord un module linguistique pour traiter les entités nommées dans les documents sources provenant du Web puis le module d'apprentissage génère une classification de ces entités nommées sur la base de modèles contextuels. Cette approche s'inspire du travail réalisé sur SemTag mais outre le fait que SmartWeb soit capable d'enrichie la base de connaissance, il est aussi capable d'enrichir l'ontologie en créant de nouvelles sous-classes à partir des annotations.

- **Onto-H Interactive Editor**

Onto-H [BEN 05] est un projet financé par le Ministère Espagnol des Sciences et de la Technologie et réalisé par la société Isoco[80]. Ce projet cherche à développer un ensemble de composants pour aider un utilisateur à peupler une ontologie tout en lui fournissant des fonctionnalités avancées pour le processus d'annotation. Onto-H développe un service pour faire appel à des moteurs d'extraction d'information externes, une interface utilisateur pour afficher les documents et leurs annotations, un moteur d'annotation et une base de connaissance capable de contrôler la cohérence des nouvelles instances. Onto-H doit être disponible soit sous la forme d'un plugin pour Protégé, soit comme application indépendante. Pourtant, nous n'avons pu avoir accès à aucune de ces implémentations

[77] http://tcc.itc.it/projects/ontotext/
[78] http://www.itc.it/irst
[79] http://smartweb.dfki.de/main_pro_en.pl?infotext_en.html
[80] http://www.isoco.com/en/innovation/projects_national.html

car il semblerait que cet outil ait été développé pour une application cliente particulière, non disponible publiquement. Ce projet apparaît toutefois très similaire à l'initiative CREAM et ses dérivés.

- **h-TechSight**

h-TechSight[81] [MAY 05a] est un projet financé dans le cadre du programme de l'Information Society Technologies[82] (IST) de l'Union Européenne. Il a pour but de mettre en place une plateforme de gestion des connaissances pour les entreprises. Cette plateforme doit être capable de produire des synthèses décrivant la dynamique des concepts et de leurs instances dans les articles analysés, à travers le temps. Ils emploient pour cela un système d'extraction d'information basé sur une implémentation de GATE pour annoter les documents et enrichir automatiquement les bases de connaissances des diverses ontologies de domaine. Ces ontologies, au gré des résultats fournis par l'outil d'extraction, peuvent être manuellement modifiées et enrichies.

- **AktiveDoc**

AktiveDoc[83] [LAN 05] est un nouvel outil développé par le groupe Natural Language Processing de l'université de Sheffield. AktiveDoc se veut un véritable environnement d'édition de pages Web Sémantique tel MS World pour les documents textuels. Pour devenir cet éditeur de pages sémantiques, AktiveDoc allie l'annotation manuelle, et/ou semi-automatique via l'utilisation d'Amilcare, à l'activité d'écriture des pages Web traditionnelles via une seule et même interface utilisateur. Il permet également la réutilisation d'annotations provenant de serveurs d'annotations ou de bases de connaissances externes, connectés via des web services. AktiveDoc a récemment été renommé AktiveMedia car il étend son service d''annotation à des documents multimédias. Une version téléchargeable de l'outil est annoncée sur leur site Web mais non encore disponible.

[81] http://www.h-techsight.org/
[82] http://cordis.europa.eu/ist//home.html
[83] http://nlp.shef.ac.uk/wig/aktivedoc.htm

fiche signalétique			Les Ressources					Les Traitements					Les Ontologies					Le Stockage			L'Interfaçage		
Année de parution	Equipe / Projet	Nom de l'outil	Type de contenu	Format en entrée	Provenance des documents	Information concernée	Format en sortie	Méthode d'annotation	Outil d'extraction d'information	Niveau de l'apprentissage	Niveau d'automatisation	Montant du travail manuel	Provenance de l'ontologie	Ontologie de référence	Langage ontologie	Eléments de l'ontologie pouvant être créés	Domaine(s) d'application	Support de stockage des annotations	Annotations dans ou hors document	Base de Connaissance utilisée	Interfaces utilisateurs	Interopérabilité	open source
2000	Mindswap	**SHOE Knowledge Annotator**	Textuel	HTML	URLs, fichiers en local	segments textuels correspondant à des entrées de la	SHOE	Non applicable	Non applicable	Non applicable	manual	Annotation manuelle par l'utilisateur	URLs & locale several ontologies	ontologie de domaine	SHOE	concepts, relations & claims (ie. énoncés)	Tout domaine	Système de fichier local	insérées dans le document ou séparées du	Non applicable	Vérification de la syntaxe SHOE pour détecter les erreurs	Exposé, SHOE Search, Semantic Search.	applet ou application indépendante
2001	Ubot	**AeroDAML**	Textuel	HTML	URLs, fichiers en local	Noms propres & relations de haut niveau	DAML+OIL	TAL, règles d'extraction, patrons	AeroText	Non	auto	Ecriture des règles pour AeroText	Ontologie générique, locale	WordNet par défaut ou ontologie de domaine	DAML+OIL	classes, instances & propriétés	Par défault, indépendant d'un domaine	Système de fichier local	insérées dans le document	AeroText's CommonKB		Non applicable	Acces Web (plus disponible)
2001	W3C	**Amaya**	Textuel	XML, HTML, XHTML	URLs	Métadonnées DublinCore + commentaires libres	RDF	Non applicable	Non applicable	Non applicable	manuel	Annotation purement manuelle	?	ontologie de domaine	RDF(S)	DublinCore + free text statements	Tout domaine	Système de fichier local Entrepôt RDF (Annotea)	séparées du document	Annotea	navigateur Web et éditeur		
2001	W3C	**Annotea**	Textuel	HTML, XML	URLs	Métadonnées DublinCore +	RDF	Non applicable	Non applicable	Non applicable	manuel	Annotation purement manuelle	?	ontologie de domaine	RDF(S)	DublinCore + free text statements	Tout domaine	Système de fichier local Entrepôt RDF	séparées du document	Serveur d'Annotation			
2001	UNI Manchester & UNI Southampton EPSRC project	**COHSE**	Textuel	HTML	URLs, fichiers en local	Entités Nommées + noms communs	DAML+OIL (RDF triples)	Non applicable	Non applicable	Non applicable	manuel & auto	Annotation manuelle par l'utilisateur	URLs & locale	ontologie de domaine	OIL, DAML+OIL	classes, instances & propriétés	Domaine d'évaluation : site web du tutoriel Java	hyperlinks enregistrés dans le Distributed Links Service	insérées dans le document ou séparées du document	Serveur d'ontologie, lexiques	plug-in pour Internet Explorer et Mozilla, Pas de vérification des contraintes	Mozilla, navigateur Web, Annotea	plug-in
2001	Uni Karlsruhe (AIFB)	**CREAM**	Textuel + deep Web	HTML, XML	URLs, fichiers en local	Noms propres & relations de haut niveau	RDF, OWL, DAM+OIL	Non applicable	Non applicable	Non applicable	manuel	Annotation manuelle par l'utilisateur	Non applicable	ontologie de domaine	DAML+OIL, OWL	classes, instances, attributs, relations	Tout domaine	Système de fichier local Serveur d'annotation	insérées dans le document ou séparées du document	Non applicable	éditeur d'ontologie, Document Viewer, Crawler, Document Manager		
2001	Open University (KMi) AKT Project	**MnM**	Textuel + deep Web	HTML, XML, texts, DBs	URLs, fichiers en local	Entités Nommées	XML, RDF, DAML+OIL, OCML	TAL (POS, NER) + Wrapper induction	Lazy-TAL algorithm * Amilcare * Crystal (ML) + Marmot (POS) + Badger (IE)	apprentissage supervisé	manuel & semi	Annotation manuelle d'un corpus d'entraînement	URLs & locale several ontologies	ontologie KMi	OCML (RDF & DAML+OIL in future ?)	classes, instances, attributs	Domaine d'évaluation: articles parus dans KMi Planet sur les activités et événements au laboratoire KMi	Système de fichier local BC : serveur d'ontologie	insérées dans le document	WebOnto server	navigateur Web, interface de validation, pas de vérification des contraintes	WebOnto OCML repository	open API
2001	AIFB + Ontoprise	**OntoAnnotate**	Textuel	HTML	URLs, fichiers en local	Entités Nommées (+ relations ?)	HTML-A, RDF, DAML+OIL	wrapper	OntoMatcher + SMES	Non	manuel, semi, auto	?	locale	ontologie de domaine	F-Logic, RDF(S)	classes, instances, attributs, relations	Tout domaine Domaine d'évaluation : Ontologie SWRC	Système de fichier local Serveur d'annotation : OntoBroker		OntoBroker	drag & drop, vérification des contraintes	OntoBroker	
2001	Uni Karlsruhe (AIFB)	**Ont-O-Mat (V1) AIFB**	Textuel + deep Web	HTML	URLs, fichiers en local		DAML+OIL, OWL, SQL	Non applicable	Non applicable	Non applicable	manual	Annotation manuelle d'un corpus d'entraînement	URLs & locale	ontologie de domaine	DAML+OIL, F-Logic, RDF(S)	classes, instances, attributs, relations	Tout domaine	Système de fichier local Serveur d'annotation : OntoBroker	insérées dans le document ou séparées du	OntoBroker : annotation inference server	drag & drop, créer et annoter un document, vérification des contraintes	OntoBroker, OntoEdit	
2002	UNI Sheffield (NLP group) AKT Project	**Melita**	Textuel	HTML, txt	fichiers en local	Entités Nommées	RDF(S), DAML+OIL	TAL (POS, NER) + Wrapper induction	Amilcare utilisant ANNIE (GATE) + LP²	apprentissage supervisé	semi	Annotation manuelle d'un corpus d'entraînement	locale	ontologie de domaine	?	classes, instances	Domaine d'évaluation : Petites annonces d'embauche	?	insérées dans le document	?	possibilité de définir des seuils pour les résultats de l'outil d'extraction		
2002	Uni Karlsruhe (AIFB)	**S-CREAM**	Textuel + deep Web	HTML, DBs	URLs, fichiers en local	Entités Nommées (+ relations ?)	DAML+OIL	TAL (POS, NERs) + Wrapper induction	Amilcare utilisant ANNIE (GATE) + LP²	apprentissage supervisé	manual & semi	Annotation manuelle d'un corpus d'entraînement	URLs & locale	ontologie de domaine	DAML+OIL	classes, instances, attributs, relations	Tout domaine	Système de fichier local Bases de données	insérées dans le document ou séparées du	OntoBroker : annotation inference server	drag & drop, créer et annoter un document, vérification des contraintes	OntoBroker	Non, cf. OntoAnnotate
2003	UNI Sheffield (TAL group) AKT project Dot.Kom	**Armadillo**	Textuel	HTML	URLs, DBs	Entités Nommées	RDF	TAL (POS + NER) + wrappers + Googling	Amilcare utilisant ANNIE (GATE) + LP²	apprentissage non supervisé	semi & auto	Ecriture des règles	?		RDF(S)	classes, instances & propriétés	Domaine d'évaluation : informations sur les travailleurs du département informatique de l'université	BC : Entrepôt RDF Doc: Document Server	insérées dans le document	Entrepôt RDF			
2003	AKT project	**ArtEquAkt Project**	Textuel				RDF	TAL (POS, NERs)	Apple Pie parser + GATE					ontologie de domaine		classes, instances & propriétés	Tout domaine Domaine d'évaluation : artistes & peintures (Ontologie CRM du CIDOC)	Enrichissement de bases de connaissance		WordNet			
2003	IST project	**CROSSMARC project**	Textuel	HTML	URLs	Entités Nommées	?	TAL (POS, NER) + modèles HMM	NER system + COCLU (clustering)	apprentissage non supervisé	semi	Paramétrage manuel des modèles HMM + Validation des annotations	locale ?	ontologie de domaine	XML-dialect	classes, instances, attributs	Tout domaine Domaine d'évaluation : magasins de vente en ligne (ontologie des ordinateurs)	BC : ?	?	?			
2003	OntoText (SIRMA SA)	**KIM**	Textuel	HTML	URLs, fichiers en local	Entités Nommées + relations de haut niveau	RDF	TAL (POS tagging, NER), règles d'extraction,	GATE PRs + JAPE	Non	auto	Ecriture des règles GATE (règles JAPE)	Ontologie générique, locale	KIMO --> PROTON	RDF(S), OWL	classes, instances, relations	Par défaut, independant d'un domaine, domaine d'évaluation :	BC: Sesame (entrepôt RDF)	insérées dans le document ?	Sesame : entrepôt RDF	divers plug-ins, tris des résultats par pertinence	Lucene IR	yes + plug-in IE
2000	Open University (KMi) AKT Project	**Magpie**	Textuel	HTML	URLs, DDs, locale	Entités Nommées	OCML, RDF	TAL (NERs) + Wrapper induction	Espotter (NER) MnM	Non	auto	enrichissement de parties spécifiques des lexiques	locale	ontologie de domaine	RDF(S), OCML	classes, instances	Domaine d'évaluation: articles parus dans KMi Planet sur les activités et événements au laboratoire	DC: Semantic Log KB	Non, temps réel	lexiques d'Entités Nommées	plug-in navigateur Web, web services, surlignage des annotations	MnM	
2003	Uni Karlsruhe (AIFB)	**Ont-O-Mat (V2) Amilcare**	Textuel + deep Web	HTML	URLs, fichiers en local	Entités Nommées	DAML+OIL, OWL, SQL	TAL (POS, NERs) + Wrapper induction	Amilcare utilisant ANNIE (GATE) + LP²	apprentissage supervisé	manual & semi	Annotation manuelle d'un corpus d'entraînement	URLs & locale	ontologie de domaine	DAML+OIL, F-Logic, RDF(S)	classes, instances, attributs, relations	Tout domaine	Système de fichier local Serveur d'annotation : OntoBroker	insérées dans le document ou séparées du	OntoBroker : annotation inference server	drag & drop, créer et annoter un document, vérification des contraintes	OntoBroker, OntoEdit	
2003		**Semantic Word**	Textuel	MSWord	fichiers en local	segments textuels	DAML	TAL + wrappers	AeroDAML	apprentissage supervisé ?	manuel & semi	Annotation purement manuelle	?	ontologie de domaine	DAML+OIL	classes, instances & propriétés	Tout domaine	Système de fichier local	insérées dans le document	Non applicable	éditeur avec des patrons prédéfinis	MsWord	
2003	Standford University	**SemTag**	Textuel	HTML	URLs	segments textuels correspondant à des entrées de la taxonomie	RDF	TAL, règles d'extraction + calculs de similarité + Googling	TaxoNonmy Based Disambiguation algorithm (TBD) + Seeker (IE)	apprentissage non supervisé	auto	Non	Ontologie générique	Taxoonomie TAP	RDF(S) ?	entrées des lexiques	Indépendant du domaine	Serveur d'annotation : Label Bureau	insérées dans le document	Lexiques TAP		API to public repositories	
2004	IST project	**Dot.Kom project**	Textuel	HTML+D14	fichiers en local	Entités Nommées (+ relations de haut niveau ?)	DAML+OIL, RDF?	TAL + Wrapper induction	OntoMat AnNontizer, Amilcare, MnM	apprentissage supervisé	manual & auto	Annotation manuelle d'un corpus d'entraînement	URLs & locale	ontologie de domaine	DAML+OIL	classes, instances, attributs, relations	Tout domaine Domaine d'évaluation : Juridique et BioTechnologies		insérées dans le document ou séparées du	OntoBroker?		TextToOnto, Magpie	
2004	Uni Karlsruhe (AIFB)	**Ont-O-Mat (V3) PANKOW**	Textuel + deep Web	HTML, DBs	URLs, fichiers en local	Entités Nommées	DAML+OIL, OWL, SQL	TAL (POS) + Hearst patterns + Googling	PANKOW	apprentissage non supervisé	semi & auto	Non	URLs & locale	ontologie de domaine	DAML+OIL	classes, instances	Tout domaine Domaine d'évaluation : projet GETESS sur le Tourisme	Système de fichier local Serveur d'annotation : OntoBroker	insérées dans le document ou séparées du	OntoBroker : annotation inference server	drag & drop, créer et annoter un document, vérification des contraintes		
2004	Open University (KMi)	**OntoSophie**	Textuel	HTML, textes	?	Entités Nommées	XML	TAL + Classifieurs	Marmot (TAL) + Crystal (induction tool) + Badger (IE)	apprentissage supervisé	semi	manual training of classifiers + validation of proposed classification/anNont	?	ontologie KMi	?	classes, instances	Tout domaine Domaine d'évaluation : KMi Planet newsletter au sujet des activités et événements au laboratoire Kmi (Ontologie	Enrichissement de bases de connaissance	?	KB?			
2005	UNI Sheffield (Wit & TAL groups)	**AktiveDoc**	Textuel	HTML	?	Entités Nommées + commantaires libres	RDF	TAL (POS + NER) + Wrapper induction	* Amilcare utilisant ANNIE (GATE) + LP² * Armadillo	apprentissage supervisé et non supervisé	manuel & semi	Annotation manuelle par l'utilisateur	?	?	?	classes, instances & propriétés	?	MySQL DB	séparées du document	MySQL DB	Environnement d'édition intégré, propose des suggestions à l'utilisateur	Melita, MagPie, Armadillo, Amilcare	
2005	UNI Sheffield (NLP group)	**h-TechSight**	Textuel	HTML	URLs	Entités Nommées	DAML+OIL, RDF	TAL (POS tagging, NER), règles d'extraction,	GATE PRs + JAPE	Non	semi	Ecriture des règles GATE (règles JAPE)	locale	ontologie de domaine	DAML+OIL, RDF	classes, instances	Domaine d'évaluation : emploi	BC: MSAccess DB		MS Access DB	portail KM, éditeur d'ontologie		
2005	iSOCO SA	**Onto-h**	Textuel	HTML, RTF, txt	fichiers en local, JDBC	?	RDF	TAL, règles d'extraction,	Moteur d'inférence	?	semi		locale	ontologie de domaine	RDF, RDF(S)	classes, instances, attributs, relations	Domaine d'évaluation : ontologie Sciences	Système de fichier local		Non	éditeur + vérification des contraintes	Non applicable	plug-in + application
2005	Mondeca + Uni Paris 4 (Lalicc)	**OntoPop**	Textuel	HTML, MSWord, Texte, XML, PDF	URLs, fichiers en local	Entités Nommées + relations de domaine	RDF, Topic Maps, OWL	TAL, règles d'extraction, patrons	* Insight Discoverer Extractor * GATE PRs	Non applicable	semi & auto	Ecriture des règles d'acquisition de connaissance pour IDE ou GATE	incluse dans ITM	ontologie de domaine	OWL, RDF, Topic Maps	classes, instances, attributs, relations	Tout domaine Domaines d'évaluation : Intelligence Economique, l'Edition Juridique, la Presse	Système de fichier local BC : ITM	séparées du document	ITM KB	Interface de validation, vérification des contraintes de restriction et de cohérence	ITM Systèmes de gestion de contenu	Non
2005	ITC-irst project	**Ontotext Project**	Textuel	HTML	URLs	Noms propres + relations temporelles	XML?	TAL (POS, ...)	?	?	?	?	?	ontologie de domaine	?	classes, instances, attributs	Tout domaine Domaine d'évaluation : événements parus dans les journaux de Trentino	?	?	I-CAB (Italian Content Annotation Bank)		People ON-Line web port	
2005	BMBF project	**SmartWeb project**	Textuel	HTML	fichiers en local	segments textuels correspondant à des classes de l'ontologie	RDF	TAL (POS, lemmatiseur) + classifieurs	WEKA	apprentissage non supervisé	auto	Non	?	ontologie de domaine	RDF(S), OWL	classes, instances	Tout domaine Domaine d'évaluation : coupe du monde de foot 2006 (ontologie du Foot)	Base de Connaissance : entrepôt RDF	?	entrepôt RDF			
2006	IST project	**SEKT project**	Textuel	HTML	URLs, fichiers en local	Noms propres + relations de haut niveau	RDF	TAL	KIM	Non	auto	Ecriture des règles JAPE pour GATE	locale	Ontologie générique : PROTON	RDF(S), OWL	classes, instances, attributs, relations	Indépendant du domaine	BC : OWLIM (entrepôt RDF)	insérées dans le document ?	KIM : entrepôt RDF (Sesame)	KIM Document Store + KIM plug-in pour Internet Explorer	Sesame, Lucene	open source
	Ubot	**AeroSWARM**	Textuel	HTML	URLs, fichiers en local	Noms propres & relations de haut niveau	OWL	TAL, règles d'extraction, patrons	AeroText	Non	auto	Ecriture des règles pour AeroText	Ontologie générique, locale	WordNet par défaut ou ontologie de	OWL	classes, instances & propriétés	Par défault, indépendant d'un domaine	Serveur d'annotation : serveur RDF (comme Annotea)	insérées dans le document	AeroText's CommonKB		Annotea, Amaya	Acces Web (plus disponible)
	Mindswap	**SMORE**	Multimédia	HTML, textes, emails (MailSMORE) & images (PhotoSMORE)	URLs	segments textuels correspondant à des entrées de la taxonomie	RDF, OWL	Non applicable	Non applicable	Non applicable	manual	Annotation manuelle par l'utilisateur	URLs & locale several ontologies	ontologie de domaine	RDF, OWL	classes, instances & propriétés	Tout domaine	Système de fichier local	insérées dans le document	Non applicable	navigateur Web, éditeur d'ontologie, vérification des restrictions de domaine et de portée, vérification des triplets invalides	navigateur et éditeur d'ontologie SWOOP	

Annexe II. Analyse d'un arbre conceptuel généré à partir de l'outil IDE

Nous avons analysé l'ensemble de l'article « Famille Coppola, l'esprit de clan » paru dans le magasine « Elle » le 14/07/2003 afin de comparer ce qui aurait dû être étiqueté par l'outil d'extraction IDE et ce qui a été réellement étiqueté. Dans un premier temps, nous avons surligné manuellement à l'aide de différentes couleurs les éléments pertinents relatif au domaine de la « Presse People » et en fonction des formulaires MUC : Entités Nommées, Attributs et Relations [GRI 96]. Voici l'explication des couleurs ayant servi au surlignage de ces éléments :

- Entité nommée Personne
- Entité nommée Personnage
- Entité nommée Œuvre
- Entité nommée Société
- Attribut Parenté
- Attribut Lieu
- Attribut Date
- Relation Mariage
- Relation Naissance
- Relation Filmographie
- Relation Bibliographie
- Relation Inspiration
- Relation Goût
- Relation Nécrologie
- Relation Santé

Ensuite, nous avons analysé l'article à l'aide de l'outil d'extraction IDE qui l'étiquette automatiquement. Nous avons enfin comparé les résultats afin de déterminer quels étaient les éléments extraits correctement, la façon dont l'article a été étiqueté et les difficultés d'extraction non résolues par l'outil.

FAMILLE COPPOLA L'ESPRIT DE CLAN

Il y a le « parrain », Francis, bien sûr.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse: reconnaissance d'EN

Et tous les autres membres de la famille, qui ont su, à leur tour, imposer leur patte : Sofia, Roman, les enfants, Nicolas Cage, le neveu, ou encore Spike Jonze, le gendre.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)

/Personnalite (Sofia)
/REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/REFERENCE-ACTEUR (Nicolas Cage)
/ActorNamed (Nicolas Cage)
/Personnalite (Nicolas Cage)
/REFERENCE-ACTEUR (Spike Jonze)
/ActorNamed (Spike Jonze)
/Personnalite (Spike Jonze)
Résultat analyse : reconnaissance d'EN, perte des informations de parenté

Une tribu qui a connu ses drames et ses grands bonheurs, et qui semble soudée pour toujours. Et ce n'est pas du cinéma. RUGGIERI MARION En 1972, dans leur maison à San Francisco, Eleanor, Francis et leurs trois enfants : Sofia, Giancarlo et Roman./ 2001.
IDE : /REFERENCE-ACTEUR (RUGGIERI Marion)
/ActorNamed (RUGGIERI Marion)
/Personnalite (RUGGIERI Marion)
/REFERENCE-ACTEUR (San Francisco)
/ActorNamed (San Francisco)
/ NomDePersonnePotentiel (San Francisco)
/ ProperName(San)
/Prenom(Franscisco)
/SEXE_MASCULIN(Francisco)
/REFERENCE-ACTEUR (Eleanor)
/ActorNamed (Eleanor)
/Prenom (Eleanor)
/REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
/REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/REFERENCE-ACTEUR (Giancarlo)
/ActorNamed (Giancarlo)
/Prenom (Giancarlo)
Résultat analyse: reconnaissance d'EN, perte des informations de parenté, perte de 'Roman', identification de 'San Francisco' comme une 'Personne'

A la première de « Moulin Rouge », au Festival de Cannes, Roman, Sofia, Francis et Eleanor./
IDE : /Cinema (Moulin Rouge)
/REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN, perte de 'Eleanor'

Sofia, la relève Coppola./ 1.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : reconnaissance de l'EN

L'oncle de Francis, le compositeur Anton Coppola.
IDE : /REFERENCE-ACTEUR (L' oncle de Francis , le compositeur Anton Coppola)

/ActorNamed (L' oncle de Francis , le compositeur Anton Coppola)
/QualificationPersonne (L' oncle de Francis)
/Famille (oncle de Francis)
/FamilleAttestée (oncle de Francis)
/LienParente (oncle)
/Personnalite (Francis)
/Personnalite (Anton)
/ProperName (Coppola)
Résultat analyse : reconnaissance des EN et de la relation de parenté entre les deux personnes

2. Roman. 3. Sofia et Nicolas Cage.
IDE : /REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/REFERENCE-ACTEUR (Sofia et Nicolas Cage)
/ ActorPlural
/ActorNamed (Sofia)
/Personnalite (Sofia)
/ActorNamed (Nicolas Cage)
/Personnalite (Nicolas Cage)
Résultat analyse : reconnaissance des EN

4. Talia, la sœur de Francis, avec Stallone.
IDE : /REFERENCE-ACTEUR (Talia , la sœur de Francis ,)
/ActorNamed (Talia , la sœur de Francis ,)
/Prenom (Talia)
/SEXE_FEMININ (Talia)
/QualificationPersonne (la sœur de Francis)
/Famille (sœur de Francis)
/FamilleAttestée (sœur de Francis)
/LienParente (sœur)
/Personnalite (Francis)
/REFERENCE-ACTEUR (Stallone)
/ActorNamed (Stallone)
/Personnalite (Stallone)

Résultat analyse : reconnaissance des EN et de la relation de parenté entre les deux personnes

5. Roman et Frankie Rizer./
IDE : /REFERENCE-ACTEUR (Roman et Frankie)
/ ActorPlural (Roman et Frankie)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
/ActorNamed (Frankie)
/Prenom (Frankie)
/SEXE_MASCULIN (Frankie)
Résultat analyse : reconnaissance des EN

Sofia (1), actrice dans « Le Parrain III » aux côtés d'Andy Garcia (2), réalise « The Virgin Suicides » avec Kirsten Dunst (3). Digne fille de papa (4). F
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/REFERENCE-ACTEUR (Andy Garcia)
/ActorNamed (Andy Garcia)
/ Personnalite (Andy Garcia)
/REFERENCE-ACTEUR (Kirsten Dunst)
/ActorNamed (Kirsten Dunst)
/ Personnalite (Kirsten Dunst)

Résultat analyse : reconnaissance des EN 'Personne', mais pas des EN 'Œuvre' ni des relations entre ces personnes et ses oeuvres

Le parrain, c'est moi ! Tel pourrait être le pitch de la vie de Francis Ford Coppola.
IDE : /REFERENCE-ACTEUR (Francis Ford Coppola)
/ActorNamed (Francis Ford Coppola)
/Personnalite (Francis Ford Coppola)
Résultat analyse : reconnaissance des EN 'Personne'

Tant la famille y a toujours tenu le premier rôle dans ses longs-métrages comme dans ses grandes maisons. Il faut dire que les membres du « clan » ont de quoi faire pâlir d'envie n'importe quel directeur de casting. Il y a la mère, Italia, comédienne pour De Sica.
Résultat analyse : pas de reconnaissance des EN 'Personne', ni de l'attribut 'parenté'

La sœur, Talia, l'inoubliable Adrienne de « Rocky ».
IDE : /REFERENCE-ACTEUR (Talia)
/ActorNamed (Talia)
/Personnalite (Talia)
/OEUVRE_D_ART (l' inoubliable Adrienne de « Rocky »)
/ActorNamed (l' inoubliable Adrienne)
/Personnalite (Adrienne)
/Oeuvre (Rocky)
/ActorNamed (Rocky)
/Personnalite (Rocky)
Résultat analyse : reconnaissance des EN 'Personne' et 'Oeuvre' mais fausse relation entre les arguments, manque l'actrice pour pouvoir le relier à son rôle dans le Film. Perte de l'attribut 'parenté'

Le neveu, Nicolas Cage, qu'on ne présente plus.
IDE : /REFERENCE-ACTEUR (Nicolas Cage)
/ActorNamed (Nicolas Cage)
/Personnalite (Nicolas Cage)
Résultat analyse : reconnaissance des EN 'Personne', mais perte de l'attribut 'parenté'

Et, bien sûr, les enfants, Sofia et Roman, tous deux réalisateurs courus.
IDE : /REFERENCE-ACTEUR (Sofia et Roman)
/ ActorPlural (Sofia et Roman)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : reconnaissance des EN 'Personne', mais perte de l'attribut 'parenté'

Sans oublier le gendre, le réalisateur Spike Jonze, et la belle-fille, le top model Frankie Rizer...
IDE : /REFERENCE-ACTEUR (le réalisateur Spike Jonze)
/ActorNamed (le réalisateur Spike Jonze)
/Personnalite (Spike Jonze)
/REFERENCE-ACTEUR le top model Frankie Rizer)
/ActorNamed (le top model Frankie Rizer)
/Prenom (Frankie)
/SEXE_MASCULIN (Frankie)
/ProperName (Rizer)
Résultat analyse : reconnaissance des EN 'Personne'

Avec une pareille équipe, Francis, le mythique auteur du « Parrain » et d'« Apocalypse Now », aurait pu signer le plus « heureux » de ses films.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)

/ Bibliographie (auteur)
/Musique (heureux)
Résultat analyse : reconnaissance des EN 'Personne', bruit et silence dans la reconnaissance des 'Œuvre', non relation entre un auteur et ses oeuvres

Mais c'était compter sans le destin, les tournages impitoyables, les revers de fortune, et un fils tragiquement disparu. Moteur...

LE CLAN COPPOLA
IDE : /REFERENCE-ACTEUR (LE CLAN COPPOLA Francis)
/ActorNamed (LE CLAN COPPOLA Francis)
/Personnalite (COPPOLA Francis)
Résultat analyse : mauvais découpage du texte, reconnaissance des EN 'Personne'

Francis Coppola naît le 7 avril 1939 à Detroit, dans le Michigan.
IDE : / DATE-NAISSANCE (Coppola naît le 7 avril 1939 à Detroit)
/Person (Coppola)
/ ProperName (Coppola)
/Naissance (naît)
/DATE (le 7 avril 1939)
/Location (Detroit)
/America (Detroit)
/UnitedStates (Detroit)
/REFERENCE-ACTEUR (Michigan)
/ActorNamed (Michigan)
/Prenom (Michigan)
/SEXE_FEMININ (Michigan)
Résultat analyse : reconnaissance des EN 'Personne', erreur dans la reconnaissance des 'Location', reconnaissance d'un événement Naissance

Il est le deuxième des trois enfants de Carmine et Italia Coppola.
IDE : /REFERENCE-ACTEUR (Carmine et Italia Coppola)
/ ActorPlural (Carmine et Italia Coppola)
/ActorNamed (Carmine)
/Prenom (Carmine)
/SEXE_MASCULIN (Carmine)
/ActorNamed (Italia Coppola)
/ProperName (Italia Coppola)
Résultat analyse : reconnaissance des EN

Son père, originaire de New York, est chef d'orchestre.
Résultat analyse : non reconnaissance des EN 'Lieu'

Francis fera appel à lui pour composer la musique du « Parrain », en 1972.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne', silence dans la reconnaissance des 'Œuvre', non relation entre un auteur et ses œuvres

Sa mère, elle, est la fille du célèbre compositeur napolitain Francesco Pennino, auteur de l'opéra « Senza Mamma », dont un extrait figure dans « Le Parrain II ».
IDE : /REFERENCE-ACTEUR (Francesco Pennino)
/ActorNamed (Francesco Pennino)
/NomDePersonnePotentiel (Francesco Pennino)
/Prenom (Francesco)
/SEXE_MASCULIN (Francesco)
/ ProperName (Pennino)
/ Bibliographie (auteur)

Résultat analyse : reconnaissance des EN 'Personne', bruit et silence dans la reconnaissance des 'Œuvre', non relation entre un auteur et ses oeuvres

Comédienne, elle joue dans plusieurs films de Vittorio De Sica, avant d'embrasser la carrière de « mamma ».
IDE : /REFERENCE-ACTEUR (Vittorio De Sica)
/ActorNamed (Vittorio De Sica)
/Personnalite (Vittorio De Sica)
/Cinema (mamma)
Résultat analyse : reconnaissance des EN 'Personne', bruit dans la reconnaissance des 'Œuvre'

Francis a de qui tenir !
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

A 8 ans, « Francie », comme l'appelle sa maman, s'intéresse au cinéma et monte ses premiers essais en 8 mm...
IDE : /REFERENCE-ACTEUR (Francie)
/ActorNamed (Francie)
/Personnalite (Francie)
Résultat analyse : reconnaissance des EN 'Personne'

Une attaque de polio le cloue au lit pendant plusieurs mois. Il dévore les bandes dessinées, apprend le ventriloquisme et l'art des marionnettes. Un sujet qui sera au cœur du premier film de son gendre Spike Jonze, le mari de Sofia...
IDE : /REFERENCE-ACTEUR (son gendre Spike Jonze , le mari de Sofia)
/ActorNamed (son gendre Spike Jonze , le mari de Sofia)
/Famille (son gendre)
/Personnalite (Spike Jonze)
/QualificationPersonne (le mari de Sofia)
/Famille (mari de Sofia)
/FamilleAttestée (mari de Sofia)
/LienParente (mari)
/Personnalite (Sofia)
Résultat analyse : reconnaissance des EN et de la relation de parenté entre les deux personnes

Francis poursuit son apprentissage à l'Université d'Hofstra, où il acquiert une formation théâtrale, puis à l'école de cinéma de l'UCLA, où il devient l'assistant de Roger Corman, un producteur-réalisateur prolifique.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
/REFERENCE-ACTEUR (Roger Corman)
/ActorNamed (Roger Corman)
/Personnalite (Roger Corman)
Résultat analyse : reconnaissance des EN 'Personne'

Les trois enfants seront de tous les combats, c'est-à-dire de tous les films. « C'était comme faire des photos de famille », raconte Sofia.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : reconnaissance des EN 'Personne'

A 23 ans, il connaît toutes les ficelles du métier et rencontre celle qui deviendra sa seule et unique femme malgré bien des infidélités : Eleanor.
IDE : /REFERENCE-ACTEUR (Sofia)

/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : reconnaissance des EN 'Personne'

Ensemble, ils auront trois enfants : Giancarlo, dit Gio, Roman et Sofia.
IDE : /ENFANT (ils auront trois enfants)
/Parent (ils)
/ CoupleActeur (ils)
/Naissance (auront trois enfants)
/REFERENCE-ACTEUR (Giancarlo)
/ActorNamed (Giancarlo)
/ Prenom (Giancarlo)
/SEXE_MASCULIN (Giancarlo)
/REFERENCE-ACTEUR (Roman et Sofia)
/ ActorPlural (Roman et Sofia)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
/ActorNamed (Sofia)
/ProperName (Sofia)
Résultat analyse : reconnaissance des EN, impossibilité d'instancier relation parenté

Ils seront de tous les combats, c'est-à-dire de tous les films. Roman naît à l'Hôpital Américain à Neuilly, pendant le tournage de « Paris brûle-t-il ? », dont Francis a rédigé le scénario.
IDE : /REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
/REFERENCE-ŒUVRE (Paris)
/REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/ Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance de l'événement 'Naissance' ni des EN 'Œuvre' et bruit dans la reconnaissance des EN 'Lieu'

L'enfant en conservera un goût prononcé pour la « nouvelle vague » française, qu'on retrouvera par touches dans son premier film « CQ », dont l'action se déroule à Paris.
IDE : /Cinema (nouvelle vague)
/ REFERENCE-OEUVRE (CQ)
Résultat analyse : reconnaissance des EN 'Oeuvre', bruit dans la reconnaissance des EN 'Oeuvre', non reconnaissance des EN 'Lieu'

Sofia n'a pas 1 an quand elle fait sa première apparition en garçonnet dans « Le Parrain I ».
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/ Personnalite (Sofia)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance des EN 'Œuvre'

Dix-neuf ans plus tard, elle revient dans « Le Parrain III », où elle interprète la fille d'Andy Garcia, une composition qui lui vaudra les lazzis de la presse.
IDE : /REFERENCE-ACTEUR (Andy Garcia)
/ActorNamed (Andy Garcia)
/ Personnalite (Andy Garcia)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance des EN 'Œuvre'

« Pour lui, nous mettre dans ses films, c'était comme faire des photos de famille. » Les enfants participeront aussi activement au tournage d'« Apocalypse Now ».
IDE : / REFERENCE-OEUVRE (Apocalypse)
Résultat analyse : mauvaise reconnaissance des EN 'Oeuvre'

Trois ans d'enfer aux Philippines.

Les conditions de travail sont si rudes que Francis hésite à jeter l'éponge.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/ Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Un soir de détresse, il balance ses oscars par la fenêtre. Les enfants tentent de les récupérer sous une pluie diluvienne. Trop tard, les statuettes sont cassées.

Francis aussi, il a perdu 40 kilos et sa femme... Ou presque.
IDE : /SANTE (Francis aussi , il a perdu 40 kilos)
/ActorNamed (Francis)
/ Personnalite (Francis)
/Regime (a perdu 40 kilos)
/PerteDePoids (perdu)
/Poids (40 kilos)
Résultat analyse : reconnaissance des EN 'Personne' et de l'événement 'Santé'

Eleanor est la plus discrète de la famille, mais aussi la plus influente.
IDE : /REFERENCE-ACTEUR (Eleanor)
/ActorNamed (Eleanor)
/Prenom (Eleanor)
/SEXE_FEMININ (Eleanor)
Résultat analyse : reconnaissance des EN 'Personne'

Alors que Talia Shire, la sœur de Francis, deviendra célèbre grâce à son rôle d'Adrienne dans « Rocky » et que Nicolas Cage, le neveu de Francis (de son vrai nom Nicolas Kim Coppola), deviendra l'un des plus grands acteurs de sa génération, Eleanor préfère l'ombre à la lumière.
IDE : /REFERENCE-ACTEUR (Talia , la sœur de Francis ,)
/ActorNamed (Talia , la sœur de Francis ,)
/Prenom (Talia)
/SEXE_FEMININ (Talia)
/QualificationPersonne (la sœur de Francis)
/Famille (sœur de Francis)
/FamilleAttestée (sœur de Francis)
/LienParente (sœur)
/Personnalite (Francis)
/OEUVRE_D_ART (Adrienne dans « Rocky »)
/ActorNamed (Adrienne)
/Personnalite (Adrienne)
/Oeuvre (Rocky)
/ActorNamed (Rocky)
/Personnalite (Rocky)
/REFERENCE-ACTEUR (Nicolas Cage , le neveu de Francis)
/ActorNamed (Nicolas Cage , le neveu de Francis)
/Personnalite (Nicolas Cage)
/QualificationPersonne (le neveu de Francis)
/Famille (neveu de Francis)
/FamilleAttestée (neveu de Francis)
/LienParente (neveu)
/Personnalite (Francis)
/REFERENCE-ACTEUR (Nicolas)
/ActorNamed (Nicolas)
/Personnalite (Nicolas)
/REFERENCE-ACTEUR (Kim)
/ActorNamed (Kim)
/Personnalite (Kim)

/REFERENCE-ACTEUR (Eleanor)
/ActorNamed (Eleanor)
/Prenom (Eleanor)
/SEXE_FEMININ (Eleanor)
Résultat analyse : reconnaissance des EN et desrelations de parenté, non reconnaissance du rôle d'une 'Personne' dans une 'Œuvre'

Ecrivaine, peintre, photographe, mais aussi réalisatrice de documentaires (elle filme le making of d'« Apocalypse Now »), elle ne sortira de sa réserve qu'une fois, dans un livre, pour expliquer comment le tournage d'« Apocalypse Now » faillit briser son ménage.
IDE : /REFERENCE-OEUVRE (Apocalypse)
Résultat analyse : mauvaise reconnaissance des EN 'Oeuvre'

« Francis se mit à éprouver une très grande soif de possession physique, femme ou nourriture. »
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Il finit par lui avouer qu'il avait une maîtresse, ce qui plongea Eleanor dans une « véritable rage hystérique ».
IDE : /REFERENCE-ACTEUR (Eleanor)
/ActorNamed (Eleanor)
/Prenom (Eleanor)
/SEXE_FEMININ (Eleanor)
Résultat analyse : reconnaissance des EN 'Personne'

Mais, comme elle le conclura elle-même : « Ceux qui font des films se battent avec la vie. Ils ne restent pas assis sous des parasols à boire du thé glacé. » Ce ne fut ni la première ni la dernière incartade de Francis, et pourtant, comme le notera sa mère lors du tournage du « Parrain III », où elle tient un petit rôle à 78 ans :
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance de la relation entre une 'Personne' et une 'Œuvre', non reconnaissance des EN 'Œuvre'

« Même quand Francie était avec l'Autre, il aimait Ellie.
IDE : /REFERENCE-ACTEUR (Francie)
/ActorNamed (Francie)
/Personnalite (Francie)
/GOUT (il aimait Ellie)
/ActorUnknown (il)
/MascSing (il)
/ObjetGout (Ellie)
/ActorNamed (Ellie)
/Prenom (Ellie)
/SEXE_FEMININ (Ellie)
/VIE_SENTIMENTALE (il aimait Ellie)
/ActorUnknown (il)
/MascSing (il)
/Amour (aimait)
/ActorNamed (Ellie)
/Prenom (Ellie)
/SEXE_FEMININ (Ellie)
Résultat analyse : reconnaissance des EN 'Personne' et bruit sur les relations 'Gout' et 'Vie Sentimentale'

L'Autre voulait qu'il la quitte, mais pas lui. Ellie est restée et elle a gagné ! »

IDE : /REFERENCE-ACTEUR (Ellie)
/ActorNamed (Ellie)
/Prenom (Ellie)
/SEXE_FEMININ (Ellie)
Résultat analyse : reconnaissance des EN 'Personne'

Des trois enfants, seul Giancarlo n'apparaîtra pas à l'écran.
IDE : /REFERENCE-ACTEUR (Giancarlo)
/ActorNamed (Giancarlo)
/Prenom (Giancarlo)
/SEXE_MASCULIN (Giancarlo)
Résultat analyse : reconnaissance des EN 'Personne'

Un funeste présage pour un garçon dont l'image disparaîtra trop vite.
IDE : /NECROLOGIE (l' image disparaîtra)
/AnonymousActor (l'image)
/Deces (disparaîtra)
Résultat analyse : bruit dans la reconnaissance de l'événement 'Funérailles'

Comme le reste de la famille, Gio suit le parrain au gré de ses tournages. Après avoir été au sommet et remporté un oscar pour « Le Parrain », deux pour « Le Parrain II » et la palme d'or pour « Apocalypse Now », Francis connaît un sévère revers.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance des EN 'Œuvre'

Suite à l'échec de « Coup de cœur », en 1981, où Gio joue les assistants, son studio, Zoetrope, est saisi.
Résultat analyse : non reconnaissance des EN 'Œuvre', 'Personne' et 'Société'

Francis a 50 millions de dettes.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Il accepte de tourner à la suite huit films de commande. Parmi eux : « Jardins de pierre ».
Résultat analyse : non reconnaissance des EN 'Œuvre'

Lors de ce tournage, en 1986, le drame survient. Gio part pour une balade en bateau avec le fils de Ryan O'Neal, au large d'Annapolis.
IDE : /REFERENCE-ACTEUR (Ryan O'Neal)
/ActorNamed (Ryan O'Neal)
/Personnalite (Ryan O'Neal)
Résultat analyse : reconnaissance des EN 'Personne'

Les deux gamins s'amusent. La vie est belle à Napa Valley, Californie, où Francis s'est finalement posé dans une grande maison.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Le soleil est étincelant, les garçons slaloment entre bouées et rafiots. Tout d'un coup, une chaîne qui relie deux bateaux surgit en face d'eux. Trop vite, trop tard. Gio meurt décapité.

Il a 23 ans. Francis ne sera plus jamais le même. « Mes blessures ne se cicatriseront jamais. »

IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Le clan se ressoude. Pourtant, désormais, c'est un peu comme si Francis attendait de passer le relais.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

Il continue de tourner, mais on parle davantage des premiers succès de son excentrique neveu que des siens. Un an après la mort de Gio, Nicolas Cage rencontre la comédienne Patricia Arquette.
IDE : /REFERENCE-ACTEUR (Nicolas Cage)
/ActorNamed (Nicolas Cage)
/Personnalite (Nicolas Cage)
/REFERENCE-ACTEUR (la comédienne Patricia Arquette)
/ActorNamed (la comédienne Patricia Arquette)
/Personnalite (Patricia Arquette)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance de l'événement 'Vie sentimentale'

Les amants terribles se marient, mais font maisons séparées. Tout ce petit monde se retrouve régulièrement pour dîner à Napa Valley. Il y a le clan Coppola, les Arquette (Patricia est la sœur des comédiens Rosanna et David), ainsi que des personnages aussi éclectiques que David Lynch, Sylvester Stallone, Martin Scorsese, Marlon Brando ou Jim Carrey, meilleur ami de Nic'.
IDE : /REFERENCE-ACTEUR (le clan Coppola)
/ActorNamed (le clan Coppola)
/ProperName (Coppola)
/REFERENCE-ACTEUR (Patricia)
/ActorNamed (Patricia)
/Personnalite (Patricia)
/REFERENCE-ACTEUR (la sœur des comédiens Rosanna et David)
/ActorNamed (la sœur des comédiens Rosanna et David)
/Famille (sœur des comédiens Rosanna et David)
/FamilleAttestée (sœur des comédiens Rosanna et David)
/LienParente (sœur)
/Prenom(Rosanna)
/SEXE_FEMININ (Rosanna)
/Personnalite (David)
/REFERENCE-ACTEUR (David Lynch)
/ActorNamed (David Lynch)
/Personnalite (David Lynch)
/REFERENCE-ACTEUR (Sylvester Stallone)
/ActorNamed (Sylvester Stallone)
/Personnalite (Sylvester Stallone)
/REFERENCE-ACTEUR (Martin Scorsese)
/ActorNamed (Martin Scorsese)
/Personnalite (Martin Scorsese)
/REFERENCE-ACTEUR (Marlon Brando)
/ActorNamed (Marlon Brando)
/Personnalite (Marlon Brando)
/REFERENCE-ACTEUR (Jim Carrey)
/ActorNamed (Jim Carrey)
/Personnalite (Jim Carrey)
/REFERENCE-ACTEUR (Nic)
/ActorNamed (Nic)
/Prenom (Nic)
/SEXE_INCONNU (Nic)
Résultat analyse : reconnaissance des EN 'Personne', mauvaise reconnaissance des liens de parenté

« Les repas de famille ressemblaient à des castings... de qualité ! », se souvient Sofia.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : reconnaissance des EN 'Personne'

Tandis que Sofia se cherche, Roman se trouve.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : reconnaissance des EN 'Personne'

Lui aussi a d'abord fait l'acteur pour son père à 8 ans dans « Le Parrain II » : « Mes chaussures me faisaient mal.
Résultat analyse : non reconnaissance des EN 'Oeuvre'

Je devais me lever très tôt pour qu'on me frise les cheveux au fer chaud. Les coiffeuses m'ont brûlé le cou plusieurs fois. » Puis il réalise pubs et vidéo-clips. Durant les années 90, il met en images tout ce qui se fait de pointu en matière de scène techno et rock : Fatboy Slim, Daft Punk, Mellow ou encore, récemment, les Strokes.
IDE : /REFERENCE-ACTEUR (Fatboy Slim)
/ActorNamed (Fatboy Slim)
/ Personnalite (Fatboy Slim)
Résultat analyse : silence dans la reconnaissance des EN 'Personne'

Pendant ce temps-là, Sofia se tricote une réputation dans la mode.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : reconnaissance des EN 'Personne'

Après avoir fait plusieurs stages chez Chanel aux côtés de Karl Lagerfeld, elle lance sa propre ligne, Milk Fed, des basiques branchés, distribués en France chez A.P.C.
IDE : /REFERENCE-ACTEUR (Karl Lagerfeld)
/ActorNamed (Karl Lagerfeld)
/Personnalite (Karl Lagerfeld)
Résultat analyse : reconnaissance des EN 'Personne', silence dans la reconnaissance des EN 'Société'

Avec Roman, elle fréquente la scène techno, monte sur les planches (à roulettes, tous ses copains font du skate) et voyage à en perdre la tête, pour se poser le plus souvent à Paris, où Francis achète un pied-à-terre sur les quais, à deux pas du Quartier latin.
IDE : /COUPLE (Roman , elle fréquente la scène techno , monte sur les planches (à roulettes , tous ses copains)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
/Union (copains)
/REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

C'est un copain skate-boarder qui lui passe le roman de Jeffrey Eugenides, « The Virgin Suicides ».
IDE : /REFERENCE-ACTEUR (Jeffrey Eugenides)

/ActorNamed (Jeffrey Eugenides)
/Personnalite (Jeffrey Eugenides)
Résultat analyse : reconnaissance des EN 'Personne', non reconnaissance des EN 'Œuvre', non reconnaissance de l'événement 'Bibliographie'

L'histoire de cinq adolescentes d'une famille moyenne américaine qui mettent fin à leurs jours. Sofia, qui ne s'est jamais vraiment remise de la disparition de son grand frère, est touchée par l'histoire qui reprend des thèmes chers : la brièveté de la vie, la fragilité des instants heureux, les apparences parfois trompeuses, et puis la mort, inexplicable, qui vous laisse seule avec vos souvenirs et tant d'amour à ne pas partager.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/NECROLOGIE (la disparition de son grand frère)
/Deces (disparition)
/ActorUnknown (son grand frère)
/Famille (son grand frère)
Résultat analyse : reconnaissance des EN 'Personne', bruit dans la reconnaissance de l'événement 'Funérailles'

Tandis que son frère se fiance au top model Frankie Rizer, une grande brune aux yeux azur, et que son cousin, Nicolas, s'apprête à divorcer de Patricia, Sofia épouse Spike Jonze.
IDE : /COUPLE (son frère se fiance au top model Frankie Rizer , une grande brune)
/ActorUnknown (son frère)
/Famille (son frère)
/Fiancailles (se fiance)
/ActorNamed (Frankie Rizer , une grande brune)
/Prenom (Frankie)
/SEXE_MASCULIN (Frankie)
/ProperName (Rizer)
/CouleurCheveux (brune)
/COUPLE (son cousin , Nicolas , s' apprête à divorcer de Patricia)
/ActorNamed (son cousin , Nicolas)
/Famille (son cousin)
/Personnalite (Nicolas)
/EvenementImminent (s' apprête à)
/Divorce (divorcer)
/ActorNamed (Patricia)
/Personnalite (Patricia)
/COUPLE (Sofia épouse Spike Jonze)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/Mariage (épouse)
/ActorNamed (Spike Jonze)
/Personnalite (Spike Jonze)
Résultat analyse : reconnaissance des EN 'Personne', reconnaissance des 'Couple'

De son vrai nom Adam Spiegel, Spike est l'héritier d'un empire de vente par correspondance.
IDE : /REFERENCE-ACTEUR (son vrai nom Adam Spiegel)
/ActorNamed (son vrai nom Adam Spiegel)
/Personnalite (Adam Spiegel)
/REFERENCE-ACTEUR (Spike)
/ActorNamed (Spike)
/Prenom (Spike)
/SEXE_INCONNU (Spike)
Résultat analyse : reconnaissance des EN 'Personne'

Mais il est surtout le réalisateur de clips le plus demandé de sa génération, juste devant son beau-frère, Roman !
IDE : /REFERENCE-ACTEUR (Roman)

/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : reconnaissance des EN 'Personne'

Björk, Fatboy Slim, Daft Punk, les Beastie Boys : tous sont passés devant sa caméra.
IDE : /REFERENCE-ACTEUR (Björk)
/ActorNamed (Björk)
/Personnalite (Björk)
/REFERENCE-ACTEUR (Fatboy Slim)
/ActorNamed (Fatboy Slim)
/Personnalite (Fatboy Slim)
/REFERENCE-ACTEUR (Beastie Boys)
/ActorNamed (Beastie Boys)
/Personnalite (Beastie Boys)
Résultat analyse : silence dans la reconnaissance des EN 'Personne'

Le jeune prodige, aux faux airs de gringalet, tourne les têtes et des pubs, comme la célèbre partie de tennis qui oppose Agassi à Pete Sampras en plein Manhattan.
IDE : /REFERENCE-ACTEUR (Pete Sampras)
/ActorNamed (Pete Sampras)
/Personnalite (Pete Sampras)
Résultat analyse : silence dans la reconnaissance des EN 'Personne'

Ils se disent « oui » durant l'été 1999.
IDE : /Cinema (oui)
Résultat analyse : mauvaise reconnaissance des EN 'Oeuvre'

Un mariage intime et champêtre dans la maison familiale des Coppola à Napa Valley, sous le regard bienveillant de papa, fier d'avoir un gendre dont la maman était... flûtiste !
Résultat analyse : non reconnaissance des EN 'Personne'

Spike a 30 ans, il aime le skate, la magie et les cascades.
IDE : /REFERENCE-ACTEUR (Spike)
/ActorNamed (Spike)
/Prenom (Spike)
/SEXE_INCONNU (Spike)
/GOUT (30 ans , il aime le skate , la magie et les cascades)
/ActorUnknown (30 ans , il)
/QualificationPersonne (30 ans ,)
/Age (30 ans)
/MascSing (il)
/ObjetGout (le skate)
/ObjetGout (la magie)
/ObjetGout (les cascades)
Résultat analyse : reconnaissance des EN 'Personne', reconnaissance du fait 'Gout'

Ensemble, ils forment le couple le plus mode, le plus chic, le plus discret, celui qui rend les journalistes hystériques, d'autant que Spike s'apprête à sortir son premier long-métrage, « Dans la peau de John Malkovich », avec John Malkovich et Cameron Diaz.
IDE : /COUPLE (ils forment le couple)
/CoupleActeur (ils)
/MascPlu (ils)
/Union (forment le couple)
/REFERENCE-ACTEUR (Spike)
/ActorNamed (Spike)
/Prenom (Spike)
/SEXE_INCONNU (Spike)
/REFERENCE-ACTEUR (John Malkovich)

/ActorNamed (John Malkovich)
/Personnalite (John Malkovich)
/REFERENCE-ACTEUR (John Malkovich et Cameron Diaz)
/ActorPlural (John Malkovich et Cameron Diaz)
/ActorNamed (John Malkovich)
/Personnalite (John Malkovich)
/ActorNamed (Cameron Diaz)
/Personnalite (Cameron Diaz)
Résultat analyse : silence dans la reconnaissance des EN 'Personne', non reconnaissance de l'événement 'Filmographie'

Le film sera culte. Dans la foulée, Spike donne un coup de main à Sofia en plein tournage de « Virgin Suicides » qui sort en 2000.
IDE : /REFERENCE-ACTEUR (Spike)
/ActorNamed (Spike)
/Prenom (Spike)
/SEXE_INCONNU (Spike)
/REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
/REFERENCE-OEUVRE (Virgin Suicides)
/Oeuvre-LongMetrage (Virgin Suicides)
Résultat analyse : silence dans la reconnaissance des EN 'Personne', non reconnaissance de l'événement 'Filmographie'

Roman dirige la seconde équipe, le groupe de french techno Air est à la musique, et papa veille au montage.
IDE : /REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : silence dans la reconnaissance des EN 'Personne'

James Woods, à l'affiche avec Kathleen Turner et Kirsten Dunst, dira qu'il n'a jamais été aussi bien dirigé de sa vie.
IDE : /REFERENCE-ACTEUR (James Woods)
/ActorNamed (James Woods)
/Personnalite (James Woods)
/REFERENCE-ACTEUR (Kathleen Turner et Kirsten Dunst)
/ActorPlural (Kathleen Turner et Kirsten Dunst)
/ActorNamed (Kathleen Turner)
/Personnalite (Kathleen Turner)
/ActorNamed (Kirsten Dunst)
/Personnalite (Kirsten Dunst)
Résultat analyse : reconnaissance des EN 'Personne'

Le film fait l'unanimité de la critique et du public. Un style est né. Sofia plus que Coppola.
IDE : /REFERENCE-ACTEUR (Sofia)
/ActorNamed (Sofia)
/Personnalite (Sofia)
Résultat analyse : silence dans la reconnaissance des EN 'Personne'

« Nous formons une tribu, explique Roman. Nos films sont très différents, mais nous partageons le même goût de la liberté. »
IDE : /REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : reconnaissance des EN 'Personne'

Mais le parrain n'est jamais loin. Alors que son neveu, Nicolas, épouse Lisa Marie Presley, la fille d'Elvis et l'ex de Michael Jackson, sur une plage de Hawaï (le mariage, procédure de divorce comprise, durera trois mois !), il vole au secours de son fils, Roman.
IDE : /REFERENCE-ACTEUR (son neveu , Nicolas)
/ActorNamed (son neveu , Nicolas)
/Famille (son neveu)
/Personnalite (Nicolas)
/REFERENCE-ACTEUR (Lisa Marie Presley)
/ActorNamed (Lisa Marie Presley)
/Personnalite (Lisa Marie Presley)
/COUPLE (Elvis et l' ex de Michael Jackson , sur une plage de Hawaï (le mariage)
/ActorNamed (Elvis)
/Prenom (Elvis)
/SEXE_MASCULIN (Elvis)
/ActorNamed (l' ex de Michael Jackson)
/Famille (ex de Michael Jackson)
/LienParente (ex)
/Personnalite (Michael Jackson)
/Mariage (mariage)
/REFERENCE-ACTEUR (son fils , Roman)
/ActorNamed (son fils , Roman)
/Famille (son fils)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : mauvaise reconnaissance des EN 'Personne', mauvaise reconnaissance de l'événement 'Mariage', mauvaise reconnaissance des liens de parenté

Qui ne parvient pas à boucler son premier long-métrage, « CQ », avec Gérard Depardieu, Billy Zane, Elodie Bouchez et le top Angela Lindvall (avec qui on lui prête une liaison).
IDE : /Cinema (CQ)
/REFERENCE-ACTEUR (Gérard)
/ActorNamed (Gérard)
/Personnalite (Gérard)
/REFERENCE-ACTEUR (Billy Zane)
/ActorNamed (Billy Zane)
/Personnalite (Billy Zane)
/COUPLE (Elodie Bouchez et le top Angela Lindvall (avec qui on lui prête une liaison)
/ActorPlural (Elodie Bouchez et le top Angela)
/ActorNamed (Elodie Bouchez)
/ Personnalite (Elodie Bouchez)
/ActorNamed (le top Angela)
/Personnalite (Angela)
/Union (liaison)
Résultat analyse : mauvaise reconnaissance des EN 'Personne', mauvaise reconnaissance de l'événement 'Couple'

C'est donc Francis qui donnera le clap de fin.
IDE : /REFERENCE-ACTEUR (Francis)
/ActorNamed (Francis)
/Personnalite (Francis)
Résultat analyse : reconnaissance des EN 'Personne'

L'accueil est mitigé, mais bienveillant. Une fois de plus, tout le monde a mis la main à la pâte. Son beau-frère s'est occupé des scènes d'action. Sa sœur a suivi le casting en plus de faire une apparition dans le film. Sa mère a fait le making of. Son père a produit. Et un des acteurs n'est autre que Jason Schwartzman... son cousin !
IDE : /REFERENCE-ACTEUR (Jason)
/ActorNamed (Jason)
/Personnalite (Jason)
Résultat analyse : reconnaissance des EN 'Personne'

« Nous formons une tribu, expliquera Roman.
IDE : /REFERENCE-ACTEUR (Roman)
/ActorNamed (Roman)
/Prenom (Roman)
/SEXE_MASCULIN (Roman)
Résultat analyse : reconnaissance des EN 'Personne'

Nos films sont très différents, mais nous partageons le même goût de la liberté, le même désir de surprendre. Il y a tellement de films interchangeables aujourd'hui. Moi, j'aime les films qui sont uniques, personnels, même s'ils sont ratés. » Le parrain peut se reposer en paix. La relève est assurée.

Annexe III. Résultats des évaluations

Cette annexe regroupe l'ensemble des résultats obtenus pour les évaluations des projets de la presse people et de l'édition juridique.

III.1 L'évaluation du projet de la Presse People

- **Présentation du corpus de validation :**

document	taille fichier	temps (ms)	classes xtm	attributs xtm	relations xtm	rôles xtm	prédicats rdf
Coppola	13	32500	135 personnalités + 7 œuvres	1 date naissance + 1 lieu naissance + 7 liens parenté	3 mariages	6 conjoints	57 personnalités + 2 pdc + 7 œuvres + 1 lieu
doc1	32	109266	219 personnalités + 8 œuvres	4 dates naissance + 2 liens parenté + 2 dates mariage	5 mariages + 1 casting	7 conjoints + 1 œuvre + 1 acteur	137 personnalités + 13 pdc + 8 oeuvres + 1 lieu
doc2	32	74234	172 personnalités + 1 œuvre	2 dates naissance	0	0	95 personnalités + 13 pdc + 1 œuvre
doc4	31	69563	174 personnalités + 6 œuvres	1 date mariage	3 mariages + 1 famille	5 conjoints + 2 parents + 1 enfant	112 personnalités + 2 pdc + 6 œuvres + 1 lieu
doc5	32	59500	185 personnalités + 9 œuvres	2 dates mariage	4 mariages + 2 castings	7 conjoints + 2 œuvres + 1 acteur	101 personnalités + 11 pdc + 9 œuvres
doc6	32	62454	176 personnalités + 4 œuvres	4 liens parenté + 3 dates mariage	3 mariages + 1 famille	5 conjoints + 1 parent + 2 enfants	103 personnalités + 12 pdc + 4 œuvres + 1 lieu
doc7	32	95875	153 personnalités + 4 œuvres	0	0	0	92 personnalités + 3 pdc + 4 œuvres
doc8	32	49328	217 personnalités + 3 œuvres	4 dates naissance + 1 lieu naissance + 2 liens parenté +	4 mariages	6 conjoints	114 personnalités + 13 pdc + 3 œuvres + 1 lieu
doc9	32	74594	182 personnalités + 4 œuvres	1 lien parenté	4 mariages + 1 casting	6 conjoints + 1 œuvre + 1 acteur	101 personnalités + 7 pdc + 4 œuvres
doc11	33	50734	120 personnalités + 3 œuvres	0	0	0	77 personnalités + 1 pdc + 3 œuvres
doc12	34	55594	162 personnalités + 1 œuvre	2 dates naissance + 4 liens parenté + 5 dates mariage +	7 mariages	14 conjoints	130 personnalités + 15 pdc + 1 œuvre + 2 lieux
doc14	33	72188	212 personnalités + 3 œuvres	2 dates naissance + 1 lien parenté + 2 dates mariage	12 mariages	19 conjoints	152 personnalités + 3 pdc + 3 œuvres
doc16	34	105375	222 personnalités + 3 œuvres	2 dates naissance + 5 liens parenté	1 famille	1 parent + 1 enfant	149 personnalités + 24 pdc + 3 œuvres
doc17	34	53500	164 personnalités + 5 œuvres	1 lien parenté	6 mariages	12 conjoints	106 personnalités + 15 pdc + 5 œuvres
doc18	31	43266	136 personnalités + 3 œuvres	2 dates naissance + 2 lieux naissance + 1 lien parenté	2 mariages	3 conjoints	68 personnalités + 5 pdc + 3 œuvres + 1 lieu
doc19	32	56047	192 personnalités + 1 œuvre	1 lien parenté + 1 date mariage	1 mariage	1 conjoint	128 personnalités + 7 pdc + 1 œuvre
doc20	32	51016	162 personnalités + 1 œuvre	1 lien parenté	1 casting	1 œuvre + 1 acteur	57 personnalités + 3 pdc + 1 œuvre
doc23	32	54922	214 personnalités + 5 œuvres	2 dates naissance + 4 dates mariage	7 mariages	12 conjoints	142 personnalités + 25 pdc + 5 œuvres
doc24	38	56797	178 personnalités + 2 œuvres	2 dates naissance	0	0	94 personnalités + 3 pdc + 2 œuvres
doc25	32	87781	227 personnalités + 6 œuvres	4 dates naissance + 4 liens parenté + 1 date mariage	6 mariages	11 conjoints	181 personnalités + 17 pdc + 6 œuvres
doc26	32	51531	211 personnalités + 3 œuvres	1 lien parenté + 1 date mariage	2 mariages	3 conjoints	141 personnalités + 11 pdc + 3 œuvres
doc28	31	54578	195 personnalités + 5 œuvres	2 dates naissance	0	0	118 personnalités + 7 pdc + 5 œuvres
doc29	32	42688	149 personnalités + 3 œuvres	0	0	0	33 personnalités + 1 pdc + 3 œuvres
doc30	32	53375	273 personnalités + 2 œuvres	2 dates naissance + 3 liens parenté + 2 dates mariage	5 mariages + 1 famille	4 conjoints + 1 parent + 1 enfant	172 personnalités + 21 pdc + 2 œuvres + 1 lieu
doc31	32	56047	206 personnalités + 1 personn	2 dates naissance + 1 lien parenté	1 mariage	2 enfants	126 personnalités + 4 pdc + 6 œuvres + 1 lieu
doc32	32	45610	194 personnalités + 5 œuvres	3 liens parenté + 4 dates mariage	5 mariages	6 conjoints	110 personnalités + 9 pdc + 5 œuvres
doc33	32	56703	193 personnalités + 4 œuvres	2 liens parenté + 4 dates mariage	1 famille	1 parent + 1 enfant	147 personnalités + 7 pdc + 4 œuvres
doc34	32	57703	201 personnalités + 1 œuvre	1 lien parenté	1 famille	1 parent + 1 enfant	145 personnalités + 3 pdc + 1 œuvre
doc35	32	49438	209 personnalités + 6 œuvres	2 liens parenté	5 mariages + 1 casting	10 conjoints + 1 œuvre + 1 acteur	144 personnalités + 10 pdc + 6 œuvres
doc36	35	52500	223 personnalités + 2 œuvres	2 dates naissance + 2 liens parenté + 3 dates mariage	4 mariages	8 conjoints	141 personnalités + 11 pdc + 2 œuvres + 2 lieux
doc37	32	53859	124 personnalités + 3 œuvres	0	1 mariage	2 conjoints	81 personnalités + 4 pdc + 3 œuvres
doc38	33	51000	123 personnalités + 3 œuvres	2 liens parenté + 1 date mariage	4 mariages	6 conjoints	89 personnalités + 5 pdc + 3 œuvres
doc39	32	54781	188 personnalités + 2 œuvres	2 dates naissance	3 mariages + 1 famille	3 conjoints + 1 parent + 1 acteur	113 personnalités + 7 pdc + 2 œuvres + 1 lieu
doc40	32	63485	144 personnalités + 1 œuvre	4 dates naissance	4 mariages	8 conjoints	87 personnalités + 6 pdc + 1 œuvre
doc41	43	61375	119 personnalités + 2 œuvres	2 dates naissance + 1 lien parenté + 1 lieu décès	2 mariages	4 conjoints	85 personnalités + 4 pdc + 2 œuvres + 1 lieu
doc42	33	64859	137 personnalités	1 lien parenté + 1 date décès	0	0	121 personnalités + 7 pdc
doc43	34	66500	207 personnalités	1 lien parenté + 1 date mariage	4 mariages	7 conjoints	147 personnalités + 6 pdc
doc44	43	76172	181 personnalités	0	4 mariages	4 conjoints	143 personnalités + 2 pdc
doc45	34	60375	134 personnalités + 1 œuvre	1 date décès	0	0	108 personnalités + 1 pdc + 1 œuvre
doc46	45	64796	196 personnalités	5 dates naissance + 2 liens parenté + 1 date décès	2 mariages	4 conjoints	157 personnalités + 3 pdc
doc47	41	56860	203 personnalités + 1 œuvre	0	0	0	129 personnalités + 1 pdc + 1 œuvre + 1 lieu
TOTAL	**1357**	**2508769**					
MOYENNE	**33,09756098**	**61189,4878**					
			Rappel global :	**0,953066**			
			Précision globale :	**0,910601**			
			F-Mesure globale :	**0,931349702**			

- **Complexité du domaine de la Presse People :**

Type de l'élément dans l'ontologie	Element de l'ontologie	Nombre de RAC pour cet élément	Nombre d'instances dans corpus	Nombre RAC * Nombre Instances d'éléments
Classes	Personnalités	4	7412	29648
	Œuvres	1	124	124
	Personnages	1	1	1
	Article	1	41	41
	Sous-Total	*7*	*7578*	*29814*
Attributs	date naissance	1	48	48
	lieu naissance	1	4	4
	lien parenté	2	56	112
	date mariage	1	39	39
	lieu mariage	1	0	0
	date décès	1	4	4
	lieu décès	1	1	1
	date funéraille	1	0	0
	lieu inhumation	1	0	0
	indexation personnalité	4	3533	14132
	indexation œuvre	2	119	238
	indexation lieu géographique	4	15	60
	plan classement personnalités	23	267	6141
	Sous-Total	*43*	*4086*	*20779*
Relations	Mariage	7	113	791
	Famille	1	7	7
	Distribution	2	6	12
	Equipe Technique	1	0	0
	Inspiration	2	0	0
	Agent	1	0	0
	Sous-Total	*14*	*126*	*810*
Rôles	conjoint	6	183	1098
	parent	3	8	24
	enfant	2	9	18
	œuvre concernée distribution	1	6	6
	acteur	4	6	24
	rôle	1	0	0
	œuvre concernée équipe	1	0	0
	créateur	3	0	0
	scénariste	3	0	0
	producteur	3	0	0
	œuvre concernée inspiration	2	0	0
	personnalité inspiration	1	0	0
	agent	3	0	0
	personne concernée	3	0	0
	Sous-Total	*36*	*212*	*1170*
TOTAL		**100**	**12002**	**52573**
	Moyenne de RAC par élément:	**2,702702703**		
	Complexité du domaine:	**4,380353274**		

- **Tableau des résultats de l'application des RACs pour la tâche de peuplement d'ontologie :**

F mesure topics	nb attributs existants	nb attributs correctement acquis	nb attributs acquis	rappel attributs	précision attributs	F mesure attributs	nb relations existantes	nb relations correctement acquises	nb relations acquises	rappel relations	précision relations	F mesure relations	nb roles existants	nb roles correctement acquis	nb roles acquis	rappel roles	précision roles	F mesure roles
0,965986	9	9	9	1	1	1	2	2	3	1	0,666667	0,8	4	4	6	1	0,6666667	0,8
0,982684	6	6	8	1	0,75	0,8571429	6	6	6	1	1	1	9	9	9	1	1	1
0,994253	1	1	2	1	0,5	0,6666667	0	0	0				0	0	0			
0,980716	1	1	1	1	1	1	3	3	4	1	0,75	0,857143	6	6	8	1	0,75	0,857143
0,97733	2	2	2	1	1	1	5	4	6	0,8	0,666667	0,727273	8	8	10	1	0,8	0,888889
0,978261	4	4	5	1	0,8	0,8888889	4	3	4	0,75	0,75	0,75	8	8	10	1	0,8	0,888889
0,996825	0	0	0				0	0	0				0	0	0			
0,986547	5	5	8	1	0,625	0,7692308	4	3	4	0,75	0,75	0,75	7	4	6	0,57143	0,6666667	0,615385
0,981333	1	1	1	1	1	1	6	4	5	0,666667	0,8	0,727273	10	6	8	0,6	0,75	0,666667
1	0	0	0				0	0	0				0	0	0			
0,987879	8	8	12	1	0,6666667	0,8	4	4	7	1	0,571429	0,727273	8	8	14	1	0,5714286	0,727273
0,988506	4	3	5	0,75	0,6	0,6666667	8	8	12	1	0,666667	0,8	11	11	19	1	0,5789474	0,733333
0,965217	6	6	7	1	0,8571429	0,9230769	3	1	1	0,333333	1	0,5	6	2	2	0,33333	1	0,5
0,99705	1	1	1	1	1	1	4	4	6	1	0,666667	0,8	8	8	12	1	0,6666667	0,8
0,989324	4	3	5	0,75	0,6	0,6666667	3	2	2	0,666667	1	0,8	5	3	3	0,6	1	0,75
0,984456	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0,984894	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
1	3	3	6	1	0,5	0,6666667	6	6	7	1	0,857143	0,923077	10	10	12	1	0,8333333	0,909091
0,991736	1	1	2	1	0,5	0,6666667	0	0	0				0	0	0			
0,987179	8	7	9	0,875	0,7777778	0,8235294	4	4	6	1	0,666667	0,8	7	7	11	1	0,6363636	0,777778
1	2	2	2	1	1	1	2	2	2	1	1	1	3	3	3	1	1	1
0,982716	1	1	2	1	0,5	0,6666667	2	0	0	0			6	0	0	0		
0,996721	0	0	0				0	0	0				0	0	0			
0,987387	5	5	7	1	0,7142857	0,8333333	4	4	5	1	0,8	0,888889	6	6	8	1	0,75	0,857143
0,990654	2	2	3	1	0,6666667	0,8	1	1	1	1	1	1	2	2	2	1	1	1
0,980296	8	7	7	0,875	1	0,9333333	5	5	5	1	1	1	6	6	6	1	1	1
0,98995	4	4	6	1	0,6666667	0,8	1	1	1	1	1	1	2	2	2	1	1	1
0,992629	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
0,993072	2	2	2	1	1	1	4	4	6	1	0,666667	0,8	8	8	12	1	0,6666667	0,8
0,991189	5	7	9	1,4	0,7777778	1	3	3	4	1	0,75	0,857143	6	6	8	1	0,75	0,857143
0,992188	0	0	0				1	1	1	1	1	1	2	2	2	1	1	1
0,992126	3	3	3	1	1	1	4	3	4	0,75	0,75	0,75	7	4	6	0,57143	0,6666667	0,615385
0,987013	1	1	2	1	0,5	0,6666667	4	4	4	1	1	1	5	5	5	1	1	1
0,993151	2	2	4	1	0,5	0,6666667	2	2	4	1	0,5	0,666667	4	4	8	1	0,5	0,666667
0,991803	3	3	4	1	0,75	0,8571429	1	1	2	1	0,5	0,666667	2	2	4	1	0,5	0,666667
0,996364	2	2	2	1	1	1	0	0	0				0	0	0			
0,99759	2	2	2	1	1	1	4	4	4	1	1	1	7	7	7	1	1	1
1	0	0	0				4	4	4	1	1	1	4	4	4	1	1	1
0,992647	1	1	1	1	1	1	0	0	0				0	0	0			
0,994924	6	6	8	1	0,75	0,8571429	1	1	2	1	0,5	0,666667	2	2	4	1	0,5	0,666667
1	0	0	0				0	0	0				0	0	0			
0,989028	**117**	**115**	**151**	**0,982906**	**0,7615894**	**0,858209**	**108**	**97**	**125**	**0,898148**	**0,776**	**0,832618**	**184**	**162**	**216**	**0,88043**	**0,75**	**0,81**
0,989332	**2,8536585**	**2,804878049**	**3,6829268**	**0,99**	**0,8000567**	**0,8707473**	**2,634146**	**2,365853659**	**3,0487805**	**0,900505**	**0,821205**	**0,851815**	**4,487805**	**3,951219512**	**5,2682927**	**0,89928**	**0,8141689**	**0,845129**

- **Tableau des résultats de la consolidation des instances de l'ontologie :**

document	nb topics validés	nb attributs validés	nb relations validées	nb rôles validés	nb topics tampon	nb attributs tampon	nb relations tampon	nb rôles tampon
Coppola	59	9	3	6	6	1	0	0
doc1	153	7	3	6	5	1	3	3
doc2	118	2	0	0	4	0	0	0
doc4	120	0	3	7	3	1	1	1
doc5	90	1	4	8	4	1	2	2
doc6	123	5	3	7	2	1	1	1
doc7	98	0	0	0	2	0	0	0
doc8	110	6	2	4	4	4	2	2
doc9	115	1	1	2	5	0	6	8
doc11	57	0	0	0	2	0	0	0
doc12	109	12	7	14	3	0	0	0
doc14	159	5	7	14	5	0	3	3
doc16	125	7	1	2	3	0	0	0
doc17	115	1	6	12	4	0	0	0
doc18	90	3	1	2	4	1	1	1
doc19	112	0	0	0	7	1	1	1
doc20	67	1	1	1	3	0	0	0
doc23	124	6	5	10	4	0	2	2
doc24	73	2	0	0	1	0	0	0
doc25	112	8	5	10	7	1	1	1
doc26	92	1	1	2	5	1	1	1
doc28	93	2	0	0	3	0	0	0
doc29	61	0	0	0	0	0	0	0
doc30	147	7	3	6	5	0	2	2
doc31	117	3	1	2	7	0	0	0
doc32	101	3	1	2	5	4	4	4
doc33	100	6	1	2	3	0	0	0
doc34	138	1	1	2	4	0	0	0
doc35	133	2	6	12	3	0	0	0
doc36	141	7	4	8	8	0	0	0
doc37	89	0	1	2	1	0	0	0
doc38	95	2	2	4	3	1	2	2
doc39	124	2	1	2	4	0	3	3
doc40	98	4	4	8	4	0	0	0
doc41	78	3	2	4	5	1	0	0
doc42	93	2	0	0	0	0	0	0
doc43	95	1	3	6	6	1	1	1
doc44	101	0	0	0	10	0	4	4
doc45	86	1	0	0	1	0	0	0
doc46	95	8	2	4	2	0	0	0
doc47	63	0	0	0	2	0	0	0
TOTAL	**4269**	**131**	**85**	**171**	**159**	**20**	**40**	**42**
MOYENNE	**104,1219512**	**3,19512**	**2,07317**	**4,170732**	**3,87804878**	**0,4878049**	**0,9756098**	**1,0243902**

	acquis	validés et créés	rejetés tampon	supprimés
classe	7548	4269	159	3120
attribut	151	131	20	0
relation	125	**85**	**40**	0
rôle	216	**171**	**42**	3

- **Tableau des résultats de l'application des RACS pour la tâche d'annotation sémantique :**

document	nb predicats existants	nb predicats matchés	predicats mal matchés	rappel attributs	précision attributs	F mesure attributs
Coppola	70	67	67	0,9571429	1	0,9781022
doc1	166	159	159	0,9578313	1	0,9784615
doc2	109	109	109	1	1	1
doc4	125	121	121	0,968	1	0,9837398
doc5	127	121	121	0,9527559	1	0,9758065
doc6	126	120	120	0,952381	1	0,9756098
doc7	101	99	99	0,980198	1	0,99
doc8	140	131	131	0,9357143	1	0,9667897
doc9	117	112	112	0,957265	1	0,9781659
doc11	82	81	81	0,9878049	1	0,993865
doc12	158	148	148	0,9367089	1	0,9673203
doc14	166	158	158	0,9518072	1	0,9753086
doc16	181	176	176	0,9723757	1	0,9859944
doc17	134	126	126	0,9402985	1	0,9692308
doc18	80	77	77	0,9625	1	0,9808917
doc19	139	136	136	0,9784173	1	0,9890909
doc20	65	61	61	0,9384615	1	0,968254
doc23	180	172	172	0,9555556	1	0,9772727
doc24	100	99	99	0,99	1	0,9949749
doc25	208	204	204	0,9807692	1	0,9902913
doc26	161	155	155	0,9627329	1	0,9810127
doc28	131	130	130	0,9923664	1	0,9961686
doc29	39	37	37	0,9487179	1	0,9736842
doc30	202	196	196	0,970297	1	0,9849246
doc31	140	137	137	0,9785714	1	0,9891697
doc32	125	124	124	0,992	1	0,9959839
doc33	158	158	158	1	1	1
doc34	152	149	149	0,9802632	1	0,9900332
doc35	161	160	160	0,9937888	1	0,9968847
doc36	163	156	156	0,9570552	1	0,9780564
doc37	90	88	88	0,9777778	1	0,988764
doc38	103	97	97	0,9417476	1	0,97
doc39	130	123	123	0,9461538	1	0,972332
doc40	99	94	94	0,9494949	1	0,9740933
doc41	99	92	92	0,9292929	1	0,9633508
doc42	128	128	128	1	1	1
doc43	154	153	153	0,9935065	1	0,9967427
doc44	147	145	145	0,9863946	1	0,9931507
doc45	110	110	110	1	1	1
doc46	161	160	160	0,9937888	1	0,9968847
doc47	132	132	132	1	1	1
TOTAL	**5359**	**5201**	**5201**	**0,9705169**	**1**	**0,9850379**
MOYENNE	**130,7073**	**126,85366**	**126,85366**	**0,9695107**	**1**	**0,9844002**

Rappel pour l'annotation :	**0,97051689**
Précision pour l'annotation :	**1**
F-mesure pour l'annotation :	**0,98503788**

- **Tableau des résultats de la consolidation des annotations:**

document	nb prédicats validés	nb prédicats tampon
Coppola	42	9
doc1	132	27
doc2	82	10
doc4	80	10
doc5	61	19
doc6	85	11
doc7	61	11
doc8	80	6
doc9	72	8
doc11	31	10
doc12	86	14
doc14	105	11
doc16	95	12
doc17	90	14
doc18	58	7
doc19	76	18
doc20	44	6
doc23	97	11
doc24	34	10
doc25	90	17
doc26	67	11
doc28	67	9
doc29	31	2
doc30	95	15
doc31	73	22
doc32	73	12
doc33	72	8
doc34	91	13
doc35	98	12
doc36	91	21
doc37	59	8
doc38	73	9
doc39	74	11
doc40	59	13
doc41	54	11
doc42	66	22
doc43	72	13
doc44	70	14
doc45	54	12
doc46	63	11
doc47	38	11
TOTAL	**2941**	**501**
MOYENNE	**71,7317073**	**12,2195122**

	acquises	validées et créées	rejetées tampon	supprimées
annotations	5201	2941	501	1759
pourcentage	100	56,54681792	9,63276293	33,820419

- **Tableau des résultats de l'expérience menée avec les étudiants du M2 en Information et Communication :**

Annotateur	Temps	Classes					Relations								Attributs			
	(min)	Personnalité	Personnage	Œuvre	Film	Société	Mariage	Naissance	Filmographie	Bibliographie	Inspiration	Goût	Nécrologie	Santé	Parenté	Lieu	Date	TOTAL
1	35	145	2	2	25	3	2	1				2	1		26	18	8	**235**
2	40	120	10	16	28	30	17	3	12	12	3	6	4	4	52	17	17	**351**
3		21		3	5	1	14	1	14		5	5	2	1	30	1	3	**106**
4	40	71	3	1	23	3	7	3	3			2	1	2	14	18	7	**158**
5	45	139	3	9	29	4	7		3		1	1	2	2	49	17	8	**274**
6	25	137	2	11	23		8	2	1	2	2	1	3	2	29	18	15	**256**
7	30	160	3	5	21	1	11	1	13		3	2	1	3	46	22	19	**311**
8	29	108	6	1	26		10		5			2	6		24	18	7	**213**
9	30	98	1	1	24	4	6	2	5		4	1	1	1	12	13	6	**179**
10	30	177	4	2	26	4	6	5	11	2	3	4	1	1	47	19	16	**328**
11	30	81	2	3	26	0	7	1			1	1	2	1	31	16	9	**181**
12	36	46	2	2	24	3	5	2		1		1	1		29	15	10	**141**
13	40	145	6	1	25	2	6	2		2		10	2	1	44	20	8	**274**
14	18	98	3	2	25	4	7	4	5	1	2	7	2	2	15	15	12	**204**
15	40	111	3	1	23	4	13	2	21	4	3	8	5	1	34	18	17	**268**
16	30	136	5	17	33		17	1	2	1	1	3	7	2	60	26	19	**330**
17	37	149	3	2	26	3		2	1	1		2	1		27	23	6	**246**
18	36	143	2	2	27	4	10	1				9	5	2	58	24	10	**297**
19	35	158	5	2	26	1		1	2						47	18	8	**268**
20		*152*	*2*	*2*	*26*	*4*	*9*	*2*	*30*	*1*	*1*	*2*	*2*	*2*	*37*	*18*	*7*	**297**
Moyenne	**33,667**	**119,75**	**3,526315789**	**4,25**	**24,6**	**4,41176**	**9**	**2**	**8,533333333**	**2,7**	**2,5454545**	**3,63**	**2,57894737**	**1,8**	**35,55**	**17,7**	**10,6**	**243,158**
OntoPop	**0,325**	**135**		**7**			**3**	**1**	**1**					**1**	**7**	**1**		**156**

III.2 L'évaluation du projet de l'édition juridique

- **Présentation du corpus de validation :**

document Balisage Renvois	taille fichier
refsource 2005-1	123
refsource 2005-2	112
refsource 2005-3	107
refsource 2005-4	144
TOTAL	**486**
MOYENNE	**121,5**

document Balisage Renvois	taille fichier
CAAbordeaux_ch1	8
CAAbordeaux_ch2	7
CAAbordeaux_ch3	5
CAAbordeaux_ch4	15
CAAbordeaux_ch5	6
CAAbordeaux_jugeAppelReferes	10
CAAbordeaux_jugeReconduites	6
CAABordeaux_jugeReferes	6
CAABordeaux_pdt	12
CAABordeaux_PdtCh1	4
CAABordeaux_PdtCh2	6
CAADouai_ch1	7
CAADouai_ch2	10
CAADouai_ch3	9
CAADouai_pdtDeleguePdt	16
CAADouai_plen	8
CAALyon_audPlen	11
CAALyon_ch1	12
CAALyon_ch2	6
CAALyon_ch3	7
CAALyon_ch4	7
CAALyon_ch6	10
CAALyon_formPlen	11
CAAMarseille_ch1	10
CAAMarseille_ch2	6
CAAMarseille_ch3	21
CAAMarseille_ch4	7
CAAMarseille_ch5	30
CAAMarseille_ch6	19
CAAMarseille_plen	6
CAANancy_ch1	9
CAANancy_ch2	10
CAANancy_ch3	12
CAANancy_ch4	13
CAANancy_plen	14
CAANantes_ch1	14
CAANantes_ch2	14
CAANantes_ch3	12
CAANantes_ch4	6
CAANantes_plen	10
CAAParis_ch1	7
CAAParis_ch2	14
CAAParis_ch3	12
CAAParis_ch4	11
CAAParis_ch5	5
CAAParis_formPerm	9
CAAParis_plen	7
CAAversailles_ch1	9
CAAversailles_ch2	9
CAAversailles_ch3	19
CAAversailles_magistratDelegue	13
CASS_assplen	11
CASS_avis	4
CASS_civ1	5
CASS_civ2	5
CASS_civ3	5
CASS_com	14
CASS_crim	18
CASS_soc	4
CAversailles_ch1	16
CAversailles_ch2	18
CAversailles_ch3	22
CAversailles_ch4	13
CAversailles_ch5	26
CAversailles_ch6	18
CAversailles_ch9	262
CAversailles_ch12	47
CAversailles_ch13	11
CAversailles_ch14	11
CAversailles_ch16	27
CAversailles_ch17	20
CAversailles_ch24	20
CAversailles_chCommReunies	32
CE_consdEtatDeleguePdtCont	5
CE_jugeReferes	13
CE_pdtSectCont	4
CE_sectCont_SS3et8	9
CE_sectCont_SS6	7
TCONFL	12
TAparis_sect1	16
TAparis_sect2	12
TAparis_sect3	9
TAparis_sect4	8
TAparis_sect5	9
TAparis_sect6	16
TAparis_sect7	7
TAparis_magistratDelegue	8
TAparis_magistratDesigne	14
TAparis_jugeReferes	7
TOTAL	**307**
MOYENNE	**9,9032258**

Rappel global :	**0,992457**
Précision globale :	**0,994632**
F-Mesure globale :	**0,99354331**

- **Complexité du Domaine de l'Edition Juridique :**

Type de l'élément dans l'ontologie	Element de l'ontologie	Nombre de RAC pour cet élément	Nombre d'instances dans corpus	Nombre RAC * Nombre Instances d'éléments
Classes	Content Unit	1	93	93
	Référence DE Jurisprudence	1	89	89
	Référence A Légistation TC	1	1	1
	Référence A Légistation TC art	1	0	0
	Référence A Légistation TNC	2	16	32
	Référence A Légistation TNC art	2	0	0
	Référence A Publication	1	3	3
	Référence A Publication CU	4	7	28
	Référence A Jurisprudence	2	123	246
	Sous-Total	*15*	*332*	*492*
Attributs	document pivot	1	93	93
	classification term	1	49	49
	type formation	3	181	543
	siège formation	3	2	6
	type juridiction	3	276	828
	siège juridiction	3	203	609
	numéro jurisprudence	18	216	3888
	type décision	5	85	425
	date décision	4	225	900
	NOR	2	0	0
	partie	14	142	1988
	numéro jurisdata	1	0	0
	nom code	2	1	2
	numéro article TC	1	0	0
	type Texte	6	16	96
	appellation	3	0	0
	numéro texte	8	8	64
	émetteur rattachement	10	1	10
	NOR	3	0	0
	date texte	6	16	96
	numéro article TNC	3	0	0
	nom publication	3	10	30
	émetteur	6	0	0
	numéro publication	2	1	2
	date publication	2	10	20
	page début	1	7	7
	matricule	2	0	0
	Sous-Total	*116*	*1542*	*9656*
Relations	Références de CU	1	89	89
	Renvoi Simple	5	150	750
	Renvoi Intervalle	2	0	0
	Sous-Total	*8*	*239*	*839*
Rôles	CU reference	1	89	89
	referenced CU	1	89	89
	référence	1	150	150
	source	2	150	300
	référence début	2	0	0
	référence fin	2	0	0
	Sous-Total	*9*	*478*	*628*
TOTAL		**148**	**2591**	**11615**
	Moyenne de RAC par élément:	**4**		
	Complexité du domaine:	**4,482825164**		

- **Tableau des résultats pour la tâche de balisage des renvois (application des RACs) :**

document	nb topics existants	nb topics correctement annotés	nb topics annotés	rappel topics	précision topics	F mesure topics	nb attributs existants	nb attributs correctement annotés	attributs annotés	rappel attributs	précision attributs	F mesure attributs	nb relations existantes	nb relations correctement annotés	nb relations annotés	rappel relations	précision relations	F mesure relations	nb roles existants	nb roles correctement annotés	nb roles annotés	rappel roles	précision roles	F mesure roles
refsource 2005-1	54	52	54	0,962963	0,962963	0,962963	228	228	228	1	1	1	50	50	50	1	1	1	100	100	100	1	1	1
refsource 2005-2	51	51	51	1	1	1	187	187	187	1	1	1	50	50	50	1	1	1	100	100	100	1	1	1
refsource 2005-3	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0				0	0	0			
refsource 2005-4	51	46	51	0,901961	0,901961	0,901961	169	169	169	1	1	1	50	50	50	1	1	1	100	100	100	1	1	1
TOTAL	**157**	**150**	**157**	**0,955414**	**0,955414**	**0,955414**	**585**	**585**	**585**	**1**	**1**	**1**	**150**	**150**	**150**	**1**	**1**	**1**	**300**	**300**	**300**	**1**	**1**	**1**
MOYENNE	**39,25**	**37,5**	**39,25**	**0,966231**	**0,966231**	**0,966231**	**146,25**	**146,25**	**146,25**	**1**	**1**	**1**	**37,5**	**37,5**	**37,5**	**1**	**1**	**1**	**75**	**75**	**75**	**1**	**1**	**1**
	Rappel pour le balisage :		0,988854																					
	Précision pour le balisage :		0,988854																					
	F-mesure pour le balisage :		0,988854																					

- **Tableau des résultats pour la tâche d'identification des décisions de jurisprudence (application des RACs) :**

document	nb topics existants	nb topics correctement annotés	nb topics annotés	rappel topics	précision topics	F mesure topics	nb attributs existants	nb attributs correctement annotés	nb attributs annotés	rappel attributs	précision attributs	F mesure attributs	nb relations existantes	nb relations correctement annotés	nb relations annotés	rappel relations	précision relations	F mesure relations	nb roles existants	nb roles correctement annotés	nb roles annotés	rappel roles	précision roles	F mesure roles
CAAbordeaux_ch1	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_ch2	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_ch3	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_ch4	2	2	2	1	1	1	13	13	13	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_ch5	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_jugeA	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAbordeaux_jugeR	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAABordeaux_jugeR	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAABordeaux_pdt	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAABordeaux_PdtCI	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAABordeaux_PdtCI	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAADouai_ch1	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAADouai_ch2	2	2	2	1	1	1	12	12	12	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAADouai_ch3	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAADouai_pdtDelegu	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAADouai_plen	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_audPlen	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_ch1	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_ch2	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_ch3	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_ch4	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_ch6	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAALyon_formPlen	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch1	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch2	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch3	2	2	2	1	1	1	12	12	12	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch4	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch5	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_ch6	2	2	2	1	1	1	14	14	14	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAMarseille_plen	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANancy_ch1	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANancy_ch2	2	2	2	1	1	1	13	13	13	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANancy_ch3	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANancy_ch4	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANancy_plen	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANantes_ch1	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANantes_ch2	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANantes_ch3	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANantes_ch4	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAANantes_plen	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_ch1	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_ch2	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_ch3	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_ch4	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_ch5	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_formPerm	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAParis_plen	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAversailles_ch1	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAversailles_ch2	2	2	2	1	1	1	12	11	11	0,9166667	1	0,9565217	1	1	1	1	1	1	2	2	2	1	1	1
CAAversailles_ch3	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAAversailles_magis	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_assplen	2	2	2	1	1	1	12	12	12	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_avis	2	2	2	1	1	1	4	4	4	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_civ1	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_civ2	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_civ3	2	2	2	1	1	1	8	8	8	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_com	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_crim	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CASS_soc	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch1	2	2	2	1	1	1	9	8	8	0,8888889	1	0,9411765	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch2	2	2	2	1	1	1	12	11	11	0,9166667	1	0,9565217	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch3	2	2	2	1	1	1	10	9	9	0,9	1	0,9473684	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch4	2	2	2	1	1	1	11	10	10	0,9090909	1	0,952381	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch5	2	2	2	1	1	1	10	9	9	0,9	1	0,9473684	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch6	2	2	2	1	1	1	10	9	9	0,9	1	0,9473684	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch9	2	2	2	1	1	1	9	8	8	0,8888889	1	0,9411765	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch12	2	2	2	1	1	1	13	12	12	0,9230769	1	0,96	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch13	2	2	2	1	1	1	9	8	8	0,8888889	1	0,9411765	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch14	2	2	2	1	1	1	10	9	9	0,9	1	0,9473684	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch16	2	2	2	1	1	1	11	10	10	0,9090909	1	0,952381	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch17	2	2	2	1	1	1	12	11	11	0,9166667	1	0,9565217	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_ch24	2	2	2	1	1	1	11	10	10	0,9090909	1	0,952381	1	1	1	1	1	1	2	2	2	1	1	1
CAversailles_chComr	2	2	2	1	1	1	11	10	10	0,9090909	1	0,952381	1	1	1	1	1	1	2	2	2	1	1	1
CE_consdEtatDelegu	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CE_jugeReferes	2	2	2	1	1	1	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CE_pdtSectCont	2	2	2	1	1	1	5	5	5	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CE_sectCont_SS3et	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
CE_sectCont_SS6	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TCONFL	2	2	2	1	1	1	9	9	9	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect1	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect2	2	2	2	1	1	1	13	13	13	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect3	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect4	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect5	2	2	2	1	1	1	17	17	17	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect6	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_sect7	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_magistratDe	2	2	2	1	1	1	13	13	13	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_magistratDe	2	2	2	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TAparis_jugeReferes	2	2	2	1	1	1	11	11	11	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
TOTAL	**178**	**178**	**178**	**1**	**1**	**1**	**862**	**847**	**847**	**0,9825986**	**1**	**0,9912229**	**89**	**89**	**89**	**1**	**1**	**1**	**178**	**178**	**178**	**1**	**1**	**1**
MOYENNE	**2**	**2**	**2**	**1**	**1**	**1**	**9,6853933**	**9,516853933**	**9,5168539**	**0,9840012**	**1**	**0,9915965**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**1**	**1**	**1**

Rappel pour l'identification :	0,99564965
Précision pour l'identification :	1
F-mesure pour l'identification :	0,99782008

Références bibliographiques

[ALA 03] – ALANI H., KIM S., MILLARD D. E., Weal M. J., Hall W., Lewis P. H. & Shadbolt N., Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation, in *Proceedings of the Knowledge Markup and Semantic Annotation Workshop (SEMANNOT'03)*, Sanibel, Florida, 2003.

[AMA 04] – AMARDEILH F. & FRANCART T., A Semantic Web Portal with HLT Capabilities, in *Actes du Colloque Veille Stratégique Scientifique et Technologique (VSST'04)*, Toulouse, France, 2004, Vol.2, pp. 481-492.

[AMA 05a] – AMARDEILH F., LAUBLET P. & MINEL J.-L., Annotation documentaire et peuplement d'ontologie à partir d'extractions linguistiques, in *Actes de la Conférence Ingénierie des Connaissances (IC'05)*, Nice, France, 2005, 12 p.

[AMA 05b] – AMARDEILH F., LAUBLET P. & MINEL J.-L., Document Annotation and ontology population from linguistic extractions, in *Proceedings of the International Conference on Knowledge Capture (KCAP'05)*, Banff, Canada, 1-5 octobre 2005, p. 161-168.

[AMA 06a] – AMARDEILH F. & FRANCART T., Enrichissement de bases de connaissances par l'annotation sémantique, in Ingénierie des Systèmes d'Information (ISI), Edition spéciale « Systèmes d'information stratégiques », 11(2), Editions Hermès, 2006, pp. 53-70.

[AMA 06b] – AMARDEILH F., OntoPop or how to annotate documents and populate ontologies from texts, in *Proceedings of the Workshop on Mastering the Gap: From Information Extraction to Semantic Representation (ESWC'06)*, Budva, Montenegro, 2006, CEUR Workshop Proceedings, ISSN 1613-0073, 2006.

[AMA 06c] – AMARDEILH F., CARLONI O. & NOEL L., PressIndex: a Semantic Web Press Clipping Application, in *Proceedings of the ISWC 2006 Semantic Web Challenge*, Athens, Georgia, USA, 2006.

[APP 99] – APPELT D. & ISRAEL D., Introduction to information extraction technology, in *International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, 1999.

[ARP 01] – ARPIREZ J.C., CORCHO O., FERNÁNDEZ-LÓPEZ M. & GÓMEZ-PÉREZ A., WebODE : a Workbench for Ontological Engineering, in *Proceedings of the International Conference on Knowledge Capture (K-CAP'01)*, Victoria, Canada, 2001.

[AUS 00a] – AUSSENAC N. & SEGUELA P., Les relations sémantiques : du linguistique au formel, in Cahiers de grammaire, Numéro spécial « Sémantique et Corpus », Volume 25, Presses de l'UTM, Toulouse, 2000, pp. 175-198.

[AUS 00b] – AUSSENAC-GILLES N., BIEBOW B. & SZULMAN S., Revisiting ontology design: a method based on corpus analysis, in *Proceedings of the European Knowledge Acquisition Conference (EKAW'2000)*, Springer-Verlag LNCS 1937, 2000, pp. 172-188.

[AUS 03] – AUSSENAC-GILLES N. & CONDAMINES A., Rapport final de l'action spécifique « Corpus et Terminologie », 2003.

[BAC 00] – BACHIMONT B., Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances, in Ingénierie des Connaissances : Evolutions récentes et nouveaux défis, CHARLET J., ZACKLAD M., KASSEL G. & BOURIGAULT D. (Eds.), Eyrolles, 2000, pp. 305-323.

[BAC 00] – BACHIMONT B., Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances, in Ingénierie des Connaissances : Evolutions récentes et nouveaux défis, CHARLET J., ZACKLAD M., KASSEL G. & BOURIGAULT D. (Eds.), Eyrolles, 2000.

[BAC 01] – BACHIMONT B., Modélisation linguistique et modélisation logique des ontologies : l'apport de l'ontologie formelle, in *Actes des journées francophones d'Ingénierie des Connaissances (IC'2001)*, Presse Universitaire de Grenoble, 2001.

[BAC 02] – BACHIMONT B., ISAAC A. & TRONCY R., Semantic Commitment for Designing Ontologies : A Proposal, in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02)*, Springer-Verlag LNAI 2473, Sigüenza, Spain, 2002.

[BAC 05] – BACHIMONT B., Corpus et connaissances : de l'extraction linguistique à la modélisation conceptuelle, in Sémantique et corpus, Condamines A. (Ed.), Hermès, Traité IC2, Cognition et Traitement de l'Information, Hermès, 2005, pp. 319-346.

[BAG 04] – BAGET J.-F., CANAUD E., EUZENAT J. & HACID M.-S., Les langages du Web Sémantique, in Le Web sémantique, CHARLET J., LAUBLET P. & REYNAUD C. (Ed.), Hors série de la Revue Information - Interaction - Intelligence (I3), 4(1), Cépaduès, Toulouse, 2004, pp. 21-43.

[BAN 07] – Baneyx A., Construire une ontologie de la pneumonie : aspects théoriques, modèles et expérimentations, Thèse de doctorat, Université Paris 6, 2007, 216 p.

[BEC 01] – BECHHOFER S. & GOBLE C., Towards Annotation using DAML+OIL, in *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (KCAP'01)*, Victoria, Canada, 2001.

[BEC 02] – BECHHOFER S., CARR L., GOBLE C., KAMPA S. & MILES-BOARD T., The Semantics of Semantic Annotation, in *Proceedings of the 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems (ODBASE'01)*, LNCS Volume 2519, Springer-Verlag, Irvine, Californie, 2002, pp. 1151-1167.

[BEC 03] – BECHHOFER S., GOBLE C., CARR L., KAMPA S. & HALL W., COHSE: Conceptual Open Hypermedia Service, in Annotation for the Semantic Web, HANDSCHUH S. & STAAB S. (Eds.), Frontiers in Artificial Intelligence and Applications, Volume 96, IOS Press, Springer-Verlag, 2003, pp. 193-211.

[BEN 99] – BENJAMINS V.R., FENSEL D., DECKER S. & GÓMEZ-PÉREZ A., (KA)2: building ontologies for the internet: a mid term report, in International Journal of Human Computer Studies, Volume 51, 1999, pp.687-712.

[BEN 02] – BENJAMINS V.R., CONTRERAS J., CORCHO O. & GÓMEZ-PÉREZ A., Six challenges for the semantic web, in *Proceedings of the Workshop on the Semantic Web (KR'02)*, Toulouse, France, 2002.

[BEN 05] – BENJAMINS V.R., CONTRERAS J., BLÁZQUEZ M., NIÑO M., GARCÍA A., NAVAS E., RODRÍGUEZ J., WERT C., MILLÁN R. & DODERO J.M., ONTO-H: A collaborative semiautomatic annotation tool, in *Proceedings of the 8th International Protégé Conference*, Madrid, Espagne, 2005.

[BER 98] – BERNERS-LEE T., Weaving the Web, Harper Eds, San Francisco, 1998, 226 p.

[BIE 99] – BIEBOW M. & SZULMAN S., TERMINAE : a method and a tool to build of a domain ontology, in *Proceedings of the 11th European Knowledge Acquisition Workshop (EKAW'99)*, Springer, 1999.

[BLA 98] – BLÁZQUEZ M., FERNÁNDEZ M., GARCÍA-PINAR J.M. & GÓMEZ-PÉREZ A., Building Ontologies at the Knowledge Level using the Ontology Design Environment, in *Proceedings of the 11th Knowledge Acquisition Workshop (KAW'98),* Banff, Canada, 1998.

[BLO 05] – BLOEHDORN S., PETRIDIS K., SAATHOFF C., SIMOU N., TZOUVARAS V., AVRITHIS Y., HANDSCHUH S., KOMPATSIARIS Y., STAAB S. & STRINTZIS M. G., Semantic Annotation of Images and Videos for Multimedia Analysis, in *Proceedings of the 2nd European Semantic Web Conference (ESWC'05)*, LNCS 3532, Springer-Verlag, Heraklion, Crête, Grèce, 2005, pp. 592-607.

[BOH 05] - BOHRING H. & AUER S., Mapping XML to OWL Ontologies, in *Proceedings of the 13th Leipziger Informatik-Tage (LIT'05)*, Leipzig, Allemagne, 2005.

[BON 03] – BONTCHEVA K. & CUNNINGHAM H., The Semantic Web: A New Opportunity and Challenge for Human Language Technology, in *Proceedings ot the Workshop on Human Language Technology for the Semantic Web and Web Services at the Second International Semantic Web Conference (ISWC'03)*, Sanibel Island, Floride, 2003.

[BOU 98] – BOUILLON P., Traitement automatique des langues naturelles, Editions Duculot, 1998, 248 p.

[BOU 03] – BOURIGAULT D. & AUSSENAC-GILLES N., Construction d'ontologies à partir de textes, in *Actes de la 10ème conférence annuelle sur le Traitement Automatique des Langues (TALN'03)*, Volume 2, Batz-sur-Mer, 2003, pp. 27-50.

[BOU 04] – BOURIGAULT D., AUSSENAC-GILLES N. & CHARLET J., Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas, In Techniques Informatiques et Structuration de Terminologies, PIERREL J.-M. ET SLODZIAN M. (Ed.), Numéro Spécial de la Revue d'Intelligence Artificielle (RIA), 18(1), Hermès, Paris, 2004, pp. 87-110.

[BUI 03] – BUITELAAR P., DECLERCK T., CALZOLARI N. & LENCI A., Towards A Language Infrastructure for the Semantic Web, in *Proceedings of the Workshop on Human Language Technology for the Semantic Web and Web Services in the International Semantic Web Conference (ISWC'03)*, Sanibel Island, Florida, 2003.

[BUI 05] – BUITELAAR P. & RAMAKA S., Unsupervised Ontology-based Semantic Tagging for Knowledge Markup, in *Proceedings of the Workshop on Learning in Web Search (ILMC'05)*, Bonn, Allemagne, 2005.

[CEL 04] – CELJUSKA D. & VARGAS-VERA M., Semi-automatic Population of Ontologies from Text, in *Proceedings of the Workshop on Data Analysis (WDA'04)*, Slovaquie, 2004, pp. 33-49.

[CHA 00] – CHARLET J., ZACKLAD M., KASSEL G. & BOURIGAULT D., Ingénierie des Connaissances : Evolutions récentes et nouveaux défis, Eyrolles, Paris, 2000.

[CHA 02] – CHARLET J., L'Ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales, Habilitation à diriger des recherches, Université Paris VI, 2002, 127 p.

[CHA 04] – CHARLET J., BACHIMONT B. & TRONCY R., Ontologies pour le Web Sémantique, in Le Web sémantique, CHARLET J., LAUBLET P. & REYNAUD C. (Ed.), Hors série de la Revue Information - Interaction - Intelligence (I3), 4(1), Cépaduès, Toulouse, 2004, pp. 69-100.

[CHA 05] – CHARNOIS T. & ENJALBERT P., Compréhension automatique, in Sémantique et traitement automatique du langage naturel, ENJALBERT P. (Ed.), Hermès, Traité IC2, Cognition et Traitement de l'Information, Paris, 2005, pp. 267-308.

[CIM 04] – CIMIANO P., HANDSCHUH S. & STAAB S., Towards the Self-Annotating Web, in *Proceedings of the 13th International World Wide Web Conference (WWW'04)*, ACM Press, New-York, USA, 2004, pp. 462-471.

[CIR 02] – CIRAVEGNA F., DINGLI A., PETRELLI D. & WILKS Y., User-System Cooperation in Document Annotation based on Information Extraction, in *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW'02)*, LNCS 2473, Springer-Verlag, Madrid, Espagne, 2002, pp. 122-138.

[CIR 03a] – CIRAVEGNA F., DINGLI A., GUTHRIE D. & WILKS Y., Integrating Information to Bootstrap Information Extraction from Web Sites, in *Proceedings of the Workshop on Information Integration on the Web (IJCAI'03)*, Acapulco, Mexique, 2003.

[CIR 03b] – CIRAVEGNA F. & WILKS Y., Designing Adaptive Information Extraction for the Semantic Web in Amilcare in Annotation for the Semantic Web, HANDSCHUH S. & STAAB S. (Eds.), Frontiers in Artificial Intelligence and Applications, Volume 96, IOS Press, Springer-Verlag, 2003, pp. 112-127.

[CIR 04] – CIRAVEGNA F., CHAPMAN S., DINGLI A. & WILKS Y., Learning to Harvest Information for the Semantic Web, in *Proceedings of the 1st European Semantic Web Symposium (ESWS'04)*, Springer-Verlag, Heraklion, Crête, Grèce, 2004.

[CON 03] – CONTRERAS J. & BENJAMINS R., Annotation tools and services, Délivrable 3.1, Esperonto Services Project, 2003, 67 p.

[CON 05] – CONDAMINES A., Sémantique et corpus, quelles rencontres possibles ?, in Sémantique et corpus, Condamines A. (Ed.), Hermès, Traité IC2, Cognition et Traitement de l'Information, Hermès, 2005, pp. 15-38.

[COI 96] – COIRIER P., GAONAC'H D. & PASSERAULT J.-L., Psycholinguistique textuelle : une approche cognitive de la compréhension et de la production des textes, Armand Colin, Paris, 1996, 297 p.

[COR 06] – CORCHO O., Ontology based document annotation: trends and open research problems, in International Journal of Metadata, Semantics and Ontologies, 1(1), Inderscience, 2006, pp. 47-57.

[CRI 03] – CRISPINO G., Une plate-forme informatique de l'Exploration Contextuelle : modélisation, architecture et réalisation (ContextO) – Application au filtrage sémantique de textes, Thèse de doctorat, Université Paris IV-Sorbonne, Paris, 2003.

[CUN 99] – CUNNINGHAM H., Information Extraction - a User Guide, Research Memo, 2nde édition, Université de Sheffield, UK, 1999, http://www.dcs.shef.ac.uk/~hamish/IE/.

[CUN 99] – CUNNINGHAM H., MAYNARD D. & TABLAN V., JAPE: a Java Annotation Patterns Engine (Second Edition), Technical report CS--00--10, Université de Sheffield, UK, 2000.

[CUN 02a] – CUNNINGHAM H., GATE - a General Architecture for Text Engineering, in Computers and the Humanities, Volume 36, 2002, pp.223–254.

[CUN 02b] – CUNNINGHAM H., MAYNARD D., BONTCHEVA K. ET AL., GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, ACM Press, Philadelphie, Pennsylvanie, 2002, pp. 168-175.

[DES 91] – DESCLES J.-P., JOUIS C., OH-JEONG H.-G. & REPPERT D., Exploration Contextuelle et sémantique : un système expert qui trouve les valeurs sémantiques des temps de l'Indicatif dans un texte, in Knowledge modeling and expertise transfer, Amsterdam, 1991, pp. 371-400.

[DES 93] – DESCLES J.-P. & JOUIS C., L'exploration contextuelle : une méthode linguistique et informatique pour l'analyse informatique de textes, in *Proceedings of the International Language Natural (ILN'93)*, Nantes, 1993.

[DES 97] – DESCLÉS J.-P., CARTIER E., JACKIEWICZ A. & MINEL J.-L., Textual Processing and Contextual Exploration Method, in *Proceedings of CONTEXT'97*, Rio de Janeiro, Brésil, 1997, pp. 189-197.

[DILL 03a] – DILL S., EIRON N., GIBSON D., GRUHL D., GUHA R., JHINGRAN A., KANUNGO T., RAJAGOPALAN S., TOMKINS A., TOMLIN J. A. & ZIEN J. Y., SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation, in *Proceedings of the 12th International World Wide Web Conference (WWW'03)*, ACM Press, Budapest, Hongrie, 2003, pp. 178-186.

[DILL 03b] – DILL S., EIRON N., GIBSON D., GRUHL D., GUHA R., JHINGRAN A., KANUNGO T., MCCURLEY K. S., RAJAGOPALAN S. & TOMKINS A., A Case for Automated Large-Scale Semantic Annotation, in Journal of Web Semantics, Science, Services and Agents on the World Wide Web, 1(1), Elsevier, 2003, pp. 115-132.

[DIN 03a] – DINGLI A., Next Generation Annotation Interfaces for Adaptive Information Extraction, in *Proceedings of the 6th Annual Computer Linguists UK Colloquium (CLUK'03)*, Edinburgh, Royaume-Uni, 2003.

[DIN 03b] – DINGLI A., CIRAVEGNA F. & WILKS Y., Automatic Semantic Annotation using Unsupervised Information Extraction and Integration, in *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (KCAP'03)*, Sanibel, Floride, 2003.

[DOA 04] – DOAN A.-H., MADHAVAN J., DOMINGOS P. & HALEVY A., Ontology matching: a machine learning approach, in Handbook of ontologies, STAAB S. & STUDER R. (Eds.), International handbooks on information systems, Springer Verlag, Berlin, 2004, pp. 385-404.

[DOA 05] - DOAN A.-H. & HALEVY A., Semantic integration research in database community: a brief survey, in Artificial Intelligence Magazine, Special issue on Semantic Integration, 26(1), 2005, pp. 83-94.

[DOM 98] – DOMINGUE J., Tadzebao and WebOnto : Discussing, Browsing, and Editing Ontologies on the Web, in *Proceedings of the 11th Knowledge Acquisition Workshop (KAW'98),* Banff, Canada, 1998.

[DOM 03] – DOMINGUE J., DZBOR M. & MOTTA E., Semantic Layering with Magpie, in Handbook on Ontologies, STAAB S. & STUDER R. (Eds.), Frontiers in Artificial Intelligence and Applications, Volume 96, IOS Press, Springer-Verlag, 2003, pp. 533-554.

[DUT 03] – DUTOIT D., PICAND Y., DE TORCY P. & ROGER G., Natural Language Processing and Multimedia Browsing, Concrete and Potential Contributions, in *Proceedings of the European Symposium on Ambient Intelligence*, Eindhoven, Pays-Bas, 2003.

[DZB 03] – DZBOR M., DOMINGUE J. & MOTTA E., Magpie – towards a semantic web browser, in *Proceedings of the 2nd International Semantic Web Conference (ISWC'03)*, International Handbooks on Information Systems, Springer-Verlag, 2003, pp. 690-705.

[EHR 04] - EHRIG M. & STAAB S., QOM - quick ontology mapping, in *Proceedings of the 3rd International Semantic Web Conference (ISWC'04)*, Hiroshima, Japan, 2004.

[ENJ 05a] – ENJALBERT P., Sémantique et TALN : première approche, in Sémantique et traitement automatique du langage naturel, ENJALBERT P. (Ed.), Traité IC2, Cognition et Traitement de l'Information, Hermès, Paris, 2005, pp. 27-52.

[ENJ 05b] – ENJALBERT P., L'extraction d'information, in Sémantique et traitement automatique du langage naturel, ENJALBERT P. (Ed.), Traité IC2, Cognition et Traitement de l'Information, Hermès, Paris, 2005, pp. 309-334.

[ENJ 05c] – ENJALBERT P. & VICTORRI B., Les paliers de la sémantique, in Sémantique et traitement automatique du langage naturel, ENJALBERT P. (Ed.), Traité IC2, Cognition et Traitement de l'Information, Hermès, Paris, 2005, pp. 53-96.

[EUZ 04] – EUZENAT J., State of the art on ontology alignement, Délivrable D2.2.3, Knowledge Web Project, 2004, 79 p.

[EUZ 05] – EUZENAT J., L'annotation formelle de documents en 8 questions, in Ingénierie des connaissances, TEULIER R., CHARLET J & TCHOUNIKINE P., L'Harmattan, Paris, 2005, pp. 251-271.

[EUZ 06] – EUZENAT J., An API for Ontology Alignment, 2006, http://alignapi.gforge.inria.fr/

[FEL 98] – FELLBAUM C., WordNet an Electronic Lexical Database for English, MIT Press, Cambridge, 1998, 423 p.

[FER 97] – FERNANDEZ M., GOMEZ-PEREZ A. & JURISTO N., METHONTOLOGY : from ontological art towards ontological engineering, in *Proceedings of the Spring Symposium Series on Ontological Engineering (AAAI'97)*, AAAI Press, Stanford, USA, 1997.

[FIL 68] – FILLMORE C., The case for the case, in Universals in linguistic theory, Bach E. & Harms R. (Eds.), Holt Rinehart & Winston, Chicago, 1968, pp. 1-90.

[FRE 98] – FREITAG D., Machine Learning for Information Extraction in Informal Domains. Thèse de doctorat, Université Carnegie Mellon, 1998.

[FUC 93] – FUCHS C., Linguistique et Traitements automatiques des langues, Hachette, Paris, 1993, 304 p.

[GAN 99] – GANGEMI A., PISANELLI D. M. & STEVE G., An Overview of the ONIONS project: Applying Ontologies to the Integration of Medical Terminologies, in Data and Knowledge Engineering, 31(2), 1999.

[GAR 03] – GARSHOLD L. M., Living with topic maps and RDF, in *XML Europe 2003*, London, UK, 2003.

[GRI 96] – GRISHMAN R. & SUNDHEIM B., Design of the MUC-6 Evaluation, in *Proceedings of the Annual Meeting of the Association for Computationnal Linguistic (ACL)*, Vienna, Virginie, 1996, pp. 413-422.

[GRI 01a] – GRIVEL L., GUILLEMIN-LANNE S., LAUTIER C. & MARI A., La construction de composants de connaissance pour l'extraction et le filtrage de l'information sur les réseaux, in *Actes du 3ème congrès du Chapitre français de l'International Society for Knowledge Organization (ISKO)*, Paris, France, 2001.

[GRI 01b] – GRIVEL L., GUILLEMIN-LANNE S., COUPET P. & HUOT C., Analyse en ligne de l'information : une approche permettant l'extraction d'informations stratégiques basée sur la construction de composants de connaissance, in *Actes de la Conférence Veille Stratégique Scientifique et Technique (VSST'01)*, Barcelona, Spain, 2001.

[GRU 91] – GRUBER T. R., The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases, in *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, Cambridge, USA, Morgan Kaufmann, 1991, pp. 601-602.

[GRU 93] – GRUBER T., A translation approach to portable ontology specifications, in Knowledge Acquisition Journal, 5(2), Academic Press, 1993, pp. 199-220.

[GRU 95] – GRUNINGER M. & FOX M. S., Methodology for the design and evaluation of ontologies, in *Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing (IJCAI'95*), Montréal, 1995.

[GUA 92] – GUARINO N., Concepts, Attributes and Arbitrary Relations : Some linguistic and ontological criteria for structuring knowledge bases, in Data and Knowledge Engineering, 8(3), 1992.

[GUA 00] – GUARINO N. & WELTY C., A Formal Ontology of Properties, in *Proceedings of European Knowledge Acquisition Conference (EKAW'2000)*, Springer-Verlag LNCS 1937, 2000, pp. 97-112.

[HAB 06] – HABERT B., TAL sur corpus: histoire, acquis, défis, in Compréhension des langues et interaction, SABAH G. (Ed.), Traité IC2, série Cognition et traitement de l'information, Hermès, 2006.

[HAB 05] – HABERT B., Instruments et ressources électroniques pour le français, Collection "L'essentiel Français", Ophrys, Paris, 2005, 169 p.

[HAN 01] – HANDSCHUH S., STAAB S. & MAEDCHE A., CREAM - Creating relational metadata with a component-based, ontology-driven annotation framework, in *Proceedings of the Knowledge Capture Conference (KCAP'01)*, Banff, Canada, 2001, pp. 76-83.

[HAN 02] – HANDSCHUH S., STAAB S. & CIRAVEGNA F., S-CREAM - Semi-automatic CREAtion of Metadata, in *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW'02)*, LNCS 2473, Springer-Verlag, Madrid, Espagne, 2002.

[HAN 03a] – HANDSCHUCH S., STAAB S. & STRUDER R., Leveraging metadata creation for the Semantic Web with CREAM, in *Proceedings of the Annual German Conference on Artificial Intelligence (KI'03)*, 2003.

[HAN 03b] – HANDSCHUH S. & STAAB S., Annotating of the Shallow and the Deep Web, in Annotation for the Semantic Web, HANDSCHUH S. & STAAB S. (Eds.), Frontiers in Artificial Intelligence and Applications, Volume 96, IOS Press, Springer-Verlag, 2003, pp. 25-45.

[HAN 05] – HANDSCHUH S., Creating Ontology-based Metadata by Annotation for the Semantic Web, Thèse de doctorat, Université de Karlsruhe, 2005, 225 p.

[HAR 97] – HARPRING, P., Proper Words in Proper Places: The Thesaurus of Geographic Names, MDA Info, 2(3), 1997.

[HEF 01] – HEFLIN J. & HENDLER J. A., A Portrait of the Semantic Web in Action, in IEEE Intelligent Systems, 16(2), IEEE, 2001, pp. 54-59.

[HER 05] – HERNANDEZ N., Ontologies de domaine pour la modélisation du contexte en Recherche d'information, Thèse de doctorat, Université Paul Sabaltier de Toulouse, 2005, 248 p.

[HOB 97] – HOBBS J. R., APPELT D., BEAR J., ISRAEL D., KAMEYAMA M., STICKEL M. & TYSON M., FASTUS: A Cascaded Finite-State Transducer for Extraction Information from Natural-Language Text, in Extracting Information for Natural-Language Text, ROCHE E. & SCHABES Y. (Eds.), Finite-state Language Processing, Language, Speech and Communication, MIT Press, Cambridge, 1997, pp. 381-406.

[HOV 98] – HOVY E., Combining and standardizing largescale, practical ontologies for machine translation and other uses, In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, Grenade, Espagne, 1998, pp. 535-542.

[IBM 06] – IBM, Unstructured Information Management Architecture (UIMA), SDK User's Guide and Reference, 2006, 364 p., disponible à l'adresse URL suivante:

http://dl.alphaworks.ibm.com/technologies/uima/UIMA_SDK_Users_Guide_Reference.pdf

[KAH 01] – KAHAN J., KOIVUNEN M.R., PRUD'HOMMEAUX E. & SWICK R., Annotea: An Open RDF Infrastructure for Shared Web Annotations, in *Proceedings of the 10th International World Wide Web Conference (WWW'01)*, ACM Press, Hong-Kong, 2001, pp. 623-632.

[KAL 03a] – KALFOGLOU Y. & SCHORLEMMER M., Ontology mapping: the state of the art, in The Knowledge Engineering Review, 18(1), 2003, pp. 1-31.

[KAL 03b] – KALYANPUR A., HENDLER J., PARSIA B. & GOLBECK J., SMORE – Semantic Markup, Ontology, and RDF Editor, 2003, non publié mais disponible à l'adresse URL suivante : http://www.mindswap.org/papers/SMORE.pdf

[KAL 05] – KALYANPUR A., PARSIA B., SIRIN E., CUENCA-GRAU B. & HENDLER J., "Swoop: A 'Web' Ontology Editing Browser", In Journal of Web Semantics, 4(2), 2005.

[KAS 02] – KASSEL G., OntoSpec : une méthode de spécification semi-informelle d'ontologies, in *Actes des journées francophones d'Ingénierie des Connaissances (IC'2002)*, Rouen, 2002, pp. 75-87.

[KAT 02] – KATZ B., LIN J. & QUAN D., Natural Language Annotations for the Semantic Web, in *Proceedings of the International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02)*, Irvine, Californie, 2002, pp. 1317-1331.

[KIM 02] – KIM S., ALANI H., HALL W., LEWIS P., MILLARD D., SHADBOLT N. & WEAL M., Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web, in *Proceedings of the Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM'02)*, Lyon, France, 2002, pp. 1-6.

[KIR 05] – KIRYAKOV A., POPOV B., TERZIEV I., MANOV D., KIRILOV A. & GORANOV M., Semantic Annotation, Indexing, and Retrieval, in Journal on Web Semantics, Science, Services and Agents on the World Wide Web, 2(1), Elsevier, 2005, pp. 49-79.

[KOG 01] – KOGUT P. & HOLMES W., AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages, in *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (KCAP'01)*, Victoria, Canada, 2001.

[KUS 97] – KUSHMERICK N., WELD D. & DOORENBOS B., Wrapper induction for information extraction, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.

[LAN 05] – LANFRANCHI V., CIRAVEGNA F. & PETRELLI D., Semantic Web-Based Document: Editing and Browsing in AktiveDoc, in *Proceedings of the 2nd European Semantic Web Conference (ESWC'05)*, LNCS 3532, Springer-Verlag, Heraklion, Crête, Grèce, 2005, pp. 623-632.

[LAU 02] – LAUBLET P., REYNAUD C. & CHARLET J., Sur Quelques Aspects du Web Sémantique, in *Assises du GDR I3*, Cépadues, Nancy, 2002, pp. 59-78.

[LAU 07] – LAUBLET P., Web Sémantique et Ontologies, in Nouvelles technologies cognitives et concepts des sciences humaines et sociales, Volume 1, Humanités Numériques, Hermès, Paris, à paraître en 2007.

[LAV 04] – LAVELLI A., CALIFF M. E., CIRAVEGNA F., FREITAG D., GIULIANO C., KUSHMERICK N. & ROMANO L., IE evaluation: Criticisms and recommendations, In *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM 2004)*, San Jose, Californie, 2004.

[LAS 01] – LASSILA O. & MCGUINNESS D., The Role of Frame-Based Representation on the Semantic Web, Technical Report KSL-01-02, Knowledge Systems Laboratory, Stanford University, Californie, 2001.

[LEP 00] – LE PRIOL F., Extraction et capitalisation automatique de connaissances à partir de documents textuels. SEEK-JAVA : identification et interprétation de relations entre concepts, Thèse de doctorat, Université Paris IV-Sorbonne, 2000.

[LI 00] – LI W-S. & CLIFTON C., Semint: a tool for identifying attribute correspondences in heterogeneous databases using neural networks, in Data Knowledge Engineering, 33(1), 2000, pp. 49-84.

[LIG 94] – LIGOZAT G., Représentation des connaissances et linguistique, Armand Colin, 1994, 144 p.

[LUK 00] – LUKE S. & HEFLIN J.D., SHOE 1.01. Proposed Specification, Technical Report, Parallel Understanding Systems Group, Department of Computer Science, University of Maryland, 2000. http://www.cs.umd.edu/projects/plus/SHOE/spec.html

[MAG 05] – MAGNINI B., NEGRI M., PIANTA E., ROMANO L., SPERANZA M., SERAFINI L., GIRARDI C., BARTALESI V. & SPRUGNOLI R., From Text to Knowledge for the Semantic Web: the ONTOTEXT Project, in *Proceedings of the Semantic Web Applications and Perspectives Conference (SWAP'05)*, Trento, Italie, 2005.

[MAN 06] – MANOV D. & POPOV B., Massive Automatic Annotation, Délivrable D2.6.1, SEKT Project, 2006, 24 p.

[MAY 05a] – MAYNARD D., YANKOVA M., KOURAKIS A. & KOKOSSIS A., LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies, in *Proceedings of the End User Aspects of the Semantic Web Workshop (ESWC'05)*, Heraklion, Crête, Grèce, 2005.

[MAY 05b] – MAYNARD D., Benchmarking ontology-based annotation tools for the Semantic Web, in *Proceedings of the Workshop "Text Mining, e-Research and Grid-enabled Language Technology" in the UK e-Science Programme All Hands Meeting (AHM2005)*, Nottingham, UK, 2005.

[MEL 02] – MELNIK S., GARCIA-MOLINA H. & RAHM E., Similarity flooding: a versatile graph matching algorithm, In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, San Jose, Californie, 2002.

[MIN 02] – MINEL J.-L., Filtrage sémantique. Du résumé à la fouille de textes, Hermès, Paris, 2003, 202 p.

[MOT 99] – MOTTA E., Reusable Components for Knowledge Modelling: Principles and Case Studies in Parametric Design, IOS Press, Amsterdam, Pays-Bas, 1999.

[NAZ 05] – NAZARENKO A., Sur quelle sémantique reposent les méthodes automatiques d'accès au contenu textuel ?, in Sémantique et corpus, Condamines A. (Ed.), Hermès, Traité IC2, Cognition et Traitement de l'Information, Hermès, 2005, pp. 211-244.

[NAZ 06] – NAZARENKO A., Le point sur le TAL, in Compréhension des langues et interaction, Sabah G. (Ed.), Traité IC2, Cognition et Traitement de l'Information, Hermès Science - Lavoisier, Paris, 2006, pp. 31-70.

[NOY 99] – NOY N. F. & MUSEN M. A., An algorithm for merging and aligning ontologies: automation and tool support, in *Proceedings of the Workshop on Ontology Management (AAAI'99*), AAAI Press, 1999.

[NOY 00] – NOY N. F., FERGERSON R. W. & MUSEN M. A., The knowledge model of Protégé2000 : combining interoperability and flexibility, in *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW'00)*, Juan-Les-Pins, 2000.

[NOY 01] – NOY N. F. & MCGUINNES, D.L., Ontology Development 101: A Guide to Creating Your First Ontology, Rapport technique SMI-2001-0880, SMI, 2001.

[NOY 03] – NOY N. F. & MUSEN M. A., The PROMPT suite : interactive tools for ontology merging and mapping, in International Journal of Human-Computer-Studies, 59(6), 2003.

[NOY 04] – NOY N. F., Semantic Integration: A Survey of Ontology-Based Approaches, in SIGMOD Record, Special Issue on Semantic Integration, 33(4), 2004, pp. 65-70.

[PIE 00] – PIERREL J.-M., Ingénierie des langues, Traité IC2, Informatique et Systèmes d'Information, Hermès, Paris, 2000, 354 p.

[POP 03] – POPOV B., KIRYAKOV A., MANOV D., KIRILOV A., OGNYANOFF D. & GORANOV M., Towards Semantic Web Information Extraction, in *Proceedings of the Human Language Technologies Workshop (ISWC'03)*, Sanibel, Floride, 2003, pp. 1-22.

[PRI 04] – PRIE Y. & GARLATTI S., Méta-données et annotations dans le Web sémantique, in Le Web sémantique, CHARLET J., LAUBLET P. & REYNAUD C. (Ed.), Hors série de la Revue Information - Interaction - Intelligence (I3), 4(1), Cépaduès, Toulouse, 2004, pp. 45-68.

[RIN 03] – RINALDI F., DOWDALL J., HESS M., ELLMAN J., ZARRI G.-P., PERSIDIS A.,, BERNARD L., KARANIKAS H., Multilayer annotations in Parmenides, in *Proceedings of the Knowledge Markup and Semantic Annotation Workshop*, Sanibel, Florida, USA, 2003, pp.33-40.

[ROD 06] – RODRIGUES T., ROSA P. & CARDOSO J., Mapping XML to Exiting OWL ontologies, in *Proceedings of the International Conference WWW/Internet (WWW'06)*, Murcia, Espagne, 2006.

[SAB 90] – SABAH G., L'intelligence artificielle et le langage, in Représentation des connaissances, Volume 1, Hermès, Paris, 1990, 357 p.

[SAZ 05] – SAZEDJ P. & PINTO S., Time to evaluate: Targeting Annotation Tools, in *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation (semAnnot'05)*, volume 185, 2005.

[SCH 97] – SCHULTZ D. J., IEEE Standard for Developing Software Life Cycle Processes, Technical Report 1074-1997, IEEE Computer Society, New-York, 1997.

[SHV 05] – SHVAIKO P. & EUZENAT J., A Survey of Schema-based Matching Approaches, in Journal of Semantics, Volume 3730, Springer-Verlag, Berlin, 2005, pp. 146-171.

[SOW 00] – SOWA J., Knowledge Representation: Logical, Philosophical and computational foundations, Brooks Cole Publishing Co., Pacific Grove, 2000, 594 p.

[STA 01a] – STAAB S., MAEDCHE A. & HANDSCHUH S., An Annotation Framework for the Semantic Web, in *Proceedings of the 1st International Workshop on MultiMedia Annotation*, Tokyo, Japon, 2001.

[STA 01b] – STAAB S., MAEDCHE A. & HANDSCHUH S., Creating Metadata for the Semantic Web: An Annotation Framework and the Human Factor, Technical Report, Institut AIFB, Université de Karlsruhe, Allemagne, 2001, 25 p.

[STE 03] – STEVENSON M. & CIRAVEGNA F., Information Extraction as a Semantic Web Technology: Requirements and Promises, in *Proceedings of the Adaptive Text Extraction and Mining Workshop at the 14th European Conference on Machine Learning (ECML'03)*, Cavtat-Dubrovnik, Croatia, 2003.

[STO 02] – STOJANOVIC L., STOJANOVIC N. & HANDSCHUH S., Evolution of Metadata in Ontology-based Knowledge Management Systems, in *Proceedings of Experience Management 2002*, Berlin, 2002.

[STU 98] – STUDER R., BENJAMINS V.R. & FENSEL D., Knowledge engineering: principles and methods, in IEEE Transactions on Data and Knowledge Engineering, 25(1&2), 1998, pp.161-197.

[SUR 02] – SURE Y., ERDMANN M., ANGELE J., STAAB S., STUDER R. & WENKE D., OntoEdit : Collaborative Ontology Development for the Semantic Web, in *Proceedings of the International Semantic Web Conference (ISWC 2002)*, Sardinia, Italia, 2002.

[SUR 03] – SURE Y., AKKERMANS H., BROEKSTRA J., DAVIES J., DING Y., DUKE A., ENGELS R., FENSEL D., HORROCKS I., IOSIF V., KAMPMAN A., KIRYAKOV A., KLEIN M., LAU T., OGNYANOV D., REIMER U., SIMOV K., STUDER R., VAN DER MEER J. & VAN HARMELEN F., On-To-Knowledge: Semantic Web enabled Knowledge Management, in Web Intelligence, Springer-Verlag, 2003, pp. 277-300.

[SZU 02] – SZULMAN S., BIEBOW B., AUSSENAC-GILLES N., Structuration de Terminologies à l'aide d'outils d'analyse de textes avec TERMINAE, in Traitement Automatique de la Langue (TAL), Numéro spécial « Structuration de Terminologie », 43(1), Hermès, 2002, pp. 103-128.

[TAL 03] – TALLIS M., Semantic Word Processing for Content Authors, in *Proceedings of the Knowledge Markup and Semantic Annotation Workshop (SEMANNOT'03)*, Sanibel, Floride, 2003.

[TOL 05] – TOLKSDORF R., NIXON L. J. B., LIEBSCH F., MINH NGUYEN D., PASLARU BONTAS E. ET NIXON L. J. B., Enabling real world Semantic Web applications through a coordination Middleware, in *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, Heraklion, Greece, 2005.

[TRO 02] – TRONCY R. & ISAAC A., DOE : une mise en œuvre d'une méthode de structuration différentielle pour les ontologies, in *Actes des 13èmes Journées Francophones d'Ingénierie des Connaissances (IC'02)*, Rouen, 2002, pp. 63-74.

[USC 95] – USCHOLD M. ET KING M., Towards a Methodology for Building Ontologies, in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI-95)*, Montréal, Canada, 1995.

[URE 06] – UREN V., CIMIANO P., HANDSCHUH S., VARGAS-VERA M., MOTTA E. & CIRAVEGNA F., Semantic annotation for knowledge management: requirements and a survey of the state of the art, in Journal of Web Semantics, Science, Services and Agents on the World Wide Web, 4(1), Elsevier, 2006, pp.14-26.

[VAL 03] – VALARAKOS A., SIGLETOS G., KARKALETSIS V.& PALIOURAS G., A Methodology for Semantically Annotating a Corpus Using a Domain Ontology and Machine Learning, in *Proceedings of*

the Recent Advances in Natural Language Processing International Conference (RANLP'03), Bulgarie, 2003, pp. 495-499.

[VAL 04] – VALARAKOS A., PALIOURAS G., KARKALETSIS V. & VOUROS G., Enhancing Ontological Knowledge through Ontology Population and Enrichment, in *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW'04)*, LNAI 3257, Springer-Verlag, Whittlebury Hall, Royaume-Uni, 2004, pp. 144-156.

[VAN 83] – VAN DIJK T. A. & KINTSCH W., Strategies of Discourse Comprehension, Academic Press, New York, 1983, 418 p.

[VAN 85] – VAN DIJK T. A., Handbook of Discourse Analysis, Volume 1, Academic Press, New York, 1985, 302 p.

[VAR 01] – VARGAS-VERA M., DOMINGUE J., MOTTA E., BUCKINGHAM SHUM S. & LANZONI M., Knowledge Extraction by using an Ontology-based Annotation Tool, in *Proceedings of the Workshop Knowledge Markup & Semantic Annotation (KCAP'01)*, Victoria, Canada, 2001.

[VAR 02a] – VARGAS-VERA M., MOTTA E., DOMINGUE J., LANZONI M., STUTT A. & CIRAVEGNA F., MnM: Ontology Driven Tool for Semantic Markup, in *Proceedings of the Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM'02)*, Lyon, France, 2002.

[VAR 02b] – VARGAS-VERA M., MOTTA E., DOMINGUE J., LANZONI M., STUTT A. & CIRAVEGNA F., MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup, in *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW'02)*, LNCS 2473, Springer-Verlag, Madrid, Espagne, 2002, pp. 379-391.

[VAR 04] – VARGAS-VERA M. & CELJUSKA D., Event Recognition on News Stories and Semi-Automatic Population of an Ontology, in *Proceedings of the IEEE/ACM International Joint Conference on Intelligent Agent and Web Intelligence (WI'04)*, IEEE Computer Society Press, Beijing, Chine, 2004, pp. 615-618.

[VAT 03] – VATANT B., Cooking for the Semantic Web: OWL and Topic Map Pudding, Rapport Technique, Mondeca, Paris, 2003. http://www.mondeca.com/owl/owltm.htm

[VAT 04] – VATANT B., Ontology-driven Topic Maps, In *XML Europe 2004*, Amsterdam, Netherlands, 2004.

[VIC 05] – VICTORRI B., Le calcul de la référence, in Sémantique et TALN, Enjalbert P. (Ed.), Traité IC2, Cognition et Traitement de l'Information, Hermès, Paris, 2005, p.133-172

[VOL 04] – VOLZ R., HANDSCHUH S., STAAB S., STOJANOVIC L. & STOJANOVIC N., Unveiling the hidden bride: Deep Annotation for Mapping and Migrating Legacy Data to the Semantic Web, in Journal of Web Semantics, Science, Services and Agents on the World Wide Web, 1(2), Elsevier, 2004, pp. 187-206.

[VOO 98] – VOORHEES E. M., Using WordNet for Text Retrieval, in WordNet: An Electronic Lexical Database, Fellbaum (Ed.), MIT Press, 1998, pp. 285-303.

[WEH 97] – WEHRLI E., L'analyse syntaxique des langues naturelles : problèmes et méthodes, Informatique, Masson, Paris, 1997, pp. 1-54.

[WER 05] – WERLI S., Agent de détection de faits et d'événements majeurs dans les textes de la toile. Une application à la veille : le système ALSEM, Thèse de doctorat, Université Paris IV, 2005.