



HAL
open science

DOE : une mise en oeuvre d'une méthode de structuration différentielle pour les ontologies

Raphaël Troncy, Antoine Isaac

► **To cite this version:**

Raphaël Troncy, Antoine Isaac. DOE : une mise en oeuvre d'une méthode de structuration différentielle pour les ontologies. Actes 13e journées francophones sur Ingénierie des Connaissances (IC), May 2002, Rouen, France. pp.63-74. hal-00922312

HAL Id: hal-00922312

<https://inria.hal.science/hal-00922312>

Submitted on 25 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DOE : une mise en œuvre d'une méthode de structuration différentielle pour les ontologies

Raphaël Troncy^{*†}, Antoine Isaac^{*‡}

* Institut National de l'Audiovisuel, Direction de la Recherche
4, Av. de l'Europe - 94366 Bry-sur-Marne
{rtroncy, aisaac}@ina.fr et <http://www.ina.fr/>

† INRIA Rhône-Alpes, Action EXMO
655, Av. de l'Europe - 38334 Saint-Ismier cedex
<http://www.inrialpes.fr/exmo>

‡ CAMS, équipe LaLICC, Université de Paris-Sorbonne
96 Boulevard Raspail, 75006 Paris
<http://www.lalicc.paris4.sorbonne.fr>

Thèmes : *Méthodes de résolution de problèmes, ontologies et langages de spécification de connaissances Hypertextes, hypermédia et Ingénierie des connaissances documentaires*

Résumé

L'INA s'intéresse aux ontologies car celles-ci peuvent être utilisées comme des sources de descripteurs permettant d'indexer des documents audiovisuels. Les méthodologies et les outils pour construire de tels objets existent, mais peu proposent une aide concrète pour organiser les concepts entre eux et expliciter le sens attribué aux termes mobilisés. Cet article propose d'utiliser une méthodologie basée sur la sémantique différentielle pour normaliser le sens des termes manipulés dans l'ontologie. Il présente un éditeur d'ontologies, *DOE*, dont l'objectif est de prendre en charge cette méthodologie, et avec lequel plusieurs ontologies sont actuellement en cours de construction.

Mots clef : Construction d'ontologies, éditeur d'ontologies, *DOE*, méthodologie, sémantique différentielle.

1 Introduction

1.1 Contexte

L'accès à l'information passe aujourd'hui de plus en plus par le biais de systèmes techniques permettant de traiter des contenus numérisés. L'Institut National de l'Audiovisuel (INA), qui a pour mission la gestion de larges collections audiovisuelles¹, n'échappe pas à la règle. En effet, l'Institut capte maintenant directement les programmes des chaînes en numérique et a entrepris un vaste plan de numérisation de ses fonds anciens. Au sein de l'Institut, la Direction de la Recherche et Expérimentation doit accompagner les mutations du système technique audiovisuel provoquées par

l'irruption de ces technologies numériques. En particulier, elle doit préparer la mise en place d'un «*système numérique intégral*», où les outils de production et d'exploitation reposent sur ces technologies. Ceci implique une réflexion sur le procédé fondamental de description et de reformulation qui permet ensuite d'atteindre et d'utiliser l'information contenue dans un document : l'*indexation*.

Actuellement, plusieurs équipes de documentalistes visionnent et décrivent les émissions diffusées. Cette description s'effectue à travers la rédaction de *notices documentaires*², résultat d'une interprétation textuelle du programme. La recherche documentaire s'effectue ensuite par l'intermédiaire de ces notices. La numérisation permet de faire évoluer la pratique de l'indexation et l'utilisation de son résultat. En effet, les descriptions peuvent à présent se comporter comme des véritables *méta-données*, compléments d'information aux documents (ou segments de documents) audiovisuels dont elles partagent le support. Plus précises – on peut très naturellement connecter une méta-donnée à un seul segment du document – et plus riches – on peut attribuer à une méta-donnée une signification computationnelle qui permettra par exemple d'effectuer des raisonnements automatiques à partir des documents [7] –, les descriptions seront la clé de voûte d'importantes bases de données multimédias qui seront autant de ressources permettant de retrouver ou d'éditer de nouveaux documents. Les fonctions de l'indexation seront alors largement étendues car la composition de ces nouveaux documents ne sera possible que si les segments les composant sont décrits par leur contenu.

La description du contenu des documents s'effectue par un ensemble de descripteurs dont la sémantique doit être explicite. En spécifiant la nature, l'organisation et le compor-

1. L'INA a une mission patrimoniale qui s'articule autour :

- du dépôt légal des chaînes de télévision hertziennes et des radios publiques depuis 1995 (loi de 1992),
- et de l'archivage, à des fins commerciales, des programmes radio-télévisés des chaînes publiques.

2. Celles-ci sont plus ou moins complètes, en fonction du genre de l'émission décrite.

tement des objets relatifs à un domaine particulier, les ontologies permettent le partage d'une compréhension de ce domaine, et donc peuvent garantir la validité du recours à un ensemble de descripteurs reflétant cette compréhension. Récemment, de nombreuses ontologies ont fait leur apparition, qu'elles aient été élaborées dans un but de recherche pure ou bien pour être appliquées dans des projets industriels. Néanmoins, construire une ontologie demeure une activité difficile, et il n'y a toujours pas en la matière de guide méthodologique complet et universellement reconnu [10]. De nombreux outils de création d'ontologies ont aussi fait leur apparition. Parmi ceux-ci, quelques-uns essaient de guider leur utilisateur en suivant une méthodologie de conception plus ou moins complète. Cependant, force est de constater qu'aucun de ces outils n'a réussi à s'imposer. La réflexion sur la construction des ontologies reste donc ouverte.

1.2 Positionnement

Comme nous l'avons dit plus haut, les ontologies fournissent le vocabulaire propre à un domaine et fixent – avec un degré de formalisation variable – le sens des concepts et des relations entre ceux-ci. Ces concepts (et ces relations) sont organisées *via* la relation de subsumption pour former une taxinomie. L'usage montre que cette taxinomie – elle n'est jamais absente, et quelques fois la définition des ontologies se résume même à sa seule construction – constitue la porte d'accès à l'ontologie. Il est donc extrêmement important de la structurer correctement : la compréhension du domaine visé doit réellement émerger des concepts qui y sont présentés. C'est à ce problème précis que cet article tente d'apporter un élément de solution.

C'est cette préoccupation qui a tout d'abord orienté notre revue des méthodes proposées par la communauté (section 2). Nous verrons qu'en l'absence de réponse réellement satisfaisante à notre problème, nous nous sommes tournés vers la méthodologie proposée par Bruno Bachimont, qui introduit un processus de *normalisation sémantique* des concepts (et des relations). Cette méthodologie, nous montrerons comment nous avons cherché à la mettre en œuvre dans un outil d'aide à la structuration d'ontologies, **DOE**³, qui se veut complémentaire des environnements de conception d'ontologies déjà existants (section 3). Nous détaillerons donc les différentes étapes de la méthodologie (section 4, 5 et 6) en mettant l'accent sur le retour d'expérience offert par la réalisation d'une ontologie concernant le cyclisme, qui nous a guidé en ce qui concerne les choix d'implémentation et la résolution de quelques problèmes méthodologiques qui ont été rencontrés à cette occasion. Nous conclurons finalement en évoquant les problèmes et les perspectives dégagés lors de cette mise en œuvre (section 7).

3. L'outil est disponible gratuitement à <http://opales.ina.fr/public/>.

4. <http://babage.dia.fi.upm.es/ontoweb/wp1/OntoRoadMap/index.html>

5. <http://www.ontoknowledge.org>.

2 Quels guides méthodologiques pour construire une taxinomie ?

2.1 Un problème souvent écarté

On doit préciser ici que parmi les – nombreux – travaux en la matière, nous nous sommes surtout intéressés à ceux qui cherchaient à prendre en charge la construction *ex nihilo* des ontologies. Effectivement, si les travaux relatifs aux ontologies affichent pour ambition de faciliter la réutilisation d'un grand nombre de ces artefacts, nous ne sommes pas encore sûrs de la viabilité d'une telle démarche, surtout si les ontologies sont construites dans un flou méthodologique assez important.

Parmi les nombreuses productions en matière de conception d'ontologies (on peut se reporter pour un répertoire relativement complet au travail du OntoWeb Technical Road-Map⁴), on peut cependant regretter que peu présentent des guides précis pour la construction des hiérarchies produites. En effet, des travaux comme ceux d'Ushold et Grüniger [15], ceux de l'équipe du LAI de Madrid [9] ou ceux des chercheurs impliqués dans le projet On-To-Knowledge⁵, s'intéressent plus à l'ensemble du processus de conception de l'ontologie, et se concentrent sur des questions comme celles du cycle de vie des ontologies construites, de l'ordonnancement des étapes auquel l'ontologiste devra se soumettre pour valider son travail (définition des questions auxquelles devra répondre le système, collecte de documents pertinents pour l'acquisition des connaissances, évaluation des résultats...). Répondre à ces questions est bien évidemment important si l'on veut voir un jour émerger un véritable « génie ontologique », mais nous pensons que la phase de conceptualisation proprement dite, celle où les concepts de l'ontologie sont dégagés, définis par un certains nombres de propriétés et organisés entre eux, gagnerait à être guidée de manière plus précise qu'elle ne l'est dans ces réflexions. Par exemple, la méthode METHONTOLOGY, alors qu'elle propose un grand nombre de *représentations intermédiaires* [5] afin de mieux conduire la construction des ontologies au niveau des connaissances, n'insiste pas sur la manière de structurer ses *arbres de classification de concepts*, l'une de ces représentations.

2.2 Une réponse possible : donner plus de place à la langue

Parmi les méthodes qui ont été présentées, peu proposent des aides concrètes pour guider les utilisateurs à organiser les concepts entre eux : tout repose sur leur bonne intuition du domaine. Ce constat est somme toute normal puisqu'aucune des méthodologies exposées ne prend réellement en charge l'explicitation des concepts sous sa forme la plus naturelle : le langage.

En effet, si à peu près toutes préconisent l'utilisation de celui-ci pour tenter de préciser le sens des concepts manipulés, que ce soit dans des commentaires présents dans l'ontologie elle-même ou dans des documents produits lors du processus de modélisation, elles sont peu à énoncer des consignes précises pour la rédaction de ces compléments [10]. En fait, malgré les efforts de commentaires, les termes utilisés comme primitives de connaissances peuvent toujours être sujets à des interprétations multiples, ce qui nuit fortement à la compréhension de l'ontologie et à son utilisation.

Le problème réside dans le fait qu'aucune méthodologie ne force l'ontologiste à expliciter clairement le sens qu'il attribue aux concepts : les commentaires restent très informels. Bruno Bachimont propose, dans une méthodologie introduite dans le cadre du projet MENELAS [16], de contraindre l'utilisateur à un *engagement sémantique* [2] en introduisant une *normalisation sémantique* des termes manipulés dans l'ontologie. Cette méthodologie n'avait cependant toujours pas fait l'objet d'une implémentation satisfaisante.

Après avoir formulé ce besoin, il nous a paru judicieux de poursuivre notre parcours de l'état actuel de l'ingénierie des ontologies par un rapide examen des outils logiciels existants. Peut-être certains proposent-ils des pistes pour assurer le support de cette démarche méthodologique...

3 Vers une solution implémentée

3.1 Quels sont les outils disponibles ?

Il existe un certain nombre d'outils permettant de construire des ontologies, et on peut les regrouper en deux catégories. Les premiers conçus (le serveur Ontolingua⁶ [8], par exemple) permettent de spécifier les ontologies au niveau symbolique : une grande partie des définitions des objets se fait directement dans un langage de représentation de connaissances donné (pour Ontolingua, il s'agit de KIF), auquel le créateur et l'utilisateur de l'ontologie doivent se plier. Les seconds tiennent compte de l'importance du niveau des connaissances : ils proposent à leur utilisateur de créer l'ontologie de manière relativement indépendante de tout langage implémenté et prennent ensuite automatiquement en charge l'opérationnalisation de l'ontologie, en la transposant dans divers langages. C'est cette catégorie qui nous a le plus intéressés : il semble en effet naturel de chercher à s'abstraire – dans un premier temps – du niveau symbolique si on veut obtenir une ontologie permettant un réel partage d'une compréhension.

Parmi ces outils de seconde génération, on peut citer **Protégé2000**⁷ [13], un environnement graphique et hors-

ligne de développement d'ontologies développé par le SMI de Stanford. Dans le modèle des connaissances de *Protégé*, les ontologies consistent en une hiérarchie de *classes* qui ont des attributs (*slots*), qui peuvent eux-mêmes avoir certaines propriétés (*facets*). L'édition des listes de ces trois types d'objets se fait par l'intermédiaire de l'interface graphique, sans avoir besoin d'exprimer ce que l'on a à spécifier dans un langage formel : il suffit juste de remplir les différents *formulaires* correspondant à ce que l'on veut spécifier. Ce modèle autorise d'ailleurs une liberté de conception assez importante, puisque le contenu des formulaires à remplir peut être modifié suivant les besoins *via* un système de *méta-classes*, qui constituent des sortes de « patrons » de connaissance. L'interface, très bien conçue, et l'architecture logicielle permettant l'insertion de *plug-ins* pouvant apporter de nouvelles fonctionnalités (par exemple, la possibilité d'importer et d'exporter les ontologies construites dans divers langages opérationnels de représentation) ont participé au succès de *Protégé2000*, qui regroupe une communauté d'utilisateurs assez importante et constitue une référence pour beaucoup d'autres outils.

L'un d'entre eux est **OILed**⁸ [4], développé sous la responsabilité de l'université de Manchester pour éditer des ontologies dans le langage de représentation OIL. Officiellement, il n'a pas d'autre ambition que de construire des exemples montrant les vertus du langage pour lequel il a été créé. Néanmoins, il offre la plus grande partie de ce que l'on peut attendre d'un éditeur d'ontologies. On peut créer des hiérarchies de classes et spécialiser les rôles, et utiliser avec l'interface les types d'axiomes les plus courants (*OILed* utilise les expressions définies dans les logiques de description). Cet éditeur offre également les services d'un raisonneur, *FaCT*, qui permet de tester la satisfaisabilité des définitions de classes et de découvrir des sous-omissions restées implicites dans l'ontologie. Il permet enfin d'exporter les ontologies construites dans d'autres langages de représentation d'ontologies en passe de devenir – ou déjà devenus – des standards : *DAML+OIL*⁹ et *RDFS*¹⁰.

OntoEdit [12], contrairement aux deux précédents, n'est pas disponible gratuitement¹¹. Il présente les fonctionnalités essentielles communes aux autres éditeurs (hiérarchie de concepts, expression de lois, export de l'ontologie dans des langages divers) et a le mérite de s'appuyer sur une réflexion méthodologique significative. La modélisation des axiomes a fait l'attention de soins particuliers pour pouvoir être effectuée – en tout cas pour les types les plus répandus – indépendamment d'un formalisme privilégié, et cela pour faciliter la traduction d'un langage de représentation à un

6. <http://www-ksl-svc.stanford.edu:5915/>

7. <http://protege.stanford.edu/index.shtml>

8. <http://oiled.man.ac.uk/>

9. <http://www.daml.org/language/>

10. <http://www.w3.org/TR/rdf-schema/>

11. Une version de démonstration est disponible sur le site d'Ontoprise, la société qui le développe en collaboration avec l'AIFB de Karlsruhe.

autre. Il propose également une gestion originale des *questionnaires de compétences*. Des questions pour les réponses desquelles l'ontologie doit fournir le matériel conceptuel, on peut extraire les termes appelés à intégrer l'ontologie. Un petit outil fait une comparaison lexicale des termes extraits des différentes questions pour en déduire automatiquement d'éventuelles subsumptions. Le procédé semble cependant loin d'être fiable, puisqu'il repose sur l'hypothèse que le nom d'un concept se retrouve parfois dans le nom de ses spécialisations.

Le dernier outil évoqué ici sera **WebODE**¹² [1]. Cette plate-forme en ligne développée par le LAI de Madrid se place au niveau méthodologique dans la lignée d'ODE, un éditeur qui assurait fidèlement le support de METHONTOLOGY, la méthodologie maison. Elle illustre bien l'évolution des outils de construction d'ontologies, puisque les nombreuses tables de son prédécesseur ont été remplacées par une interface très travaillée, réalisant un pas supplémentaire vers une conception au niveau des connaissances. On peut cependant regretter que cette évolution se soit faite au détriment de l'application des contraintes méthodologiques : les *représentations intermédiaires* utilisées dans le processus de conception sont désormais moins mises en avant, à tel point que le guide de l'utilisateur ne les signale que pour «*assurer la compatibilité conceptuelle avec ODE*». L'accent a plus été mis sur la possibilité d'un travail collaboratif ou sur la mise à disposition d'outils complémentaires, comme un moteur d'inférences.

3.2 DOE, un outil d'aide à la construction d'ontologies

Si tous ces outils peuvent être considérés – en tout cas dans une première approche – comme satisfaisants en matière d'expressivité, d'interface... on peut affirmer qu'il existe toujours un certain vide méthodologique. Le problème que nous évoquions, à savoir la structuration des taxinomies produites, est en particulier très peu pris en charge : on ne peut pas dire que les outils existants guident réellement l'utilisateur. La piste d'un traitement plus complet des informations véhiculées par le langage ne semble pas non plus pouvoir être suivie grâce à ces éditeurs, puisque les commentaires demeurent toujours accessoires.

A ce constat est venue s'ajouter une préoccupation spécifique à un projet de l'INA, OPALES (pour *Outils pour des Portails Audiovisuels Educatifs et Scientifiques*). Ce projet a en effet pour but de mettre à disposition de communautés d'utilisateurs des documents audiovisuels qu'ils pourront indexer en fonction de leurs besoins spécifiques. La considération des enjeux sémantiques de l'indexation et le besoin d'un outil spécifique au projet ont achevé de nous convaincre de mettre en pratique des idées de conception s'appuyant sur l'expression des connaissances en langage

naturel. Nous avons donc implémenté notre propre outil, *DOE* (pour *Differential Ontology Editor*). Cet outil, programmé en Java, n'a pas pour ambition de concurrencer les grands environnements existants, mais plutôt de fournir un début d'implémentation de la méthodologie de structuration retenue. Cette méthodologie propose trois étapes (Figure 1) dont nous allons à présent rappeler les principes et présenter la mise en œuvre dans notre éditeur.

4 Normalisation sémantique

L'étape de normalisation sémantique a pour but d'expliquer le sens des libellés linguistiques pour qu'ils fonctionnent comme des primitives. Ces libellés doivent avoir leurs significations normées pour pouvoir être utilisés. Nous allons voir en quoi l'utilisation d'une sémantique différentielle apporte une réponse à ce problème (section 4.1) avant de montrer comment cette utilisation s'intègre dans l'outil d'édition que nous avons développé à l'INA (section 4.2). Nous concluons en évoquant l'une des difficultés rencontrées lors de l'utilisation de la méthodologie : la modélisation des concepts de haut niveau (section 4.3).

4.1 De la recherche de termes à la création d'une ontologie différentielle

4.1.1 Termes et sémantique différentielle

L'usage de la langue naturelle est l'accès privilégié aux connaissances d'un domaine. En effet, même à l'INA, ce sont avant tout à travers des ressources textuelles que les documentalistes décrivent les programmes radio-télévisés. Le document audiovisuel n'ayant pas, par défaut, d'unités discrètes significatives (l'équivalent des mots pour le texte), et n'offrant pas de signification conventionnelle établie (il n'y a pas par exemple de dictionnaire d'images), les documentalistes ont recours à un ensemble de documents baptisés *péritextes*¹³ accompagnant le programme [11]. Il paraît donc logique de commencer par utiliser des procédés d'extraction terminologique afin de trouver des «candidats-termes» (ces candidats peuvent être le résultat d'outils de type *Lexter* [6]).

Cette piste a été suivie dans le cadre de travaux sur l'extraction de structures conceptuelles à partir de documents textuels relatifs au cyclisme. Pour construire l'ontologie de ce domaine, nous avons eu accès à un corpus de plus de 500 000 mots, soit environ 850 articles. Ce corpus est composé d'articles issus de la presse généraliste (*Le Monde*, *Libération*, *Le Parisien*), de la presse spécialisée (*L'Équipe*) ou encore de dépêches d'agences de presse (*AFP*) qui ont couvert le Tour de France 1999. Il est plutôt homogène, tant sur le plan stylistique que sur les différents thèmes abordés (descriptions géographique, vie de la course, classements, dopage, etc.). A titre d'exemple, l'analyse sur une partie de

12. <http://delicias.dia.fi.upm.es/webODE/>

13. A titre d'illustration, on peut citer les dossiers de production, de presse ou ceux fournis par le diffuseur.

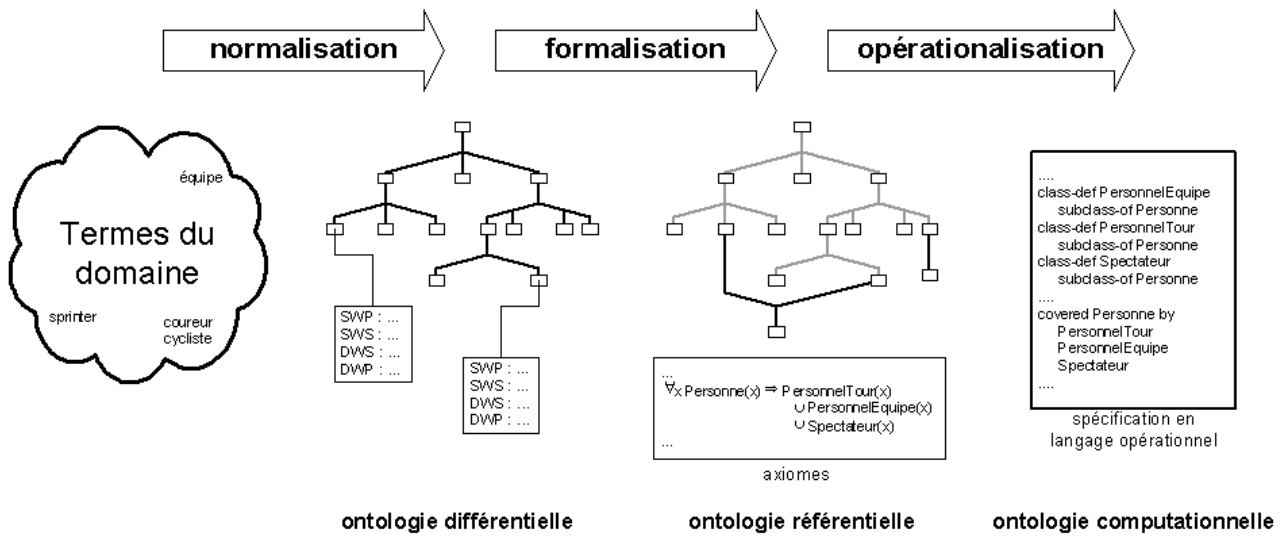


FIG. 1 – Les 3 étapes de la méthode de construction d’ontologies

ce corpus a fait ressortir un certain nombre de termes qualifiant une personne humaine (Figure 2), mais qui ne jouent manifestement pas tous le même rôle dans une épreuve cycliste.

coureur cycliste - spectateur - directeur sportif -
médecin équipe - journaliste - commissaire de
course - grimpeur - rouleur - sprinteur

FIG. 2 – Quelques termes désignant une personne

Après avoir isolé un certain nombre de libellés, il faut veiller à leurs donner une signification précise et donc utiliser une théorie sémantique appropriée. Bruno Bachimont propose dans [2] d’utiliser la sémantique différentielle présentée par François Rastier dans [14]. Dans ce paradigme, le sens est intra-linguistique : il se construit par des relations d’opposition entre les unités du système linguistique. En fait, il s’agit d’attribuer un sens aux termes grâce à des traits sémantiques ou *sèmes*. Ces sèmes que l’on va associer à une unité sont regroupables en deux catégories :

- les *sèmes génériques* qui permettent de regrouper les unités entre elles ;
- les *sèmes spécifiques* qui permettent à une unité de se distinguer de celles avec qui on l’a regroupée.

Par exemple, dans le domaine du cyclisme, on va pouvoir associer le sème «monte bien les côtes dures et longues en montagne» au signifiant grimpeur, et le sème «endurant et rapide dont les qualités se révèlent surtout dans les courses sur terrain plat» au signifiant rouleur. Ces deux sèmes permettent bien de distinguer ces deux unités l’une de l’autre. En revanche, elles vont partager le même sème «homme dont le métier est de

participer aux courses cyclistes» – associé au signifiant coureur cycliste. Ces deux types de sèmes permettent donc de définir une unité par les identités et les différences qu’elle entretient avec ces plus proches voisins. C’est cette position dans le système qui donne un sens à l’unité considérée.

Il faut cependant signaler que ce sens différentiel est fortement dépendant du contexte, qui encadre l’interprétation, le mécanisme d’attribution des sèmes aux termes. Or, pour obtenir une primitive réellement exploitable, il faut contrôler cette interprétation : il s’agit bien d’une *normalisation sémantique*.

4.1.2 Ontologie et principes différentiels

Comment concrètement parvenir à cette normalisation du sens des unités manipulées ? Le réseau d’identités et de différences évoqué dans le paragraphe précédent reste à structurer, si on veut réellement le rendre opérationnel, pour aboutir à la construction de ce que l’on va appeler une *ontologie différentielle*. Tout d’abord, on observe qu’une unité (dorénavant, on parlera de *notion*) se définit en premier lieu par une relation de subsomption avec une autre unité : celle dont elle hérite tous ses sèmes génériques. Par exemple, on pourra affirmer que coureur cycliste subsume grimpeur et rouleur. En pratique, le créateur d’ontologies doit donc pouvoir exprimer les identités et les différences de chaque notion dans son voisinage proche : la notion-parente et les notions-sœurs. Il obtiendra finalement une taxinomie¹⁴ de notions où la signification d’une unité s’obtiendra en collectant les identités et les différences permettant de caractériser les notions rencontrées sur le chemin qui mène de la notion racine (la plus générique) à cette unité.

14. Il s’agit d’un arbre : l’héritage multiple est interdit dans la mesure où l’on associe aux notions d’une fratrie des sèmes spécifiques qui sont en opposition.

Bruno Bachimont propose quatre principes permettant d'expliciter ces informations :

- le principe de *communauté avec le père* («*similarity with parent*» ou **SWP**) : on explicite pourquoi le fils hérite des propriétés de la notion qui le subsume ;
- le principe de *différence avec le père* («*difference with parent*» ou **DWP**) : on explicite la différence qui permet de distinguer le fils du père ;
- le principe de *différence avec les frères* («*difference with siblings*» ou **DWS**) : on indique ici la propriété qui permet de distinguer la notion considérée de ses notions-sœurs ;
- le principe de *communauté avec les frères* («*similarity with siblings*» ou **SWS**) : on donne enfin la propriété – admettant plusieurs valeurs exclusives – qui a permis de caractériser les notions d'une même fratrie, justifiant ainsi le principe précédent.

Pour reprendre l'exemple donné dans la Figure 2, on s'aperçoit que les termes grimpeur, rouleur et sprinter renvoient à un coureur cycliste. De plus, toutes les personnes habituellement présentes sur le Tour de France n'ayant pas le même statut, on peut les regrouper selon les raisons qui les ont poussées à y participer. Ainsi, la notion *Personne* peut se spécialiser en trois nouvelles notions – *Personnel Épreuve*, *Personnel Équipe* et *Spectateur* – selon les principes différentiels présentés dans la Figure 3. Le sous-arbre complet de la notion *Personne* est donné dans la Figure 4.

→ <i>Personnel Épreuve</i>
SWP: c'est un individu humain
SWS: une propriété précise la raison de la personne sur les lieux de l'épreuve
DWS: est accrédité par la direction de l'épreuve
DWP: joue un rôle particulier par rapport à l'épreuve : la personne est accréditée par la direction de celle-ci
→ <i>Personnel Équipe</i>
SWP: c'est un individu humain
SWS: une propriété précise la raison de la personne sur les lieux de l'épreuve
DWS: est employé par une des équipes participantes
DWP: joue un rôle particulier par rapport à l'épreuve : la personne est employée par une équipe participante à l'épreuve
→ <i>Spectateur</i>
SWP: c'est un individu humain
SWS: une propriété précise la raison de la personne sur les lieux de l'épreuve
DWS: n'est ni accrédité par la direction de l'épreuve, ni employé par une des équipes participantes
DWP: joue un rôle particulier par rapport à l'épreuve : se contente d'y assister

FIG. 3 – Principes différentiels associés aux spécialisations directes de *Personne*

Par le processus de normalisation sémantique exposé ci-dessus, on passe d'un candidat-terme à une notion dont la

signification, ancrée dans le domaine et l'application envisagée, est invariable et peut donc fonctionner comme une primitive exprimant une connaissance.

4.2 La gestion des ontologies différentielles dans DOE

Dans cette première étape de modélisation, l'utilisateur doit pouvoir naviguer dans une hiérarchie arborescente de notions. L'éditeur offre donc une représentation graphique de cet arbre et permet d'interagir avec la hiérarchie : consulter des notions déjà représentées, les modifier et en ajouter de nouvelles. Se pose alors la question du contenu de ces notions, de ce que l'on peut y définir (en plus évidemment de son nom).

4.2.1 Modélisation sémantique des notions

Il faut d'abord définir le contenu conceptuel des notions à inclure dans l'ontologie différentielle. L'utilisateur doit pouvoir consulter et définir les attributs correspondants aux quatre principes dégagés par la méthodologie. L'outil développé se propose d'assister l'utilisateur dans cette saisie, voire d'en automatiser une partie. En effet, au cours de la construction de l'ontologie du cyclisme, quelques faits intéressants ont pu être observés, conduisant à accorder un statut méthodologique différent – encore à justifier théoriquement – à chacun des principes.

En premier lieu, il y a souvent partage de *SWP* et *SWS* par l'ensemble des concepts membres d'une même fratrie. En effet, toutes les notions d'une même fratrie sont filles d'un même père, et toutes admettent une propriété discriminante déterminée à partir d'un axe sémantique commun : cela est nécessaire à la cohérence de la structure locale de l'ontologie. L'éditeur remplit donc automatiquement ces deux champs pour toute une fratrie dès lors qu'une des notions possède l'information. Il ne faut néanmoins pas supprimer toute liberté de modifier ces attributs : l'utilisateur doit pouvoir reformuler un axe sémantique commun (boutons *edit*). L'éditeur laisse cette possibilité en répercutant alors les éventuels changements dans la fratrie à laquelle la notion appartient. Si l'ontologiste veut déplacer une notion dans la taxinomie, les deux principes de similarité de celle-ci sont adaptés à sa nouvelle fratrie.

Ensuite, le principe de différence avec le père (*DWP*) apparaît souvent comme la somme de la spécification de l'axe commun aux frères (*SWS*) et de celle de la différence avec les frères (*DWS*). En effet, on se donne d'abord le moyen de créer une différence, que l'on concrétise ensuite pour achever la définition du concept. L'éditeur propose donc de construire lui-même la différence avec le père (*DWP*) en concaténant simplement les énoncés des principes *DWS* et *SWS*. Mais ici, il ne doit pas y avoir de contrainte définitive : l'utilisateur a tout loisir de changer ce qui ne reste qu'une simple suggestion de l'outil (bouton *build*).

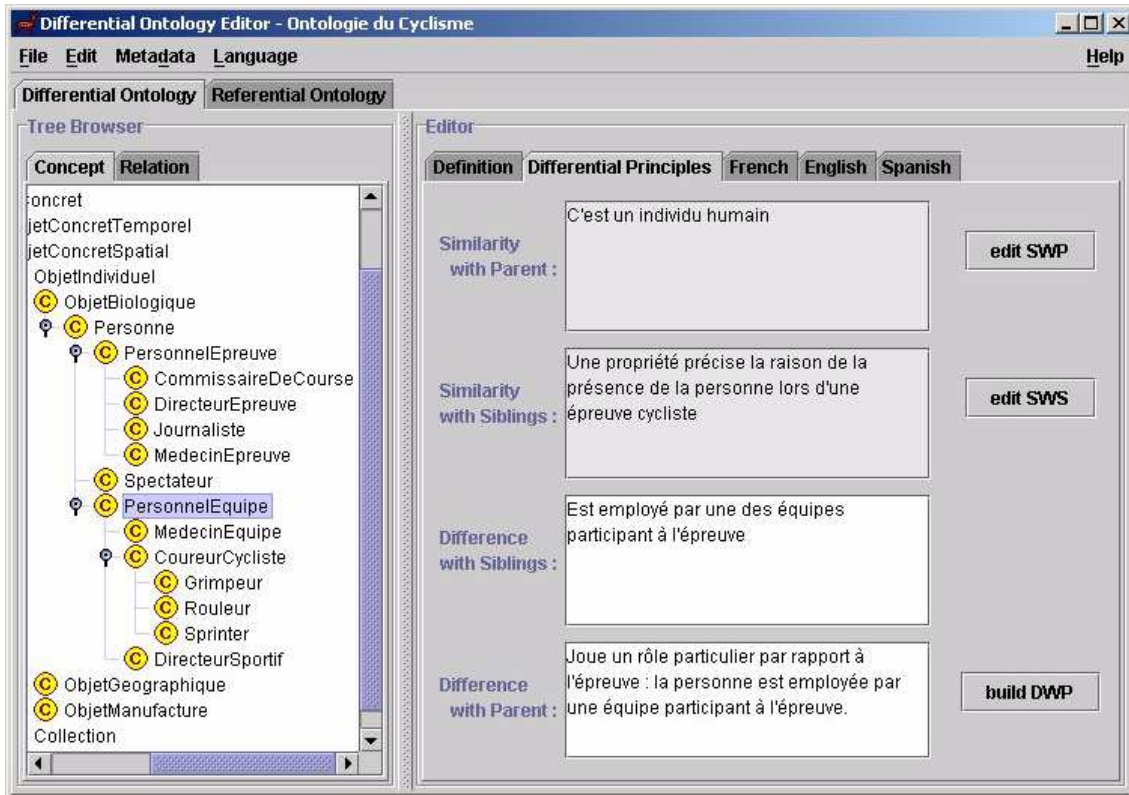


FIG. 4 – Les principes différentiels de la notion Personnel Équipe dans DOE

Enfin, il faut signaler que les principes sont souvent considérés par l'utilisateur dans le même ordre, à savoir *SWP*, *SWS*, *DWS* et *DWP*. On commence effectivement par préciser que la notion éditée est bien subsumée par une autre. On affirme ensuite ce qui fonde son originalité : choisir le fils plutôt que le père, c'est insister sur une caractéristique donnée (*SWS*) et en indiquer une valeur possible (*DWS*) qui l'oppose aux autres frères, et donc qui différencie ses instances de toutes les autres instances du père. L'interface de l'éditeur reflète ce cheminement, comme le montre la Figure 4 qui illustre l'exemple déroulé jusqu'à présent pour le concept *Personnel Équipe*.

On a vu que la modélisation différentielle d'une notion permettait de définir celle-ci par l'interprétation de sa position dans la structure hiérarchique globale de l'ontologie. A titre informatif, l'éditeur génère donc, dans l'onglet *Définition*, une *définition différentielle* des notions, en affichant le chemin menant de la racine de l'arbre à la notion considérée et en récapitulant les valeurs des différences avec les pères rencontrés sur ce chemin.

4.2.2 Informations complémentaires

Dans un second temps, l'éditeur propose d'attacher à chacune des notions des aspects propres à une langue donnée. Deux éléments terminologiques peuvent être saisis dans l'éditeur : le *terme préféré*, celui qui est utilisé pour désigner

le plus souvent la notion (et qui a donc de fortes chances d'apparaître également en tant que libellé de la notion), et une liste des différents *synonymes* de celle-ci.

Comme les ontologies construites ont pour vocation le partage d'une conceptualisation au sein d'une communauté regroupant des individus utilisant parfois des langues différentes, les informations terminologiques doivent apparaître dans chacune des langues concernées. L'éditeur gère donc la possibilité de rajouter de nouvelles langues dans la portée de l'ontologie et adapte en conséquence les informations qui sont à saisir pour chaque notion. Cela se matérialise dans l'éditeur par un nouvel onglet dont l'intitulé est la langue choisie (la Figure 4 indique que ces informations peuvent être données en français, anglais ou espagnol).

Parmi les attributs de la notion propre à une langue, nous avons également choisi de faire figurer une *définition encyclopédique*. Celle-ci n'est pas issue d'un processus de modélisation linguistique, mais se veut l'équivalent d'un commentaire, un complément informel de la définition élaborée à l'aide des principes différentiels.

Cette gestion de plusieurs langues, pour un certain nombre d'informations attachées aux notions, n'a pas pour ambition de rendre l'éditeur « multilingue ». Ce problème est en effet beaucoup plus complexe ; des notions verbalisables dans une langue pourraient ne pas l'être immédiatement dans une autre. L'éditeur se borne donc actuellement à don-

ner la possibilité de mettre en correspondance des termes et des définitions informelles. De plus, il faut rappeler qu'un certain nombre d'informations centrales (les principes différentiels, notamment) ne peuvent être explicitées que dans une seule langue, qui prendra alors le statut de langue de référence pour l'ontologie. Toutes les langues ne sont donc pas traitées de la même manière.

4.2.3 Gestion des relations

On a récapitulé l'ensemble des informations nécessaires pour caractériser les notions de l'ontologie différentielle. A ce stade, on peut remarquer qu'il n'existe pas de différence fondamentale entre le traitement des notions dénotant des «concepts» et celles dénotant des «relations entre concepts» : celles-ci sont en effet définies comme les concepts à l'aide des quatre principes. Nous considérons qu'une des caractéristiques fondamentales des relations, à savoir la spécification des concepts dont elles lient les instances, ne doit apparaître qu'à l'étape suivante. Convaincus que la distinction est fondamentale pour la compréhension de tout domaine, nous choisissons tout de même de faire figurer les deux types de notions dans deux arbres différents et le modèle sous-jacent de l'ontologie entretient deux listes.

4.3 Normalisation sémantique et notions de haut niveau

Lors de la construction de l'ontologie du cyclisme, nous nous sommes heurtés à un problème de poids : comment exprimer les principes différentiels pour les notions de haut niveau de l'ontologie ? En effet, la modélisation d'un domaine fait rapidement ressortir un ensemble de concepts «majeurs», d'assez haut niveau, que l'on a spécialisés pour former des sous-arbres dans la taxinomie. Ainsi, dans le cyclisme, émergent des concepts tels que *Personne*, *Épreuve* ou encore *Incident*. Mais, d'après la méthodologie exposée ci-dessus, ces concepts ne peuvent pas appartenir à une même fratrie dont le père serait le concept le plus général *AnyConcept* : il est en effet impossible de dégager un axe sémantique permettant de les réunir et par conséquent d'exprimer les principes différentiels censés justifier cet arbre. La solution passe donc par l'introduction de nouveaux concepts permettant d'effectuer cette jonction entre le sommet de l'ontologie et les concepts propres au domaine modélisé.

On retrouve ici les problèmes liés à ce qu'on appelle les *top-level ontologies*. Quelques tentatives comme *UpperCyc* et dernièrement l'initiative du SUO¹⁵ sous l'égide de l'IEEE ont pour but de standardiser les concepts d'une telle ontologie afin que celle-ci puisse resservir à tous ceux qui en ont besoin. Nous nous sommes donc inspirés de ces travaux pour compléter l'ontologie du cyclisme.

15. SUO: *Standard Upper Ontology*, <http://suo.ieee.org/>.

16. Encore que d'autres types de sémantiques formelles soient envisageables.

Ce travail de modélisation est difficile car il exige des connaissances qui relèvent plus de la philosophie que du domaine étudié. La manipulation des principes différentiels s'avère en effet ardue dans les premières ramifications de l'arbre. La Figure 5 montre partiellement le résultat obtenu dans les taxinomies des concepts et des relations. Peut-être critiquables, les choix qui ont été effectués ont au moins le mérite de pouvoir être expliqués par l'énonciation des principes différentiels pour chacun des nœuds de l'arbre. De plus, le gain en termes de ré-utilisabilité est indéniable : d'autres ontologies actuellement en développement à l'INA profitent de l'effort effectué en reprenant des parties hautes de l'ontologie.

5 Formalisation des connaissances

L'arbre ontologique construit dans la première étape a permis de fixer un certain nombre de notions en normalisant leurs sens. Cependant, le travail n'est toujours pas terminé ; on ne peut pas vraiment utiliser cet ensemble de notions dans un véritable SBC. Il faut en effet introduire des concepts obéissant à une sémantique formelle et extensionnelle¹⁶ pour que ceux-ci servent en tant que primitives dans un langage formel de représentation de connaissances. Nous allons donc détailler quelles manipulations vont permettre d'atteindre cet objectif à partir de l'ontologie différentielle (section 5.1) avant de montrer comment ces opérations sont partiellement prises en charge dans l'éditeur *DOE* (section 5.2). Nous verrons alors comment les deux premières étapes de la méthodologie s'articulent dans l'éditeur (section 5.3).

5.1 Construction d'une ontologie référentielle

Les notions dont on dispose dans l'ontologie différentielle sont de nature linguistique, toujours soumises à des processus d'interprétation (même si à présent cette interprétation est adaptée à la pratique ciblée lors de la construction de l'ontologie). Le passage à une sémantique extensionnelle va permettre de lier ces notions à un ensemble de référents dans le monde, donnant lieu à ce qu'on va appeler dans la suite une *ontologie référentielle*, composée de *concepts* qui vont agir comme des primitives formelles.

Ces primitives ne sont plus définies par les principes différentiels. Elles n'ont donc pas intrinsèquement de sens interprétatif ; elles n'en acquièrent que parce qu'elles sont reliées – par leur libellé – aux notions différentielles. Cependant, ce sont des primitives et elles peuvent, grâce aux mécanismes de composition de sens d'une sémantique extensionnelle, servir à définir de nouveaux concepts formels. Chaque concept étant lié par référence à un ensemble d'objets du domaine (son *extension*), on peut avoir recours à

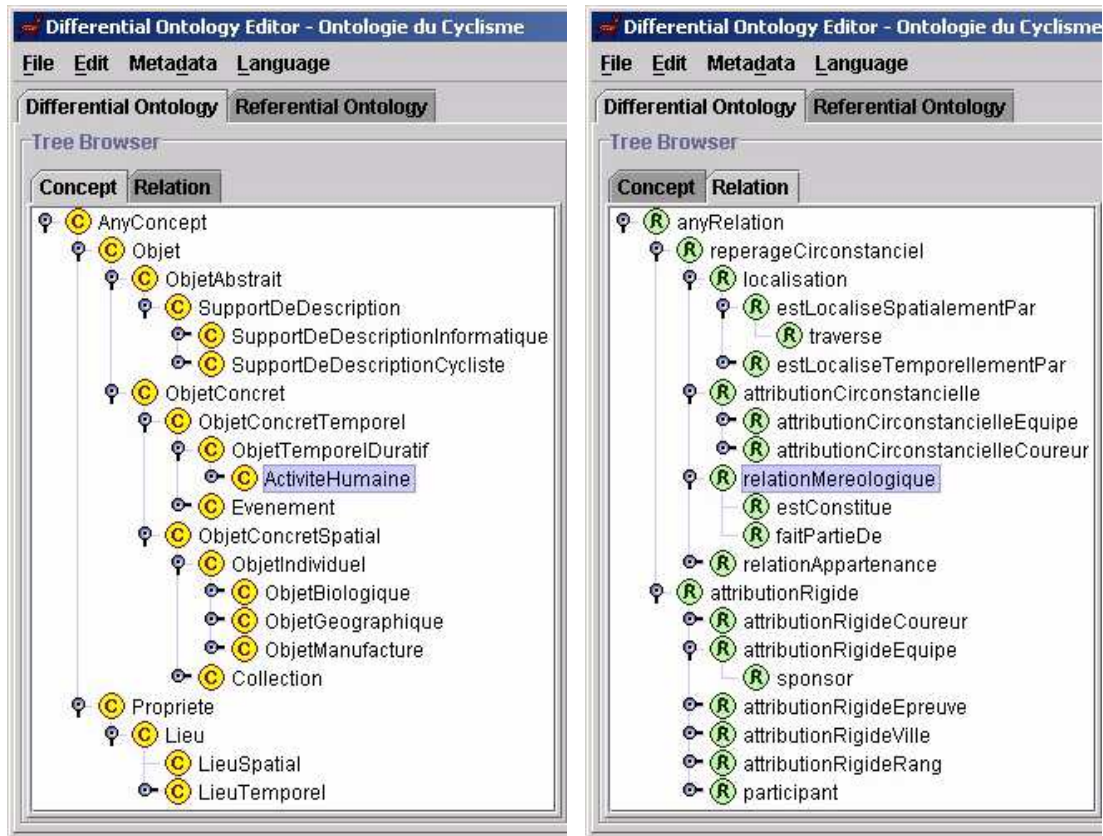


FIG. 5 – Les concepts et les relations de haut niveau utilisés dans l'ontologie du cyclisme

des opérations de composition de sens qui utilisent les opérations qui existent pour les ensembles (réunion, intersection ou complémentaire).

La comparaison des extensions permet de définir une relation d'héritage extensionnelle entre les concepts : un concept sera subsumé par un autre si et seulement si son extension est incluse dans celle de son parent. On peut alors se poser la question de la structure de la hiérarchie de subsumption obtenue. S'il paraît naturel d'affirmer que les relations d'héritage définies dans l'ontologie différentielle tiendront toujours, on peut se demander dans quelle mesure l'ajout de nouveaux nœuds va modifier la structure arborescente. En effet, les «transcriptions» des notions différentielles en concepts formels peuvent admettre des extensions qui ont un sous-ensemble commun. Ainsi, *rouleur* et *grimpeur* sont bien des notions qui s'excluent, mais les concepts formels correspondants ont des extensions qui peuvent avoir en commun plusieurs individus : par exemple, l'individu Lance Armstrong est habituellement répertorié dans ces deux catégories. On peut donc définir dans l'ontologie référentielle un concept formel *Grimpeur_Rouleur* dont la référence sera l'intersection des extensions des concepts *Grimpeur* et *Rouleur*, et dont Lance Armstrong sera un élément. L'héritage multiple devient donc possible,

et la structure hiérarchique obtenue est celle d'un treillis¹⁷.

On a vu que la sémantique référentielle nous permet de définir de nouveaux concepts en fonction d'autres concepts. Il faut signaler que l'on peut également exprimer de telles définitions pour les concepts formels directement issus de la transposition de l'ontologie différentielle. C'est notamment à cette étape que l'on va définir les relations par la donnée de leur arité et de leur domaine, ce qui les associera de fait à des produits cartésiens de références de concepts.

On peut aussi introduire des axiomes logiques, lois qui régissent les comportements des individus qui constituent l'extension des concepts formels. Par exemple, on peut spécifier que les sous-concepts d'un concept forment une partition disjointe de celui-ci. Ainsi, dans l'ontologie du cyclisme, les concepts *Personnel Épreuve*, *Personnel Équipe* et *Spectateur* forment une partition disjointe du concept *Personne*. On peut enfin exprimer des lois concernant les relations, comme leur attacher des propriétés algébriques : réflexivité, transitivité... Il est clair que l'ontologie référentielle doit expliciter toutes les lois logiques induites par la conceptualisation que l'on veut spécifier.

17. Il est en effet possible de clore l'ensemble des concepts par le concept «vide».

5.2 La gestion des ontologies référentielles dans *DOE*

Concrètement, l'ontologie référentielle est le «produit fini» de la construction de l'ontologie au niveau des connaissances. On doit pouvoir attacher aux notions de l'ontologie différentielle une signification formelle permettant de fixer la référence de ces notions, et qui pourra être traduite de manière automatique dans un langage compréhensible par l'ordinateur.

La première opération de cette étape consiste à récupérer les deux taxinomies établies pour l'ontologie différentielle. Aucune des informations apportées dans l'ontologie référentielle ne peut remettre en cause la hiérarchie de subsumption : les deux arbres de concepts et de relations construits précédemment réapparaissent donc tels quels dans *DOE*.

L'utilisateur peut ajouter des concepts ou des relations qui ne présenteront plus les informations sémantiques des notions différentielles. Ces nouveaux objets peuvent être ajoutés en tant que spécialisation d'une entité issue de l'ontologie différentielle, mais aussi que spécialisation d'une entité introduite lors de l'étape référentielle. L'héritage multiple étant possible, l'éditeur tient à jour, pour chaque entité, la liste de ses parents et permet d'en ajouter ou d'en retirer. Par contre, les entités introduites à ce stade dans *DOE* restent *primitives*. La possibilité de définir des concepts par conditions nécessaires et suffisantes avec des combinaisons booléennes, ou encore par l'expression de contraintes relationnelles, n'est pas proposée dans *DOE*. D'autres outils (comme *Protégé2000* ou *OilEd*) pouvant assurer le support de telles fonctionnalités, nous préférons recourir à des mécanismes d'exports adaptés (en utilisant par exemple DAML+OIL, un langage d'encodage et d'échange d'ontologies proposé dans le cadre des travaux relatifs au Web sémantique) plutôt que de refaire ce qui est déjà bien pris en compte (ou en voie de l'être) ailleurs. Il en va de même pour les types d'axiomes simples qui s'appliquent aux relations (symétrie, réflexivité, ...).

Nous avons choisi dès le début de différencier les relations des concepts correspondant aux entités du domaine. Cela a une répercussion au niveau de l'ontologie référentielle. La description de la référence des relations est effectivement spécifique : pour une relation donnée, il faut pouvoir préciser son arité ainsi que ses domaines. L'éditeur propose de rentrer ces informations, et le choix des domaines se fait parmi la liste des concepts déjà entrés dans l'ontologie. A ce stade, l'éditeur prend en compte les relations de spécialisation entre relation : l'arité, quand elle a été définie, doit être propagée correctement le long de la hiérarchie de subsumption. De même, les domaines d'une relations doivent hériter des domaines de la relation dont elle hérite. En cas d'héritage multiple, les domaines de la relation fille doivent être sous-concepts des domaines de toutes les relations parentes.

5.3 Articulation des étapes différentielle et référentielle

La méthodologie exposée jusqu'à présent insiste bien sur les différences entre les deux premières étapes. Les objets qui y sont manipulés sont de nature différente (linguistique pour les uns, logique pour les autres), ce qui a des répercussions sur la manière dont on les construit. Il ne faut cependant pas oublier qu'il y a des liens forts entre les deux ontologies : les notions différentielles servent à donner une signification linguistique aux concepts référentiels auxquels ils sont directement associés. La conséquence directe est d'ailleurs la réutilisation dans l'ontologie référentielle de la hiérarchie de subsumption établie dans l'ontologie différentielle. Une question s'est alors posée : l'éditeur doit-il présenter ces deux étapes de front, ou l'une après l'autre ?

D'après notre expérience, il semble préférable de construire l'ontologie référentielle au fur et à mesure de l'élaboration de la taxinomie différentielle. En effet, les notions qui y sont présentes ont pour but de préciser le sens des entités référentielles correspondantes. Mais ces entités doivent pouvoir servir à l'introduction d'autres entités (concepts ou relations) dès qu'elles sont définies. Attendre que toute la taxinomie différentielle soit entièrement terminée pour introduire des concepts référentiels qui auraient pu l'être bien plus tôt peut se révéler préjudiciable à la sérénité supposée de la construction de l'ontologie. Par exemple, le concept *Maillot A Pois Rouge* désigne sur le Tour de France le coureur cycliste leader du classement de la montagne. Ce concept peut être introduit dès que le concept *Coureur Cycliste* et la relation *est Classé Coureur Montagne* sont définis. L'éditeur permet donc de mener ces deux étapes simultanément en passant à tout moment de la construction d'une ontologie à l'autre.

Il faut alors souligner les problèmes de cohérence que cela induit. Si une évolution monotone de l'ontologie différentielle ne pose pas de problèmes (les relations de subsumption restent valides dans l'ontologie référentielle), le déplacement ou la suppression d'une de ses branches peut créer des concepts référentiels orphelins. L'éditeur prend en charge cette vérification et informe l'utilisateur le cas échéant. Enfin, il est à noter que l'introduction des entités référentielles ne permet pas de remettre en cause les données de l'ontologie différentielle puisque ces dernières sont issues d'une étape qui est logiquement antérieure.

6 Opérationnalisation des connaissances

Dans cette troisième et dernière étape, on va chercher à munir les concepts présents dans l'ontologie référentielle d'une signification en termes d'opérations informatiques. On spécifie ainsi le comportement des objets informatiques présents dans le système qui utilise l'ontologie, pour aboutir

à ce que l'on va appeler une *ontologie computationnelle*.

Les concepts computationnels se définissent par les inférences, les calculs que pourra effectuer un système à partir de la donnée des individus qu'ils couvrent dans le monde. Le système utilise alors un langage opérationnel de représentation des connaissances¹⁸ qui fait appel à des capacités d'inférence précises répondant aux besoins exprimés lors de la spécification du système. Pour un langage de représentation reposant sur les graphes conceptuels, il s'agira d'opérations de manipulation de graphes (jointure, projection, etc.). Pour un langage fondé sur le paradigme des logiques de description, il s'agira plutôt des tests de subsomption permettant la classification des individus introduits dans la base de connaissances. L'expression des primitives de l'ontologie référentielle dans un de ces langages assigne donc une véritable signification computationnelle aux concepts manipulés par le système. Par exemple, si on veut exprimer le fait qu'une *Personne* est soit un membre d'une équipe, soit employée par la direction d'une épreuve, soit un *Spectateur*, on peut recourir à la déclaration suivante, utilisant le langage DAML+OIL :

```
<daml:Class rdf:about="#Personne">
  <daml:disjointUnionOf
    rdf:parseType="daml:collection">
    <daml:Class rdf:about="#PersonnelEpreuve"/>
    <daml:Class rdf:about="#Personnel"/>
    <daml:Class rdf:about="#Spectateur"/>
  </daml:disjointUnionOf>
</daml:Class>
```

Dans l'éditeur *DOE*, ce passage à une ontologie computationnelle se fera par un export de l'ontologie référentielle dans un certain nombre de langages opérationnels. Par exemple, dans le cadre du projet *Opales*, l'éditeur assure une traduction en langage CGXML, un langage permettant de spécifier des supports de graphes conceptuels, utilisé par COGITANT, une plate-forme graphes conceptuels développée au LIRMM de Montpellier. Cette traduction s'effectue grâce à une feuille de style XSLT appliquée au format de sauvegarde XML de l'éditeur. Pour être tout à fait complet, il faut noter que l'éditeur permet aussi de sauvegarder un certain nombre de métadonnées concernant l'ontologie elle-même. Il suit ici les propositions du *Dublin Core*¹⁹.

A ce stade, on peut observer qu'il existe une certaine ambiguïté sur le statut des mécanismes d'export. D'un côté, ils sont utilisés lors de l'étape référentielle, où ils permettent de définir plus précisément l'ontologie en utilisant des outils plus complets sur ce plan. De l'autre, ils peuvent servir à spécifier l'ontologie computationnelle dès lors que les informations apparaissant dans l'ontologie référentielle sont satisfaisantes pour l'application envisagée. Il faut néanmoins souligner que cette difficulté se place plus au niveau de l'implémentation qu'à celui de la méthodologie elle-même,

18. Pour un comparatif récent des langages opérationnels, on peut se reporter à [3]. L'action *ESCRIRE* (<http://www.inrialpes.fr/exmo/cooperation/escrire/>) compare elle, les trois grandes classes de formalismes de représentation des connaissances (logiques de descriptions, graphes conceptuels et paradigme objet) dans le contexte de la représentation du contenu de documents.

19. <http://www.dublincore.org>

puisqu'elle résulte du choix de ne pas mettre en œuvre des fonctionnalités d'expressivité formelle ou computationnelle prises en charge par d'autres environnements.

7 Conclusion et perspectives

Nous avons vu que les méthodes et les outils en circulation dans la communauté se préoccupant de la conception des ontologies présentaient un point faible : rien ne force les utilisateurs à exprimer clairement le sens qu'ils veulent attribuer aux concepts, les commentaires restant très informels. Nous avons alors décidé d'utiliser une méthodologie, proposée par Bruno Bachimont, qui s'appuie clairement sur des considérations linguistiques (elle utilise la sémantique différentielle) pour expliciter le sens des notions présentes dans l'ontologie. Nous avons mis en œuvre cette méthodologie dans un outil d'édition, *DOE*, afin de confirmer, par l'usage, l'intérêt qu'elle présente. Cet outil, plutôt que de constituer un environnement complet, se veut complémentaire des éditeurs existants en mettant l'accent sur la structuration des taxinomies.

Plusieurs ontologies ont déjà été modélisées (pour le projet *OPALES*) avec cet outil. Cependant, il ne faut pas perdre de vue que la portée de l'outil est assez limitée et un retour d'expérience supplémentaire de son utilisation est indispensable pour pouvoir le faire évoluer. Parmi les améliorations souhaitables permettant de mieux insérer *DOE* dans une chaîne complète de conception d'ontologies, on peut déjà signaler :

- en amont, l'articulation avec les résultats d'outils d'acquisition des candidats-termes (travail terminologique) est une piste à explorer très sérieusement ;
- il faudrait chercher s'il n'existe pas, pour la phase de construction de l'ontologie référentielle, des fonctionnalités requises et non prises en charge par d'autres éditeurs (par exemple, la spécification des axiomes simples pour les relations ne rentre pas dans ce cadre) ;
- finalement, il serait intéressant de pouvoir importer des ontologies conçues avec d'autres outils pour les valider selon notre méthodologie.

Références

- [1] J. Arpirez, O. Corcho, M. Fernández-López et A. Gómez-Pérez. *WebODE : a Workbench for Ontological Engineering*. In *First international Conference on Knowledge Capture (K-CAP'01)*, Victoria, Canada, 2001.

- [2] Bruno Bachimont. Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. In J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, éditeurs, *Ingénierie des Connaissances : Evolutions récentes et nouveaux défis*. Eyrolles, 2000.
- [3] Catherine Barry, Christian Cormier, Gilles Kassel et Jérôme Nobécourt. Évaluation de langages opérationnels de représentation d'ontologies. In *Ingénierie des Connaissances (IC'01)*, pages 309–327, Grenoble, 2001.
- [4] S. Bechhofer, I. Horrocks, C. Goble et R. Stevens. Oiled: a Reason-able Ontology Editor for the Semantic Web. In Springer-Verlag LNAI, éditeur, *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*, volume (2174), pages 396–408, Vienne, Autriche, 2001.
- [5] M. Blázquez, M. Fernández, J.M. García-Pinar et A. Gómez-Pérez. Building Ontologies at the Knowledge Level using the Ontology Design Environment. In *11th Banff Knowledge Acquisition Workshop (KAW'98)*, Banff, Canada, 1998.
- [6] Didier Bourigault. *Extraction et structuration automatiques de terminologie pour l'aide à l'acquisition des connaissances à partir de textes*. PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1994.
- [7] Thomas Dechilly et Bruno Bachimont. Une ontologie pour éditer des schémas de description audiovisuels, extension pour l'inférence sur les descriptions. In *Ingénierie des Connaissances (IC'00)*, Toulouse, 2000.
- [8] A. Farquhar, R. Fikes, W. Pratt et J. Rice. Collaborative Ontology Construction for Information Integration. Technical Report KSL-95-63, Knowledge Systems Laboratory Department of Computer Science, 1995.
- [9] M. Fernández, A. Gómez-Pérez et N. Juristo. METHONTOLOGY : From Ontological Art Towards Ontological Engineering. In *AAAI97 Spring Symposium Series on Ontological Engineering*, Stanford, USA, 1997.
- [10] Antoine Isaac. Vers la mise en œuvre informatique d'une méthode de conception d'ontologies. Rapport de DEA, Institut des Sciences Humaines Appliquées, Université Paris IV, Septembre 2001.
- [11] Karine Lespinasse, Benoit Habert et Bruno Bachimont. Le péritexte, un sésame pour les données audiovisuelles ? L'analyse exploratoire d'un corpus hétérogène de notices documentaires interprétant des documents audiovisuels. In *5^{èmes} Journées Internationales d'Analyse Statistiques des Données Textuelles (JADT'00)*, Lausanne, Suisse, 2000.
- [12] A. Maedche, H.-P. Schnurr, S. Staab et R. Studer. Representation Language-Neutral Modeling of Ontologies. In *Proceedings of the German Workshop «Modellierung»*, Koblenz, Allemagne, 2000.
- [13] N.F. Noy, R.W. Ferguson et M.A. Musen. The knowledge model of Protégé2000 : Combining interoperability and flexibility. In *12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'00)*, Juan-les-Pins, 2000.
- [14] F. Rastier, M. Cavazza et A. Abeillé. *Sémantique pour l'analyse*. Masson, Paris, 1994.
- [15] M. Uschold et M. Gruninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 1996.
- [16] P. Zweigenbaum et Consortium MENELAS. MENELAS: an access system for medical records using natural language. *Computer Methods and Programs in Biomedicine*, (45), 1994.