DAFOE: A Multimodel and Multimethod Platform for Building Domain Ontologies*

Jean Charlet
INSERM UMR_S 872, Eq. 20 ; Université Pierre et Marie Curie ; AP-HP - France
jean.charlet@spim.jussieu.fr

Sylvie Szulman
LIPN - UMR 7030, Université Paris 13 - CNRS - France
Sylvie.Szulman@lipn.univ-paris13.fr

Guy Pierra
LISI-ENSMA, Université de Poitier - France
pierra@ensma.fr

ABSTRACT

The concept of ontologies, appeared in the nineties, constitute a key point to represent and share the meaning carried by formal symbols. Yet building of such an ontology is quite difficult. A way to do so is to use preexistent elements (textual corpus, taxonomies, norms or other ontologies) and operate them as a basis to define the field ontology. Still, there is neither accepted process nor set of tools to progressively built ontologies from the field's resources in a traceable and explicit way. We report in this paper several propositions developed within the framework of the ANR DaFOE4App project to support emergence of such tools. Accounting for several needs, methods and models, our proposal rely on 3 key points: 1) to define a general methodological framework to integrate the realization of several design scenarios; 2) to define a modeling structure allowing various templates; 3) to specify and develop a platform able to integrate various kind of tools currently used in an autonomous way within a modeling structure. Moreover, the platform should insure persistence and traceability and allow to built formal ontologies from the analysis of textual corpus using NLP (Natural Language Processing) annotated texts.

Categories and Subject Descriptors

D.2.11 [**Software Architectures**]; I.2.4 [**Knowledge Representation Formalisms and Methods**]; H.2.1 [**Database Management**]: Logical DesignData models

General Terms

Design

Keywords

Ontology Building, Ontology Editor, Meta-Modelization, Data Model

*DaFOE4App <http://dafoe4app.fr> est un projet ANR TLOG 010 qui vise à construire une plateforme de construction d'ontologie, DaFOE.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
JFO 2008 December 1-3, 2008, Lyon, France
Copyright 2008 ACM X-XXXXX-XX-X/XX/XX ...$5.00.

1. INTRODUCTION

Depuis son émergence, au début des années 1990, dans les recherches en modélisation de connaissances, la notion d'ontologie s'est rapidement diffusée dans un grand nombre de domaines de recherche en informatique. Définie comme la représentation formelle, et consensuelle au sein d'une communauté d'utilisateurs, des concepts propres à un domaine et des relations qui les relient [17], la notion d'ontologie apparaît ainsi comme la pierre philosophale permettant de représenter explicitement, et de partager, la signification dénotée par des symboles informatiques. Compte tenu du caractère très prometteur de cette notion, de nombreux travaux ont visés à permettre son utilisation dans des domaines aussi divers que le traitement automatique de la langue naturelle, la recherche d'information, le commerce électronique, le web sémantique, la spécification des composants logiciels et l'intégration de système d'information.

L'efficacité de toutes ces approches présuppose néanmoins l'existence d'une ontologie de domaine susceptible d'être développée, ou d'être mise en œuvre, au sein de l'application cible. Or la conception d'une telle ontologie s'avère particulièrement difficile, surtout si l'on souhaite qu'elle fasse l'objet de consensus dans une communauté assez large. Un moyen très largement utilisé pour atteindre cet objectif est de partir d'éléments préexistants dans le domaine : corpus textuels, taxonomies, normes ou fragments d'ontologie préexistants, et de les exploiter comme base pour définir progressivement l'ontologie du domaine. La construction d'ontologie à partir de textes fait l'objet d'études depuis plusieurs années dans le domaine de l'ingénierie des ontologies. Un cadre méthodologique en quatre étapes (constitution d'un corpus de documents, analyse linguistique du corpus, conceptualisation, opérationnalisation de l'ontologie) est commun à la plupart des méthodes de construction d'ontologies à partir de textes (Terminae[1] [2], [3], Text2Onto [14]). Ces méthodes sont implémentées dans des outils qui se distinguent par leur approche de la phase de conceptualisation plus ou moins automatique [20]. Cependant s'il existe des outils largement utilisés, tels que Protégé, pour représenter formellement une ontologie supposée déjà conçue, et s'il existe également plusieurs plateformes de traitement automatique de la langue (TAL) permettant d'analyser automatiquement les corpus et de les an-

[1] http://www-lipn.univ-paris13.fr/~szulman/logi/index.html

noter tant du point de vue syntaxique que statistique, il n'existe actuellement aucune procédure généralement acceptée, ni a fortiori aucun ensemble cohérent d'outils supports, permettant de concevoir de façon progressive, explicite et traçable une ontologie de domaine à partir d'un ensemble de ressources informationnelle relevant de ce domaine.

Le but de ce papier de présenter les propositions développées, au sein du projet ANR DaFOE 4app, pour favoriser l'émergence d'un tel ensemble d'outils. Compte tenu de la diversité des besoins, des méthodes et des modèles utilisés par les différents auteurs, il est vite apparu que cet ensemble ne pouvait être fondé ni sur une méthode unique, ni sur une structure de modélisation statique. Les propositions présentées ont donc visées à atteindre les trois objectifs suivants :

1. Définir un cadre méthodologique général susceptible d'intégrer la réalisation des scénarios de conception très divers correspondant à l'ensemble des besoins identifiés.

2. Définir une structure de modélisation permettant de s'adapter à des modèles différents.

3. Spécifier puis développer une plateforme capable d'accueillir et d'intégrer les différentes catégories d'outils actuellement utilisés de façon autonomes au sein d'une structure de modélisation unique, et permettant d'aller de l'analyse d'un corpus, supposé annoté par une plateforme de TAL, à la définition d'une ontologie formelle du domaine.

4. Garantir, au sein de cette plateforme, la persistance, la traçabilité et le dimensionnement des modèles (plusieurs milliers de concepts).

Le plan du papier est le suivant. La section 2 présente l'état de l'art dans le domaine des outils de conception d'ontologie et identifie les besoins à satisfaire. La section 3 présente un ensemble de scénarios tests devant pouvoir être assistés par la plateforme et elle propose un cadre méthodologique recouvrant les différents cheminements possibles pour concevoir une ontologie. La section 4 propose une formalisation des étapes par lesquelles peut passer une démarche de conception d'ontologie. Elle spécifie, pour chaque étape, la structure de modélisation proposée. La section 5 présente les mécanismes définis pour supporter la diversité: architecture de meta-modélisation de type MOF pour la représentation des données, structure à greffons (*plug-ins*) pour l'adaptabilité fonctionnelle. Enfin nous décrivons, en conclusion, l'apport d'une telle approche par rapport à l'existant et nous invitons tous les membres de la communauté à collaborer avec le projet pour intégrer dans la plateforme les outils existants.

2. ÉTAT DE L'ART

L'état de l'art décrit ici a été conduit dans le cadre de l'étude des besoins de la plateforme DaFOE pour expliciter les besoins des utilisateurs au regard des outils existants. Nous avons analysé trois types d'outils (les logiciels de traitement automatique des langues (TAL), les plateformes de construction d'ontologies à partir de texte et les éditeurs d'ontologies). Les problèmes d'intégration et d'interopérabilité entre outils sont rapidement apparu, ce qui justifiait le projet de création d'une nouvelle plateforme.

2.1 Les logiciels de traitement automatique des langues

Les logiciels de TAL jouent permettent d'extraire des textes des éléments de connaissances à représenter dans une ontologie. Les expériences en la matière sont très diverses et nous ne faisons pas ici un inventaire des outils existants[2]. Nous analysons plutôt ce que les fonctionnalités les plus importantes[3] peuvent apporter à la construction d'ontologies et nous justifions les objectifs de DaFOE

2.1.1 Extraction de candidats termes

L'analyse terminologique d'un corpus permet d'identifier des syntagmes qui semblent avoir un fonctionnement terminologique, c'est-à-dire qui relèvent d'un vocabulaire spécialisé doté d'une sémantique relativement stable et consensuelle au sein d'une situation de communication déterminée. Dans le processus de construction d'ontologie, l'extraction terminologique permet de repérer les concepts qui sont mentionnés par le corpus. Elle sert donc au repérage du vocabulaire conceptuel.

Les outils d'extraction existants sont nombreux et variés. Dans la mesure où l'on manque de recul et de méthodes pour évaluer et comparer leurs approches, nous ne faisons pas le choix exclusif d'un extracteur contre un autre : nous prévoyons de pouvoir intégrer dans la plateforme DaFOE des résultats de différents extracteurs.

2.1.2 Pondération des candidats termes

Une fois recensés les candidats termes du corpus, il est important de leur associer un poids et de les trier. Cela permet de focaliser le travail de construction d'ontologie sur un petit nombre de termes et de commencer par les éléments les plus signifiants. En effet, l'extraction produit généralement des listes très importantes de candidats termes (jusqu'à quelques dizaines de milliers de candidats pour un corpus de 100 000 mots).

Certains extracteurs intègrent la pondération dans leur résultat mais il paraît cependant plus pertinent de dissocier les deux fonctionnalités dans DaFOE par souci de modularité. Cela permet de combiner différents outils de TAL et de choisir les critères de saillance en fonction des applications visées.

2.1.3 Validation des éléments terminologiques

La pondération des termes peut servir à préparer la validation des termes par un expert mais il doit néanmoins retravailler manuellement les résultats. Dans certains cas, on se contente d'explorer les résultats sans passer par une phase systématique de validation. Comme ces tâches sont longues et subjectives, il est important de penser la présentation des résultats, les fonctionnalités et l'ergonomie des interfaces de manière à accélérer le travail de validation, faciliter la navigation et en assurer la cohérence. DaFOE devra tenir compte des possibilités que suscitent les interfaces existantes comme le traitement des candidats termes à travers la navigation *via* les têtes ou les expansions que permet un outil comme Termonto.

[2]Le lecteur pourra se référer aux ouvrages de Maedche [19], Buitelaar *et al.* [11] et Cimiano [13]

[3]A noter que le problème important de l'extraction des entités nommées n'est pas une priorité pour DaFOE qui met l'accent sur la construction d'ontologie plus que sur son peuplement, même si des extensions sont envisagées.

2.1.4 Normalisation des termes

Les termes étant des entités textuelles, ils subissent différentes modifications de surface qui nuisent à leur repérage en corpus et à leur visualisation. Il est donc important, ne serait-ce que pour mesurer leur fréquence et pour les visualiser, de pouvoir regrouper les différentes formes d'un même terme sous une forme canonique commune (une sorte de "lemme" de terme). Cette normalisation n'est pas toujours prise en charge par les extracteurs et elle peut l'être de différentes manières. Dans DaFOE, nous ne faisons pas de présupposition sur la nature de la normalisation mais nous prévoyons de différencier la forme canonique de ses variantes.

2.1.5 Extraction de relations conceptuelles

Les textes expriment également des informations sur les relations sémantiques que les termes entretiennent en eux. Si l'on considère que les termes représentent les concepts de l'ontologie à construire, les relations sémantiques qu'ils entretiennent peuvent être considérées comme le reflet de relations conceptuelles. Repérer ces relations aide à structurer l'ontologie. L'extraction des relations terminologiques est une tâche complexe du fait de la diversité des relations sémantiques à prendre en compte et de la diversité des méthodes mises en œuvre. La mise en relation des termes est une des étapes sur laquelle il est prévu de travailler dans le cadre de DaFOE. L'idée est de construire un greffon qui permette d'appeler des méthodes classiques à partir de règles contextuelles d'extraction – *e.g.* "patrons" qui sont la plupart du temps spécifiques des méthodes.

2.1.6 Regroupement de classes sémantiques

De nombreuses recherches visent à construire des classes sémantiques de mots à partir de l'analyse de la distribution des mots en corpus. L'approche est d'autant plus fructueuse que le corpus de départ est plus homogène. Les méthodes diffèrent par la définition du contexte sur lequel elles s'appuient (simple fenêtre de mot, contexte syntaxique, etc.), par la mesure de similarité qu'elles prennent en compte et par la démarche de classification qu'elles adoptent (supervisée ou non, nombre de classes visés, etc.). Les approches numériques et symboliques étant complémentaires, le défi consiste à déterminer comment tirer profit de cette complémentarité dans le processus très concret de construction d'ontologies (méthode et outils) tel que le proposera la plateforme DaFOE.

2.1.7 Gestion du multilinguisme

Il est important de prendre en compte le multilinguisme dans la construction d'ontologies. Deux approches sont possibles : 1) faire le même travail de construction d'ontologies en parallèle en prenant appui sur des corpus relevant du même domaine dans les différentes langues cibles puis intégrer les ontologies résultantes ou 2) partir de corpus parallèles, créer une première ontologie à partir du sous-corpus d'une langue et trouver des correspondances traductionnelles des termes associés aux concepts de la première ontologie pour enrichir le niveau lexical (on parle alors *localisation*). Dans les deux cas, c'est une tâche lourde et difficile. Dans un premier temps, nous retenons la première solution.

2.2 Les plateformes de construction d'ontologies à partir de texte

Différents systèmes visent à permettre la construction d'ontologies à partir de textes.

Text2Onto[4] se présente comme une plateforme de construction d'ontologies à partir de textes. Il repose sur l'architecture GATE[5] pour pré-traiter les textes et extraire des concepts (en réalité des termes), des relations entre concepts et des instances de concepts. Pour ces opérations, on peut faire appel à différents outils et combiner leurs résultats. En tant que boîte à outils, Text2Onto se rapproche des objectifs de DaFOE mais la construction d'ontologies est vue comme un processus automatique et le travail de conceptualisation (choix et organisation des concepts, de leurs propriétés et de leurs relations) ne sont pas guidées. D'où la nécessité de coupler un autre système [20] pour affiner l'ontologie produite.

A l'opposé des systèmes comme ASIUM [16] ou OntoGen[6] proposent une approche incrémentale et interactive. Ils reposent tous les deux sur un processus de classification supervisé. Dans ASIUM, l'approche repose sur l'analyse des distributions des mots du corpus : les classes de mots obtenues sont des ébauches de concepts que l'utilisateur peut retoucher avant de poursuivre le processus de classification ascendant. L'approche descendante d'OntoGen part au contraire des concepts les plus généraux pour construire une ontologie thématique destinée à l'indexation documentaire. Dans les deux cas, la méthode de construction d'ontologie, bien que semi-automatique, est figée. La conception de la plateforme DaFOE repose au contraire sur l'idée qu'il n'existe pas approche unique et générale pour la construction d'ontologies : la section suivante montre une grande diversité des pratiques, liée à la diversité des applications et des ressources disponibles.

D'autres outils occupent une position intermédiaire. Comme Text2Onto, Terminae [2] intègre des outils de TAL qui sont interchangeables. Comme ASIUM et OntoGen, il repose sur une conception interactive de la construction d'ontologies. Le projet DaFOE s'est inspiré de l'expérience antérieure de Terminae, mais l'objectif est à la fois d'élargir la palette d'outils de TAL exploitables et de supporter des ontologies de taille plus importante. De la même manière l'expérience de ITM[7] pour la construction de référentiels métier a guidé celle de DAFOe mais l'objectif est de pouvoir construire une gamme plus riche d'ontologies.

Le développement de la plateforme DaFOE répond donc à des objectifs plus ambieux en termes d'interopérabilité des outils de TAL et d'extraction de connaissances, de robustesse et d'ouverture (on doit pouvoir tester différents scénarions de construction) que les outils existants.

2.3 Les éditeurs d'ontologies

Au-delà des outils de construction d'ontologies à proprement parler, il existe d'assez nombreux éditeurs d'ontologie qui prennent en charge la partie formelle du processus de construction d'ontologies mais en supposant d'avoir déjà identifié les connaissances à modéliser. Les éditeurs utilisent un brouillon pour mettre au point les concepts et relations. Ils permettent de structurer un modèle et n'ont, pour la plupart, qu'une sortie possible à un niveau de structuration

[4] http://ontoware.org/projects/text2onto/
[5] http://gate.ac.uk/
[6] http://ontogen.ijs.si/
[7] http://www.mondeca.com/index.php/fr/intelligent_topic_manager

élevé, proche de l'ontologie formalisée.

Les principaux éditeurs testés sont SWOOP[8], pOWL[9], PLIBEditor[10], OntoEdit[11], Protégé-OWL[12], DOE[13], Hozo[14], WebODE[15], TERMINAE, KAON[16]. Les critères selon lesquels ont été évalués ces outils sont, en désordre et résumé, ceux de l'architecture, l'interopérabilité entre outils ou entre ontologies, la méthodologie sous-jacente, les critères relatifs aux inférences, la visualisation du modèle, les sources possibles et outils pour les intégrer, la caractérisation des modèles possibles / produits, les modules de réutilisation/adaptation de BDs, ...

2.4 Bilan

Le bilan de cet état de l'art couplé à une étude ergonomique non décrite ici [12], a permis d'expliciter les spécificités attendues de la plateforme DaFOE. Nous citons principalement :

- considérer les outils de TAL en dehors de DaFOE;
- favoriser leur utilisation sous forme de greffons via des interfaces ;
- Pouvoir gérer de gros volumes de données (un corpus a couramment une taille de 600 000 mots) ;
- Pouvoir gérer des thésaurus, ontologies et autres ressources utiles dans la plateforme tout en considérant qu'il n'y en a qu'une seule qui est construite (pas d'algorithmes d'alignement mis en œuvre) ;
- Pouvoir gérer des instances des classes (occurrences des concepts vues au niveau formel) ;
- Combiner réutilisation et modélisation à partir de textes ;
- Permettre de caractériser les genres et différences de concepts au sein d'une ontologie différentielle (car hiérarchisés en fonction des principes différentiels nous précisant les différences et similitudes de chaque concept avec les concepts pères et frères);
- Assurer la traçabilité des textes aux modèles et inversement;
- Avoir différentes couches de modélisation ;
- Pourvoir éditer une ontologie formelle ("à la PROTÉGÉ ") ;
- Pouvoir exploiter des extracteurs de termes comme SYNTEX ou YATEA [1] et extracteurs de relations comme CAMELEON [9, 22] ;
- Construire une plateforme flexible ;
- ...

Ces spécificités sont reprises pour justifier le modèle de données et l'architecture de la plateforme (*cf.* § 4 et 5).

3. DÉFINIR DES SCÉNARIOS DE CONSTRUCTION D'ONTOLOGIES

Afin de décrire au mieux les besoins et usages des futurs utilisateurs de la plateforme, nous avons défini des scénarios de construction d'ontologie. Les 8 partenaires impliqués dans le projet DaFOE 4app, ont alors chacun proposé un scénario de construction d'ontologie rendant compte d'une pratique qui est la leur. Autant que faire se peut, les scénarios ont été décrits comme si l'utilisateur se servait de la plateforme DaFOE. Nous ne présentons ici que les grandes lignes des scénarios qui permettent d'expliciter les principales fonctionnalités attendues de la plateforme.

3.1 Scénarios à partir de corpus uniquement textuel

Scénario Terminae.

Ce premier scénario décrit la construction d'une ontologie formelle suivant la méthode TERMINAE enrichie. Cette méthode permet de concevoir une ontologie à partir d'un corpus textuel.

La première étape consiste à construire le corpus pour l'analyser à l'aide de la plateforme linguistique disponible dans l'outil TERMINAE. Après annotation du corpus, la plateforme linguistique fournit une liste de candidats termes parmi lesquels l'utilisateur choisira les termes à conceptualiser pour l'élaboration de son ontologie. Dans un second temps, on définit les concepts terminologiques et les relations termino-ontologiques liant ces termino-concepts. Les concepts terminologiques sont complétés par des informations lexicales (définition, synonymes...). L'étape suivante correspond à la description des termino-concepts dans un langage semi-formel à l'aide des informations lexicales issues de l'étape précédente. Dans le cadre de cette semi-formalisation, on précise les différences et similitudes de chaque concept avec les concepts pères et frères. On passe ensuite à un niveau formel en organisant les concepts dans une ontologie que l'on aligne sur une ontologie de haut niveau.

Scénario Relations.

Dans ce scénario, la problématique de repérer les candidats termes se complique par un repérage des relations.

La première étape consiste à identifier les sources de connaissances en fonction de leur niveau de structuration et conceptualisation (corpus, thésaurus, ontologie, ...). La deuxième étape est le traitement préliminaire du corpus grâce à des outils de TAL dont les résultats sont récupérables dans la DaFOE grâce à des API. La 3e étape est le dépouillement des résultats des outils de TAL en particulier, l'examen des relations lexicales rendues par un extracteur de relations lexicales.

Dans ce scénario appliqué à la construction d'une terminologie dans le domaine de l'archéologie, il est indispensable de pouvoir d'abord traiter le plus possible tout ce qui vient des textes, puis "nettoyer" selon des critères de différenciation. L'idéal étant de pouvoir passer facilement d'une tâche à l'autre sans préjuger de l'ordre dans lequel on les réalise

[8] http://www.mindswap.org/2004/SWOOP/
[9] http://aksw.informatik.uni-leipzig.de/Projects/Powl
[10] http://www.plib.ensma.fr/
[11] http://www.ontoprise.de/content/e1276/index_eng.html
[12] http://protege.stanford.edu
[13] http://homepages.cwi.nl/~troncy/DOE/
[14] http://www.hozo.jp/
[15] http://webonto.open.ac.uk/
[16] http://kaon2.semanticweb.org/

et d'avoir des modèles vides que l'ont rempli en fonction du travail qu'on a envie de faire.
Par rapport aux relations, DaFOE doit permettre de faire des choix sur les types de relations qui nous intéressent et sur les patrons qui vont permettre de les découvrir. DaFOE doit aussi permettre d'afficher l'ensemble des phrases du corpus retournées par la projection de tous les patrons associés à ce type de relation.

Scénario pneumologie.

Il s'agit dans ce scénario de construire une ontologie régionale de la pneumologie [6] à partir de ressources textuelles et en utilisant la méthodologie ARCHONTE [5].
Avant toute chose, il est nécessaire d'élaborer le corpus de référence. Ce corpus est ensuite traité par l'analyseur syntaxique SYNTEX-UPRERY [10] [8]. Le module SYNTEX extrait du corpus des syntagmes nominaux et construit un réseau de mots et syntagmes et analysant chaque phrase du corpus. Le module UPERY propose ensuite une analyse distributionnelle sur le réseau fournit par SYNTEX. On utilise alors TERMONTO qui est une interface d'accès aux résultats de l'outil SYNTEX-UPRERY. TERMONTO propose à l'issu de l'analyse syntaxique une liste de candidats termes. On sélectionne, parmi ces derniers, les termes à conceptualiser pour construire notre ontologie. Les termes choisis sont, dans une prochaine étape, organisés en concepts termino-ontologiques hiérarchisés entre eux à l'aide des principes de similarité et dissimilarité vis-à-vis des concepts pères et frères comme le préconise la méthodologie ARCHONTE. Dans un même temps, on définit des relations termino-conceptuelle liant ces termino-concepts entre eux. Enfin, la dernière étape consiste à utiliser l'éditeur PROTÉGÉ afin de passer à un niveau formel.

Scénario licence de contenu audiovisuel.

Dans ce scénario, on élabore une ontologie nous permettant d'exprimer de façon générique des licences représentatives des intentions des détenteurs de contenus [21].

Là encore, la première phase est la constitution du corpus dont on partira pour dégager nos premiers termino-concepts. En revanche, dans cet exemple on ne procède pas à une analyse automatique ou semi-automatique du corpus. Les candidats termes sont sélectionnés manuellement. Le corpus étant une succession de dictionnaires de données juridiques non uniformes, il est, en effet, difficile de le traiter autrement que manuellement.

Afin de pouvoir vérifier la pertinence du choix de nos termino-concepts, on commence à les hierarchiser en se basant sur la méthodologie ARCHONTE. On a donc des concepts termino-ontologiques, organisés dans une ontologie différentielle. Dans un même temps on complète les termino-concepts par des informations lexicales (définition encyclopédique, synonymes, et labels dans d'autres langues). On définit également les relations termino-conceptuelles liant ces concepts termino-ontologiques.

Pour finir, on formalise notre ontologie à l'aide de l'éditeur PROTÉGÉ. C'est durant cette dernière étape que l'on précise les éventuelles contraintes à associer à chaque concept (conjonction, disjonction, union...) et à chaque relation ontologique (transitivité, symétrie).

3.2 Scénario à partir de corpus de toute structuration

Scénario patrimoine.

Ce scénario a pour particularité de reposer sur une demande de marché. Il s'agit pour un industriel de mettre en œuvre une ontologie dédiée à la décoration de maison. Suite à cette modélisation, les clients désirent disposer de différents services tels qu'un moteur de recherche enrichi, du fait de l'ontologie, d'un outil permettant de classifier des images et vidéos en utilisant le vocabulaire de l'ontologie, et de voir ce travail réalisé en multilingue (français et anglais). Le demandeur développant une série de portails web thématique, c'est à partir de ces sites que sera élaboré le corpus documentaire. Ce corpus est ensuite traité par des outils de TAL qui retournent une liste de candidats termes. Si cette liste ne semble pas complète à un expert du domaine, il est alors possible de compléter le corpus précédent et de recommencer l'étape de traitement. Le modélisateur choisi parmi les candidats termes, ceux qu'il désire conceptualiser. Durant cette étape, il peut également relier différents termes par une relation de synonymie.

L'étape suivant consiste à manipuler les concepts terminologiques issus de l'étape précédente, à savoir, les modifier, les supprimer, ou compléter ces derniers à l'aide d'attributs (langue, libellé, nom abrégé, identifiant...). On note que la structure terminologique est la même quelle que soit la langue dans laquelle on modélise l'ontologie. Pour un même concept dans des langues différentes, on changera les attributs (langue, libellé...). Dans un même temps, l'ontologue définit les relations termino-conceptuelles liant les concepts termino-ontologiques.

On passe ensuite à un niveau formel. La gestion des concepts à ce niveau peut être subdivisées selon que les concepts formels représentent un attribut, une association ou une classe de concept. On génère dans ce même temps les relations entre concepts formels.

Scénario de création de deux ontologies pour le traitement des images satellitaires.

Dans le cadre de ce scénario, on essaie d'automatiser l'indexation d'images satellitaires dont la production dépasse de beaucoup les capacités en temps des photo-interprètes. Le contenu d'une image satellitaire peut-être analysé en fonction a) de caractéristiques de couleurs, textures et géométrie et b) du contenu sémantique qui peut être abstrait des précédentes caractéristiques.
Un tel scénario nécessite de construire 2 ontologies, 1 ontologie des objets des scènes observées au travers des images satellitaires et 1 ontologie dite "graphique" issue des traitements appliqués à l'image (en particulier calcul des attributs couleur/forme/texture et classification), mise en relation avec l'ontologie de la scène par apprentissage. À partir de là, on pourra exploiter un moteur de raisonnement dans un processus d'interrogation interactif de la base d'images satellitaires.
Une première étape est d'identifier les ressources terminologiques concernant la description des scènes vues par satellite (thesaurus, pages html, terminologies ou ontologies existantes). Un seconde étape consiste à identifier les ressources concernant les outils de traitement d'images (à partir d'ontologies existant dans la littérature) – que la plateforme DaFOE doit permettre de récupérer. La troisième étape est l'organisation en concepts primaires / secondaires et l'identification des relations majeures (notamment de hiérarchie) à partir d'une analyse des textes identi-

fiés ou d'ontologies existantes.
La plateforme DaFOE doit permettre de regrouper les termes associés à un même concept (soit par méthodes automatiques de clusterisation, soit en donnant accès à des outils d'édition manuelle) et faire émerger des caractéristiques sur ces regroupements afin de faciliter l'exploration d'autres concepts en relation avec le dernier défini. La plateforme DaFOE doit aussi permettre le peuplement des ontologies construites, en particulier celle des objets ou scènes observées.

3.3 Scénarios à partir de données structurées

Scénario de création d'une ontologie d'un schéma de bases de données.

Le développement d'une ontologie à partir d'un schéma de base de données (BD) peut avoir plusieurs objectifs parmi lesquels on peut citer 1) réutiliser la connaissance représentée dans ce schéma ou 2) concevoir l'ontologie locale susceptible de définir, ensuite, le contenu d'une base de données préexistante. Dans un tel contexte, on suppose que le schéma de la BD est lisible, par exemple traduit en OWL. Une fonctionnalité indispensable est aussi de lire un schéma SQL, que ce soit nativement ou par un greffon.

À partir de là, on a besoin de générer l'ontologie (cas 2) et de supposer des représentations : *a*) une table est représentée comme une classe primitive ; *b*) un attribut est représenté comme une propriété ; *c*) une association est représentée comme une propriété ; *d*) une vue est représentée comme une classe définie ; *e*) les contraintes de clés sont représentées, de même que les dépendances fonctionnelles, et les cardinalités de clefs étrangères.

Certaines modifications d'interprétations peuvent être demandées par l'utilisateur, p. ex. l'interprétation d'une table non comme une classe primitive, mais comme un type défini par énumération. De plus, un certain nombre d'opérations doivent être possible pour modifier ou documenter l'ontologie. Ces opérations portent principalement sur les créations, suppressions, déplacement de classes, les créations/suppressions de propriétés. Par ailleurs, des opérations spécifiques doivent être possibles : *a*) absorption d'une classe par une classe qui la référence. Toutes les propriétés de la première sont redéfinies avec la seconde comme domaine ("dé-normalisation"); *b*) modification de la hiérarchie, soit qu'une classe soit découpée en une hiérarchie de subsomption avec répartition des propriétés, soit qu'une partie de hiérarchie soit "mise à plat" avec transfert des propriétés sur la classe résultante.

Ces opérations doivent pouvoir être complétées par un greffon en fonction de besoins spécifiques.

Scénario de maintenance d'une ontologie formelle.

Ce scénario décrit une possibilité d'usage de la plateforme pour la gestion de l'évolution d'une ontologie en cours de construction. Le processus de construction défini à travers les couches du modèle de données (*cf.* § 4) peut être conduit de manière itérative et incrémentale mais les constructions intermédiaires ne sont pas suffisamment formelles pour que la gestion de leur évolution puisse être assurée selon des procédés prédéfinis (des scénarios de modifications, des types de changements, ...) et de manière automatique et contrôlée (validation automatique avec analyse et résolutions des impacts de changements). Ainsi, ce n'est qu'au niveau ontologique/formel du modèle de données qu'il est possible de proposer une approche de gestion d'évolution qui part d'un changement formellement spécifié, l'analyse pour étudier ses caractéristiques et ses impacts sur l'ontologie et l'applique de manière automatisée (minimisant l'intervention de l'ingénieur), tout en gardant la traçabilité des évolutions. La maintenance se fait en 4 étapes :
1) spécification d'un changement : cela nécessite une fonction [enrichir/modifier ontologie] permettant de déclencher le début d'une opération de changement sur l'ontologie formelle; 2) spécification formelle du changement : le système définit la spécification formelle du changement : son type (ajout, modification, suppression, ...) ; le type des entités concernées (classes, propriétés, instances), etc; 3) vérification de la consistance : le changement n'est pas enregistré, il est appliqué de manière temporaire à une version test de l'ontologie. Si des inconsistances sont détectées, le système les signale et donne la main à l'utilisateur s'il veut modifier la spécification de son changement, dans ce cas on revient à étape 1; 4) confirmation du changement et application. Les opérations de mise à jour sont enregistrées dans l'historique d'évolution.

3.4 Construction d'un cadre méthodologique

À partir de ces scénarios, il était évident que les attentes des partenaires étaient diverses et ne pouvaient se satisfaire de la mise au point d'un méthode de construction d'ontologie unique. Un cadre méthodologique a été élaboré durant la définition de la plateforme. Il a été utilisé d'une façon cyclique, à savoir comme cadre permettant d'avoir une description commune des processus mis en jeu en même temps que modèle évoluant pour être à même de ternir compte des desiderata de tous exprimés entre autres via les scénarios. La figure 1 décrit ce cadre. On y voit apparaître les différents niveaux d'entrée dans la plateforme et les processus mis en jeu entre ces niveaux : en vert, le niveau du corpus (0), en violet, les niveaux correspondant à des produits de plus en plus élaborés (1) des réseaux terminologiques s'organisant durant l'analyse des données ("Réseaux termino-ontologiques" et "Modèle conceptuel"), (2) un niveau termino-ontologique où les concepts sont organisés ("Ontologie") et (3) un niveau où l'ontologie est formalisée ("Ontologie formelle").

Ce cadre méthodologique, présenté sous la forme où il a été élaboré, est la source de la définition de la plateforme, que ce soit le modèle de données qui précisera et explicitera un modèle à 4 couches (*cf.* § 4) ou l'architecture qui prendra en compte des fonctionnalités extensibles par greffon (*cf.* § 5). Enfin, les scénarios décrits ci-dessus seront une partie des critères de recettage de la plateforme qui devra effectivement permettre de les mettre en œuvre.

4. MODÈLE DE DONNÉES

Le modèle de données de la plateforme DaFOE permet de prendre en compte l'interopérabilité d'outils hétérogènes. Il intègre l'ensemble des informations nécesaires à la conception d'une ontologie. Il prend en compte plusieurs aspects :

- l'architecture ouverte permettant l'ajout de greffons;
- l'utilisation de la plateforme à partir d'un niveau quelconque ;
- la production par la plateforme de différentes

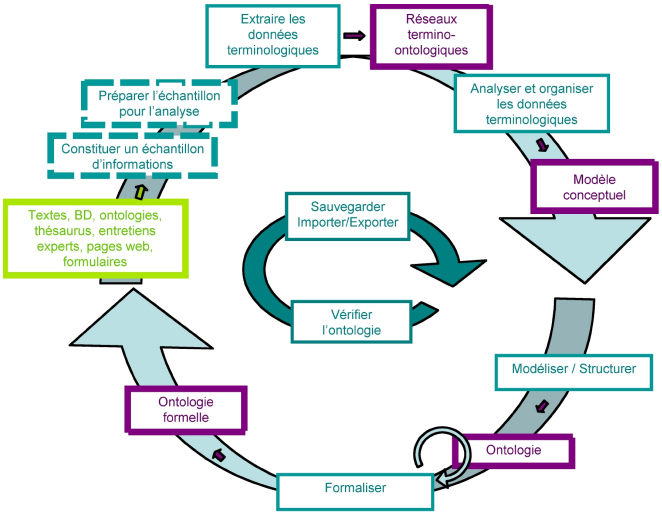

Figure 1: Cadre méthodologique de l'élaboration d'une ontologie.

ressources (ressources termino-ontologiques ou ontologies formelles).

Le modèle de données suit le cadre méthodologique. Il se décompose en quatre couches (voir figure 2).

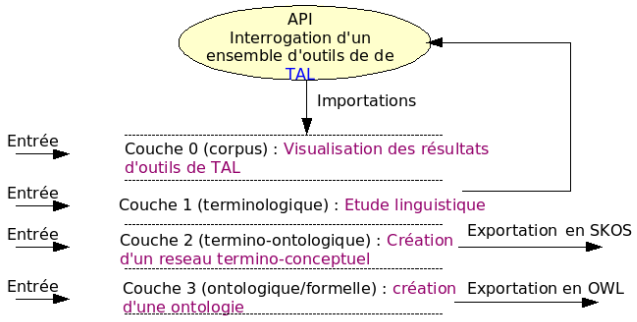

Figure 2: Structure en couches du modèle de données de DaFOE

Pour illustrer les différentes notions décrites ci-dessous, nous nous appuyons sur la construction d'une ontologie destinée à être exploitée par des sites internet du domaine de l'ameublement.

4.1 Couche corpus (0) ou couche textuelle

Cette couche est utile pour un utilisateur de la plateforme qui travaille à partir de textes. Celui-ci doit constituer un corpus textuel brut à partir de ses différents textes. S'il possède des outils de TAL, il pourra visualiser les résultats de ses outils dans DaFOE. Sinon le texte brut est découpé en phrases qui sont repérées chacune par un identifiant unique. Si le corpus d'entrée est déjà segmenté par un outil de TAL, la segmentation proposée par l'outil est conservée et un lien est fait entre les identifiants fournis par DaFOE et ceux fournis par l'outil. Le modèle de données prévu pour le noyau prend en compte les informations linguistiques classiques décrites au paragraphe suivant. La plateforme fournira des outils permettant de nettoyer les résultats des outils de TAL.

4.2 Couche terminologique (1)

Cette couche du modèle de données sert à l'analyse linguistique. Celle-ci consiste à identifier les termes parmi les candidats termes (CT) (Exemples de CT: *table*, *table basse*, *table de nuit*, *table de chevet*, *table des matières*, *table de cuisson*) et les relations lexicales qui sont des relations entre termes (Exemple: chaque table appartient à une gamme de meubles). Nous nous intéressons également aux relations de composition syntaxique qui sont également des relations entre termes, considérées d'un point de vue syntaxique comme dans le syntagme nominal (*table de cuisson* où *table* est désignée comme la tête du syntagme et *cuisson* le modifieur).

4.2.1 Modèle de données pour les CT/termes

Un candidat-terme ou un terme peut être décrit par : *a*) des propriétés morpho-syntaxiques (genre, nombre, un attribut indiquant si c'est une entité nommée comme *Louis XV*) ; *b*) une forme canonique (éventuellement choisie arbitrairement à défaut d'autres critères) ; *c*) les variantes du CT/terme comme *rangement, ranger* ; *d*) les identifiants des phrases dans lesquelles apparaisse le CT/terme ; *e*) la langue qui est par défaut celle du corpus ; *f*) des critères de saillance du terme comme la fréquence du terme dans le corpus ; *g*) un attribut de validation.

Les critères de saillance du CT/terme sont enregistrés dans un vecteur. Ils sont calculés par des requêtes spécifiques faites sur le corpus annoté ou par des résultats d'outils de TAL. L'attribut de validation permet de distinguer les termes parmi les candidats-termes. Ainsi parmi les CTs présentés ci-dessus, nous éliminons le CT *table des matières* qui apparaît dans le corpus mais n'appartient pas au domaine étudié.

4.2.2 Modèle de données pour les relations lexicales

Parmi les relations lexicales, les relations sémantiques (synonymie, hyperonymie, composé/composant, antonymie, les relations propres au domaine) sont privilégiées car elles permettent de trouver les relations conceptuelles, c'est-à-dire des relations entre concepts. Ces relations lexicales peuvent être déterminées *a*) en utilisant des règles d'extraction utilisant des patrons, *b*) en exploitant la structure syntaxique des termes, enfin *c*) en utilisant des indices statistiques (termes co-occurents).

Le modèle de données permet de recueillir des patrons et des relations lexicales décrites par l'utilisateur ou un ensemble de relations de composition syntaxique fournies par un outil externe comme l'outil Syntex/Upery [8].

Cette étude linguistique est la première étape d'un processus de conceptualisation appelé normalisation [4]. La seconde étape consiste à chercher le sens des termes afin de construire des concepts en tenant compte de l'objectif pour lequel la ressource ontologique est construite. Elle est décrite dans la couche termino-ontologique.

4.3 Couche termino-ontologique (2)

Le niveau termino-ontologique permet de représenter les termes désambiguïsés appelés concepts termino-ontologiques (CTO), et les relations désambiguïsées qui sont des relations termino-ontologiques (RTO) entre concepts termino-ontologiques. C'est à ce niveau que le multilinguisme est traité.

4.3.1 Concepts termino-ontologiques

On représente un sens de terme par un CTO. Ainsi le terme *table* a deux sens, soit une table comme meuble, soit un appareil électroménager. Les différents synonymes sont représentés par le même CTO. Les termes *table de chevet* et *table de nuit* sont synonymes. Ils sont décrits par le même CTO identifié par le terme vedette *table de chevet*.
En cas d'ambiguïté au niveau terminologique, tout ou partie des occurrences du niveau terminologique sont référencées par chacun des CTOs associés à ce terme. Un CTO peut être désigné par plusieurs termes (ses synonymes) et un terme peut donner naissance à plusieurs CTOs.
Le CTO est l'élément de base permettant de passer du terme au concept, plus généralement du linguistique au conceptuel (formel). La figure 3 présente les deux parties d'un CTO. Le

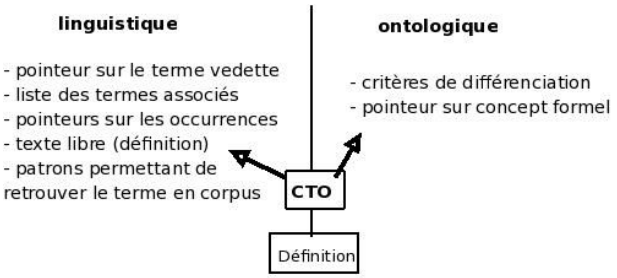

Figure 3: Description d'un concept termino-ontologique (CTO) : à gauche, la partie linguistique, à droite, la partie conceptuelle.

modèle de données reprend une partie du modèle de données de la couche terminologique (partie linguistique). Une partie ontologique lui est associée : *a*) les critères de différenciation (4 champs texte précisant la différence, la similitude avec le CTO père, la différence et la similitude avec les CTO frères) ; *b*) un pointeur sur le concept formel lorsqu'il sera créé.

4.3.2 Les relations termino-ontologiques

Une RTO est définie entre deux CTOs. La figure 4 présente le modèle de données qui lui est associé. Ces RTO

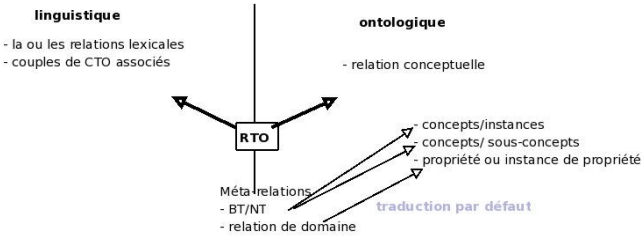

Figure 4: Description d'une relation termino-ontologique (RTO).

pourront être typées en utilisant des méta-relations usuelles des thésaurus (exprimées en anglais usuellement) :

- *Broader Than/ Narrower Than* (BT/NT) ;
- *Generic/ Specific* ;
- *Related Term/ Related Term* ;
- *Compounds* (partie/tout) ;
- relations de domaine.

Ces méta-relations permettent de catégoriser les RTOs. Une traduction par défaut au niveau ontologique est définie :

- BT/NT seront traduites par défaut comme des relations *classe (concept)/instance* ;
- *Generic/ Specific* seront traduites par défaut par une relation *classe (concept)/sous-classe(sous-concept)*.
- *Related Term/ Related Term*, *Compounds*, relations de domaine seront éventuellement traduites par une propriété, ou une instance d'une propriété.

Ainsi le terme *table* (meuble de salon) est en relation d'hyperonymie avec le terme *table de chevet*. Les RTOs permettent de préciser une sémantique permettant de faire des inférences au niveau ontologique (transitivité, relations fonctionnelles). Les CTOs et RTOs assurent la traçabilité entre les niveaux linguistiques et ontologiques. Ils permettent de passer des termes ou relations lexicales en corpus aux concepts et relations conceptuelles dans l'ontologie.

4.3.3 Le multilinguisme

Parmi les 2 deux choix possibles (*cf.* § 2.1.7), nous avons choisi la localisation qui consiste à présenter une même ressource ontologique dans différentes langues. Ainsi, pour chaque langue, un terme et les éléments linguistiques (synonymes, définition en langage naturel, ect..) dépendant de la langue sont associés au CTO.

4.4 Couche ontologique (3)

Le modèle de données permet d'accueillir les éléments d'un langage d'expression de l'ontologie au moins équivalent à celui défini dans OWL-DL. Il contient des concepts et des propriétés, des instances de concepts et des instances de propriétés et des valeurs (DataTypes). Dans une première version, les types les plus courants (*string, integer, float, boolean, date* (version simple)) seront définis. Un concept est défini ou primitif. Un concept formel ou l'un de ses attributs contient un lien sur le CTO s'il existe. Ainsi un concept dénoté par *table_meuble* est créé dans l'ontologie correspondant à un des sens du terme *table*. Il est décrit par plusieurs relations conceptuelles comme *appartientGamme* à valeur dans le concept *gammeMeuble* décrit par un style, etc. Le concept *table_chevet* est créé en relation de subsomption avec le concept *table_meuble*. Il hérite de toutes les propriétés du concept père et une propriété doit le distinguer de ce dernier comme la pièce où le meuble est utilisé (*chambre*).

5. SUPPORT DE LA DIVERSITÉ ET ARCHITECTURE DE MÉTA-MODÉLISATION

Comme nous l'avons vu dans les sections précédentes, l'architecture de la plateforme doit faire face à trois exigences : (1) supporter la représentation des formalisations à la fois terminologique (FT), termino-ontologique (FTO) et ontologique (FO) ; (2) permettre la modification des modèles de représentation utilisables dans les différentes formalisations, tant au niveau modèle qu'au niveau instance ; (3) permettre l'adaptation de la plateforme à des approches diverses mettant en œuvre des traitements différents. Nous

décrivons dans cette section, d'une part, l'architecture de méta-modélisation que nous avons définie pour faire face aux deux premières exigences, ainsi que le prototype de validation que nous avons réalisé, et, d'autre part, les interfaces de greffons définies pour assurer la flexibilité fonctionnelle de la plateforme. Enfin, dans les perspectives de développement, nous décrivons brièvement le contexte dans lequel sera développé le second prototype.

5.1 Gestion Flexible des données : architecture de méta-modélisation

L'ingénierie des systèmes informatiques s'est toujours basée sur des modèles. Ces modèles permettent de décrire les systèmes en cours de développement et leur environnement à différents niveaux d'abstraction. L'approche de l'Ingénierie Dirigée par les Modèles (IDM), plus récente, aussi appelé gestion des modèles [7] utilise des architectures analogues à celle du MOF[17], développée par l'OMG pour la programmation au niveau du modèle et la génération de code. L'architecture du MOF est caractérisée par l'existence de trois niveaux de modélisation : M0 pour les instances, M1 pour les modèles, M2 pour les méta-modèles. Chaque niveau est représenté comme instance d'une structure définie par le niveau de modélisation supérieur, le méta-méta-modèle M3 étant réflexif. Dans cette section, nous proposons l'utilisation du MOF pour définir une architecture de méta-modélisation permettant une gestion flexible des données.

5.1.1 *Notre proposition d'architecture*

Les trois étapes de formalisation – entre les 4 couches (*cf.* § 4) – nécessitent trois structures de modélisation. De plus, les modèles devant être flexibles, ceux-ci seront représentés comme instances d'un méta-modèle afin de pouvoir facilement les étendre et les modifier. Enfin, l'objectif étant de proposer une automatisation supervisée permettant de passer d'un niveau de formalisation à l'autre, les méta-modèles doivent eux-mêmes être représentés au sein d'un même méta-méta-modèle afin de permettre d'exprimer les morphismes entre méta-modèles. Par exemple, un CT/terme génèrera par défaut un concept termino-ontologique (CTO) qui à son tour génèrera un concept ontologique (CO). Nous proposons (*cf.* fig. 5) pour cela une structure de type MOF possédant trois niveaux de formalisation (horizontaux) et trois couches d'abstraction (verticales).

Cette architecture est une extension de l'architecture de Base de Données à Base Ontologique OntoDB [15]. Elle offre, pour chaque type de données géré, une structure de méta-modélisation permettant de modifier le modèle de chacune des couches d'abstraction en modifiant (ajout, suppression, modification) des instances de la couche supérieure.

La partie données (M1/M0). Elle représente les objets du domaine (M0), c'est à dire pour les FT et FTO, les occurrences de termes dans le corpus. Celles-ci sont décrites comme instance d'un terme ou d'un CTO défini dans M1 et d'un ensemble de valeurs de propriétés (par exemple le nom du document, l'adresse dans le document) applicables aux occurrences de ce terme. Par exemple, c'est à ce niveau que seront décrites les occurrences de termes.

[17] `http://www.omg.org/mof/`

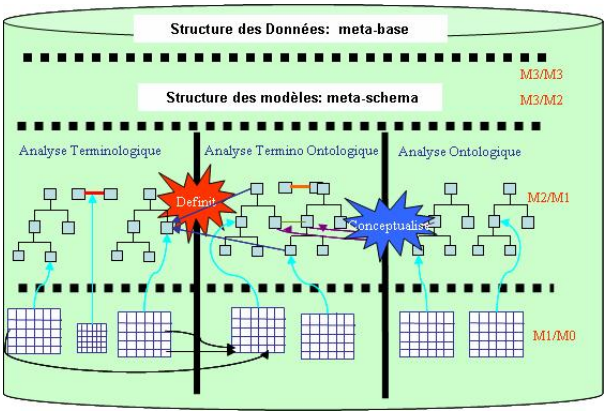

Figure 5: Architecture de méta-modélisation des données dans DaFOE.

La partie modèles (M2/M1). Cette partie contient les trois modèles de la plateforme (terminologique, termino-ontologique et ontologique) et les relations qui les unissent. Par exemple, pour le modèle terminologique, seront représentés les termes et les relations "apparentes" (ou relations extraites par les outils de TAL). Pour le modèle termino-ontologique, on trouvera le lien de chaque CTO vers le terme associé et les relations telles que l'hyperonymie ou la composition. Pour le modèle ontologique, on trouvera pour chaque CO (classe ou propriété), le lien vers le ou les concepts termino-ontologiques éventuels qui ont permis de le créer, et ainsi que les relations – p. ex. subsomption – permettant de spécifier précisément le concept. Cette partie permettra également le paramétrage de la partie données dans un contexte d'évolution.

La partie méta-schéma (M3/M3- M3/M2). Elle permet de représenter, au sein d'un modèle réflexif, à la fois les différents modèles (terminologique, termino-ontologique et ontologique) utilisés et le méta-schéma lui-même (cette partie permet de rendre générique, au travers d'une API, l'ensemble des traitements sur les modèles).

5.1.2 *Étude de cas : validation de l'architecture*

Dans cette partie qui constitue la validation de notre architecture, nous présentons un modèle de données simplifié ayant permis de développer un premier prototype de la plateforme DaFOE. Il ne prend pas en compte, pour l'instant, les relations de niveau modèle (*cf.* fig. 6).

La particularité de notre modèle est qu'il réifie les CT/termes (TermEntry), CTO (TermOnto) et CO (Concept), ce qui nous garantie un maximum de flexibilité quant à l'extension du prototype.

En effet, au niveau M2/M1 de l'analyse terminologique de notre architecture nous ne manipulons que les informations propres aux Candidats Termes (Entité TermEntry). Un CT/terme est par exemple interprété comme une classe dont les instances sont ses occurrences. La réification se fait en créant une instance dans l'entité ATTRIBUT de couche d'abstraction M3, ce qui permet de résoudre le problème de diversité des informations extraites des différents outils de TAL. Il est également possible, par paramétrage dans

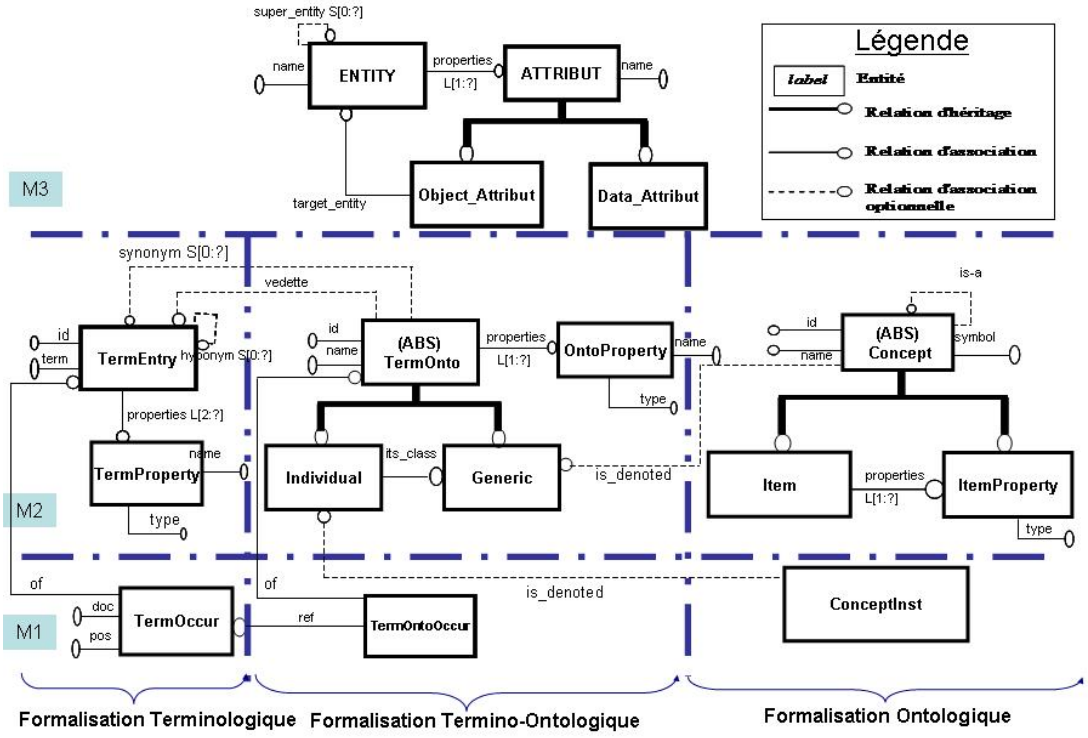

Figure 6: Vue simplifiée des couches d'abstraction et des niveaux de formalisation.

l'Entité TermProperty, d'enrichir la description des occurrences de termes.

C'est au niveau M2 de l'analyse termino-ontologique que nous gérons la polysémie des termes indépendamment du contexte. L'entité OntoTerm permet de contextualiser les sens des CTO à l'aide de référence vers les occurrences de termes de la FT et préserve la trace du texte vers les modèles. Il est également possible d'enrichir la description des instances de CTO à l'aide de l'entité OntoProperty.

Exemple. En reprenant l'exemple de l'ameublement de (*cf.* § 4), les différents CTs sont représentés au niveau modèle de la FT en tant qu'instances de *TermEntry* (niveau M2). Les occurrences de CT dans le corpus sont toutes représentées comme instances de *TermOccur* (niveau M1). Chaque instance référence une position dans le corpus, ainsi que le *TermEntry* auquel il correspond. Après la validation des CTs, le système instancie, par défaut, tous les termes validés en tant que CTOs, représentés comme instances de *TermOnto* dans la FTO (niveau M2). Chaque CTO ainsi instancié référence, en tant que terme *vedette*, le terme qui lui correspond. Par défaut, ses occurrences (*TermOntoOccur*) ne sont pas instanciées, ce qui signifie que les occurences sont celles du terme vedette correspondant. Pour les deux acceptations du terme *table*, le concepteur crée deux *TermOntos* différents. L'un décrit des meubles (*table_meuble*) et l'autre des appareils électroménagers (*table_appareil*). Les deux référencent *table* en tant que *vedette*. Le concepteur passe alors en revue toutes les occurrences du terme *table* dans le *TermOccur* correspondant pour les répartir entre les deux *TermOntos*. Chaque instance de *TermOntoOccur* (niveau M1) contient un pointeur vers une instance du *TermOccur*, ainsi qu'un pointeur vers le *TermOnto* qui lui correspond. En décidant que *table de chevet* et *table de nuit* sont synonymes, le concepteur demande au système de regrouper dans le même *TermOnto*, les deux *TermOntos* générés à partir de *table de chevet* et *table de nuit*. Le système demande alors quel est le terme vedette et met à jour les deux attributs *vedette* et *synomym* du *TermOnto* correspondant. A l'issue de cette phase, par défaut, les différents *TermOntos* génériques (instance de *Generic*) sont instanciés dans la FO comme *Item*. Le concepteur organise alors son ontologie, distinguant une hiérarchie de *meubles* et une hiérarchie d'*appareils*, et décrivant au choix la *table de nuit* et la *table basse* comme des spécialisations de *table_meuble*, ou, tous trois, comme des spécialisations disjonctives de *meuble* (niveau M2).

5.2 Évolutivité des fonctions : choix d'une architecture de développement

Pour faire face aux besoins d'extensibilité, l'architecture DaFOE est une architecture à base de greffons, centrée autour d'un noyau. Inspirée de l'architecture de PROTÉGÉ[18] [18], la plateforme contiendra pour chaque type de formalisation (FT, FTO, FO), des points d'extension décrivant le "contrat" que doit remplir une extension (ou greffon) afin d'être utilisable sur la plateforme comme contributeur pour un service donné. Nous proposons dans cette section au travers les concepts d'extension et de points d'extension, une Architecture Orientée Service comme solution au besoin d'évolutivité fonctionnelle de la plateforme DaFOE.

5.2.1 Proposition de points d'extension

[18] `http://protege.stanford.edu/`

Précisons tout d'abord que la notion de point d'extension ne correspond pas à une fonctionnalité atomique souhaitée par l'utilisateur. En effet, si tel était le cas, chaque évolution de la plateforme nécessiterait la création de nouveaux points d'extension et de ce fait le redéveloppement/redéploiement de celle-ci. Un point d'extension représente un ensemble de fonctionnalités analogues factorisées à l'aide de critères de similarité, et caractérisés par les mêmes APIs d'accès. Nous proposons les points d'extension suivants, valables pour chaque niveau de formalisation :

Traitements graphiques : *Tabwiget_Extension_Point.* Ce critère permet de spécifier les points d'extension permettant aux greffons d'enrichir les interfaces graphiques utilisateur de la plateforme en offrant de nouveaux modes de visualisation des données (arborescence, graphe, tableau, etc.) par accès au modèle de données.

Traitements des entrées de données : *Import_Extension_Point.* Ce critère permet de spécifier les points d'extension permettant aux greffons d'importer des données au sein de la plateforme. Dans la FT par exemple, pour gérer la diversité des résultats fournis par les outils de TAL, un point d'extension permettra de déployer des greffons propres à chacun de ces outils de TAL. Dans la FO, où se pose le problème de diversité des modèles d'ontologie (OWL, PLIB, RDF/S, etc.), la création d'une ontologie interne à partir d'une ontologie existante nécessitera de déployer un greffon de traduction pour les modèles d'ontologie cibles. En natif, la plateforme disposera d'entrée OWL (FO) et SKOS (FTO).

Traitements de sortie de données : *Export_Extension_Point.* Ce critère est le dual de celui du traitement des entrées de données.

Traitements de persistance : *DataAccess_Extension_Point.* Ce critère vise à supporter la diversité des SGBD sur lesquels pourrait être déployée la base de données. Le développement initial sera effectué sur le SGBD PostgreSQL et la migration vers un autre SGBD (Oracle par exemple) ne nécessitera que le développement et le déploiement d'un greffon d'accès aux données sur ce nouveau SGBD. La figure 7 illustre une vue simplifiée d'une architecture extensible identique pour les différents niveaux.

5.3 Perspectives de développement

Pour valider l'architecture de méta-modélisation proposée, un premier prototype à été réalisé par le LISI. Il s'agit d'une application Java s'appuyant sur le SGBD PostgreSQL. La seconde version, en cours de développement implantera l'architecture de développement proposée à la section 5.2. Cette version s'appuie sur le *framework* Eclipse à travers la technologie Eclipse RCP (*Rich Client Platform*) pour la construction d'applications extensibles à base de greffons. La démarche utilisée pour ce développement comportera trois aspects : 1) le maquettage (fichiers PowerPoint) des interfaces utilisateurs par le consortium, 2) le développement de la plateforme selon le cycle en spirale, avec validation

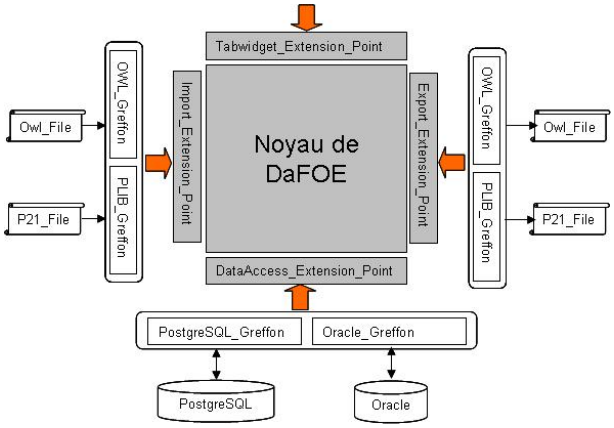

Figure 7: Architecture Orientée Service du noyau de DaFOE.

régulière par les membres du consortium tout au long des deux prochaines années et 3) le démarrage du développement de greffons de façon interne au projet (et éventuellement externe) dès la troisième année du projet.

6. CONCLUSION ET PERSPECTIVES

Nous avons proposé dans ce papier une approche pour permettre l'émergence d'une plateforme capable d'assister un concepteur d'ontologies tout au long du processus permettant d'intégrer des ressources informationnelles propres au domaine (corpus, taxonomies, schéma de base de données, normes) et de définir une ontologie formelle de ce domaine. Les points essentiels de notre contribution sont : *a*) la définition précise des différents états par lesquelles on peut passer lors de la conception d'une ontologie formelle ; *b*) un modèle de donnée permettant d'assurer à la fois la traçabilité et l'automatisation surpervisée des différentes étapes ; *c*) la fourniture d'une démarche méthodologique ouverte guidant l'ontologue sans le contraindre ; *d*) une architecture basée sur le MOF et sur des greffons permettant d'assurer l'évolutivité des modèles et des traitements autour d'un modèle noyau ; *e*) La spécification des différents niveaux d'entrée/sortie de la plateforme : corpus structuré en phrases, fichiers SKOS, fichiers OWL ; *f*) la production finale d'une ontologie à laquelle est associée une composante terminologique.

Après une phase de maquettage au cours de laquelle la faisabilité de l'approche a été établie, nous allons maintenant commencer le développement du prototypes de la plateforme. L'objet essentiel d'une telle plateforme étant de permettre à tous les chercheurs du domaine de pouvoir greffer leurs propres développements, et/ou de pouvoir mener à bien, avec l'assistance de la plateforme, leurs propres démarches de conception, les auteurs de cet article apprécieraient de recevoir toute spécification de besoins, et/ou toute description d'interface de greffon nécessaire pour atteindre cet objectif. La plateforme pourra être mise à disposition des contributeurs dès sa disponibilité en version prototype.

7. REMERCIEMENTS

Les auteur remercient l'ensemble des membres du projet DaFOE 4app pour le travail réalisé qui a pu être synthétisé dans cet article.

8. ADDITIONAL AUTHORS

Nadia Nadah Heudiasyc CNRS/UMR 6599, Université de Technologie de Compiègne, nadia.nadah@hds.utc.fr ;
Henry Valéry Teguiak LISI-ENSMA et CRITT-Informatique, teguiakh@ensma.fr
Nathalie Aussenac-Gilles, CNRS/IRIT et Université de Toulouse, Nathalie.Aussenac-Gilles@irit.fr ;
Adeline Nazarenko, LIPN - UMR 7030, Université Paris 13 - CNRS, Adeline.Nazarenko@lipn.univ-paris13.fr.

9. REFERENCES

[1] S. Aubin and T. Hamon. Improving term extraction with terminological resources. In T. Salakoski, F. Ginter, S. Pyysalo, and T. Pahikkala, editors, *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, number 4139 in LNAI, pages 380–387. Springer, August 2006.

[2] N. Aussenac-Gilles, B. Biébow, and S. Szulman. Revisiting ontology design: a methodology based on corpus analysis. In R. Dieng and O. Corby, editors, *Knowledge Engineering and Knowledge Management : Methods, Models, and Tools. Proc. of the 12th International Conference, (EKAW'2000)*, LNAI 1937, pages 172–188. Springer-Verlag, 2000.

[3] N. Aussenac-Gilles, S. Despres, and S. Szulman. *Bridging the Gap between Text and Knowledge: Selected Contributions to Ontology learning from Text*, chapter The Terminae Method and Platform for Ontology Engineering from Texts. IOS Press, 2008.

[4] B. Bachimont. Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingéniérie des connaissances. In J. Charlet, M. Zacklad, G. Kassel, and D. Bourigault, editors, *Ingénierie des Connaissances, évolutions récentes et nouveaux défis*, pages 305–323, Paris, 2000. Eyrolles.

[5] B. Bachimont. *Art et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle*. Mémoire d'habilitation à diriger des recherches, Université Technologique de Compiègne, 2004.

[6] A. Baneyx. *Construire une ontologie de la pneumonie : aspects théoriques, modèles et expérimentations*. Thèse de doctorat, Université Paris 6, 2007.

[7] P. Bernstein. Applying model management to classical meta data problems. In *Proceedings of the Conf. on Innovative Data Systems Research (CIDR)*, 2003.

[8] D. Bourigault. Upery : un outil d'analyse distributionnelle étendu pour la construction d'ontologies á partir de corpus. In *Actes de la 9iéme conférence annuelle sur le Traitement Automatique des Langues (TALN 2002)*, pages 75–84, Nancy, France, 2002.

[9] D. Bourigault and C. Fabre. Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaires*, (25):131–51, 2000. numéro spécial Sémantique et corpus.

[10] D. Bourigault, C. Fabre, C. Frérot, M.-P. Jacques, and S. Ozdowska. Syntex, analyseur syntaxique de corpus. In *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France, 2005.

[11] P. Buitelaar and P. Cimiano, editors. *Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text*. IOS Press, 2007.

[12] S. Cazabat. Recentrer la conception sur les futurs utilisateurs – exemple d'un projet de conception d'un outil informatique pour la construction d'ontologies. In *Actes de la conférence SELF 2008 – Ergonomie et Conception : Concevoir pour l'activité humaine*, Ajaccio, 17-19 sep. 2008.

[13] P. Cimiano. *Ontology Learning and Population from Text. Algorithms, Evaluation and Applications*. IOS Press, 2007.

[14] P. Cimiano and J. Volker. Text2onto - a framework for ontology learning and data-driven change discovery. In A. Montoyo, R. Munoz, and E. Metais, editors, *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain, JUN 2005. Springer.

[15] H. Dehainsala. *Explicitation de la sémantique dans les bases de données : Base de données á base ontologique et le modéle OntoDB*. Thèse de doctorat, LISI/ENSMA, 2007.

[16] D. Faure and C. Nedellec. Knowledge acquisition of predicate argument structures from technical texts using machine learning: the system asium. In D. F. et R. Stude, editor, *Proceedings of the 11th International Conference on Knowledge Engineering and Knowledge Management*, pages 329–334. Springer-Verlag, may 1999.

[17] T. R. Gruber. A translation approach to portable ontology specifications. 5:199–220, 1993.

[18] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The protégé owl plugin: An open development environment for semantic web applications. In *International Semantic Web Conference*, pages 229–243, 2004.

[19] A. Maedche. *Ontology learning for the Semantic Web*. Kluwer Academic Publisher, 2002.

[20] T. Mondary, S. Despres, A. Nazarenko, and S. Szulman. Construction d'ontologies à partir de textes : la phase de conceptualisation. In Y. Prié, editor, *19es Journées Francophones d'Ingénierie des Connaissances (IC)*, pages 87–98, 18-20 Juin 2008.

[21] N. Nadah, M. D. de Rosnay, and B. Bachimont. Licensing digital content with a generic ontology: escaping from the jungle of rights expression languages. In *ICAIL '07: Proceedings of the 11th international conference on Artificial intelligence and law*, pages 65–69, New York, NY, USA, 2007. ACM.

[22] P. Seguela. *Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques*. PhD thesis, Université Toulouse III, 2001.