

Géditerm: un logiciel pour gérer des bases de connaissances terminologiques

Les bases de connaissances terminologiques organisent des connaissances terminologiques en différenciant les aspects linguistique et conceptuel. Cet article fait le point des outillages disponibles pour gérer des BCT et délimite les besoins des terminologues en la matière; il présente *Géditerm*, logiciel de saisie et gestion de BCT que nous avons développé, ses spécifications et ses fonctionnalités; il se termine par les questions théoriques qu'a fait évoluer cette réalisation, en particulier en contribuant au choix du modèle de données et du niveau de représentation des connaissances.

Termes clés:

Bases de connaissances terminologiques; représentation des connaissances; modélisation conceptuelle.

Introduction

Voilà près de dix ans que la notion de base de connaissances terminologiques (BCT) a vu le jour, en particulier autour des travaux d'I. Meyer (1992) ou d'A. Condamines (1993). Les BCT visent à répondre au besoin de rendre compte de connaissances terminologiques, en différenciant les niveaux linguistique et conceptuel dans une même structure de données. Le concept de BCT, aujourd'hui parvenu à une certaine maturité, se stabilise dans des travaux de recherche au confluent de la terminologie et de l'intelligence artificielle. Il reste cependant peu précis, de telle sorte que les réalisations concrètes, peu nombreuses, sont toutes assez différentes. Ainsi, les BCT sont encore loin d'être répandues et diffusées, que ce soit pour des applications industrielles ou auprès des terminologues. On peut trouver plusieurs raisons à cela:

1) Leur mise au point est longue et les méthodes applicables encore en cours de définition. Pour dépasser les techniques classiques d'exploration manuelle de corpus ou de définition de termes sans référence à leur usage, les terminologues exploitent désormais des corpus selon une approche linguistique, manuelle ou automatisée à l'aide de logiciels de traitement automatique de langage naturel (TALN). Or l'utilisation judicieuse de ces outils et leur intégration requièrent un travail

méthodologique rigoureux associé à des développements informatiques complémentaires.

2) Leur intérêt, bien qu'intuitivement fort, présente plusieurs facettes dont certaines n'ont pas été complètement démontrées. Ainsi, les BCT constituent un pas en avant indéniable pour la pratique terminologique: on passe de listes de termes, chacun ayant un statut référentiel, à des termes reliés entre eux sémantiquement via un réseau conceptuel, établi en fonction de leur usage en corpus. Dans une BCT, le terme est donc rattaché à une sémantique différentielle, plus pertinente dans de nombreux cas, comme le soulignent F. Rastier (1995), M. Slodzian (1995) ou B. Bachimont (1995). En revanche, beaucoup reste à faire pour démontrer l'apport de l'utilisation des BCT au développement d'applications. Il faudrait préciser comment les exploiter au mieux et mesurer en quoi elles augmentent la qualité des modèles produits et l'efficacité de leur construction. Pour le moment, la contribution des BCT a été surtout étudiée pour construire des ontologies par D. Škuce (1994), des terminologies ou des index par C. Gros (1998).

3) Les outils informatiques adaptés à leur construction sont encore insuffisants. Ils peuvent être de deux sortes. Une première classe de logiciels⁽¹⁾ aide le terminologue-linguiste à analyser les corpus et à automatiser ses techniques d'exploration. Ce sont des logiciels de TALN, les extracteurs de listes de mots, des outils d'aide à la mise au point ou à l'application de

(1) Un panorama d'outils de ce type a été dressé par A. Condamines et J. Rebeyrolles en 1997.

marqueurs, des concordanciers ou des outils statistiques, etc. Une deuxième famille regroupe les environnements de saisie et de gestion des données d'une BCT. Ces logiciels, peu nombreux, correspondent souvent à une interprétation très informatique de la notion de BCT. Plutôt que de répondre au plus près aux besoins des linguistes-terminologues, ils privilégient la mise au point d'un réseau conceptuel et se focalisent d'avantage sur la formalisation des connaissances. Ils servent en effet souvent à construire des bases de connaissances ou des ontologies.

Nous avons développé un logiciel de saisie de BCT, dans le cadre plus large d'un projet⁽²⁾ qui vise l'évaluation de la pertinence des BCT pour différents types d'applications. En contribuant au choix du modèle de données puis du niveau de représentation des connaissances, notre intervention a fait progresser la définition de ce concept, tout en mettant en évidence à la fois ses limites et sa pertinence. L'article s'articule donc autour de trois parties: la première fait le point de logiciels disponibles pour gérer des BCT et délimite les besoins des terminologues auxquels nous avons cherché à répondre; la deuxième présente le logiciel développé, *Géditerm*, les spécifications retenues et ses fonctionnalités; la dernière met l'accent sur les questions théoriques qu'a fait évoluer cette «concrétisation informatique» du concept de BCT.

(2) Ce projet a été financé par le Gis «Sciences de la Cognition», la région Midi-Pyrénées et la DER d'EDF-GDF de 1996 à 1998.

1 Quelques systèmes de gestion informatique de BCT

Pour présenter les différents logiciels ou langages qui existent aujourd'hui pour construire des BCT, nous nous plaçons dans la perspective du linguiste-terminologue qui souhaite enregistrer des données alors qu'il analyse un corpus, identifiant des termes ou des connaissances sur un domaine. Nous mettons à part les bases de données terminologiques, versions informatiques de listes sur papier, présentant les mêmes faiblesses: les termes sont définis indépendamment les uns des autres, sans référence précise à leur usage.

Nous avons donc évalué en quoi les structures de données produites par différents systèmes étaient des BCT, quelles étaient leurs fonctionnalités et en quoi elles facilitaient la tâche de leurs utilisateurs. Nous avons retenu les critères énoncés par F. Lemaire et F. Rechenmann (1995): la structure d'une BCT doit permettre d'éviter la confusion entre relations linguistiques (grammaticales) et relations sémantiques (conceptuelles); de différencier la représentation des connaissances terminologiques et conceptuelles. L'offre actuelle est très disparate, suivant que les outils prennent en compte ou non l'utilisation qui sera faite de la BCT, et qu'ils s'appuient ou non sur une représentation formelle des connaissances. Le fait d'utiliser une représentation formelle permet, lors de la construction d'une BCT, de réduire les ambiguïtés en obligeant à formuler des critères de définition et de différenciation, de classer au fur et à mesure les concepts définis, de vérifier leur cohérence, etc. La contribution de l'intelligence artificielle est ici significative: la plupart des formalismes utilisés sont inspirés des réseaux sémantiques, des

logiques de description et des graphes conceptuels. Par contre, la formalisation impose des contraintes. Nous reviendrons sur la question de savoir si une BCT doit être formelle ou non lorsque nous présenterons nos propres choix, dans la partie 2.

Nous distinguons ici les réalisations qui visent à développer une représentation structurée en vue d'utilisations potentielles non définies de celles qui visent à développer des BCT en sachant à quoi elles vont servir (ces BCT formeront une partie d'application). Les premières s'intéressent plus au processus de construction des BCT et à la composante terminologique alors que les secondes privilégient la base de connaissances de la BCT et son utilisation.

1.1 Des logiciels pour enregistrer des BCT

Les environnements qui guident l'organisation des connaissances et la construction de la BCT, sans avoir *a priori* une idée sur l'utilisation qui en sera faite, ne reposent pas sur un profil précis de l'utilisateur, à la fois linguiste et cognicien. Le modèle des données et l'interface dissocient termes et concepts, tout en facilitant leurs associations et le retour au corpus de référence. On peut donc bien parler de BCT au sens donné par Meyer (1992), Condamines (1993) ou Rechenmann (1995). Ces travaux ont également en commun le souci de mener une étude «théorique» de ce que doit être une BCT.

HTL, l'interface hypertextuelle de consultation des résultats de *Lexter* (conçu par D. Bourigault, 1994) sert avant tout à visualiser et valider les candidats termes produits à partir d'un corpus. HTL permet aussi d'organiser les termes en familles de synonymes, désignées par un des termes, le terme-vedette, qui est

proche d'une étiquette de concept. Les vedettes peuvent d'ailleurs être associées par des relations conceptuelles étiquetées. De plus, le corpus est intégré dans HTL: il donne accès à toutes les occurrences d'un terme et permet de justifier l'organisation conceptuelle en fonction de l'usage des termes en corpus. En ce sens, les données contenues dans HTL se rapprochent d'une BCT.

La méthode d'analyse conceptuelle interactive (ACI) d'H. Assadi (1998) s'appuie sur une documentation technique et sur des outils d'analyse des termes contenus dans ce corpus pour élaborer une ontologie régionale documentée. Ce résultat est assimilable à une BCT, dans la mesure où les concepts sont organisés en réseau puis reliés à des expressions linguistiques et au corpus d'où ils sont extraits. De plus, la définition des concepts est locale au corpus. Dans le prototype d'atelier associé à la méthode ACI, le réseau conceptuel est représenté par des *frames*, gérés formellement au moyen de règles. Ce langage permet de construire progressivement l'ontologie et de garantir, par induction, la bonne organisation de la taxonomie des concepts et l'héritage de leurs propriétés.

Pour construire une BCT formelle, N. Capponi a envisagé une démarche en deux temps (1995, p. 11): construire un modèle terminologique du domaine par analyse linguistique du corpus; puis formaliser ce modèle pour en faire une BCT exploitable par un résolveur de problème. La formalisation utilise *Classic*, une logique de description, pour réduire une partie du modèle terminologique (des listes de candidats-termes produites par *Lexter*) puis le structurer en définissant des concepts et des relations. Cette BCT n'intègre pas le corpus et ne distingue pas explicitement terme et concept,

puisque les informations linguistiques sont associées aux concepts, qui sont les classes du formalisme. De ce fait et à cause des contraintes liées au langage choisi, cette BCT convient mal pour enregistrer les résultats d'une analyse linguistique plus fine.

1.2 Des outils pour utiliser des connaissances terminologiques

D'autres travaux ont recours à la définition d'une BCT parce que cette structure de données semble adaptée à certains traitements sur les connaissances contenues dans les textes: les classer, les vérifier, les comparer, les spécialiser ou les généraliser. Les exemples retenus ici illustrent des applications variées mais étroitement liées au texte: recherche documentaire, vérification de spécifications décrites dans le texte. Le réseau conceptuel est alors directement formalisé dans un langage choisi en fonction des opérations à faire sur les connaissances dans l'application envisagée. Mais, de ce fait, ces outils privilégient les utilisations du réseau conceptuel, et non les besoins liés à la construction de la BCT. Nous ne mentionnons pas certains langages et environnements de construction d'ontologies, qui gèrent les termes désignant les concepts, car ces travaux s'éloignent de notre objectif.

Terminae, logiciel développé par B. Biébow et S. Szulman (1997), sert avant tout à construire, à partir de textes, des ontologies pour une application donnée, la vérification de spécifications. Ce logiciel permet d'étudier les occurrences des termes en corpus, de les organiser et de les formaliser dans une BCT. À partir d'une lecture du texte, les concepts sont organisés dans une hiérarchie différentielle, ce qui permet de les classer et de valider leurs descriptions. Les termes associés et les liens vers le texte sont conservés. Le formalisme

utilisé, une logique de description, permet ensuite de raisonner sur ces connaissances ou de les traduire vers d'autres formalismes.

CG-KAT, logiciel développé par P. Martin (1995), permet de décrire formellement des informations pour faciliter la recherche de connaissances dans les documents d'où elles sont extraites, de présenter le document sous forme hypertextuelle et de l'indexer. Ces informations sont représentées à différents niveaux de granularité (concept, phrase, paragraphe, etc.), selon différents points de vue. À la fois proches du niveau linguistique et formalisées à l'aide de graphes conceptuels (GC), elles constituent une structure apparentée à une BCT mais inexploitable dans un autre contexte. Ces graphes sont construits par analyse automatique des textes (contenu et forme), puis enrichis en exploitant le réseau terminologique *Wordnet*.

Plus général, le logiciel *Hytropes*⁽³⁾ est une interface de gestion de bases écrites en *Tropes*, un langage de *frames* proposé par J. Euzenat (1996). Il autorise la formulation de requêtes pour retrouver les concepts d'un domaine (les classes d'objets *Tropes*) selon des critères sémantiques (leurs champs ou attributs). *Hytropes* facilite la visualisation et la structuration des concepts, de leurs sous-classes et des différents points de vue associés. Il permet donc de gérer des ontologies à rapprocher de BCT dans la mesure où des textes sont associés aux concepts (et inversement), de manière à illustrer des définitions, justifier la représentation des connaissances et faciliter la navigation au sein des connaissances.

(3) Logiciel visible sur l'Internet à l'adresse <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/euzenat/euzenat-demo.html>

Dans la lignée de Code4 (1994), le logiciel *DockMan*⁽⁴⁾, développé par D. Skuce (1998), permet de modéliser les connaissances présentes dans des textes afin de les rechercher par requêtes. *DockMan* propose un niveau d'abstraction intermédiaire pour représenter ces connaissances avant d'en construire une ontologie formelle. Son originalité est de gérer des assertions, unités de connaissances tirées du texte, plus riches que le terme ou le concept. Chaque assertion traduit de manière structurée un terme et toutes les informations linguistiques relatives au texte associé (sujet/prédictat/objets). Explorer les connaissances présentes dans le texte se fait ensuite par des requêtes sur la base des assertions, qui renvoient les différentes utilisations des termes. Une base de ce type est amorcée par une analyse syntaxique automatique et sommaire du texte, puis enrichie par un linguiste qui organise et décrit précisément les assertions, au plus près du texte.

2 Le logiciel *Géditerm*: présentation

2.1 Prise en compte des besoins des terminologues

Notre objectif a été de fournir aux terminologues *Géditerm*, un outil de saisie de connaissances pour les organiser en une BCT. Le terminologue peut extraire ces connaissances d'un corpus en utilisant ses compétences et techniques de linguiste, sa connaissance de la langue de sens commun et ses outils de traitement automatique de la langue.

(4) Logiciel accessible sur l'Internet à l'adresse <http://dkm.site.uottawa.ca/beta/dockman.html>

Méthodologie de référence

Notre logiciel a été spécifié dans le contexte d'un projet de recherche, au cours duquel les linguistes-terminologues étaient en train de mettre au point leur démarche. Celle-ci n'était donc pas figée en début de spécification, et il a été difficile d'imaginer des fonctionnalités adaptées. Finalement, le travail de définition des spécifications a participé à la réflexion méthodologique, en obligeant à préciser les méthodes de travail et la façon dont les données sont renseignées. Les linguistes savaient quels outils elles utilisaient, comment elles procédaient de manière manuelle, mais n'avaient jamais mis à plat leur démarche, ni caractérisé leurs productions lorsqu'elles utilisent des logiciels. Le fruit de cette réflexion a été publié par A. Condamines et J. Rebeyrolle (1997). Avant d'exposer les fonctionnalités de *Géditerm*, nous présentons la démarche suivie pour recueillir les besoins, les besoins retenus et les choix relatifs à la représentation des connaissances.

Démarche suivie

Au centre de notre démarche de recueil des besoins, nous avons conduit une série d'entretiens auprès des linguistes puis des réunions de travail centrées sur le modèle de données. Ces réunions ont eu pour support une formulation écrite des spécifications, affinée et corrigée tout au long des échanges. Très rapidement, une maquette a été développée pour suggérer des fonctionnalités, vérifier leur adéquation aux méthodes de travail et aux besoins des linguistes en phase de construction d'une BCT. À partir de là, des modèles UML ont été construits pour organiser les spécifications, assurer une bonne documentation du projet et

minimiser les erreurs lors des évaluations.

Une autre référence pour nous aider à spécifier les fonctionnalités a été l'analyse comparée et l'évaluation d'un logiciel proche, l'interface HTL présentée ci-dessus, que les linguistes ont utilisé pour valider la liste des candidats-termes identifiés par *Lexter*. HTL autorise une présentation des termes et la saisie de certaines de leurs propriétés en listes. Les linguistes n'ont pas utilisé ce type de présentation. Mais les termes eux-mêmes sont aussi accessibles sous forme de fiche rassemblant toutes les informations qui les concernent. Lorsqu'une vedette est associée à un terme, on peut consulter aisément l'un puis l'autre. Les différentes informations rattachées à une vedette (termes associés et relations conceptuelles) sont présentées sur un même écran. Une évaluation positive de ces dernières fonctions nous a conduit à favoriser une saisie par fiche ou carte plus que par liste.

Le modèle de données

Les besoins correspondent d'abord aux exigences théoriques (classiques) relatives à la séparation des notions clés de terme et de concept, ainsi que leur relation avec le corpus, et touchent donc le modèle de données. Ainsi, le système doit distinguer les mots en contexte des structures de terme et de concept, tout en rendant compte de la complexité de leur couplage. Les termes étant enregistrés sous une forme «standard» proche du lemme, il faut conserver leurs formes brutes pour les retrouver dans le corpus et accéder à leurs différentes occurrences. Une originalité de *Géditerm* est de conserver le corpus comme partie intégrante de la BCT, puisque, de fait, il définit le domaine couvert par cette BCT. Les informations présentes dans le corpus, l'usage des termes, donnent leur sens

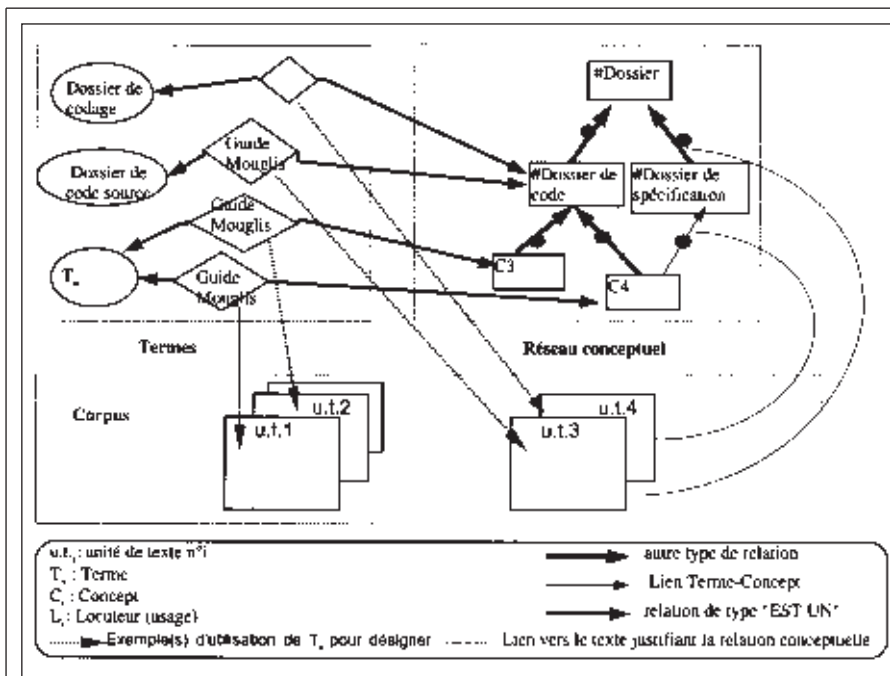


Figure 1 :

Organisation des données dans la BCT. Le réseau conceptuel, à droite, est relié aux données terminologiques, à gauche, par des liens qui renvoient à des parties du texte justifiant le fait que ce terme désigne ce concept. Les liens vers les textes partant des relations conceptuelles renvoient à des occurrences illustrant la relation.

aux concepts associés, dont les définitions ont donc une validité locale au corpus et non universelle. L'organisation des concepts est « stabilisée mais provisoire » : c'est celle révélée par le texte.

Les données terminologiques et conceptuelles ainsi que les relations entre ces données et le corpus sont enregistrées dans la BCT, selon un modèle de données qui a fait l'objet d'une étude approfondie. Une première version de ce modèle, sa justification pratique et théorique, ont été présentées par N. Aussenac et P. Séguéla (1997). Le modèle stabilisé correspondant à l'organisation actuelle des données dans le logiciel fait l'objet de la figure 1 (Simon 98). Pour organiser la base de données de la BCT, et en vue d'anticiper la prise en compte de concepts matérialisés par des sons ou des images, le modèle

conceptuel de la base comprend une entité signifiant (à ne pas prendre au sens de strict de Saussure) dont les entités terme, image et son sont des spécialisations. Dans la suite, nous assimilerons terme et signifiant.

Besoins relatifs aux fonctionnalités

Une autre partie des besoins exprimés concerne les fonctionnalités de saisie, mise à jour et consultation des données de la base. On suppose que le logiciel est utilisé soit pour saisir à la main des termes et des concepts un à un, décrits au fur et à mesure, soit qu'il récupère des données déjà validées, produites par un logiciel. L'élimination systématique de candidats qui ne seraient pas des termes parmi les listes produites par des logiciels d'extractions de candidats termes

comme *Lexter* ou *Nomino* doit se faire en amont de l'utilisation de *Géditerm*. De ce fait, on privilégie un affichage et une saisie par fiches et non sur des listes (comme c'est le cas dans HTL par exemple).

Une autre des caractéristiques du travail des linguistes est de progresser dans la description et dans la compréhension du domaine en s'intéressant successivement à des concepts ou à des types de relations conceptuelles à partir de leur traces dans le corpus. Ces concepts sont repérés par des groupes de termes ayant le même comportement en contexte et définissant des classes conceptuelles plus ou moins larges. Pratiquement, les linguistes ont donc besoin de rechercher des termes reliés, des synonymes ou antonymes, de vérifier qu'un mot dans différents contextes correspond au même terme et ensuite au même concept, etc. Ils souhaitent aussi sauvegarder toutes les relations associées à un concept dans le corpus.

Enfin, le logiciel sera d'autant plus utile qu'il sera simple à utiliser et qu'il permettra à ses utilisateurs de vérifier et récapituler rapidement ce qui a été saisi. Une visualisation du réseau conceptuel sous forme de graphes, déjà évaluée par les linguistes dans le cadre de précédentes études, semble être un moyen efficace de montrer et valider le réseau conceptuel. La sélection de sous-ensembles des données en fonction de critères sémantiques avant leur affichage est indispensable pour éviter la visualisation d'énormes graphes.

Parmi les besoins implicites, il a fallu clarifier le type d'aide, de guide que le logiciel doit apporter, que ce soit à travers la représentation des connaissances, l'enchaînement ou la nature des fonctionnalités. Avec les linguistes, nous avons choisi de ne pas définir le logiciel comme un processus actif qui impose pas à pas de suivre une méthode particulière ou d'utiliser des fonctionnalités dans un certain

ordre. Le logiciel doit proposer un ensemble de fonctions d'édition, appelées à la demande selon les informations que l'utilisateur veut enregistrer ou modifier. Par contre, il doit pouvoir consigner des éléments méthodologiques, comme les marqueurs lexico-sémantiques ayant servi à identifier les relations conceptuelles, comme les extraits du corpus justifiant les relations conceptuelles (exemples) ou justifiant les relations terme-concept (occurrences pour lesquelles le terme désigne le concept).

2.2 Principes retenus pour la représentation des connaissances

Un des principes importants retenus pour la représentation des connaissances a été de préférer une représentation non formelle, qui rende compte du réseau conceptuel sans permettre de raisonner ou d'inférer sur ces connaissances. Pour décider de ce choix, nous avons tenu compte de la pratique des linguistes que nous avons confrontée aux logiciels présentés dans la partie 1. Les rapports de S. Simon (1998) et P. Séguéla (1996) rendent compte de nos évaluations de différentes représentations formelles des connaissances. Reprenant ici leurs conclusions, nous expliquons pourquoi nous considérons qu'utiliser un langage formel est trop contraignant pour le linguiste qui construit une BCT.

Tout d'abord, la formalisation contraint le linguiste à répondre à des questions dont la réponse n'est pas forcément dans le texte ou est ambiguë. En effet, la description formelle des connaissances oblige à poser des critères de définition des concepts. Créer un nouveau concept dans la hiérarchie EST-UN doit se justifier par la présence d'un attribut ou d'une relation (rôle) qu'il ne partage ni avec ses frères ni avec son

père. Pour toute notion repérée à partir d'un terme, il faut d'abord décider comment la représenter (sous forme de concept, de relation ou d'attribut). Ensuite, si un concept est identifié, il faut trouver des indicateurs pour le placer systématiquement au bon endroit dans la hiérarchie EST-UN, puis le décrire en établissant des relations ou attributs qui le différencient de ses frères ou de son père; savoir si ces relations et attributs lui sont propres ou sont hérités, etc. Or ces connaissances ne sont pas toujours dans les textes. Elles doivent être demandées aux experts. De plus, il est difficile de garantir une «bonne définition» des concepts en l'absence de finalité précise, c'est-à-dire de critère de décision pour trancher sur la définition à retenir. Or le linguiste ne se préoccupe pas encore à ce niveau de la finalité de la BCT.

Ensuite, la formalisation impose d'avoir une vision globale des connaissances à représenter, pour distinguer d'abord les concepts et relations dits primitifs de ceux qui seront définis à partir des premiers. De plus, il faut définir les concepts dans un ordre lié à l'organisation conceptuelle des données (placer les concepts les plus généraux puis les spécialiser) et non dans l'ordre où les données sont trouvées dans le texte. Au contraire, le linguiste dépouille le corpus progressivement et souhaite les représenter au fur et à mesure, alors qu'il ne possède pas tous les éléments nécessaires. Le linguiste a donc besoin d'une structure souple, peu contraignante, qui joue le rôle d'un outil d'annotation de résultats.

La représentation formelle des connaissances serait donc utile pour aider le linguiste à procéder de manière systématique, à ne rien oublier. Mais elle exige d'anticiper l'utilisation qui sera faite des données, d'avoir recours plus souvent aux experts du domaine pour des validations et pour compléter le

corpus. De ce fait, elle conduit à enregistrer des connaissances plus éloignées du corpus, parfois sans justification linguistique, ce qui n'est pas notre objectif premier. Envisageable dans un deuxième temps, nous l'avons prévue dans l'environnement d'exploitation de BCT que nous sommes en train de développer.

2.3 Fonctionnalités

Paramètres du logiciel

De manière à pouvoir utiliser le logiciel dans différents projets, un certain nombre de données de l'application sont paramétrables: les différents statuts de validité donnés aux termes ou aux concepts, la langue du corpus, les différentes catégories grammaticales, genres et nombres possibles des termes dans cette langue, le nombre de phrases que l'on veut voir simultanément lorsqu'on consulte le corpus.

Stocker, enregistrer

Les données enregistrées correspondent aux éléments du modèle de données: *signifiants* (termes), *concepts*, *types de relations conceptuelles* et *relations conceptuelles*, *liens* entre signifiants et concepts, *textes*, ainsi que quelques éléments méthodologiques. L'enregistrement du corpus requiert un traitement préalable pour le découper en unités. Nous avons repris sur le format utilisé dans HTL, un découpage en phrases ayant chacune un code identifiant. Nous avons envisagé assez tard d'intégrer des éléments méthodologiques, alors que cela semble un atout important pour permettre d'enrichir la méthodologie des terminologues et, plus immédiatement, pour assurer une maintenance correcte de la BCT. Il s'agit essentiellement des marqueurs lexico-sémantiques utilisés pour

repérer des traces de relations conceptuelles dans le corpus. Ils serviront à retrouver ces relations sur de nouveaux textes si jamais le corpus évolue par exemple. De plus, dans chaque structure de donnée, le linguiste dispose de zones de commentaires dans lesquelles il peut consigner la justification des choix de modélisation et des occurrences.

Pour accélérer l'inventaire des termes et des unités de texte à consigner dans une BCT, il est préférable d'utiliser en amont un outil d'extraction de candidats termes comme *Lexter* ou *Nomino*. *Géditerm* offre une fonction permettant d'intégrer dans la BCT le corpus, une liste de termes candidats (préalablement validés) et même des hypothèses de concepts tirés de HTL. Cette fonction déclenche un transfert des données systématique puis une rapide validation interactive par le linguiste pour corriger leur organisation.

Consulter ou modifier : des listes aux cartes

Accès aux listes

Les composants de base sont : les signifiants, les concepts, les types de relation et les textes. Les liens conceptuelles ne sont jamais visibles indépendamment des entités qu'ils mettent en relation. Ces informations sont accessibles à partir de listes de noms, complètes ou partielles lorsqu'elles sont filtrées selon des critères précisés par l'utilisateur. Les listes de concepts ou de signifiants affichent par défaut l'intégralité des noms des structures correspondantes de la BCT. Depuis une liste, on peut créer un nouveau composant de ce type ou bien demander à consulter un composant existant.

Accès par jeux de cartes

Choisir un nom dans une liste permet de visualiser le composant

associé (un signifiant, un concept, un lien terme-concept ou une relation conceptuelle), sous forme d'un écran de saisie appelé « fiche » ou « carte » dans la suite. La carte présente les informations spécifiques à cette donnée et ses composants reliés : pour un signifiant, la liste des concepts qu'il désigne ; pour un concept, la liste des signifiants qui le désignent et la liste des concepts reliés par des relations conceptuelles. Sélectionner un de ces composants permet d'accéder à une nouvelle carte le décrivant. Les cartes se superposent (3 au maximum) à l'écran et des onglets permettent de passer rapidement des unes aux autres. L'utilisateur dispose alors des informations relatives à plusieurs composants ayant une parenté sémantique : soit un triplet concept-relation-concept, soit un triplet terme-lien-concept. L'accès au texte se fait alors depuis les liens ou les relations.

Sur la figure 2, la carte présente les informations relatives au concept *#acteur externe*. Les autres cartes présentes contiennent les informations relatives au concept *#humain*, à un signifiant, au lien

entre le signifiant et l'un des concepts *#acteur externe* ou *#humain* ainsi que la liste des concepts. En cliquant sur l'un des onglets, la carte correspondante apparaît.

Aider à la définition et à la structuration des données

Guider l'organisation des données

Pour guider l'organisation des concepts, les relations conceptuelles sont typées, et l'ensemble des types de relations est répertorié, organisé en hiérarchie et accessible via des listes ou sous forme d'arbre (Figure 3). L'organisation hiérarchique est supposée refléter le caractère plus ou moins général des relations et s'appuie sur les liens de spécialisation tels que les interprète le linguiste. Tout nouveau type de relation doit être ajouté à la hiérarchie et défini par le type des concepts qu'il relie avant d'être utilisé.

Faciliter les choix

Géditerm simplifie la saisie d'informations en présentant des listes lorsqu'il s'agit de choisir parmi un ensemble fini de données. En favorisant de plus un repérage rapide

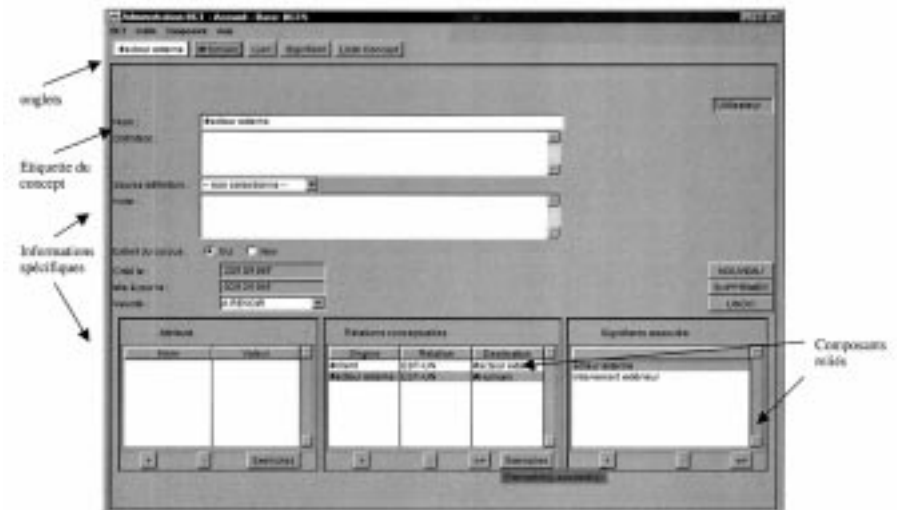


Figure 2 :
Exemple de carte de concept active : le concept *#acteur interne*.

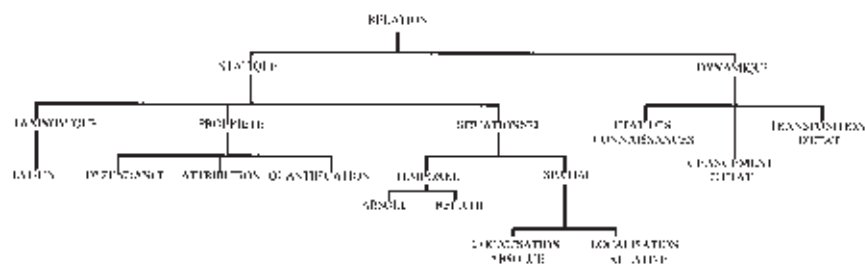


Figure 3:
La hiérarchie des types de relations conceptuelles

dans les listes (recherche des premiers caractères d'un mot) et en autorisant le paramétrage des options à choisir dans ces listes, l'interface essaie d'alléger les tâches les plus fastidieuses. Par exemple, le paramétrage permet à l'utilisateur de compléter la hiérarchie des relations avec des types de relations propres au corpus. Ensuite, lorsqu'il pose une relation entre deux concepts, il choisit son type parmi les types de cette hiérarchie.

Aider à sélectionner des données

Pour rechercher des données, on peut appliquer des filtres, ensembles de critères sémantiques, sur les listes. Ces critères portent sur les attributs des données et sur les relations entre données. Par exemple (Figure 4), les critères de sélection sur une liste de signifiants peuvent être un syntagme nominal; une variante de forme (ellipses, abréviations et formes les plus utilisées d'un terme); un locuteur; un concept (les signifiants retenus auront au moins un lien avec ce concept); un degré de validité.

Visualisation graphique

De même, on peut filtrer les données de la BCT avant de créer une vue qui sera visualisée graphiquement. Ainsi, on peut sélectionner les concepts ou les termes reliés à un concept précis ou fixer un type de relation particulier. La figure 5 présente un sous-ensemble de la BCT Mougliis: ce sont tous les

concepts reliés par une relation autre que EST-UN au concept #cycle de développement produit.

Visualisation du corpus

La consultation du texte de référence de la BCT se fait soit depuis un composant du modèle (lien terme-concept ou relation conceptuelle), soit depuis le menu principal. Les extraits du corpus sont affichés par groupes ou paragraphes (Figure 6) alors que le texte ne provenant pas du corpus est affiché de manière isolée. Un paragraphe comprend une unité textuelle et les n unités textuelles qui la précèdent et la suivent (n peut être modifié). On accède à la demande aux paragraphes précédant et suivant le paragraphe courant.

Vérifier, valider

Les spécifications ont établi l'ensemble des informations à vérifier, mais surtout les principes à suivre et

le moment de ces vérifications. Nous avons le choix entre des vérifications systématiques dès qu'une structure (terme ou concept) est modifiée, des vérifications à la demande sur ces structures ou des vérifications massives sur des listes (portant sur les structures complètes ou par critère). Nos premiers choix étaient de retenir une validation morcelée et systématique après la mise à jour de données. Pour cela, toute donnée possède un degré de validité, qui peut prendre plusieurs valeurs dont une seule est valide. Lorsqu'on valide une donnée, le processus de vérification est déclenché à la fermeture de la fiche, et les informations manquantes ou incorrectes sont signalées. Voici quelques exemples de vérifications prévues:

- Les relations conceptuelles par rapport à leur sémantique: afin de garantir que les étiquettes des types de relations conceptuelles soient interprétées de la même façon au cours de la construction de la BCT, puis à son utilisation, les types de relations comportent une «signature», c'est-à-dire que l'on indique les classes sémantiques (concepts de haut niveau) qu'ils relient. Toute relation spécifique entre deux concepts doit donc relier des concepts fils (indirectement dans la hiérarchie des concepts) de ces classes.
- La place des concepts dans la hiérarchie EST-UN. Tout concept doit, en fin de construction de la BCT, être situé dans la hiérarchie

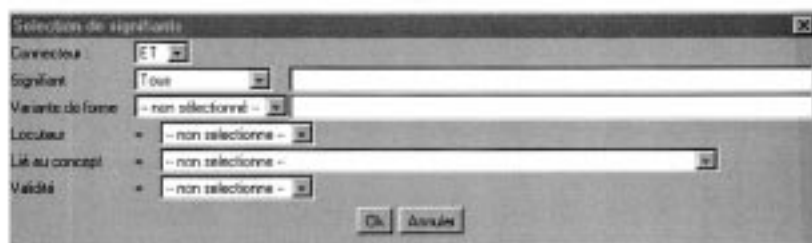


Figure 4:
Fenêtre de définition d'un filtre pour la liste des signifiants

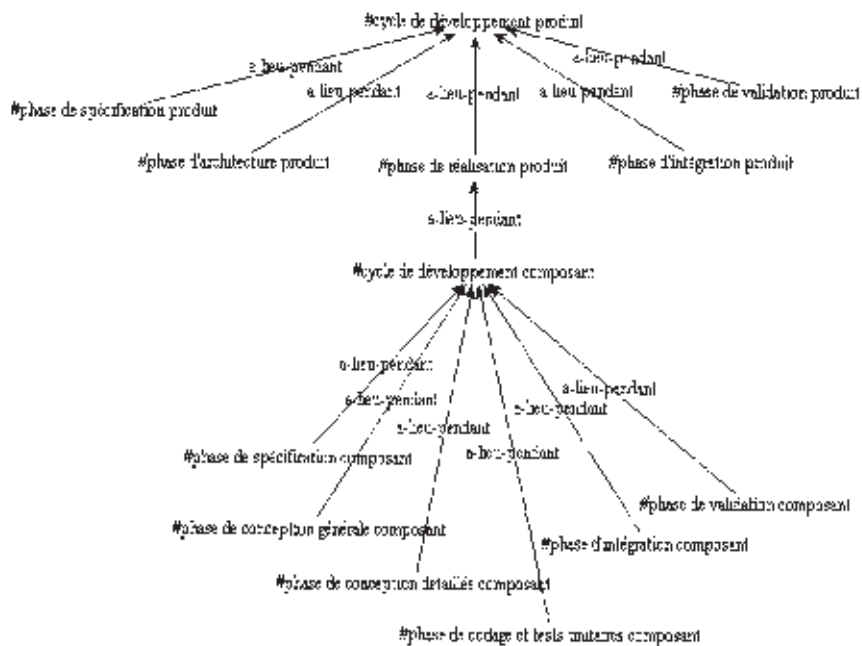


Figure 5 :
Extraits du réseau conceptuel de la BCT Mouglis.

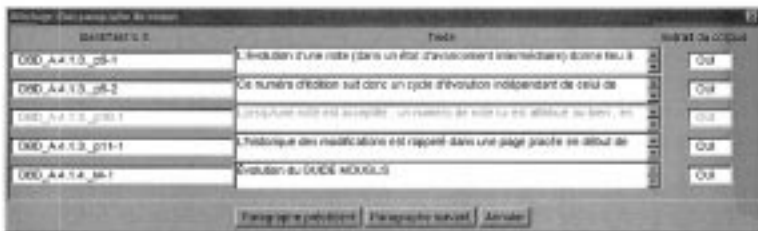


Figure 6 :
Fenêtre de consultation du corpus : un paragraphe d'unités textuelles avec n=2

EST-UN. Afin d'éviter un travail massif en fin de processus, cette vérification, initialement prévue à la création de chaque concept, n'est lancée qu'à la demande.

- La description des structures (termes ou concepts). Les spécifications ont conduit à imposer qu'un certain nombre d'informations minimales soient connues pour toute structure de données, comme son nom, mais aussi un terme relié pour un concept, sa langue pour un terme,

etc. On considère qu'une BCT n'est terminée que si ces informations minimales ont été renseignées pour tous les termes et concepts qu'elle contient. Cette vérification massive n'est lancée qu'à la demande.

Ce choix a été remis en question par la pratique des linguistes, qui, à l'utilisation de *Géditerm*, n'ont jamais déclaré les données valides afin d'éviter les vérifications. En effet, celles-ci sont trop contraignantes et demandent des connaissances parfois

non disponibles. Ces principes de validations correspondent en fait à des critères de « bonne modélisation », adaptés du principe de différenciation en vue de préparer une formalisation. Or, comme nous l'avons dit plus haut, l'application de ces règles directives requiert des connaissances pas toujours présentes dans le corpus.

2.4 Scénario d'utilisation

Pour illustrer l'utilisation des fonctions disponibles dans *Géditerm*, nous décrivons sur un exemple comment enrichir la BCT.

L'utilisation de *Géditerm* suppose d'avoir sélectionné le corpus et d'en disposer sur support informatique, découpé en phrases avec des identifiants. Si on a analysé le corpus à l'aide de *Nomino* ou *Lexter*, il est préférable de filtrer ces listes de candidats-termes avant de les importer dans la BCT.

Les points de départ de toute recherche ou saisie d'information sont les listes d'étiquettes⁽⁵⁾ de concepts et de signifiants (termes). Que ce soit directement ou à partir d'un élément relié, elles permettent de créer de nouvelles données ou de retrouver un composant. Supposons que les linguistes en soient à la phase d'identification des concepts et des relations conceptuelles. Ils appliquent des marqueurs sur le corpus, comme par exemple le marqueur suivant (TC = Terme candidat) :

Dét_déf TC1 Vêtre présent dét_indéf (partie, élément, composant...) de TC2

(5) Sur les cartes de saisie des signifiants (termes) et des concepts, le nom renvoie à l'étiquette et non à la catégorie grammaticale du terme (qui peut être un groupe nominal ou un verbe).

Ce marqueur permet de trouver le contexte suivant: *La documentation de réalisation est une composante essentielle du produit logiciel*, dans lequel « documentation de réalisation » et « produit logiciel » sont des termes qui vont être associés à des concepts. Ce contexte indique une relation de « composition » entre ces deux concepts. Enfin, d'autres occurrences de « document de réalisation » permettent d'établir que ce terme est synonyme de « dossier de réalisation ». Les termes « dossier de réalisation » (figure 7), « documentation de réalisation » et « produit logiciel » sont déjà présents dans la BCT car identifiés par *Lexter*, ainsi que le concept #*dossier de réalisation*. Le linguiste doit créer un nouveau concept à partir du terme « produit logiciel », et relier le terme « documentation de réalisation » au concept #*dossier de réalisation*.

Ensuite, il doit poser la relation conceptuelle « est-un-composant-de » entre ces deux concepts. Supposons

que ce type de relation n'ait pas encore été défini. Il doit tout d'abord le créer en modifiant la hiérarchie et situer cette relation par rapport aux relations existantes. Il doit également définir cette relation (figure 8) en indiquant les types de concepts qu'elle relie. Enfin, depuis la carte d'un des concepts concernés (figure 2), il utilise le bouton « + » associé à la liste des relations conceptuelles pour définir la relation.

Ainsi de suite, le linguiste travaille par association sémantique,

en allant d'un concept à un autre ou à un terme en fonction des données extraites du corpus. Il peut vouloir accéder à un concept ou un terme inconnu, mais caractérisé par des propriétés connues. Pour cela, à partir d'une liste, il utilise les filtres pour la réduire en fonction des propriétés recherchées, puis sélectionner par le nom le composant recherché ou une entité reliée. Enfin, pour visualiser le réseau déjà saisi, en partie ou en totalité, selon des critères particuliers, il utilise la visualisation graphique. Il faut d'abord créer une vue (sélectionner des concepts ou des termes), lancer le logiciel *Graphlet* puis visualiser la vue sous forme de graphe (figure 5). De là, on peut corriger la présentation à l'écran à l'aide d'algorithmes ou à la main, mais on ne peut pas encore modifier les données.

3 Discussion: impact de la réalisation d'un logiciel sur la réflexion théorique

3.1 Évolutions du modèle de données

Bien que les grandes lignes du modèle des données reprennent les propositions énoncées par A. Condamines et P. Amsili (1993), la réalisation concrète d'un logiciel a obligé les linguistes à préciser comment décrire termes et concepts,

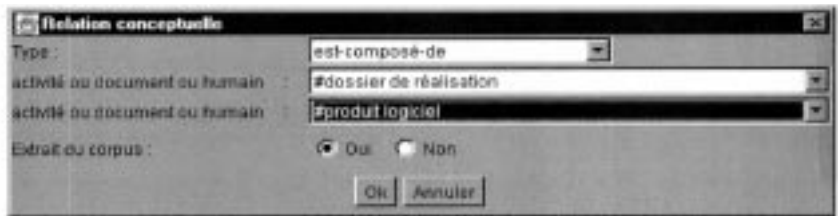


Figure 8:
Interface de définition d'un type de relation

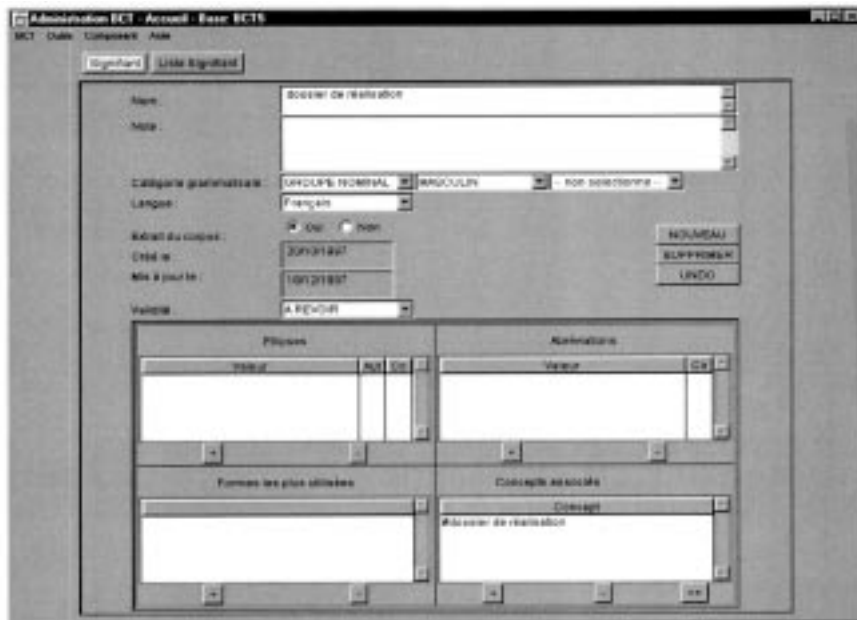


Figure 7:
La carte du terme dossier de réalisation donne accès au concept #*dossier de réalisation*

quelles informations associer au lien terme-concept et surtout comment gérer les notions d'usage et de point de vue. La notion d'usage n'existe plus en tant que telle, mais ce sont les liens entre termes et concepts qui, renvoyant à du texte, illustrent les usages. De même, la notion de point de vue n'est pas gérée explicitement. Par contre, un lien terme-concept est valide pour un ou plusieurs locuteurs. Ces locuteurs peuvent avoir des points de vue sur un concept. Dans ce cas, le terme qu'ils utilisent est polysémique : il prend un sens différent dans différents contextes. Pour les linguistes, il est important de consigner dans ce cas le point de vue dominant pour ce locuteur, c'est-à-dire le sens qu'il donne le plus couramment à ce terme. On doit donc indiquer sur les liens entre termes et concepts quels locuteurs utilisent ce terme pour désigner ce concept et, pour chacun d'eux, s'il s'agit d'un point de vue dominant.

Une dernière évolution du modèle des données a été de renforcer les liens vers le texte par rapport à ce qui était initialement prévu. Ainsi, des exemples peuvent être associés aux relations conceptuelles et aux attributs des concepts. Ils correspondent à des contextes dans lesquels on trouve des marqueurs de cette relation. Ces exemples justifient les relations conceptuelles et aident à les interpréter.

3.2 Différenciation de deux types de BCT

La construction d'une BCT avec *Géditerm* nous a conduite à vouloir renforcer le caractère neutre de l'analyse linguistique. Il nous semble important de marquer la rupture qui existe entre le point de vue du linguiste, que nous avons cherché à respecter, et celui du cogniticien qui modélise les connaissances. Le linguiste a le souci de ne pas biaiser

l'interprétation du texte en construisant une BCT. Pourtant, on sait bien que, du moment qu'il traduit son interprétation dans une structure de données, il fait des choix et construit une abstraction qui n'est pas complètement neutre par rapport au texte. De plus, pour rendre le modèle exploitable par une autre personne, l'organisation des connaissances doit être commentée. Nous appelons tout de même ce résultat une «BCT-Corpus». Nos réflexions sur la représentation des connaissances et sur les fonctions de validation nous ont confirmé qu'une BCT-C doit être informelle et incomplètement structurée.

Au contraire, si, dans un deuxième temps, on construit une BCT en vue d'une application précise, on parlera de «BCT-Applicative». Les informations y sont organisées en fonction des besoins de l'application, peuvent différer du contenu du texte, et méritent d'être formalisées pour être mieux exploitées. Les systèmes comme *CG-KAT*, *Terminae* ou *Hydropes* sont, pour nous, des outils pour construire directement des BCT-A.

3.3 Quelle validité pour les données d'une BCT?

Dans la BCT-C, c'est le corpus qui sert de justification à l'organisation des connaissances. Un autre élément de validation, plus implicite et moins facile à cerner, est la connaissance de la langue que possède le terminologue. Pour le moment, seule une vérification structurelle est possible. Or, si on veut pouvoir exploiter les connaissances, il faut imposer rapidement des critères de définition et vérifier qu'ils sont respectés. Il faut également faire valider les connaissances par des experts compétents par rapport à l'application envisagée. Il est clair que l'outillage formel est particulièrement

adapté dans ce cas, et que la traduction formelle des données du réseau conceptuel est un moyen de s'assurer de la validité des relations entre concepts par rapport à leur définition, du respect des critères de différenciation et de la bonne utilisation de la hiérarchie EST-UN pour organiser les concepts.

3.4 Vers un environnement de consultation de BCT

Le logiciel *Géditerm* est adapté à la construction d'une BCT, mais il en offre une vue trop morcelée pour pouvoir en exploiter le contenu efficacement. Le fait d'adapter le contenu d'une BCT à des contraintes spécifiques relatives à une application requiert des outils et des fonctions bien différentes, comme la possibilité de réorganiser les concepts dans la hiérarchie, de les différencier plus systématiquement, de les représenter formellement. Cette remarque confirme le bien fondé de la séparation BCT-C / BCT-A.

Pour cela, nous avons commencé à prototyper un environnement de consultation de BCT, destiné à préparer des BCT-Applicatives à partir de BCT-Corpus, comme construire un index ou formaliser les connaissances pour construire une BC. Notre hypothèse est que les personnes qui consultent une BCT sont celles qui développent une application à partir des connaissances de la BCT, pas celles qui utiliseront ces connaissances dans leur travail. Ce prototype propose des fonctions d'aide à la sélection des données dans des BCT existantes, pour les modifier et les réorganiser, en fonction d'un besoin spécifique. Il permet ainsi de construire de nouvelles BCT en fonction d'objectifs applicatifs particuliers. Pour cela, nous avons prévu de faire expliciter puis appliquer des critères de définition précis. Ce logiciel doit produire des

données d'un format facilement exportable vers des applications. Pour le moment, ce sont des tables de bases de données. On prévoit également des sorties papier sous forme de rapport de formats divers.

4 Conclusion et perspectives

Tout comme les résultats méthodologiques, les logiciels font partie de l'outillage nécessaire à l'évaluation expérimentale des BCT. Le développement de *Géditerm* représente un résultat pertinent à double titre. Tout d'abord, la réflexion approfondie menée pour sa mise au point a permis d'aborder des questions fondamentales sur la représentation des connaissances (quelles structures utiliser pour représenter des connaissances avant de les rendre interprétables par la machine? Sous quelle forme présenter ces connaissances?) et de préciser le modèle de données dans la base.

Ensuite, même si le concept de BCT s'avère très pertinent, la quantité de données requises pour leur constitution puis l'exploitation des données qu'elles contiennent ne peuvent se faire que sur support informatique. Il est donc indispensable, pour confirmer l'utilisabilité de ce concept, de fournir aussi l'environnement qui le rende exploitable concrètement, dans l'esprit d'une plate forme terminologique (moins orientée vers la modélisation que la plate-forme d'ACI proposée par H. Assadi (1998). Sinon, il restera une belle construction intellectuelle sans diffusion possible en contexte industriel. Cet environnement doit prendre en compte la démarche de construction, et donc intégrer un éditeur comme *Géditerm*. Il doit comprendre aussi des logiciels automatisant les traitements sur

corpus qui réduisent le coût de la démarche tout en maintenant le degré de validité des données. Un éditeur hypertextuel du document y est indispensable pour passer aisément du texte (d'un terme pris dans le texte) aux termes ou aux concepts, pour retrouver le texte sous sa présentation d'origine ou pour mettre en valeur les termes qu'il contient.

4.1 Perspectives techniques

Concernant le logiciel, il nous faut encore développer de nouvelles fonctionnalités, les unes liées aux mises à jour du modèle depuis la représentation graphique, les autres pour pouvoir imprimer des données sous forme de rapport. Nous devons également procéder à une évaluation systématique de *Géditerm* selon une démarche ergonomique. En particulier, il semble souhaitable de rendre plus simple et commode l'accès au corpus. Une autre piste est de rendre l'outil encore plus paramétrable, pour que le terminologue puisse ajouter ou retirer des propriétés aux termes et aux concepts en fonction des informations dont il a besoin pour les décrire. Initialement, nous avons fait l'hypothèse que ces informations soient indépendantes de l'utilisation prévue des données de la BCT. Or il est clair que le modèle actuel ne peut anticiper tout type de besoin. Par exemple, il serait insuffisant s'il fallait utiliser une BCT pour l'aide à la traduction.

4.2 Perspectives théoriques

Une première perspective concerne la représentation des connaissances. Le modèle de données actuel est limité pour rendre compte de relations complexes, faisant intervenir plus de deux concepts, ou pour rendre compte de liens possibles entre relations. Par exemple, dans une

relation de découpage en partie, on voudrait pouvoir préciser quels concepts sont complémentaires et forment ensemble l'objet entier. Autre exemple: rendre compte de schémas de type agent/verbe/objet/moyen. Dans cette perspective, deux pistes sont à étudier. L'une, dans l'esprit de *DocKMan* de D. Skuce (1998), serait de conserver, dans la partie terminologique, des schémas syntactico-sémantique ou lexico-syntaxiques au lieu de syntagmes nominaux seuls; l'autre serait de construire, dans le réseau conceptuel, leur équivalent sous forme de *frames* sophistiqués rassemblant des sous-ensembles de réseaux. La difficulté est alors de ne pas trop figer ces représentations de plus grande taille pour pouvoir revenir à leurs composants, termes et concepts.

Une autre perspective est de poursuivre l'évaluation de l'intérêt des BCT pour construire des modèles d'un domaine, des modèles de produits ou des bases de connaissances. *Géditerm* est en effet le point de départ indispensable pour ce type d'évaluation, et il doit être utilisé dans des contextes différents pour bien évaluer toutes ces fonctionnalités et permettre de favoriser l'utilisation des BCT.

Remerciements

Ce travail est le fruit d'une collaboration étroite avec une équipe de l'ERSS de Toulouse (Anne Condamines et Josette Rebeyrolle), qui est à l'origine du modèle de données retenu et qui a contribué aux spécifications de *Géditerm*. De plus, je remercie vivement Dominique Fournier qui a développé ce logiciel.

*Nathalie Aussenac-Gilles,
Irit – UMR 5505 du CNRS,
Université Paul Sabatier,
Toulouse,
France.*

Bibliographie

- Assadi (H.), (1998): *Construction d'ontologies régionales à partir de textes techniques: application aux systèmes documentaires*, Thèse de doctorat de l'Université Paris VI en Informatique, 286 p.
- Bachimont (B.), (1995) «Ontologie régionale et terminologie: quelques remarques méthodologiques et critiques», dans *Banque des mots*, n° spécial Terminologie et intelligence artificielle, 7, p. 65-84.
- Biébow (B.), Szulman (S.), (1997): «Méthodologie de création d'un noyau de base de connaissances en logique terminologique à partir de textes», dans *Actes des 2^e rencontres Terminologie et intelligence artificielle. TIA'97*, Toulouse, avril 1997, p. 69-84.
- Bourigault (D.), (1994): *Lexter, un logiciel d'extraction de terminologie: application à l'acquisition de connaissances à partir de textes*, Thèse de doctorat en informatique linguistique de l'EHESS, Paris.
- Capponi (N.), (1995): *Modélisation d'une base de connaissances terminologiques*. Mémoire de DEA de l'Univ. de Nancy I. CRIN/LORIA, Nancy, 45 p.
- Condamines (A.), Amsili (P.), (1993): «Terminologie entre langage et connaissances: un exemple de base de connaissances terminologiques», dans *Terminology and Knowledge Engineering*. Francfort, 316-323.
- Condamines (A.), Rebeyrolle (J.), (1997): «Construction d'une base de connaissances terminologique à partir de textes: expérimentation et définition d'une méthode», dans *Actes des Journées d'Ingénierie des Connaissances IC'97, mai 1997*, (INRIA), Roscoff (F), 191-206.
- Condamines (A.), Rebeyrolle (J.), (1998): «Description d'une BCTC: base de connaissances terminologiques modélisée à partir d'un Corpus», dans *Workshop COGNITERM'98, International Conference on Computational Linguistics (COLING), August 1998*, Montreal, Canada.
- Euzenat (J.), (1996): «Hytopes: a ww front-end to an object knowledge management system», dans *Knowledge Acquisition Workshop, KAW'96*, Fiche démonstration, Banff, Canada.
- Fournier (D.), (1998): *Étude et conception d'un logiciel de gestion de base de connaissances terminologiques*, Mémoire d'ingénieur CNAM, Toulouse.
- Gros (C.), Assadi (H.), (1997): «Intégration de connaissances dans un système de consultation de documentation technique», dans *Acte des 1^{ères} journées du Chapitre Français de l'ISKO*. Lille, 16-17 oct. 1997, J. Maniez et W. Mustafa el Hadi (Eds), Édition du conseil scientifique de l'université Charles de Gaulle Lille 3, Collection «Travaux et Recherche».
- Lemaire (F.), Rechenmann (F.), (1995): «Intégration de connaissances terminologiques dans les grandes bases d'objets – Exemple en biologie moléculaire», dans *Banque des mots. n° spécial Terminologie et Intelligence Artificielle*, 7, p. 103-112.
- Martin (P.), (1995): «Knowledge Acquisition using Documents, Conceptual Graphs and a Semantically Structured Dictionary», dans *Proc. of the 8th Knowledge Acquisition Workshop, KAW'95*, Banff, Canada.
- Meyer (I.), Skuce (D.), Bowker (L.), Eck (K.), (1992): «Toward a new generation of terminological Ressources: An Experiment in Building a Terminological Knowledge Base», dans *Proc. of the 13th International Conference on Computational Linguistics*, Nantes, p. 956-960.
- Rastier (F.), (1995): «Le terme, entre ontologie et linguistique», dans *Banque des mots, n° spécial Terminologie et intelligence artificielle*, 7, p. 35-64.
- Rousselot (F.), Frath (P.), Oueslati (R.), (1996): «Extracting concepts and relations from corpora», dans *12th European Conference on Artificial Intelligence, ECAI'96; Workshop on Corpus-oriented semantic analysis*, Ed. By W. Wahlster, Pub. by John Wiley & Sons Ltd.
- Séguéla (P.), Aussenac-Gilles (N.), (1997): «Un modèle de base de connaissances terminologiques», dans *Actes des 2^e rencontres Terminologie et Intelligence Artificielle. TIA'97*, avril 1997. Toulouse, p. 47-68.
- Simon (S.), (1998): *Représentation formelle de connaissances issues d'une base de connaissances terminologiques*, Mémoire de DEA, RCFR de l'Univ. P. Sabatier, Toulouse.
- Skuce D., Lethbridge (T.C.), (1994): «CODE4: A multifunctional knowledge management system», dans *Proc. of the 8th Knowledge Acquisition Workshop, KAW'94*, Banff, Canada.
- Skuce (D.), (1998): «Intelligent Knowledge Management: Integration documents, Knowledge bases, Databases and Linguistic Knowledge», dans *Proc. of the 10th Knowledge Acquisition and management Workshop, KAW'98*, April 1998, Banff, Canada.
- Slodzian (M.), (1995): «Comment revisiter la doctrine terminologique aujourd'hui?», dans *Banque des mots, n° spécial Terminologie et intelligence artificielle*, 7, p. 11-18.