Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **STIC informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Julien ROCHE

Thèse dirigée par **Christophe Roche, PR1, Université Savoie Mont-Blanc**, et codirigée par **Yangli Jia, Professeur associé, Université de Liaocheng**

préparée au sein du **Laboratoire LISTIC, équipe Condillac** dans **l'École Doctorale SISEO – Sciences et Ingénierie des Systèmes de l'Environnement et des Organisations**

Le tournant ontologique de la Terminologie

Thèse soutenue publiquement le **20 Décembre 2018**,
devant le jury composé de :

Monsieur Christophe ROCHE, Professeur, Université Savoie Mont-Blanc, Directeur de thèse
Monsieur Yangli Jia, Professeur associé, University of Liaocheng – Department of Computer Science and Technology, Co-directeur de thèse
Madame Sylvie Desprès, Professeur, Université Paris 13, Rapporteur
Monsieur Hendrik Kockaert, Professeur, Hamad Bin Khalifa University, Qatar, Rapporteur
Madame Xiaomi An, Professeur, Renmin University of China, School of Information Resource Management, Examinateur
Madame Maria Papadopoulou, Chercheur associé, Université Savoie Mont-Blanc, Examinateur
Monsieur Chengyu Fang, Professeur associé, City University of Hong Kong - Department of translation and linguistics, Examinateur

Le Tournant Ontologique de la Terminologie

- Type de doctorat : STIC informatique
- Université Grenoble-Alpes - Ecole doctorale SISEO, Université de Liaocheng, KETRC (Chine)
- Champ disciplinaire : Ingénierie des connaissances et Terminologie
- Auteur : Julien Roche
- Titre de la thèse : Le Tournant Ontologique de la Terminologie
- Directeur de recherche : Professeur Christophe Roche
- Co-directeur : Dr. Yangli Jia
- Date de soutenance : Le 20 Décembre 2018
- Membres du jury :
  - Monsieur Christophe ROCHE, Professeur, Université Savoie Mont-Blanc, Directeur de thèse
  - Monsieur Yangli Jia, Professeur associé, University of Liaocheng – Department of Computer Science and Technology, Co-directeur de thèse
  - Madame Sylvie Desprès, Professeur, Université Paris 13, Rapporteur
  - Monsieur Hendrik Kockaert, Professeur, Hamad Bin Khalifa University, Qatar, Rapporteur
  - Madame Xiaomi An, Professeur, Renmin University of China, School of Information Resource Management, Examinateur
  - Madame Maria Papadopoulou, Chercheur associé, Université Savoie Mont-Blanc, Examinateur
  - Monsieur Chengyu Fang, Professeur associé, City University of Hong Kong - Department of translation and linguistics, Examinateur

Condillac Research Group – LISTIC Lab.
Campus Scientifique – Université Savoie Mont-Blanc
73 376 – Le Bourget du Lac cedex – France

KETRC – Dept. of Computer Science
Liaocheng University
252000 – Liaocheng City – Shandong - China

REMERCIEMENTS

Je tiens à remercier Christophe Roche, mon directeur de thèse, Professeur à l'Université Savoie Mont-Blanc, pour m'avoir fait partager le domaine de recherche auquel il a dédié une grande partie de sa vie. J'ai pu ainsi bénéficier de ses connaissances, de son point de vue averti sur un sujet complexe relevant de plusieurs disciplines. Il a su trouver le fil conducteur entre son expérience et la mienne pour aboutir à des résultats qui, j'espère, apporteront leurs contributions à un domaine où le numérique tient une place de plus en plus importante.

C'est aussi une thèse qui n'aurait sans doute pas pu avoir lieu sans le concours et le soutien de Jia Yangli, mon co-directeur de thèse, Professeur Associé à l'Université de Liaocheng. Il m'a accueilli au Département informatique de son Université et m'a offert l'opportunité d'y enseigner et d'y effectuer mes travaux de recherche dans les meilleures conditions.

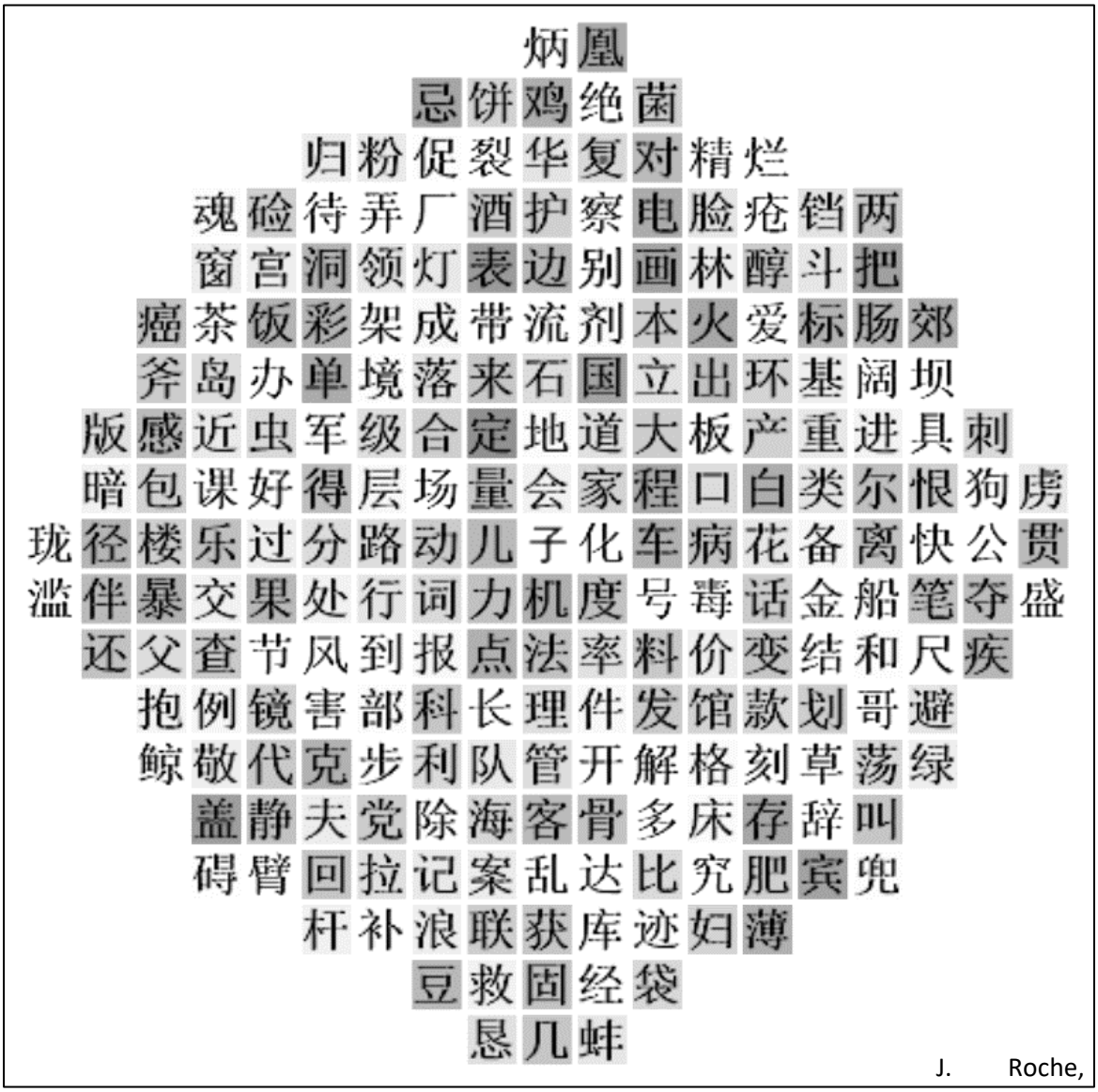
Je remercie également les personnes qui m'ont fait l'honneur de participer à mon jury, Madame Sylvie Desprès, Professeur à l'Université Paris 13, Monsieur Hendrik Kockaert, Professeur à l'Université Hamad Bin Khalifa au Qatar, Madame Maria Papadopoulou, Chercheur associé à l'Université Savoie Mont-Blanc, Madame Xiaomi An, Professeur à l'Université du Peuple de Pékin, Monsieur Chengyu Fang, Professeur associé à l'Université City de Hong Kong.

Je dédie cette thèse à mes parents qui m'ont toujours soutenu et encouragé, à mon fils, mais aussi à mon épouse pour m'avoir imprégné d'une autre culture et d'une autre langue.

LIMINAIRE

Cette thèse est faite en codirection, dans le cadre d'un accord interuniversitaire entre l'Université Savoie Mont-Blanc (Grenoble-Alpes) et l'Université de Liaocheng dans la province du Shandong en Chine. C'est pourquoi, on retrouvera un certain nombre de références à la langue chinoise.

Le caractère chinois, invariant et porteur de sens est à la base de la formation de termes plus complexes grâce une agrégation ordonnée de caractères bien choisis. Le terme chinois est alors une combinaison unique de caractères "essentiels" qui participent à sa définition.



J. Roche,

Ci-dessus, sont présents des caractères "racines" classés en fonction du nombre de mots qu'ils permettent de former. Au centre, figure le caractère impliqué dans la formation du plus grand nombre de mots ; coïncidence ou non, il signifie notamment "graine".

Table des matières

Introduction 14
Problématique : La Terminologie à l'ère numérique : une nécessaire opérationnalisation 14
Objectif : Proposer une théorie « consensuelle » du concept et une modélisation computationnelle, puis en étudier les conséquences sur la Terminologie dans ses principes et dans ses méthodes. .. 15
Contexte 15
ISO (TGT), Normalisation ISO (AFNOR) 15
L'Equipe de recherche Condillac, collaboration avec l'Université de Liaocheng 16
Structure de la Thèse 17
Plan de la thèse 20
Partie 1 : Etat de l'art 22
1.1. Ce qu'on entend par « Terminologie » 22
1.1.1. Définition de la Terminologie au sens général 22
1.1.2 La double dimension de la Terminologie 23
1.1.3 Une terminologie de corpus ou terminologie textuelle 25
1.1.4 Une terminologie conceptuelle 26
1.1.5 La Théorie générale de la Terminologie, ISO (TC37) 26
1.2 Ce qu'on entend par « Ontologie » 29
1.2.1 Introduction : L'Ontologie a deux acceptions 29
1.2.2 L'Ontologie au sens premier du terme 29
1.2.2.1 Etymologie 29
1.2.2.2 Définition 29
1.2.2.3 L'approche phénoménologique 29
1.2.2.4 L'approche essentialiste 30
1.2.2.5 Que retenir de ces deux approches pour notre travail ? 30
1.2.3 L'Ontologie de l'ingénierie des connaissances 31
1.2.3.1 Définition 31
1.2.3.2 Typologie et exemples 31
1.2.3.2.1 Core ontologie 32
1.2.3.2.2 Les ontologies universelles 32
1.2.3.2.3 Les ontologies de domaine 36
1.2.3.2.4 Les ontologies appliquées, dédiées à une tâche particulière 37

1.2.3.2.5 Les ontologies de type réseaux de termes 37
1.2.3.3 Langages de représentation de la connaissance 39
1.2.3.3.1 Langue naturelle 39
1.2.3.3.2 Langages semi-formels 40
1.2.3.3.2.1 la représentation graphique de la norme ISO 1087-1 41
1.2.3.3.2.2 UML pour la representation graphique de schémas conceptuels en Terminologie 42
1.2.3.3.3 Langages formels (langages de l'Intelligence artificielle) 43
1.2.3.3.4 Langages strictement-formels (Les logiques de description) 45
1.2.3.4 Protégé 45
1.2.3.6 Les formats d'échange informatique du W3C pour la représentation des connaissances 48
1.2.3.6.1 RDF (Ressource Description Framework) 49
1.2.3.6.2 RDFS (RDF Schema) 49
1.2.3.6.3 OWL (Ontology Web Language) 50
1.2.3.6.4 SKOS (Simple Knowledge Organization System) 51
1.2.3.7 Conclusion : quel langage choisir ? 51
Partie 2 : Le tournant ontologique de la Terminologie 54
2.1 Introduction 56
2.1.1 Tournant textuel 56
2.1.2 Intérêt pour l'Ontologie 56
2.1.2.1 Cogniterm 56
2.1.2.2 Termontographie de Rita Temmerman 57
2.1.2.3 Ressources termino-ontologiques 58
2.1.2.4 L'ontoterminologie 58
2.2 Tournant Ontologique 60
2.2.1 Introduction : lorsque le concept redevient le point de départ 60
2.2.2 Critiques de la norme ISO 1087-1 61
2.2.2.1 Les Normes iso 1087-1 :2000 et ISO 704 :2009 61
2.2.2.2 Problème de cohérence entre les normes 62
2.2.2.3 Problème de cohérence dans la manière de représenter le monde (Théorie du concept) 63
2.2.3 Proposition d'un modèle conceptuel 65
2.2.3.1 Modèle de concept (théorie du concept) : des principes épistémologiques à spécifier 65

2.3.1.1 Point de vue, domaine et sous-domaine 66
2.3.1.2 Caractéristique 66
2.3.1.2.1 Caractéristique essentielle 67
2.3.1.2.2 Caractéristique descriptive 68
2.3.1.3 Axe d'analyse (Type de caractéristique) 69
2.3.1.4 Concept 69
2.3.1.5 Classe 70
2.3.1.6 L'objet, connaissance singulière 71
2.3.1.7 Relation 72
2.3.1.7.1 Les relations entre les concepts 72
2.3.1.7.2 Les relations entre connaissance plurielle et connaissance singulière 78
2.3.1.7.3 Les relations entre les objets 79
2.3.1.8 Désignation 79
2.2.3.2 Caractéristiques essentielles : comment les représenter ? 79
2.2.3.2.1 Représentation des caractéristiques essentielles avec CmapTools 80
2.2.3.2.2 Représentation des caractéristiques essentielles avec des systemes à base de schémas 80
2.3.3.2.3 Représentation des caractéristiques essentielles avec Protégé 81
2.2.3.3 Nommer les concepts 81
2.2.3.3.1 Le terme chinois et sa formation 82
2.2.3.3.2 Le système qui régit la formation des termes chinois est-il adapté pour nommer les concepts ? 83
2.2.3.4 Nécessité d'un langage formel pour manipuler le système conceptuel 85
2.3 Apports pour la Terminologie 86
2.3.1 Apports Méthodologiques (ISO 704) 86
2.3.2 Alignement d'ontologies pour l'harmonisation des terminologies (ISO 860) 87
2.3.3 Applications informatiques 87
2.3.3.1 Thésaurus 88
2.3.3.2 Moteurs de recherche sémantiques multilingues de documents 89
Partie 3 : Mise en œuvre 92
3.1 Cahier des charges 94
3.1.1 La nécessité d'un langage de programmation interprété 94
3.1.2 Une architecture web client/serveur 95
3.1.3 Un modèle MVC 96
3.1.4 Une interface graphique flexible 97

3.1.5 Formats d'échange 98
3.1.6 Représentation graphique 98
3.2 Spécification UML 100
3.2.1 Représentation statique du système : le diagramme de classe UML 101
3.2.1.1 La classe UML 102
3.1.1.2 Les types de relations 102
3.1.2.1 La relation générique ou d'héritage 102
3.1.2.2 La relation d'association 103
3.1.2.3 La relation d'agrégation 103
3.1.2.4 La relation de composition 104
3.2.2 La représentation UML de la CLT 105
3.2.2.1 La classe « CLT » 105
3.2.2.2 La classe « Domain » 106
3.2.2.3 LA classe « Point of view » 108
3.2.2.4 La classe « Knowledge database » 109
3.2.2.5 La classe « Unit of Knowledge » 112
3.2.2.6 La classe « CLT concept » 112
3.2.2.7 La classe « CLT relation » 114
3.2.2.8 La classe « CLT object » 115
3.2.2.9 La classe « CLT characteristic » 117
3.2.2.10 La classe « Essential characteristic » 118
3.2.2.11 Les classes «Descriptive Characteristic» et «Object Descriptive Characteristic» 119
3.2.2.12 La classe « Axis of analysis » 120
3.2.2.13 La classe « Category » 122
3.2.2.13 La classe « CLT class » 123
3.2.2.14 La classe « Designation » 124
3.2.2.15 La classe « Metadata » 126
3.2.2.16 Représentation dynamique du système : un exemple d'instanciation 126
3.3 Implémentation en PHP5 129
3.3.1 Architecture 129
3.3.2 Une implémentation dans une perspective d'extension du système 130
3.3.3 La représentation d'un système conceptuel 131
3.3.4 L'implémentation des relations, exemple d'implémentation de la relation générique 133
3.3.5 La gestion des erreurs 134

3.3.6 L'implémentation de la vue (l'interface utilisateur) 135
3.3.7 L'implémentation du contrôleur (la gestion des évènements) 137
3.3.8 L'implémentation de fonctions d'analyse du système conceptuel 138
3.3.9 L'implémentation des fonctions de visualisation 139
3.3.10 Les problèmes de sécurité liés à l'utilisation de la fonction PHP « eval() » 144
3.4 Formats d'échange 145
3.4.1 Format d'export pour CmapTools 145
3.4.2 Format d'export pour RDF 146
3.4.3 Format d'export pour RDFS 147
3.4.4 Format d'export pour OWL 147
3.5 Application : opérationnalisation de dictionnaires 148
3.5.1 Conversion des dictionnaires CEDICT et CFDICT en instructions CLT 148
3.5.2 Visualisation de la terminologie des établissements 150
3.5.2.1 Affichage de la définition formelle d'un concept 150
3.5.2.2 Génération d'un glossaire en anglais 151
3.5.2.3 Affichage du système conceptuel 152
3.5.3 Conclusion 153
Conclusion 154
ANNEXES 156
Annexe A : Tableaux des correspondances entre la norme ISO 1087-1 et la notation adoptée dans la norme ISO 24156-1 156
Annexe B : Cas d'utilisation de la CLT : construction de la terminologie des sièges 158
Les étapes de construction de la Terminologie des sièges avec la CLT 159
La définition d'un domaine 159
La définition d'un point de vue 159
La définition des axes d'analyse 159
La définition des caractéristiques descriptives 160
La définition des concepts 160
L'affectation des caractéristiques essentielles et descriptives au niveau des concepts 160
La mise en relation des concepts 160
L'affectation des désignations au niveau des concepts 160
Visualisation de la terminologie des sièges 161
Affichage de la définition formelle d'un concept 161
Affichage de glossaires en fonction d'une langue donnée 162

Affichage du système conceptuel 163
Annexe C : Le logiciel TreeWords 164
Bibliographie..... 166

INTRODUCTION

PROBLEMATIQUE : LA TERMINOLOGIE A L'ERE NUMERIQUE : UNE NECESSAIRE OPERATIONNALISATION

Avec l'avènement de la société d'information dans l'ère du numérique, nos méthodes de travail ont profondément changé et la gestion des connaissances est devenue un enjeu majeur. De nouvelles pratiques liées au traitement de l'information sont rendues possibles grâce à l'opérationnalisation des terminologies telles que les moteurs de recherche sémantique multilingues, les dictionnaires numériques spécialisés, les applications liées au web sémantique, etc.

L'opérationnalisation des terminologies à des fins de traitement de l'information requiert une représentation computationnelle du système conceptuel. La théorie du concept sur laquelle se fonde la Terminologie ISO ne permet pas aujourd'hui une telle représentation informatique [Roche 2012]. Même si les normes ISO sur la Terminologie n'ont pas pour objectif l'opérationnalisation de terminologies mais la communication entre humains, elles doivent pouvoir – elles le précisent – servir à la modélisation des informations et des données : « L'objectif du travail terminologique, tel que décrit dans le présent document, est par conséquent la clarification et la normalisation des concepts et des terminologies pour la communication entre humains. Le travail terminologique peut servir à la modélisation des informations et des données, toutefois la présente Norme internationale ne traite pas des relations avec ces domaines. » [ISO 704 :2009, p.V].

Les résultats de disciplines telles que l'ingénierie des connaissances ont permis de mettre en évidence la nécessité de disposer d'une théorie du concept qui puisse donner lieu à une représentation informatique. Dans ce cadre, les ontologies, issues de l'ingénierie des connaissances, constituent une des perspectives les plus intéressantes pour la modélisation du système conceptuel d'une terminologie [Roche, 2005]. Elles ouvrent la voie à un tournant de la Terminologie que l'on peut qualifier de « ontologique », ce qui n'est pas sans conséquence sur la Terminologie elle-même, ses principes et ses méthodes (voir figure 1).

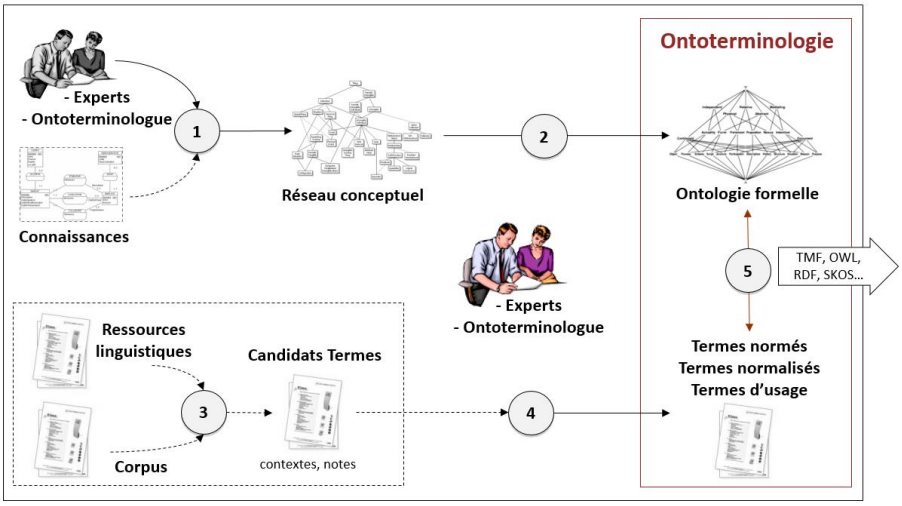


Figure 1: méthodologie de construction de terminologies centrées sur le concept

OBJECTIF : PROPOSER UNE THEORIE « CONSENSUELLE » DU CONCEPT ET UNE MODELISATION COMPUTATIONNELLE, PUIS EN ETUDIER LES CONSEQUENCES SUR LA TERMINOLOGIE DANS SES PRINCIPES ET DANS SES METHODES.

Ces nouvelles applications (moteurs de recherche sémantique multilingues, dictionnaires numériques spécialisés, applications liées au web sémantique, etc.) requièrent une opérationnalisation de la Terminologie. Elles mettent en avant le concept et la nécessité d'en avoir une représentation computationnelle. La notion de concept diffère selon le point de vue qui peut être linguistique, terminologique ou celui de l'ingénierie des connaissances. La théorie du concept de la Terminologie n'est pas opérationnalisable telle qu'elle est définie actuellement dans l'ISO [Roche, 2012]. Les objectifs de cette thèse sont multiples. Il s'agira dans un premier temps de proposer une théorie du concept qui tout en restant dans une filiation « Wüsterienne » soit « consistante » tant d'un point de vue épistémologique que logique, puis d'en proposer une modélisation computationnelle et d'en étudier les conséquences sur la Terminologie elle-même dans ses principes et dans ses méthodes. L'étude se terminera avec la présentation d'un exemple d'application et des perspectives de recherche.

CONTEXTE

ISO (TGT), NORMALISATION ISO (AFNOR)

Cette thèse se place dans le cadre de la Théorie Générale de la Terminologie et des normes ISO (International Organisation for Standardization) pour la Terminologie. La Terminologie est, dans ce cadre, régie par deux normes transversales : la norme ISO 1087-1 :2000 (Terminology work – vocabulary – part1 : Theory and application) et la norme ISO 704 (Terminology work – principles and methods). Ces deux normes, comme leur titre l'indique, décrivent le vocabulaire, les principes et les méthodes de la Terminologie dite « classique » prenant ses racines dans la Théorie Générale de la Terminologie [Felber, 1987]. Notre étude tient compte de la révision en cours de la norme ISO 1087-1 et des travaux menés au sein de la commission AFNOR/X03A « Terminologie : principes et coordination » de l'AFNOR.

Au niveau national, l'AFNOR (Association française de normalisation) est l'organisation en charge des normes et processus de normalisation (l'équivalent du CNIS[1] pour la Chine et le DIN[2] pour l'Allemagne). Elle représente la France auprès de l'ISO et du CEN (Comité européen de normalisation). La commission AFNOR/X03A « Terminologie principes et coordination », présidée par Christophe Roche (Université Savoie Mont-Blanc), est la commission nationale miroir de l'ISO/TC 37.

[1] China National Institute of Standardization

[2] Deutsches Institut für Normung

L'EQUIPE DE RECHERCHE CONDILLAC, COLLABORATION AVEC L'UNIVERSITE DE LIAOCHENG

Cette thèse s'inscrit dans un contexte de collaboration internationale (thèse en codirection) entre l'Équipe de recherche Condillac[3] « Ontologie et Terminologie » du laboratoire LISTIC[4] de l'Université Savoie Mont-Blanc (http://new.condillac.org/) et le Centre de Recherche en Ingénierie des Connaissances et en Terminologie de l'Université de Liaocheng (http://ketrc.com/ Province du Shandong, Chine). C'est pourquoi certains points (dénomination des concepts, application) sont illustrés en chinois.

L'Université Savoie Mont-Blanc et l'Université de Liaocheng ont établi une collaboration étroite sur la Terminologie, initiée dans le cadre d'un programme de coopération scientifique Xu Guangqi 2015 portant sur la norme ISO 1087-1. Cette collaboration a abouti à un accord-cadre de coopération interuniversitaire, à la mise en place de séminaires communs de recherche, de codirections de thèse ainsi qu'à la création d'un Centre de Recherche en Ingénierie des Connaissances et en Terminologie. KETRC[5], acronyme de « Knowledge Engineering & Terminology Research Center » est un nouveau centre de recherche créé le 5 Novembre 2017 avec la collaboration du Professeur Christophe Roche de l'Université Savoie Mont-Blanc pour le diriger. Les domaines de recherche du KETRC se situent dans la continuité de ceux initiés par l'équipe de recherche Condillac.

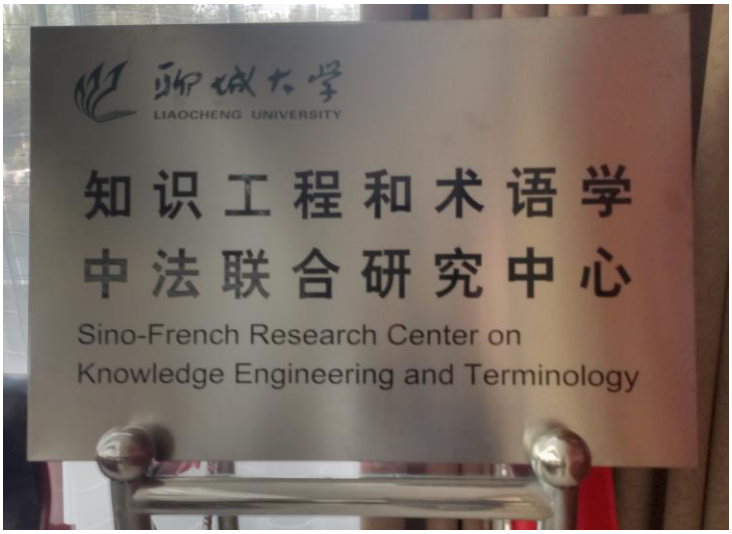


Figure 2 : KETRC, inauguré le 5 Novembre 2017 à l'Université de Liaocheng

[3] L'équipe Condillac est aussi en charge des Conférences internationales TOTh (Terminology & Ontology: Theories and applications) créées en 2007 : http://toth.condillac.org/ , et du premier Diplôme d'Université en Terminologie : http://toth.fr.condillac.org/du-terminologie

[4] Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (https://www.listic.univ-smb.fr/).

[5] « Sino-French Knowledge Engineering & Terminology Research Center » inauguré le 5 Novembre 2017 à l'Université de Liaocheng, adresse du site internet du KETRC : http://www.ketrc.com

STRUCTURE DE LA THESE

La première partie de cette thèse est dédiée à un état de l'art sur la Terminologie et l'Ontologie. Nous verrons ce qu'on entend par Terminologie, sa double dimension (linguistique et conceptuelle), ses différentes approches (textuelle et conceptuelle) et telle qu'elle est introduite dans la Théorie Générale de la Terminologie et dans les normes ISO-1087-1 et 704. L'accent sera mis sur la présentation des principes qui régissent le modèle conceptuel de la Terminologie.

L'Ontologie, au sens de l'ingénierie des connaissances, c'est-à-dire une définition des concepts et de leurs relations dans un langage intelligible par un ordinateur, constitue une des voies les plus prometteuses pour la Terminologie. Après avoir présenté ce que l'Ontologie au sens premier du terme (philosophique) peut apporter à la modélisation des connaissances, nous détaillerons les différents types d'ontologies issues de l'Intelligence Artificielle et les différents types de langages de représentation de la connaissance selon leur degré de formalisation.

La deuxième partie présente notre travail théorique. Elle est consacrée au tournant ontologique de la Terminologie. L'utilisation de l'informatique en Terminologie n'est pas nouvelle. L'analyse automatique de corpus à des fins linguistiques (extraction de candidats termes, de collocations, recherche de relations définitoires, études de concordances, etc.) a fortement impacté la Terminologie dans ses fondements et ses principes. Elle a donné lieu à un premier tournant, le « tournant textuel » de la Terminologie où on s'est davantage intéressé aux termes qui désignent des concepts en discours qu'aux concepts eux-mêmes posés comme extra linguistiques. L'opérationnalisation de terminologies à des fins de traitement de l'information, telle que nous l'entendons ici, est une problématique différente. Elle s'appuie sur une représentation computationnelle du système notionnel qui replace le concept au premier plan. C'est le « tournant ontologique » de la Terminologie qui remet en question certains principes et méthodes de la Terminologie. L'Ontologie constitue, dans un monde numérique, une des perspectives les plus prometteuses pour la Terminologie[6]. La notion « d'ontoterminologie », terminologie dont le système conceptuel est une ontologie formelle, en est directement issue.

On verra dans un premier temps les problèmes, tant épistémologiques que logiques, posés par la Théorie du Concept de la Terminologie telle qu'elle est définie dans les normes ISO 1087-1 et ISO 704, qui entravent une mise en œuvre informatique. De ce constat, nous proposerons une Théorie du Concept et un langage associé nommé CLT (Concept Language for Terminology) qui, tout en respectant dans la mesure du possible les principes de la Terminologie dite « classique », pourra donner lieu à des représentations computationnelles du système conceptuel. On introduira

[6] Savoir si on peut construire une ontologie à partir de textes est un autre problème qui ne relève pas de notre travail, et ce d'autant plus que les ontologies construites à partir de textes ne correspondent en général pas à celles établies directement par les experts du domaine dans un langage formel [Rastier 2004], [Roche 2007].

notamment une couche d'abstraction supplémentaire qui permet de distinguer les objets (instances, individus), des concepts portant sur une pluralité d'objets pouvant être éventuellement vide[7].

Dans l'approche « ontoterminologique », les *termes* et les *noms de concept* ne se confondent pas puisqu'ils relèvent de systèmes sémiotiques différents. Les *noms des concepts* en plus d'être uniques, ne sont pas limités au seul rôle *d'identifiants* pour le système formel. Ils ont une importance particulière dans un système conceptuel car ils doivent (devraient) refléter ou pouvoir indiquer la place des concepts qu'ils représentent où la nature des objets subsumés par les concepts. La manière de nommer les concepts est donc loin d'être arbitraire et peut s'inspirer de certaines langues naturelles, comme le chinois, où un mot est formé par combinaison de caractères qui représentent les caractéristiques essentielles du concept dénoté par le mot. Le système qui régit la formation des mots en chinois peut être adopté pour nommer les concepts, le terme normé pourra alors se calquer avec profit sur le nom du concept. Le programme Treewords a été développé afin d'illustrer le système de formation des termes chinois liés aux connaissances qu'ils dénotent.

Enfin, il sera question des apports méthodologiques du tournant ontologique sur la Terminologie elle-même. Expliciter le concept en mettant l'accent sur la notion de caractéristique essentielle n'est pas sans conséquence sur la façon de concevoir des terminologies, mais aussi, à travers l'alignement d'ontologies, de les harmoniser (la norme ISO 860 n'étant toujours pas révisée). Cette approche ouvre également la voie à l'opérationnalisation des terminologies comme par exemple une recherche sémantique multilingue.

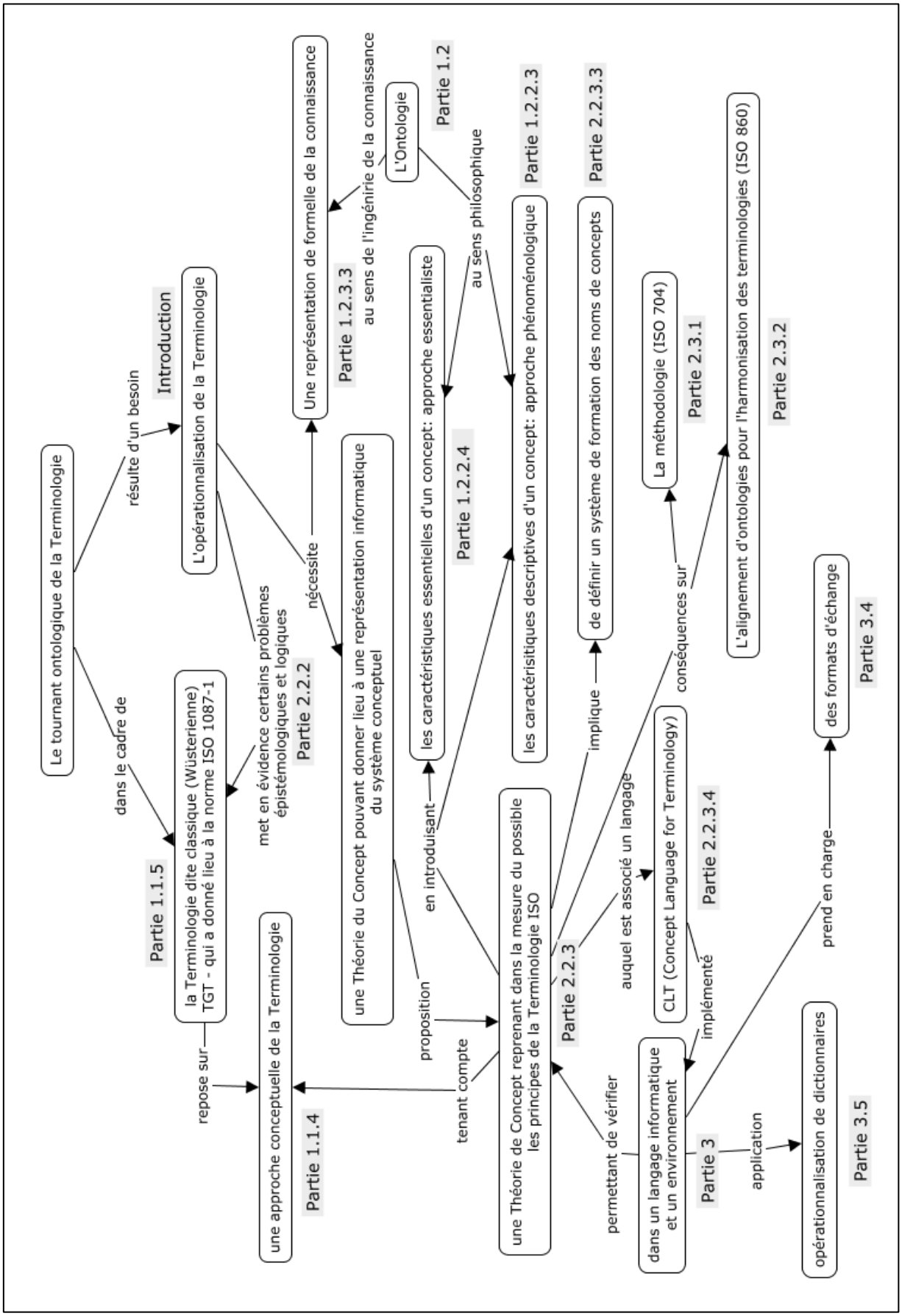
La troisième partie est consacrée à une mise en œuvre des travaux théoriques. Nous proposerons un langage informatique dédié à la CLT (Concept Language for Terminology) apte à construire un système conceptuel (définition des concepts, de leurs relations…) ce qui en soi, constitue une validation de la théorie. La CLT une fois implémentée, pourra être vue comme la spécialisation d'un langage de programmation pour la Terminologie. Le langage CLT servira à la fois de langage dédié à la création et à la manipulation de terminologies computationnelles et de format d'échange puisque compréhensible par un ordinateur. Le choix d'une architecture Client / Serveur pour l'implémentation de la CLT permet non seulement son utilisation à travers un navigateur web, mais aussi d'envisager un aspect collaboratif dans la construction du système conceptuel. Une interface graphique sera proposée afin d'établir une terminologie computationnelle sans notion de programmation. Le rôle de cette interface sera de transcrire l'action de l'utilisateur en instructions CLT. Le système conceptuel réalisé à l'aide de la CLT pourra être visualisé directement dans l'environnement et être exporté dans différents formats.

La mise en œuvre de la CLT sera illustrée à travers un exemple d'application qui montrera l'opérationnalisation d'un dictionnaire multilingue pour un domaine donné. Le système conceptuel sera construit en se servant du principe de formation des termes chinois où les termes traduisent dans leur forme ce qu'ils désignent. Cette opérationnalisation consistera à convertir les entrées du dictionnaire en instructions du langage CLT. Ce type d'opérationnalisation, basé sur le modèle de la

---

[7] Par exemple, le concept <Satellite naturel de la lune> ne correspond à aucun objet et ne peut donc, même si le concept (idée) existe, être pris en compte par les normes ISO qui font dépendre la définition du concept de la cardinalité de son extension.

CLT, pourrait ouvrir la voie à la conversion de dictionnaires spécialisés, dont les entrées sont des termes chinois, vers des terminologies computationnelles. Cela permettrait, entre autres, de vérifier la cohérence de la définition d'un terme ou de s'assurer du bien-fondé du choix d'un terme.


Le tournant ontologique de la Terminologie
résulte d'un besoin
Introduction
L'opérationnalisation de la Terminologie
dans le cadre de
Partie 1.1.5
la Terminologie dite classique (Wüsterienne)
TGT - qui a donné lieu à la norme ISO 1087-1
met en évidence certains problèmes
épistémologiques et logiques
Partie 2.2.2
nécessite
Une représentation de formelle de la connaissance
Partie 1.2.3.3
au sens de l'ingénirie de la connaissance
L'Ontologie
Partie 1.2
au sens philosophique
Partie 1.2.2.3
Partie 2.2.3.3
une Théorie du Concept pouvant donner lieu à une représentation informatique
du système conceptuel
les caractéristiques essentielles d'un concept: approche essentialiste
Partie 1.2.2.4
les caractérisitiques descriptives d'un concept: approche phénoménologique
de définir un système de formation des noms de concepts
La méthodologie (ISO 704)
Partie 2.3.1
L'alignement d'ontologies pour l'harmonisation des terminologies (ISO 860)
Partie 2.3.2
conséquences sur
implique
en introduisant
proposition
une Théorie de Concept reprenant dans la mesure du possible
les principes de la Terminologie ISO
Partie 2.2.3
auquel est associé un langage
CLT (Concept Language for Terminology)
Partie 2.2.3.4
implémenté
des formats d'échange
Partie 3.4
prend en charge
repose sur
une approche conceptuelle de la Terminologie
Partie 1.1.4
tenant compte
permettant de vérifier
dans un langage informatique
et un environnement
Partie 3
application
opérationnalisation de dictionnaires
Partie 3.5

PARTIE 1 : ETAT DE L'ART

1.1. CE QU'ON ENTEND PAR « TERMINOLOGIE »

1.1.1. Définition de la Terminologie au sens général

La terminologie au sens de système de termes est apparue dès 1837 avec la définition de Whewell (épistémologue et moraliste anglais) mais qui reste restreinte à un groupe de sciences : « système des termes employés dans la description des objets de l'histoire naturelle » [Rey, 1979]. La terminologie est alors un système de termes employés à dénommer des notions ou concepts. Une définition intéressante de « notion » est apparue en 1977 par le biais de la proposition russe de la révision de la recommandation ISO R704 : « Ensemble cohérent de jugements sur un objet, dont le noyau est constitué par les jugements reflétant les caractères inhérents de l'objet » [Rey, 1979]. Il faut comprendre ici le terme *jugement* comme opération logique. Une terminologie est un système de termes qui reflète une modélisation conceptuelle [Roche, 2005], puisque le terme permet de combiner un nom et la désignation d'un concept. La définition de terminologie telle qu'elle est définie dans la norme ISO 1087-1 (Figure 3 et 4) traduit bien cette idée, puisqu'une désignation est la représentation d'un concept par un signe qui le dénote, et un terme est un type de désignation.

3.5 Terminology	3.5 Terminologie
3.5.1 **terminology 1** set of **designations** (3.4.1) belonging to one **special language** (3.1.3)	**3.5.1** **terminologie 1** ensemble des **désignations** (3.4.1) appartenant à une **langue de spécialité** (3.1.3)

Figure 3 : définition de la terminologie dans la norme ISO 1087-1 (AFNOR)

3.5.1 **术语集 terminology 1**
专门语言(3.1.3)中的一组指称(3.4.1)的集合。

Figure 4 : équivalent en chinois de la définition de terminologie dans la norme 1087-1 (CNIS)

Il faut distinguer la Terminologie[8] en tant que discipline de la terminologie en tant qu'ensemble de désignations, résultat de l'application de cette discipline à un domaine donné (voir Figure 7). La Terminologie a eu du mal à se faire reconnaître en tant que discipline à part entière « This book

[8] Bien que la Terminologie ISO vise à imposer l'univocité des termes, le terme lui-même de « terminologie » ne l'est pas puisqu'il désigne aussi bien la terminologie en tant que discipline que la terminologie en tant que système de termes. En chinois, la distinction est clairement faite, le terme « Terminologie » (本语学) en tant que science contient le mot « étude » (学) et le terme terminologie désignant le système de termes(本语集) est construit avec le mot « collection » (集).

denies the independent status of terminology as a discipline » [Sager 1990 p1], même de nos jours [Slodzian, 2000]. Il faut attendre 1978 [Rey, 1979] pour avoir en français la définition de la Terminologie avec le sens dans lequel il est employé actuellement dans la communauté qui nous intéresse ici. L'édition de 1978 du Petit Robert définit la Terminologie comme étant « l'Etude systématique des termes […] servant à dénommer classes d'objets et concepts… ; principes généraux qui président à cette étude. »

La Terminologie est définie dans la norme ISO comme la « science étudiant la structure, la formation, le développement, l'usage et la gestion des terminologies dans différents domaines » (voir Figure 5 et 6).

3.5.2 **terminology 2** terminology science science studying the structure, formation, development, usage and management of **terminologies** (3.5.1) in various **subject fields** (3.1.2)	**3.5.2** **terminologie 2** science de la terminologie science étudiant la structure, la formation, le développement, l'usage et la gestion des **terminologies** (3.5.1) dans différents **domaines** (3.1.2)

Figure 5 : définition de la terminologie dans la norme ISO 1087-1 (AFNOR)

3.5.2 **术语学 terminology 2**
术语科学
研究各专业领域(3.1.2)中术语(3.4.3)的结构、形成、发展、用法和管理的学科。

Figure 6 : équivalent en chinois de la définition de terminologie dans la norme 1087-1 (CNIS)

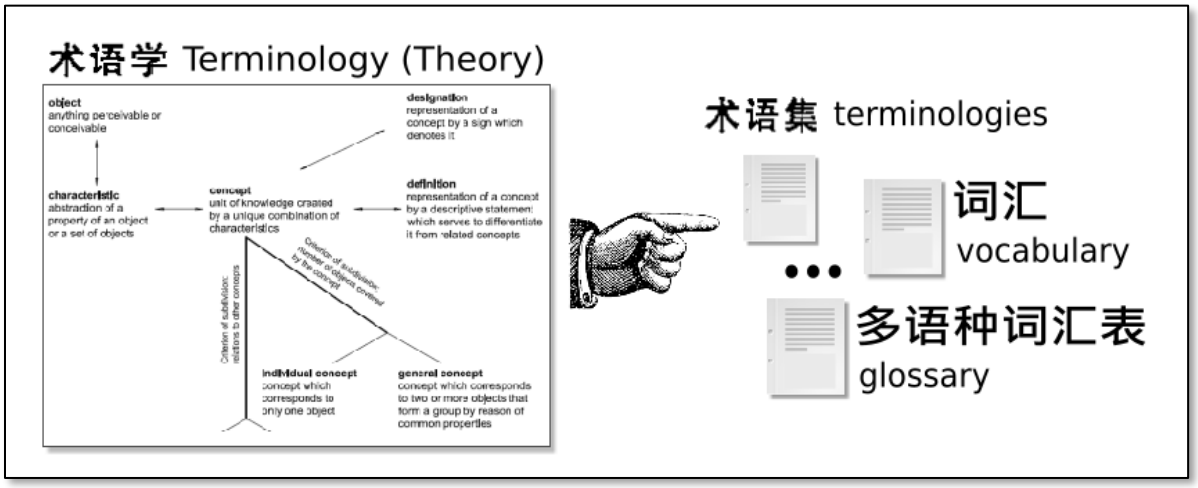


Figure 7 : différence entre Terminologie (théorie) et terminologie (système de termes)

1.1.2 LA DOUBLE DIMENSION DE LA TERMINOLOGIE

En tant que discipline scientifique à part entière, la Terminologie se base autant sur la linguistique que sur l'épistémologie pour la théorie de la connaissance : « Le travail terminologique est pluridisciplinaire et a recours à de nombreuses disciplines (logique, épistémologie, philosophie des sciences, linguistique, études en traduction, sciences de l'information et sciences cognitives) dans

l'étude des concepts et de leur représentation dans une langue de spécialité ou une langue générale. Il combine les éléments provenant de différentes approches théoriques traitant de la description, de l'organisation et du transfert des connaissances. » [Norme ISO 704, p5]. « La Terminologie est autant une science des mots qu'une science des choses » [Roche 2007] et l'on ne peut la réduire à l'une ou à l'autre, au risque de faire de l'ingénierie de la connaissance si on se focalise sur la représentation du concept ou bien de la lexicographie de spécialité si on oublie le concept au profit du terme[9] . Il faut aussi prendre conscience que le réseau de termes n'est pas superposable au réseau de concepts : "le lexique des langues ne reflète pas la conception scientifique du monde" [Rastier, 2004], « dire n'est pas concevoir » [Roche 2007], car la théorie qui régit ces deux systèmes sémiotiques, linguistique et conceptuel, est différente. On ne manipule pas les mots d'un langage naturel de la même manière que les concepts d'un langage artificiel.

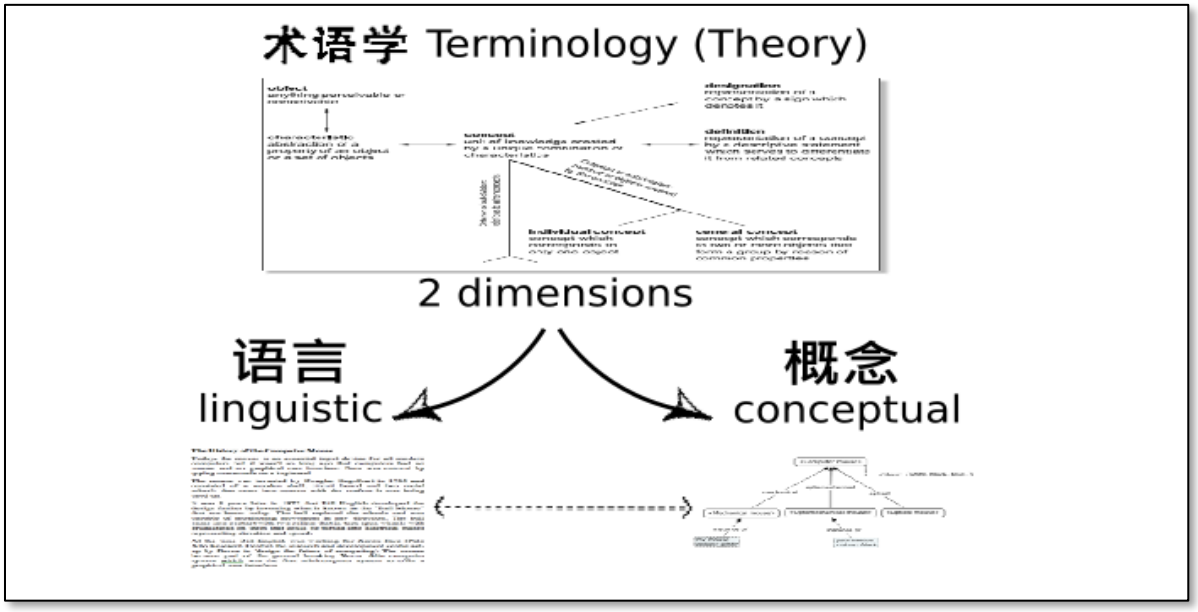


Figure 8: la double dimension de la Terminologie

Il faut donc distinguer les dimensions conceptuelle et linguistique qui composent toutes terminologies. On parle alors de la double dimension de la Terminologie [Roche 2007 ; 2012], [Costa 2013]. Le terme défini comme une « désignation verbale d'un concept général dans un domaine spécifique » ISO 1087-1 permet de relier ces deux dimensions. L'explicitation de ces deux dimensions permet de préserver la diversité langagière sur la base d'une modélisation conceptuelle partagée.

En Terminologie, on distingue deux approches, selon que l'accent est mis sur le terme ou sur le concept. L'une est dite sémasiologique où le terme est le point de départ et l'autre basée sur une démarche onomasiologique, c'est-à-dire que l'on prend comme point de départ le concept pour s'intéresser ensuite à ses désignations (termes). Deux approches différentes mais qui peuvent néanmoins se compléter.

[9] « Que l'on oublie le terme, le nom, pour décrire un mot dans son fonctionnement, que l'on cherche à atteindre par la définition le signifié de ce mot, et non pas la notion correspondant au terme, et l'on cesse (à son insu) de faire de la terminologie (-graphie) » [Rey, 1979, p. 104]

En s'inspirant du triangle sémiotique de la linguistique de Richards et Ogden (Figure 10), les terminologues de l'Ecole de Vienne adoptent un triangle sémiotique objet-concept-signe (Figure 9). Dans ce triangle sémiotique, le concept représente un savoir sur un objet, un objet est conceptualisé par un concept exprimé par un signe et « tout objet a pour nom un signe » [Lerat 89 : 56] cité dans [Otman, 1996].

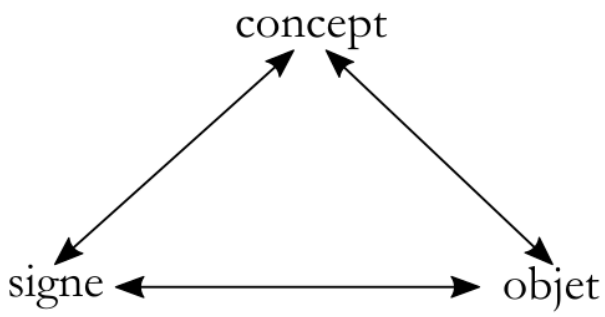


Figure 9 : Triangle sémiotique adopté par les terminologues de l'Ecole de Vienne

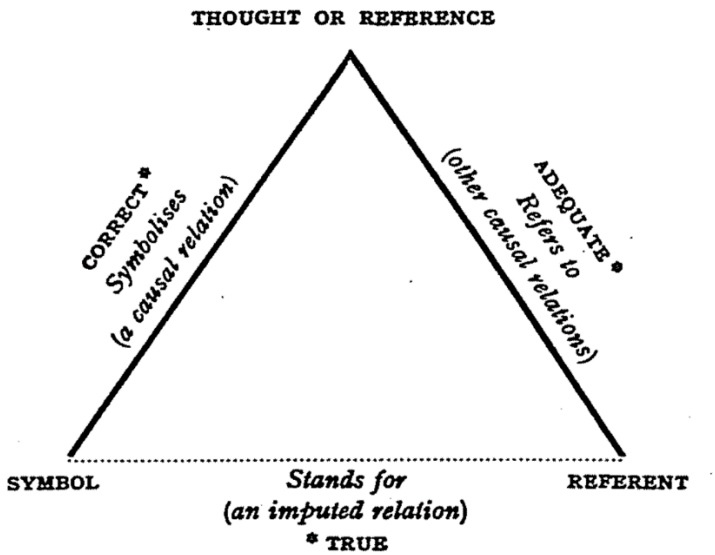


Figure 10 : Le triangle sémiotique de Richards et Ogden

1.1.3 UNE TERMINOLOGIE DE CORPUS OU TERMINOLOGIE TEXTUELLE

La Terminologie que l'on qualifiera ici de « textuelle » car s'appuyant principalement sur des corpus textuels qui constituent le point de départ, a connu et connaît encore un essor considérable. « Ce n'est que récemment, sous la pression de différents paramètres, que la constitution de terminologie à partir de textes a pris un essor considérable » [Condamines, 2005]. Elle exploite à différentes fins des méthodes de la linguistique de corpus liées au traitement automatique de la langue (TAL) mais aussi des méthodes statistiques liées à une sémantique distribution : extraction de candidats termes, étude des marqueurs linguistiques de relations conceptuelles (hiérarchiques, partitives, associatives) [Buitelaar et al., 2005]. Une communauté d'experts s'accordera sur les termes à retenir. Le texte

étant le support de la connaissance et les applications de la Terminologique surtout textuelles, il parait donc logique de ne pas s'éloigner des textes pour mieux y retourner [Bourigault, 1999] [Després & Szulman, 2008]. On parle dans ce contexte d'approche descriptive (étude de l'usage des termes dans le discours).

« La Terminologie n'a pas toujours fait très bon ménage avec les textes. La tradition wüstérienne, a même mis en garde contre l'utilisation de productions réelles pour constituer des terminologies. » [Condamines, 2005].

Puisque « les choses ne sont pas telles qu'elles sont dites » [Roche 2007] ou décrites, le risque est d'aboutir à un système conceptuel contingent, dépendant d'un corpus donné, ce qui ne correspond pas à une démarche normative telle que prônée par l'ISO. La Terminologie ne se limite pas à une lexicographie de spécialité [Rey, 1979]. Le besoin d'avoir une représentation de la connaissance indépendante de la langue naturelle semble inévitable, on parlera alors de « Terminologie conceptuelle ». C'est dans cette deuxième optique que nous nous placerons.

1.1.4 UNE TERMINOLOGIE CONCEPTUELLE

Après avoir été fortement critiquée, et parfois avec raison [Candel 2004], [Humbley 2004], [Van Campenhoudt 2006], la Terminologie que l'on qualifiera ici de « conceptuelle » prend pour point de départ le concept. Cette démarche onomasiologique s'attache à d'abord établir une modélisation conceptuelle d'un domaine pour ensuite relier les termes à des concepts. Lorsque des concepts se retrouvent sans terme(s) associé(s), de nouveaux termes peuvent être créés (néologismes) si le besoin s'en fait sentir[10]. On parle d'approche prescriptive ou normative en désignant certains termes comme dénominations standardisées. La difficulté de cette approche est de pouvoir modéliser un système de connaissances, ce qui nécessite dans tous les cas une théorie du concept et un langage spécifique dédié à la représentation et la manipulation des connaissances. Les travaux d'Eugène Wüster sont considérés comme le fondement de la théorie classique de la Terminologie en établissant une théorie générale de la Terminologie (TGT) reposant sur le concept, des travaux détaillés par son disciple Felber dans son Manuel de Terminologie [Felber, 1987].

1.1.5 LA THEORIE GENERALE DE LA TERMINOLOGIE, ISO (TC37)

La Théorie Générale de la Terminologie (TGT) reprise dans l'ISA[11] puis l'ISO dans le cadre d'une normalisation de la Terminologie, résulte des travaux de recherche entrepris par Eugègne Wüster (1898-1977), ingénieur de formation et terminologue autrichien. Il porta tout d'abord un grand intérêt à l'Esperanto et publia de nombreux articles sur les questions terminologiques et lexicographiques de l'Esperanto. Entre 1918 et 1920, alors qu'il est encore étudiant, il réalisa une encyclopédie Esperanto-germanique avec un niveau de détails encore inégalé à ce jour. Il poursuit

[10] L'existence de concepts sans désignation (vide « linguistique ») ne veut pas dire qu'on ne peut pas y accéder par paraphrases : « les sièges pour une personne » par exemple.

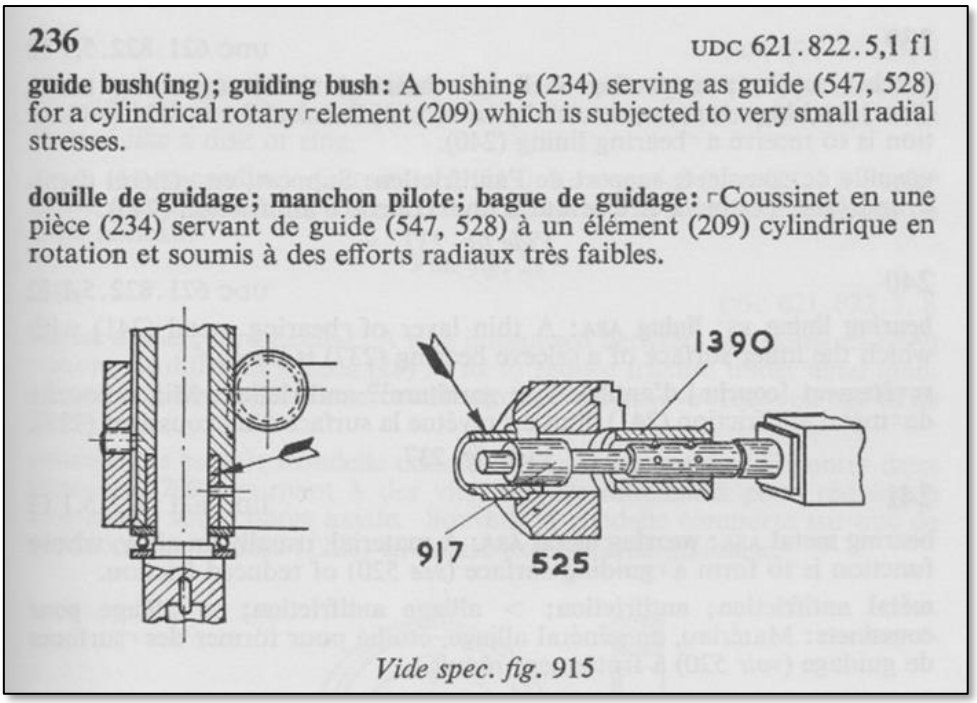
[11] L'ISA (International Federation of National Standardizing Associations) organisme qui devint après la seconde guerre mondiale l'ISO (International Organization for Standardization).

ses recherches sur la manière d'élaborer des documentations techniques multilingues qui constitue l'objet d'étude de sa thèse (International language standardization in technology, particularly in electronics) à l'Université de Stuttgart où il obtient le grade de docteur en 1931. [Campo, 2013]

Durant cette période, Wüster participa activement au comité technique de l'institut autrichien des standards et contribua considérablement au DIN (institut allemand pour la standardisation) sur les principes terminologiques.

A l'instigation de l'Union Soviétique, le livre de Wüster « International Standardization of Technical Terminlogy », publié en 1931 est à l'origine de la création du comité technique 37 « Terminologie » au sein de l'ISA (International Federation of National Standardizing Associations) qui devint après la seconde guerre mondiale l'ISO. [Rey 1979]

Son expérience acquise au cours de son expérience terminographique conduira à la publication du dictionnaire technique multilingues de la machine outils [Wüster, 1968], un dictionnaire originalement en anglais et français avec un supplément en allemand répertoriant 1401 concepts [Campo, 2013]. Ce dictionnaire est particulièrement innovant pour l'époque dans la mesure où il fait correspondre à chaque concept du domaine une représentation du concept dans un langage artificiel (schéma), une désignation du concept par des termes et une définition en langage naturel [Felber 1980].


236 UDC 621.822.5,1 f1

guide bush(ing); guiding bush: A bushing (234) serving as guide (547, 528) for a cylindrical rotary ⌐element (209) which is subjected to very small radial stresses.

douille de guidage; manchon pilote; bague de guidage: ⌐Coussinet en une pièce (234) servant de guide (547, 528) à un élément (209) cylindrique en rotation et soumis à des efforts radiaux très faibles.

Vide spec. fig. 915

Figure 11 : Dictionnaire Multilingue de la Machine-outils (Wüster 1968)

En 1975, Eugène Wüster présente à l'Université de Vienne en Autriche, la « Théorie Générale de la Terminologie – un domaine interdisciplinaire impliquant la linguistique, la logique, l'ontologie, l'informatique et les sciences des objets ». Les recherches de Wüster pour élaborer la Théorie

Générale de la Terminologie (TGT) sont, au dire de Felber, basées sur les théories de Saussure[12] (la différence entre la langue comme système et le discours, lexicologie structurale) et de Schlomann (pour le caractère systématique des termes de spécialité). Avec l'élaboration de la TGT, Wüster fait de la Terminologie une discipline « autonome » en définissant les relations impliquant à la fois la linguistique, la logique, l'ontologie, l'informatique et les sciences des choses.

L'objectif des travaux de Wüster est d'éliminer les ambiguïtés qui pouvaient apparaître dans l'usage de la langue naturelle, et de permettre ainsi une normalisation de la description des connaissances d'un domaine d'application. Pour Wüster, il est nécessaire d'avoir un centre de coordination internationale sur les travaux en Terminologie et les documents terminologiques. En 1971, Infoterm est créé sur la base d'une collaboration entre l'ISA et l'UNESCO [Campo 2013]. Les activités de Infoterm[13] sont directement liées aux travaux de d'Eugène Wüster d'une part, et au développement du TC37 de l'ISO d'autre part. Ce n'est qu'en 1996 que Infoterm devient une organisation internationale à but non lucratif, ayant pour but la promotion de la Terminologie en tant que discipline[14] [Felber 1980, Infoterm 2004].

Entre 1967 et 1973, six recommandations et une norme sont publiées par l'ISO [Felber, 1980] :

- ISO/R 1087 : contenant le vocabulaire de la théorie de la Terminologie
- ISO/R 704 et ISO/R 860 : qui traitent des principes terminologiques
- ISO/R 919 : les lignes directrices pour le travail terminologique
- ISO/R 1149 : décrivant les méthodes de la lexicographie terminologique
- ISO/R 639 et l'ISO 1951 : qui décrivent les symboles pour les langues, pays et autorités étatiques, tout comme les symboles utilisés dans les lexiques.

Les travaux de recherche d'Eugène Wüster sont repris par son disciple Helmut Felber. Il publie notamment en 1979 une compilation des cours dispensés par Eugène Wüster à l'Université de Vienne dans un livre [La Théorie Générale de la Terminologie] et par la suite, un Manuel de Terminologie [Felber, 1987].

Les principales notions introduites dans la TGT sont les suivantes [Van Campenhoudt, 2006] :

- La mise en avant d'une approche onomasiologique, en étudiant les concepts avant les termes. Le terme est la résultante d'un système de nomination (relevant de la linguistique) fondé sur un système conceptuel relevant de son propre système sémiotique.
- Les concepts se définissent de par leur place dans le système conceptuel.
- La description d'un concept se fait au moyen d'autres concepts connus en fonction de leurs liens, ce qui peut donner lieu à une définition par compréhension ou extension.

---

[12] Ce qui peut poser question dans la mesure où le concept, par définition extralinguistique au sens où il ne relève pas de la langue naturelle, n'est pas le signifié. Si le signifié se construit en discours et peut donc varier, ce n'est pas le cas de ce que désigne le terme dans le cadre du postulat de la biunivocité en Terminologie.

[13] http://www.infoterm.info/standardization/history_standardization_terminological_principles_and_methods.php

[14] La communauté TOTh gérée par l'équipe Condillac est depuis 2018 membre de Infoterm

- Une relation biunivoque entre concept et terme.
- L'étude en synchronie des termes et des concepts.

1.2 CE QU'ON ENTEND PAR « ONTOLOGIE »

1.2.1 INTRODUCTION : L'ONTOLOGIE A DEUX ACCEPTIONS

Le terme « ontologie » est utilisé dans des domaines variés avec un sens différent. La principale différence est sans nul doute dans son utilisation au sens philosophique et au sens de l'ingénierie des connaissances. Ancrée dans la philosophie aristotélicienne, son acception philosophique est ancienne et porte sur la théorie de l'Etre. Tandis que son acception dans le domaine de l'ingénierie des connaissances est récente et représente un objet informatique défini à l'aide d'un formalisme de représentation.

1.2.2 L'ONTOLOGIE AU SENS PREMIER DU TERME

1.2.2.1 ETYMOLOGIE

Le terme *ontologie* est emprunté du latin scientifique *ontologia*, lui-même composé à l'aide de *onto-*, tiré du grec *ôn*, *ontos*, « étant, ce qui est », et *-logia*, tiré du grec *logos*, « discours, traité » (Dictionnaire de l'Académie, 9ème édition). Même si l'ontologie a une très longue histoire en philosophie qui commence principalement avec les travaux d'Aristote, le terme lui-même est relativement récent. Sa création est généralement attribuée à Christian Wolff « Philosophia prima sive ontologia » (1729), mais antérieurement sous sa forme grecque à Goclenius (1613) et sous sa forme latine à Clauberg (1656).

1.2.2.2 DEFINITION

Définie aussi bien de manière succincte « *Théorie de l'être* » [Dictionnaire en ligne Larousse, consultation en date du 6 avril 2018] ou plus détaillée « *Partie de la philosophie qui a pour objet l'élucidation du sens de l'être considéré simultanément en tant qu'être général, abstrait, essentiel et en tant qu'être singulier, concret, existentiel.* » [TLFi consultation en date du 6 avril 2018], elle donne lieu de nombreuses interprétations ouvrant à des approches différentes de l'étude de la « réalité ». Nous pouvons distinguer deux grandes approches selon qu'on s'intéresse à l'objet tel qu'il est perçu à travers ses caractéristiques descriptives (approche que l'on qualifiera ici de phénoménologique) ou conçu à travers ses caractéristiques essentielles (approche dite essentialiste).

1.2.2.3 L'APPROCHE PHENOMENOLOGIQUE

L'approche phénoménologique s'intéresse à construire le monde de manière empirique (et logique) sans chercher nécessairement à en expliciter la nature, son essence, lorsque celle-ci n'est pas

carrément niée. L'objet est décrit selon une démarche « naturelle », tel qu'il nous apparaît, reposant sur la croyance qu'il existe des propriétés concrètes attachées aux objets et directement accessibles.

Les objets sont *décrits* tels qu'ils sont perçus, au travers de leurs qualités (états). L'abstraction des qualités communes permet de former des concepts et de structurer le monde. Le choix des qualités communes est fait en fonction du domaine à décrire. Les principes de l'Ontologie « phénoménologique » sont représentés dans les langages issus de l'intelligence artificielle par la logique, la théorie des relations et des attributs pouvant avoir un ensemble de valeurs pour les qualités comme par exemple la couleur, la forme, le poids[15]. Les ontologies reposant sur les logiques de description, qui portent bien leur nom, relèvent d'une telle approche.

1.2.2.4 L'APPROCHE ESSENTIALISTE

L'approche essentialiste met l'accent non pas sur ce qui qualifie les objets mais sur leur *nature* exprimée en termes de *caractéristiques essentielles*. Une caractéristique est considérée comme essentielle si retirée de la chose, la chose n'est plus ce qu'elle *est*. Par exemple, la caractéristique /pour une personne/ est essentielle au concept désigné par le terme de « chaise »[16].

Cette approche joue un rôle particulièrement important en Terminologie « classique » [Roche, 2011].

1.2.2.5 QUE RETENIR DE CES DEUX APPROCHES POUR NOTRE TRAVAIL ?

Ces deux approches, si elles ne sont pas sans conséquence sur la façon d'aborder la conceptualisation d'un domaine ne s'opposent pas mais se complètent. Elles participent à notre compréhension du monde et à sa représentation. Elles mettent en jeu des connaissances de nature différente, descriptives *versus* essentielles, pour lesquelles nous devrons proposer des modes de représentation adaptés (comment représenter en particulier une caractéristique essentielle).

La Terminologie dite « classique », au sens où nous l'avons définie au chapitre précédent (TGT, ISO), met l'accent sur la caractéristique essentielle (caractère essentiel : « caractère indispensable pour comprendre un concept » [ISO 1087-1]) dans la définition et la construction du système conceptuel (caractère distinctif : « caractère essentiel utilisé pour distinguer un concept d'autres concepts associés » [ISO 1087-1]). La priorité sera donc donnée à l'approche essentialiste, même si celle-ci est généralement absente de l'ontologie au sens de l'ingénierie des connaissances comme nous le

---

[15] Même si parfois la couleur, la forme, … peut être une connaissance essentielle et non pas uniquement descriptive.

[16] Nous suivrons ici la notation introduite par C. Roche pour distinguer ce qui relève de la dimension linguistique de ce qui relève de la dimension conceptuelle. Ainsi, les termes sont notés entre guillemets et commencent par une minuscule, par exemple « chaise », les concepts sont notés entre chevrons en commençant par une majuscule, <Siège pour une personne sans bras avec pieds et dossier> par exemple, et les caractéristiques essentielles entre barres verticales comme /pour une personne/.

verrons au chapitre suivant, ce qui ne veut évidemment pas dire qu'elle ne peut pas être représentée dans un tel système.

Pour conclure ce chapitre, nous retiendrons pour notre travail la définition suivante de l'ontologie au sens philosophique du terme : « *L'Ontologie est la science de l'être en tant qu'être, indépendamment de ses déterminations particulières* ».

1.2.3 L'ONTOLOGIE DE L'INGENIERIE DES CONNAISSANCES

1.2.3.1 DEFINITION

En ce qui nous concerne, nous nous plaçons dans cette 2ème optique, à savoir celle de l'Ingénierie des Connaissances [Staab & Studer, 2010], en prenant néanmoins en compte de l'Ontologie au sens philosophique ce qu'elle peut nous apporter d'un point de vue méthodologique. Tout comme « Terminologie » et « terminologie » on fait la distinction ici entre « Ontologie » qui se réfère à la science qui étudie la nature des choses tandis que « ontologie » désigne dans notre cas une spécification formelle d'un domaine donné : « An ontology is an explicit specification of a conceptualization » [Tom Gruber, 1993].

L'ontologie exprime un point de vue partagé par une communauté sur une conceptualisation d'un domaine. La spécification d'un domaine doit pouvoir être partagée, comprise entre différentes parties et être exprimée, dans le cadre de l'ingénierie des connaissances, dans un langage formel : « An ontology is a formal, explicit specification of a shared conceptualization » [Studer et al, 1998].

Il reste à définir ce qu'on entend par *concept* et *relation*.

Le concept peut être défini comme une unité de connaissance portant sur une pluralité de choses vérifiant une même propriété, que cette propriété porte sur la nature des choses et/ou leurs descriptions. [Roche, 2015]. L'expression de cette propriété dépendra du langage qui sera utilisé pour la définition des concepts. Les relations, qu'elles soient génériques, partitives ou associatives, permettent de structurer les concepts en un système.

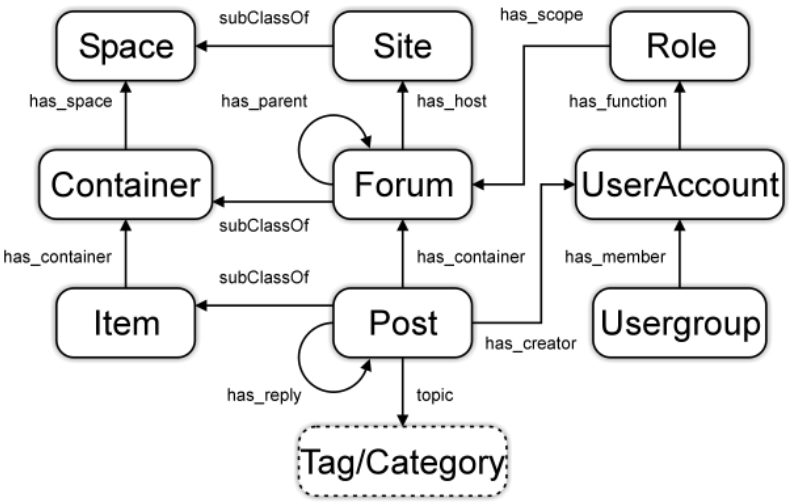
Notons que l'accès aux concepts de l'ontologie passe par l'utilisation de leur dénomination (identifiant). Ces dénominations constituent en soi une terminologie. C'est la raison pour laquelle on trouve parfois comme définition de l'ontologie celle d'un vocabulaire dont les termes sont formellement définis : " […] a vocabulary of terms and some specification of their meaning" [Uschold & Gruniger 1996]. Cependant il demeure important de ne pas confondre cette terminologie constituée des identifiants de concepts avec les termes de la dimension linguistique de la terminologie, ils ne relèvent pas des mêmes systèmes sémiotiques : <Siège pour une personne sans bras avec pieds et dossier> *versus* « chaise ».

1.2.3.2 TYPOLOGIE ET EXEMPLES

Selon le type de connaissance représenté par l'ontologie, on peut établir une typologie des ontologies [Roche, 2003] :

1.2.3.2.1 CORE ONTOLOGIE

Une « core ontologie » est une ontologie définissant les principaux concepts et propriétés nécessaires à la description d'un domaine. La « core ontologie » des communautés numériques (weblogs,wikis,…) [Berrueta et al., 2007] en est un exemple. La spécification UML du langage CLT détaillée dans la partie 3 pourrait correspondre à une « core ontologie » de la Terminologie.



La core ontologie SIOC [http://rdfs.org/sioc/spec/]

1.2.3.2.2 LES ONTOLOGIES UNIVERSELLES

Une ontologie universelle a pour particularité de vouloir représenter l'ensemble de la connaissance en proposant un seul concept comme point de départ. Plusieurs ontologies universelles ont été proposées, elles sont considérées comme une implémentation computationnelle de la philosophie naturelle. Parmi les nombreuses ontologies à visée universelle existantes [Mascardi et al., 2007], nous n'en citerons que trois.

- L'ontologie universelle **« Mikrokosmos »** a été développée dans le cadre d'un système de traduction automatique au laboratoire d'informatique de l'Université du Nouveau-Mexique. Cette ontologie répertorie actuellement 4500 concepts, structurés en trois grandes catégories : les objets, les évènements et les propriétés. L'ontologie définie dans un langage semi-formel est construite manuellement en suivant une méthodologie pour l'acquisition et le positionnement des concepts dans l'ontologie. [http://www.ilc.cnr.it/EAGLES96/rep2/node23.html#SECTION03173000000000000000]

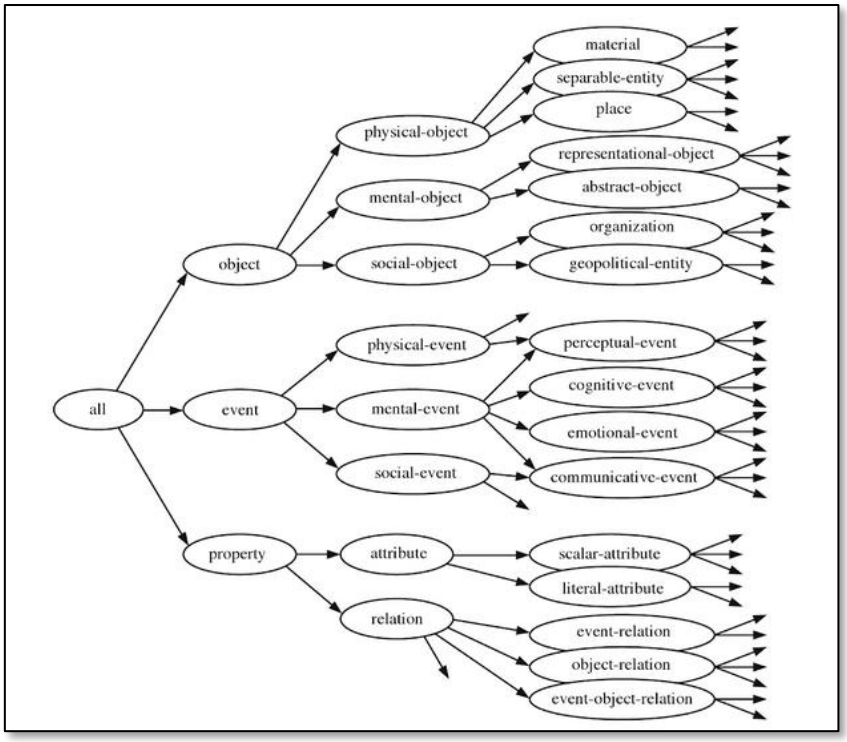


Figure 12 : ontologie universelle Mikrokosmos

Cette ontologie est intéressante car elle illustre les problèmes que pose une approche qui ne permet pas de distinguer dans les formalismes qu'elle propose ce qui relève de l'essentiel et du descriptif. Ainsi il existe, dans cette ontologie, un concept Metal-Liquid, confondant la nature de la chose avec son état : si le mercure est bien *dit* comme étant un *métal liquide*, c'est bien évidemment dû au principe d'économie de la langue, le mercure étant bien un *métal* qui, sous certaines conditions de température et de pression, se trouve dans un *état* liquide.

- **« Upper Cyc[17] »** utilise le langage de représentation cycL (basé sur la logique de prédicats de premier ordre) qui est une extension du langage KIF (Knowledge Interchange Format). Cette ontologie universelle propose des concepts généraux pour représenter la réalité à travers un consensus. Une entrée de l'ontologie commence par le nom du concept suivi d'un commentaire en anglais et les informations concernant la place du concept dans le système conceptuel. Chaque concept est représenté par une constance « Cyc » qui peut être une collection, un objet individuel, un mot dans une langue naturelle, une relation, etc.

[17] https://www.cs.nmsu.edu/~tomohara/cyc/img10.htm

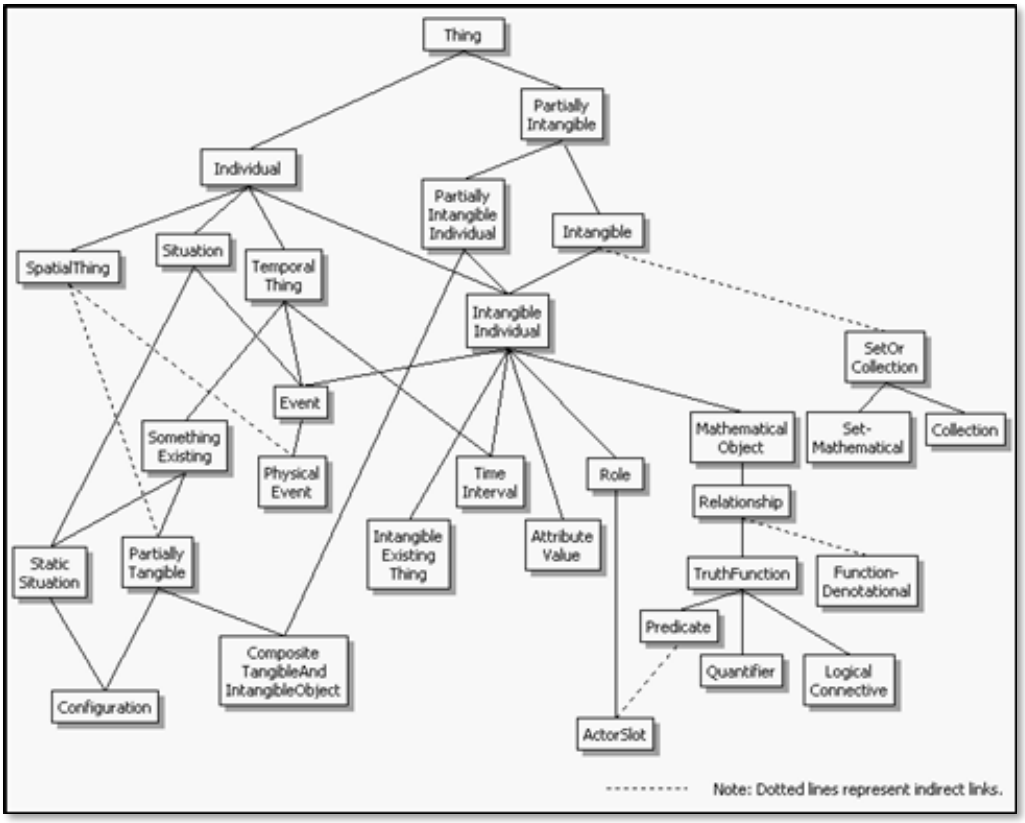


Figure 13: ontologie universelle Upper Cyc

Cyc est un projet qui occupe une place particulière due en particulier à la durée du projet initié en 1984 et à l'utilisation de KIF, avec une syntaxe à la Lisp [Gruber, 1995], possédant un fort pouvoir d'expression, malheureusement trop mal connu.

- L'ontologie universelle **« KR Ontology »** [Sowa 2000, 498] est basée sur la philosophie du processus de Whitehead et sur la sémiotique de Peirce. Sowa établit sept catégories primitives « *Independent, Relative, Mediating, Physical, Abstract, Continuant and Occurrent* ». « *Abstract* » représente une information sans se préoccuper de son support, de son existence dans le temps ou de l'espace, contrairement à «*Physical*». «*Independant*» est une entité qui existe indépendamment de toutes relations avec d'autres entités. «Relative» concerne les entités dont l'existence est en dépendance à d'autres. Une entité de type «*Mediating*» implique qu'elle est la résultante d'une relation. « *Continuant* » représente les entités dont l'existence n'est pas conditionnée par une notion de temps. «*Occurent*» regroupe les entités. L'ensemble des autres types de connaissance sont dérivées de celles-ci. Ces types ou concepts sont définis dans un langage formel, la logique, par conjonction de concepts existants : *Schema (x) ≡ Continuant (x) ∧ Form (x)*

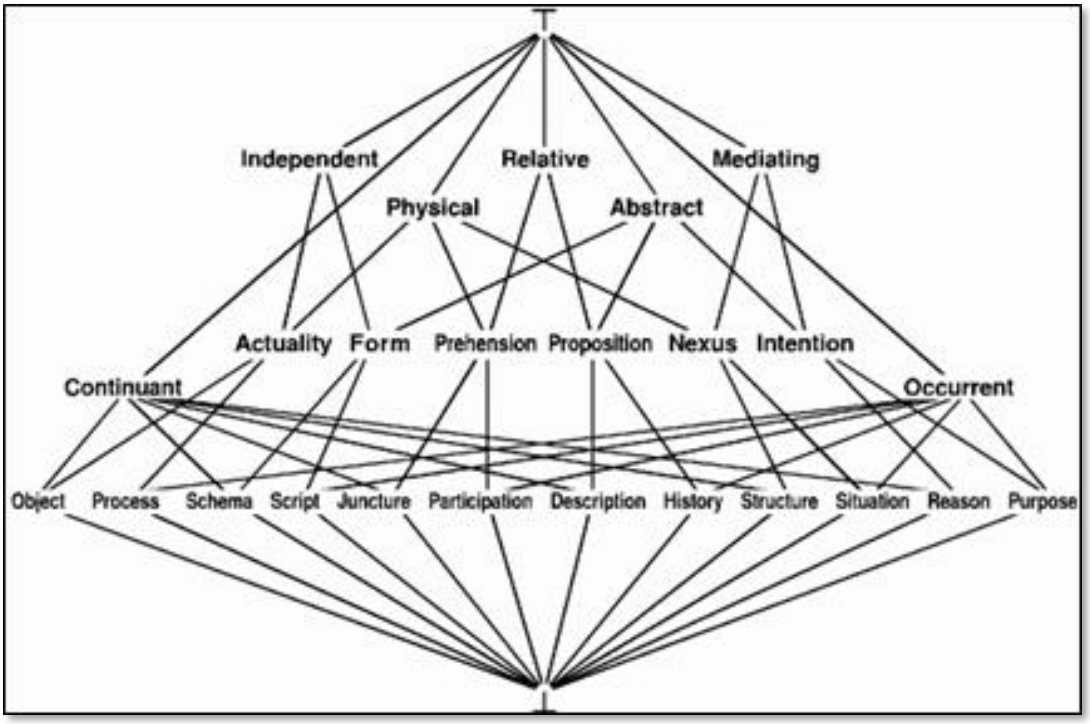


Figure 14: ontologie universelle KR Sowa

Si l'on peut ne pas être d'accord sur la vision du monde que propose Sowa (on peut noter que les trois ontologies universelles que nous venons de présenter ne découpent pas la réalité de la même façon), on peut apprécier la clarté d'une approche reposant sur une logique du premier ordre.

1.2.3.2.3 LES ONTOLOGIES DE DOMAINE

Les ontologies de domaine sont une spécialisation d'une ontologie globale ou universelle à un domaine particulier comme la médecine, la géographie, l'écologie, etc. Ce type d'ontologie permet de formaliser et représenter la connaissance d'un domaine spécifique comme ci-dessous [Benlamri 2009] :

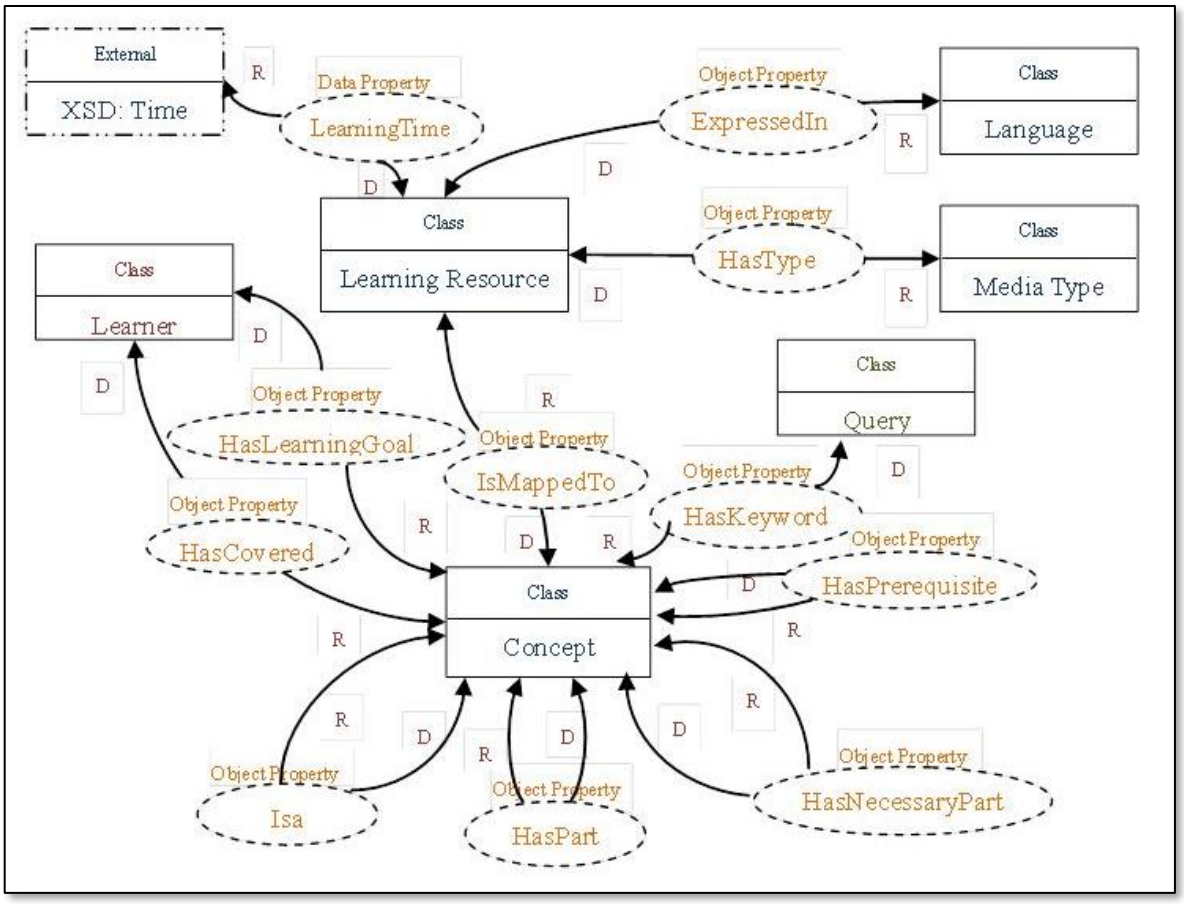


Figure 15: ontologie de domaine "e-learning"

1.2.3.2.4 LES ONTOLOGIES APPLIQUEES, DEDIEES A UNE TACHE PARTICULIERE

Elles portent sur des connaissances spécifiques à un groupe d'experts pour une application donnée. Ce type d'ontologie est en général difficilement réutilisable en dehors de son champ d'application.

- Exemple : Agitateurs (mixers) en chimie (catalogue électronique multilingue)

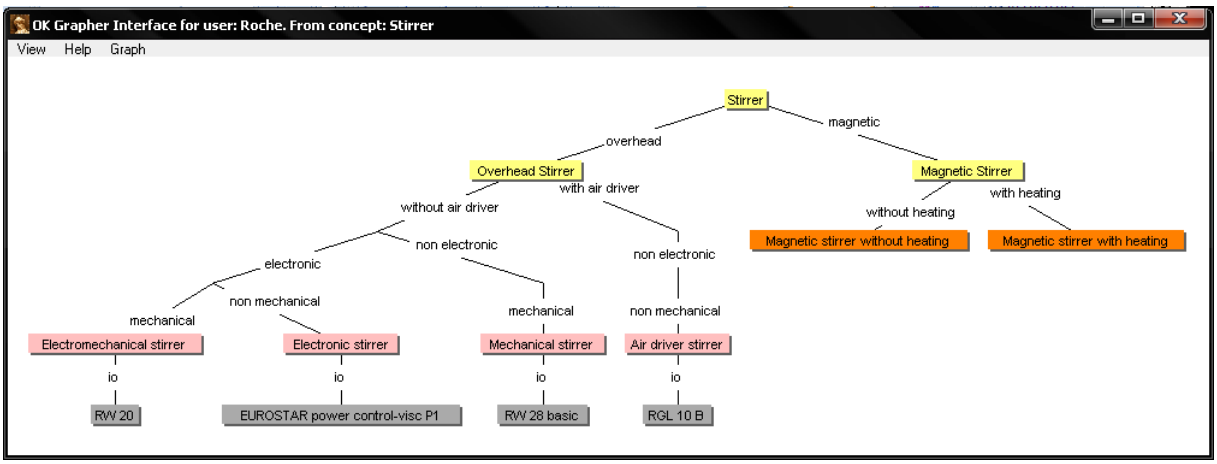


Figure 16 : catalogue électronique, cité dans le cadre de la formation TOTh 2017

1.2.3.2.5 LES ONTOLOGIES DE TYPE RESEAUX DE TERMES

Avec une acception toujours plus large du terme « ontologie », les réseaux de termes sont de plus en plus souvent considérés comme des ontologies. Nous en citerons deux parmi les plus connues.

- **WordNet**[18] est une base de données lexicale initialement créée en 1985 au laboratoire de science cognitive de l'Université de Princeton [Miller, 1998]. Wordnet est distribué sous licence BSD[19], sa base de données est accessible directement en ligne et peut également être interfacée dans les principaux langages de programmation. La version de 2012 contient 155 327 mots (noms, verbes, adjectifs et adverbes) organisés dans 175 979 « synsets ». Un « synset » accronyme de « synonym set » est un groupe de mots dénotant un sens particulier (regroupés en fonction de leur synonymie). Un « synset » contient une définition brève et quelques phrases courtes illustrant l'utilisation des mots du « synset ». Les « synset » peuvent être reliées par des relations d'hyperonymie, hyponymie ou de méronymie. C'est pourquoi WordNet est souvent assimilé à une ontologie, si l'on considère qu'un concept peut se réduire à un ensemble de termes qui en parlent (et qui sont donc synonymes puisque désignant le même concept), et les relations conceptuelles à des relations linguistiques (hyperonymie, hyponymie, méronymie)

[18] https://wordnet.princeton.edu/

[19] (Berkeley Software Distribution License) est une licence libre utilisée pour la distribution de logiciels.

Noun

- S: (n) **chair** (a seat for one person, with a support for the back) *"he put his coat over the back of the chair and sat down"*
  - direct hyponym / full hyponym
  - ***part meronym***
    - S: (n) back, backrest (a support that you can lean against while sitting) *"the back of the dental chair was adjustable"*
    - S: (n) leg (one of the supports for a piece of furniture)
  - direct hypernym / inherited hypernym / sister term
- S: (n) professorship, **chair** (the position of professor) *"he was awarded an endowed chair in economics"*
- S: (n) president, chairman, chairwoman, **chair**, chairperson (the officer who presides at the meetings of an organization) *"address your remarks to the chairperson"*
- S: (n) electric chair, **chair**, death chair, hot seat (an instrument of execution by electrocution; resembles an ordinary seat for one person) *"the murderer was sentenced to die in the chair"*
- S: (n) **chair** (a particular seat in an orchestra) *"he is second chair violin"*

Verb

- S: (v) **chair**, chairman (act or preside as chair, as of an academic department in a university) *"She chaired the department for many years"*
- S: (v) moderate, **chair**, lead (preside over) *"John moderated the discussion"*

Figure 17: Une recherche avec le mot « chair » dans Wordnet [https://wordnet.princeton.edu/]

- **BabelNet** développé par le département d'informatique et de linguistique computationnelle de l'Université de Rome, se définit comme un réseau sémantique multilingue et une ontologie lexicalisée [référence web : www.babel.org]. BabelNet utilise WordNet pour la représentation du système conceptuel, basé sur les relations entre « synset » en anglais. Les mots des « synset » de WordNet sont en partie traduits automatiquement dans diverses langues (BabelNet intègre actuellement 284 langues). Pour chaque « Babel synset », BabelNet fournit des définitions textuelles issues notamment de Wikipédia.

Figure 18: Recherche du mot "椅子" sous BabelNet [http://live.babelnet.org/ consulté le 07.10.2018]

On peut remarquer que le mot chinois « 椅子 » sous BabelNet est simplement traduit par « chaise » en français alors qu'il permet également désigner un fauteuil.

BabelNet est néanmoins une illustration intéressante de la démarche actuelle consistant à intégrer différentes ressources : Wordnet, Wikipedia, OmegaWiki, Wikidata, Wikiquote, etc. (Figure 19)

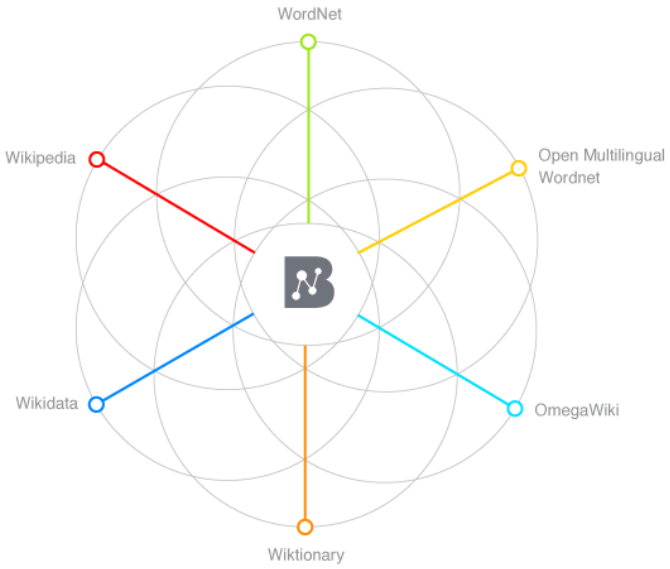


Figure 19 : https://babelnet.org/about

1.2.3.3 LANGAGES DE REPRESENTATION DE LA CONNAISSANCE

La définition d'une ontologie nécessite un langage de représentation. Le problème est alors celui du choix du langage en fonction de ce qui est visé et recherché. En effet tous les langages ne sont pas équivalents tant dans leur pouvoir d'expression que leurs propriétés. Construire une terminologie dans le domaine du tourisme par exemple, ne nécessite pas, *a priori*, de recourir à un langage formel, contrairement à une terminologie dans la sécurité nucléaire.

Nous proposons ci-dessous une typologie des différents langages selon leur degré de formalisation et de pouvoir d'expression.

1.2.3.3.1 LANGUE NATURELLE

La langue naturelle a une place privilégiée car *in fine* on produira bien des définitions en langue naturelle et on ne peut s'empêcher de « dire », de « lire » des définitions même formelles[20]. Cependant elle n'est pas le langage le plus adapté pour la définition de concept *a fortiori* si l'on

[20] On distingue ici « langue » de « langage, le premier se référant à la langue naturelle, le second à un langage artificiel.

souhaite garantir un certain nombre de propriétés. Les interprétations multiples et subjectives (au sens où ces interprétations dépendent du sujet) auxquelles elle peut donner lieu, et donc aux incompréhensions éventuelles, font qu'on ne peut retenir la langue naturelle comme langage de représentation (définition) du système conceptuel.

Citons pour exemple la définition de « chaise » tirée du TLFi :

CHAISE, subst. fém.

A.– Siège à dossier et généralement sans bras. *Chaise de salle à manger; chaise pliante; chaise cannée :*

Figure 20: Exemple de « Chaise » défini dans le TLFi (consultation en date du 13 Avril 2018)

Cette définition ne permet pas, par exemple, de savoir s'il s'agit d'un siège pour une personne ou plusieurs personnes, s'il a des pieds ou non. L'écriture d'une connaissance en langue naturelle ne permet pas de vérifier la « consistance »et la « cohérence »[21] de la définition.

1.2.3.3.2 LANGAGES SEMI-FORMELS

Les langages de représentation de la connaissance n'ont pas tous le même degré de formalisation. Parmi les langages semi-formels, on trouve les langues contrôlées (LC). Une langue contrôlée est un sous-ensemble d'une langue naturelle mais dont le vocabulaire et la grammaire ont été restreints. L'objectif est de réduire l'ambiguïté des textes. La langue contrôlée « français rationalisé de Dassault Aerospace » en est un exemple. Ces langages, bien que contrôlés, restent, c'est leur objectif, proche de la langue naturelle et constituent toujours un discours sur la connaissance.

Les langages de représentation graphiques sont également des langages semi-formels mais dont l'objectif est de représenter directement les connaissances du domaine, ici des conceptualisations, et non plus un discours sur ces connaissances. Pour cela, ces langages s'appuient sur des formalismes qui se veulent adaptés à la modélisation des connaissances, c'est-à-dire à la représentation des notions mises en jeu : concept, individu, relation. On trouve de tels langages mis en œuvre dans des éditeurs de cartes conceptuelles dont CmapTools[22] est un exemple bien connu [Cañas et al., 2004]. CmapTools est un éditeur gratuit de cartes conceptuelles représentées sous la forme de graphes orientés où les nœuds peuvent modéliser toutes connaissances (concept, individu, situation, action, etc.) et les arcs étiquetés représentent toutes relations entre ces nœuds (relations génériques, d'instanciation, partitives, associatives, etc.). Si ces environnements offrent une grande liberté dans la modélisation des connaissances, c'est souvent au prix d'un relâchement de contraintes pourtant nécessaires. Ainsi, dans CmapTools, aucune vérification n'est faite quant à d'éventuelles circularités ou doublons dans le graphe. Si de tels outils sont très utiles au démarrage d'un projet, ils ne demeurent pas toujours adaptés pour des projets un tant soit peu importants.

---

[21] Au regard de la théorie du concept qui sera choisi. Par exemple, pour la consistance, l'existence d'au moins un objet pour le concept en question, et pour la cohérence, ne pas avoir de caractéristiques exclusives entre elles dans la définition du concept.

[22] https://cmap.ihmc.us/

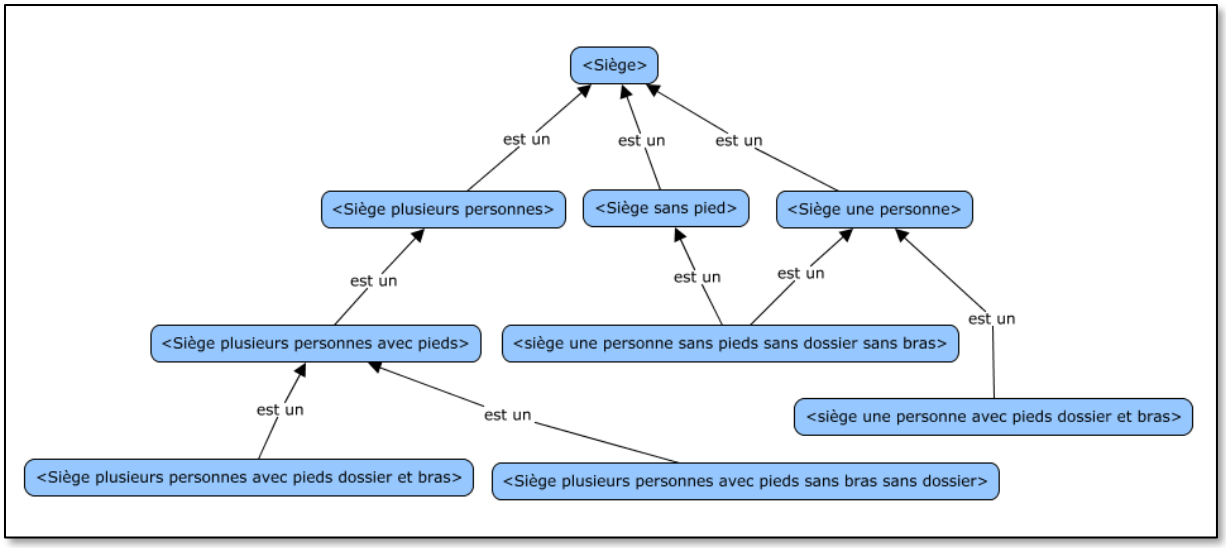


Figure 21: Exemple de la modélisation d'une conceptualisation avec CmapTools

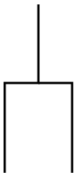
1.2.3.3.2.1 LA REPRESENTATION GRAPHIQUE DE LA NORME ISO 1087-1

La Terminologie dite « classique » (TGT, ISO) donne la préférence au concept et repose, comme nous le verrons ultérieurement, sur des principes épistémologiques clairement énoncés, même s'ils peuvent donner lieu à critiques. Elle s'est également toujours attachée à proposer une représentation « extra linguistique » du système conceptuel reposant sur des représentations graphiques qui lui sont propres.

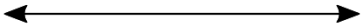
La norme pour la Terminologie ISO 1087-1 propose un langage de représentation graphique utilisé dans les schémas conceptuels. La notation utilisée est la suivante :

- Les relations génériques sont représentées sous la forme d'une arborescence :

- Les relations partitives utilisent une représentation en râteaux :

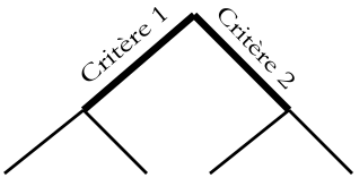


- Les relations associatives sont faites par des flèches à double sens :



- Un trait interrompu lorsqu'il s'agit d'une relation qui n'apparaît pas explicitement dans la définition d'un concept :

- Un critère de subdivision est représenté par un trait plus épais portant une indication sur le caractère distinctif :



A noter, qu'il n'existe actuellement aucun logiciel spécifiquement dédié à la notation graphique de la norme ISO 1087-1. Il serait alors utile de disposer d'un environnement logiciel à cet effet, qui permettrait au-delà d'une représentation purement graphique, de vérifier certains principes logiques impliqués dans la conception de schémas conceptuels selon les normes ISO.

1.2.3.3.2.2 UML POUR LA REPRESENTATION GRAPHIQUE DE SCHEMAS CONCEPTUELS EN TERMINOLOGIE

Avec l'essor des langages de programmation objet est né un langage de représentation graphique fréquemment utilisé par les concepteurs d'applications informatiques : UML[23] qui permet de représenter statiquement et dynamiquement un système informatique. Il peut être utilisé dans sa forme statique (diagramme de classes) pour la représentation des connaissances, mais ce n'est pas sa vocation primaire. La norme ISO 24156-1 « Graphic notations for concept modelling in terminology work and its relationship with UML » introduit l'utilisation de la notation UML pour la modélisation de schémas conceptuels en Terminologie. Elle n'a pas vocation à remplacer la notation existante de la norme ISO 1087-1 mais d'en proposer une alternative en la complétant [Kockaert H.,Antia B., 2008], [Kockaert H. et al.,2010].

L'élément de base du diagramme UML est la classe UML, représentée graphiquement par un rectangle contenant trois compartiments : le nom de la classe commençant par une majuscule, les attributs et les signatures des méthodes. La norme ISO 24156-1 reprend ce modèle à l'exception du compartiment réservé aux méthodes. Dans la première partie figure le nom du concept à la place du nom de la classe et la deuxième partie contient les caractéristiques. Nous ne détaillerons pas ici toutes les correspondances entre la norme ISO 1087-1 et les notations adoptées de la norme ISO 24156-1 (se référer au tableau des correspondances en annexe). La notation UML est mise à profit notamment pour représenter les cardinalités entre les relations associatives et partitives, une notion qui n'existe pas dans la norme ISO 1087-1. La relation partitive est définie dans la norme ISO 1087-1 comme une relation hiérarchique : « relation entre deux concepts qui est soit une relation générique,

[23] Acronyme de « Unified Modeling Language » est un langage graphique normalisé pour représenter la conception d'un système orienté objet.

soit une relation partitive » [ISO 1087-1, 3.2.20], contrairement à UML qui considère la relation partitive comme une forme plus spécifique de la relation d'association [ISO 24156-1, 5.6.3]. UML permet aussi de distinguer deux types de relation partitive : la relation d'agrégation et la relation de composition qui est une forme plus forte de l'agrégation. La relation de composition implique une dépendance d'existence entre la partie et le contenant, si le contenant est supprimé, la partie l'est aussi. L'utilisation de la notation UML rend possible une conception plus fine des schémas conceptuels de la Terminologie. Par ailleurs, il existe de nombreux outils informatiques comme ArgoUML[24] dédiés à la conception de diagrammes UML donnant la possibilité d'exporter un diagramme de classes dans un langage de description tel que XML[25].

1.2.3.3.3 LANGAGES FORMELS (LANGAGES DE L'INTELLIGENCE ARTIFICIELLE)

Les langages semi-formels, et plus précisément les langages de représentation graphiques, constituent le premier pas vers un formalisme dédié à la représentation d'une conceptualisation. Il reste à franchir le dernier pas vers un formalisme compréhensible par un ordinateur qui permettra de construire un système conceptuel « calculable » au sens où on pourra lui appliquer différentes opérations : création de concepts par spécialisation, mise en relation, inférences, etc. Dans la catégorie des langages formels pour la représentation des connaissances compréhensibles par un ordinateur, nous avons les langages issus de l'intelligence artificielle comme FRL, SRL, KIF.

- FRL (Frame Representation Language) [Minsky, 1974] est un langage informatique élaboré par Marvin Minsky et repose sur une structure appelée « Cadre » ou « Frame » (Frame de Minsky) pour représenter la connaissance, un « Frame » défini comme une « structure formelle d'informations composée d'attributs et de variables caractérisant un objet, une situation ou un concept particulier » [Otman 91 : 37]

- SRL (Schema Representation Language) [Wright & Fox, 1983], est une famille de langages de représentation de la connaissance en intelligence artificielle développée à partir de 1979 à l'institut robotique de l'Université Carnegie Mellon [Fox & Adam, 1985]. SRL écrit en LISP, utilise une unité de représentation par schémas plus connue sous le nom de « Frame » [Minsky, 1974], le terme « schéma » fait référence à la théorie du schéma introduite par Barlett en 1932 [Arbib 1992] sur laquelle SRL se base. Chaque schéma dans SRL est nommé et comporte un ensemble de « slots » (portant un nom et potentiellement une valeur) (Figure 22). Il est possible de définir un « méta-schéma » qui permettra d'avoir une information sur le « slot » en question (Figure 23). La représentation des relations se fait aussi par des « slots » (Figure 24).

[24] Logiciel libre distribué sous licence EPL1.0 et permet la création entre autre de diagramme de classes et de cas d'utilisation. http://argouml.tigris.org/ (consulté le 30/08/2018)

[25] Extended Markup Language

```
{{ Mammal
    NURSING-METHOD:
    BIRTH-PROCESS:        }}
```

The Mammal schema contains two slots, NURSING-METHOD and BIRTH-PROCESS. Slots can have values. A value can be any lisp expression. (SRL.V. écrit en Lisp, 1 expr n'est pas 1 mot)

```
{{ Mammal
    NURSING-METHOD:   breast
    BIRTH-PROCESS:        }}
```

Figure 22 : Définition d'un schéma [documentation SRL 1.5, 1983]

```
{{ Mammal:
    NURSING-METHOD:   breast
                      creator:  M.Fox }}
```

Figure 23 : définition d'un méta-schéma [documentation SRL 1.5, 1983]

```
{{ dog
    IS-A: Mammal }}
```

Figure 24 : définition d'une relation générique [documentation SRL 1.5, 1983]

- KIF (Knowledge Interchange Format) est un langage informatique avec une syntaxe à la Lisp dont le but est de réutiliser et d'échanger des systèmes conceptuels. KIF a une sémantique déclarative pour la description du monde. La connaissance est décrite à l'aide d'objets, de relations et de règles. KIF utilise la logique de premier ordre et peut intégrer un raisonneur pour vérifier la consistance des déclarations KIF. Dans l'exemple ci-dessous, « PHYSICAL-QUANTITY » est une classe et « ?q » une instance de cette classe. [Gruber, 1995]

```
(defrelation PHYSICAL-QUANTITY
  (<=> (PHYSICAL-QUANTITY ?q)
       (and (defined (quantity.magnitude ?q))
            (double-float (quantity.magnitude ?q))
            (defined (quantity.unit ?q))
            (member (quantity.unit ?q)
                    (setof meter second kilogram
                           ampere kelvin mole candela)))
```

A noter que les langages de programmation orientés objets permettent aussi de représenter des connaissances mais n'auront pas la même finalité, c'est-à-dire la conception de programmes informatiques, la résolution d'un problème donné.

1.2.3.3.4 LANGAGES STRICTEMENT-FORMELS (LES LOGIQUES DE DESCRIPTION)

Le passage à des langages de type logique correspond à un niveau de formalisation le plus élevé. On profite ainsi d'un langage à la syntaxe et à la sémantique clairement définies pour lequel il existe des raisonneurs, comme nous le verrons dans le cas de l'environnement Protégé basé sur les logiques de description, permettant par exemple de déterminer si des concepts (généralement appelés *classes* dans l'approche extensionnelle qu'est la logique) ont une extension nécessairement vide (inconsistance). De tels langages proposent une définition claire du concept.

Les logiques de description [Baader et al., 2003] font parties des langages de représentation de la connaissance largement utilisés dans la modélisation d'ontologies en particulier dans le contexte du web sémantique[26]. Ces langages basés sur une logique du premier ordre avec une sémantique formelle permettent une communication sans ambiguïtés entre humains et systèmes informatique. Ils rendent aussi possible l'utilisation de déductions logiques pour inférer des informations supplémentaires sur des faits. C'est cette particularité qui les différencie des langages de modélisation tel que UML[27]. Cette capacité d'inférer des informations supplémentaires à partir de déductions logiques (appelé raisonneur) constitue le principal atout de ces langages [Krötzsch et al., 2012]. Comme leur nom l'indique, ces logiques mettent l'accent sur la description des objets, non sur leur nature, elles correspondent à l'approche que nous avons qualifiée de phénoménologique, ainsi un individu se « définit » non pas parce qu'il est, mais par les relations qu'il entretient avec les autres. L'instanciation est ici, non pas une exemplification d'un concept, mais une opération de classification d'un individu dans la ou les classes auxquelles il peut appartenir. « The key idea that a class of individuals is described or defined by the relationships that these individuals participate in. » [Horridge 2011].

Dans les logiques de description, on distingue trois types d'entités utilisés dans la modélisation d'un domaine :

- Les concepts ou classes, regroupant un ensemble d'individus ;
- Les rôles ou propriétés, relations binaires entre individus
- Les individus

1.2.3.4 PROTEGE

Protégé[28] est un environnement libre pour la création, l'édition et l'alignement d'ontologies distribué par l'Université de Stanford dont sa première version date de 1987 [Gennari et al., 2003]. Cet environnement repose sur les logiques de description et met à disposition des opérateurs logiques

---

[26] « Le Web sémantique, concrètement, est d'abord une infrastructure pour permettre l'utilisation de connaissances formalisées en plus du contenu informel actuel du Web, même si aucun consensus n'existe sur jusqu'où cette formalisation doit aller. » [Charlet et al., 2003]

[27] Acronyme de « Unified Modeling Language » est un langage graphique normalisé pour représenter la conception d'un système orienté objet.

[28] http://protege.standford.edu/

tels que l'intersection, l'union et la négation. Son modèle permet l'utilisation d'un raisonneur ou moteur d'inférences afin de vérifier la consistance d'une ontologie. « Une ontologie peut être détectée comme inconsistante : c'est-à-dire qu'il n'existe aucune interprétation possible de l'ontologie qui soit un modèle pour cette ontologie. » [Roussey et al., 2010]

L'environnement dont l'architecture est détaillée dans la figure 28 est caractérisé par la mise à disposition d'un système de gestion de plugins à des fins d'extension. L'éditeur dispose d'une interface graphique (voir figure 29) pour la création de concepts (classes), propriétés (slots) et individus (instances). [Horridge, 2011]

- Individus (instances)

Les individus représentent les objets d'un domaine et peuvent être considérés comme des instances d'une classe. « Individuals are also known as instances. Individuals can be refered to as being 'instances of classes'. » [Horridge, 2011]. Les individus sont représentés par des losanges (Figure 25).

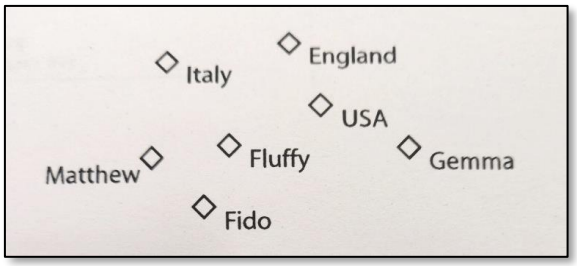


Figure 25 : représentation des individus [Horridge, 2011]

- Les propriétés ou relations (slots)

Les propriétés sont des relations binaires portant sur des individus (Figure 26). Elles peuvent être de différents types : symétriques ou asymétriques, transitives, etc. [documentation[29] de protégé 5]. Par exemple, la relation « livesIn » relie l'instance « Matthew » à l'instance « England » (relation asymétrique, irréflexive).

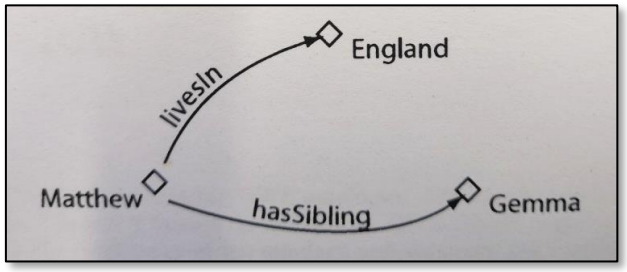


Figure 26 : Représentation des propriétés [Horridge, 2011]

[29] http://protegeproject.github.io/protege/views/object-property-characteristics/

- Les concepts (classes)

Dans Protégé, les classes ou concepts sont interprétées comme des ensembles d'individus (Figure 27). « The word *concept* is sometimes used in place of class. Classes are a concrete representation of concepts. » [Horridge, 2011]. Elles peuvent être organisées hiérarchiquement par des relations de superordination ou de subordination.

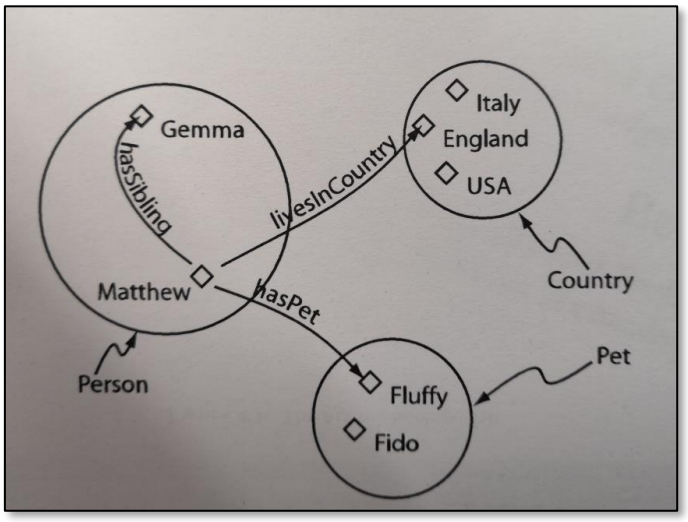


Figure 27 : Représentation des classes contenant des individus [Horridge, 2011]

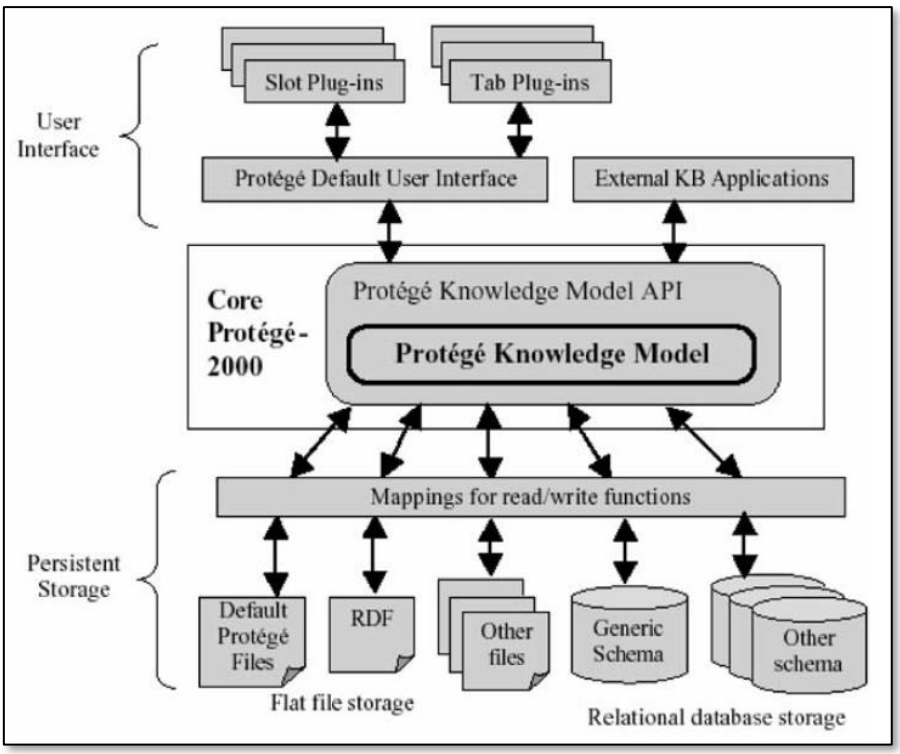


Figure 28 : Architecture de Protégé [Staab & Studer, 2010]

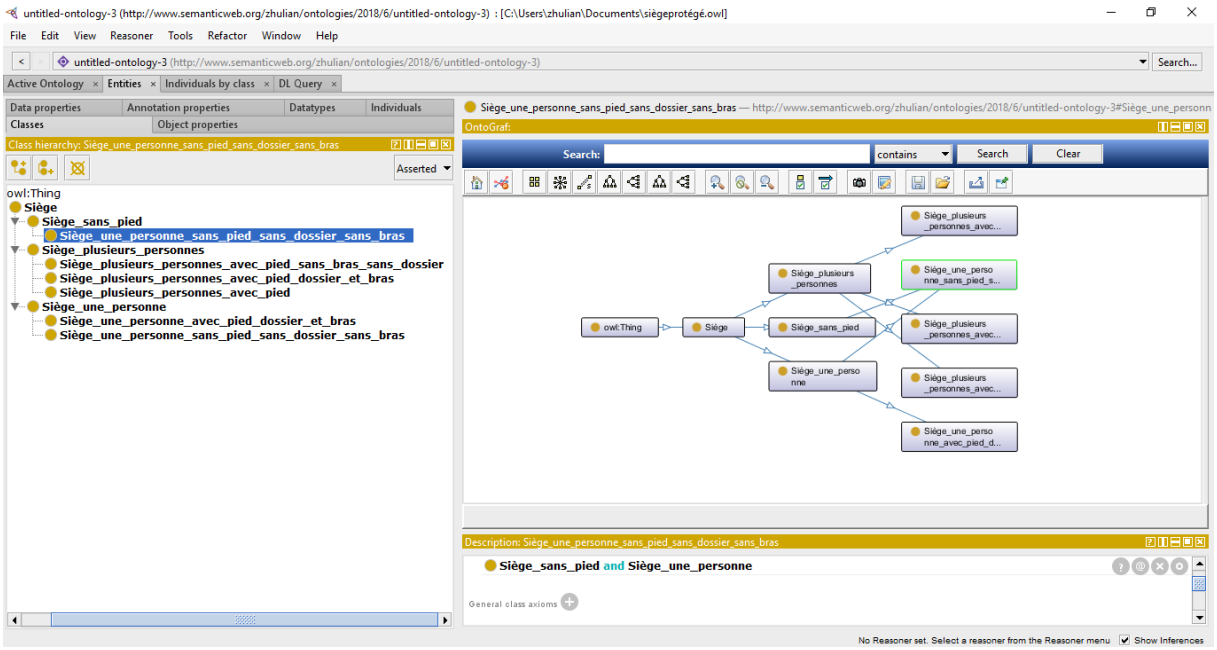
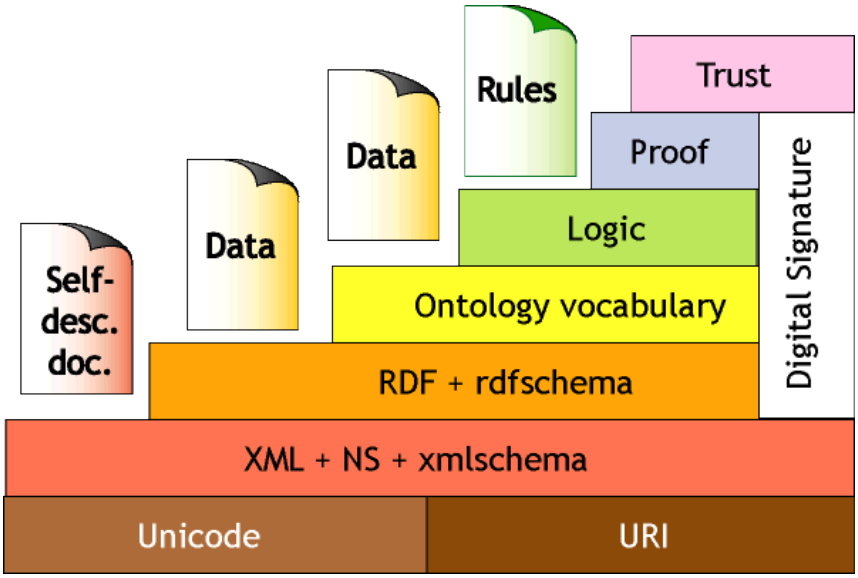


Figure 29 : Protégé - ontologie des Sièges

1.2.3.6 LES FORMATS D'ECHANGE INFORMATIQUE DU W3C POUR LA REPRESENTATION DES CONNAISSANCES

Les formats d'échange introduits ici sont définis par le W3C (The World Wide Web Consortium), un organisme qui œuvre pour la normalisation du web. Tim Berners-Lee propose une hiérarchisation de ces langages à travers le « Semantic Web Stack » [Berners-Lee, 2000].



Semantic Web Stack [Tim Berners-Lee, 2000]

1.2.3.6.1 RDF (RESSOURCE DESCRIPTION FRAMEWORK)

Basé sur le langage description XML (Extended Markup Language), RDF[30] (Ressource Description Framework) adopté comme recommandation W3C en 1999 [Lassila & Swick, 1998] est un langage permettant de représenter des ressources sous la forme de graphes orientés. Chaque ressource est identifiée par une URI (Uniform Ressource Identifier) dans une expression RDF. Les données sont décrites sous la forme de triplets : sujet, prédicat, objet. Ce format est surtout utilisé pour le web sémantique. « La manipulation des ressources du Web par des machines requiert l'expression ou la description de ces ressources. Plusieurs langages sont donc définis à cet effet, ils doivent permettre d'exprimer données et métadonnées (RDF, Cartes Topiques)… » [Charlet et al., 2003].

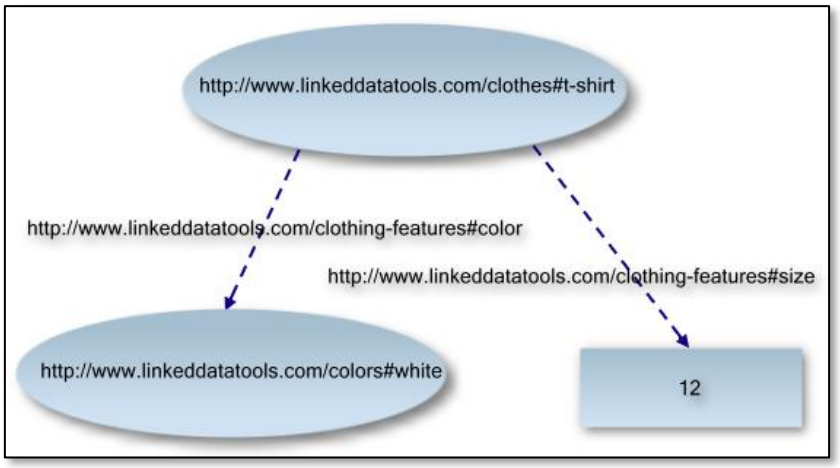


Figure 30 : exemple de représentation des données avec le format RDF (http://www.linkeddatatools.com/introducing-rdf-part-2)

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:feature="http://www.linkeddatatools.com/clothing-features#">

    <rdf:Description rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">

        <feature:size>12</feature:size>
        <feature:color rdf:resource="http://www.linkeddatatools.com/colors#white"/>

    </rdf:Description>

</rdf:RDF>
```

Figure 31 : exemple de représentation des données avec le vocabulaire RDF (http://www.linkeddatatools.com/introducing-rdf-part-2)

1.2.3.6.2 RDFS (RDF SCHEMA)

Le format RDF [McBride, 2004] peut être enrichi avec des informations sémantiques. Tout en gardant la structure des données sous forme de triplets, RDFS permet de décrire des ontologies. Ce format permet de définir entre autres des classes, des liens de super ordination entre classes, des propriétés avec la possibilité de spécifier un domaine de départ (Domain) et d'arrivée (Range).

[30] https://www.w3.org/RDF/

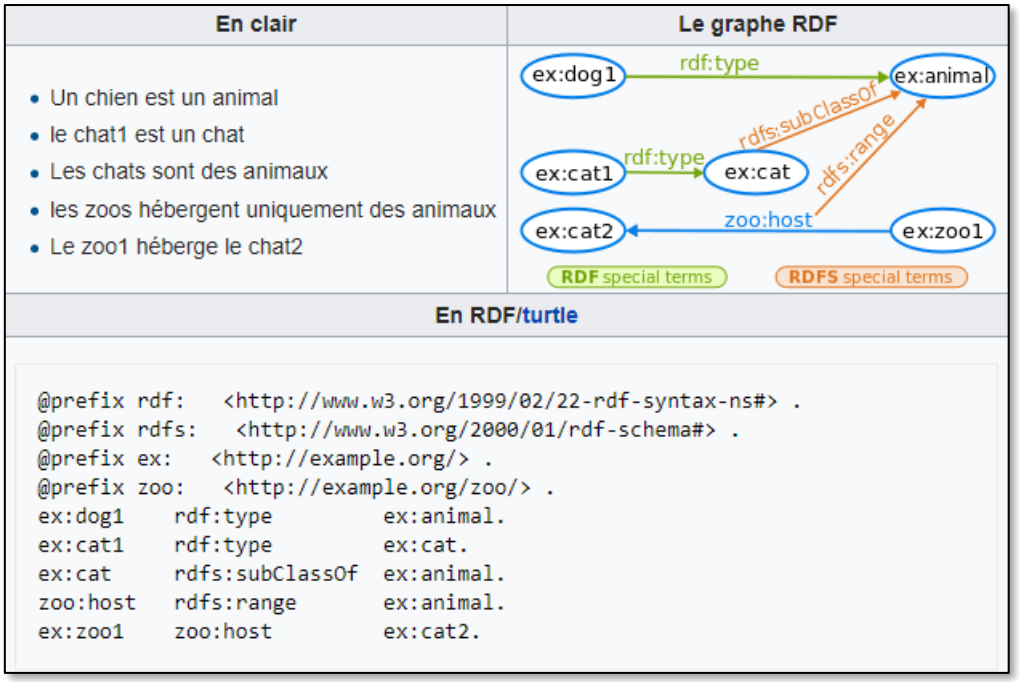


Figure 32 : Format RDFS (https://fr.wikipedia.org/wiki/RDF_Schema)

1.2.3.6.3 OWL (ONTOLOGY WEB LANGUAGE)

Le format OWL (Ontolgy Web Language) [Dean and Schreiber 2004] est le résultat d'un groupe de travail du W3C formé en 2001 et une recommandation du W3C en 2004. OWL repose sur le format RDFS (Figure 31) et le complète avec un vocabulaire plus riche. Il donne par exemple la possibilité d'utiliser des cardinalités sur les propriétés et d'utiliser des opérateurs logiques (par exemple : l'union de classes). Ce langage est fortement inspiré des logiques de description et de nombreux raisonneurs ont été développés pour ce langage. Une ontologie OWL est constitué d'individus, de propriétés et de classes. A noter que OWL est décliné en trois versions pour des degrés d'expressivité de différents niveaux (OWL Lite, OWL DL, OWL Full). [McGuinness & Van Harmelen 2004, Horridge 2011].

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <owl:Ontology rdf:about="http://www.julien-roche.fr/sieges">
        <dc:title>Example of Siege ontology</dc:title>
        <dc:description>An example ontology using OWL</dc:description>
    </owl:Ontology>
    <owl:Class rdf:about="http://www.julien-roche.fr/sieges#siegetype">
        <rdfs:label>Siege Type</rdfs:label>
    </owl:Class>
</rdf:RDF>
```

Figure 33 : Exemple utilisant le format OWL

1.2.3.6.4 SKOS (SIMPLE KNOWLEDGE ORGANIZATION SYSTEM)

SKOS[31] [Miles et al., 2005] qui signifie « système simple d'organisation des connaissances » est un format d'échange de données standardisé par le W3C[32] pour le support de KOS (knowledge organization systems) comme les thésaurus, les taxonomies ou les classifications. Le modèle de données SKOS permet le portage des systèmes d'organisation de connaissances existants vers le web sémantique. Il peut être utilisé seul ou combiné avec OWL. Les concepts peuvent être identifiés par des URIs (Uniform Resource Identifier), libellés avec des chaînes de caractères. Des notations peuvent y être associées et documentées à l'aide de plusieurs types de notes, reliés à d'autres concepts et organisés dans des hiérarchies informelles et des réseaux associatifs, agrégés dans des schémas de concepts ou regroupés dans des collections. Ce modèle de données a pour objectif d'établir un rapprochement entre plusieurs communautés de pratique impliquées dans la conception des systèmes d'organisation de connaissance en proposant un modèle commun afin d'exprimer des données compréhensibles par une machine. Les données SKOS sont exprimées sous forme de triplets RDF. A la différence de OWL, SKOS n'est pas un langage formel de représentation de connaissances, pour la simple raison qu'un thésaurus ou une classification n'exprime pas d'axiomes ou de faits mais s'attache plutôt à identifier et à décrire des concepts.

```
<A> rdf:type skos:Concept ;
  skos:prefLabel "amour"@fr ;
  skos:altLabel "adoration"@fr ;
  skos:broader <B> ;
  skos:inScheme <S> .

<B> rdf:type skos:Concept ;
  skos:prefLabel "émotion"@fr ;
  skos:altLabel "sensation"@fr ;
  skos:topConceptOf <S> .

<S> rdf:type skos:ConceptScheme ;
  dct:title "Mon Premier Thesaurus" ;
  skos:hasTopConcept <B> .
```

Figure 34 : format SKOS

1.2.3.7 CONCLUSION : QUEL LANGAGE CHOISIR ?

Les langages de représentation que nous venons de décrire ne sont pas équivalents, tant au niveau de leur formalisme que de leurs propriétés. Le choix du langage pour la modélisation du système conceptuel n'est pas anodin. Il conditionne non seulement le résultat, c'est-à-dire le système conceptuel, et les opérations qui lui seront applicables, mais surtout il conditionne la vision du

[31] https://www.w3.org/TR/2009/REC-skos-reference-20090818/

[32] W3C : Le World Wide Web Consortium (W3C) est une communauté internationale où les membres, une équipe à plein temps, et le public travaillent ensemble pour développer les standards du web.

monde à travers ce que le langage permet de décrire. En effet, un même domaine peut avoir des représentations différentes selon le langage qui sera utilisé.

Le problème posé est celui du pouvoir d'expression du langage de représentation au regard des principes épistémologiques mobilisés par la construction du système notionnel d'une Terminologie dite « classique ». On est ramené à un problème de traduction de la conceptualisation de l'expert dans un langage qui ne permet pas toujours d'en exprimer toutes les connaissances : « traduire, c'est trahir ».

Ainsi, en reprenant nos exemples de langages compréhensibles par un ordinateur, qu'ils soient formels ou strictement formels :

Les systèmes à base de schémas, en ne distinguant pas ce qui est essentiel de ce qui est descriptif (l'essentiel pourra être traduit par un attribut (slot) de valeur booléenne), peut conduire à des modélisations erronées comme dans le cas de Mikrokosmos avec le concept de 'Meta-Liquid' que nous avons déjà évoqué.

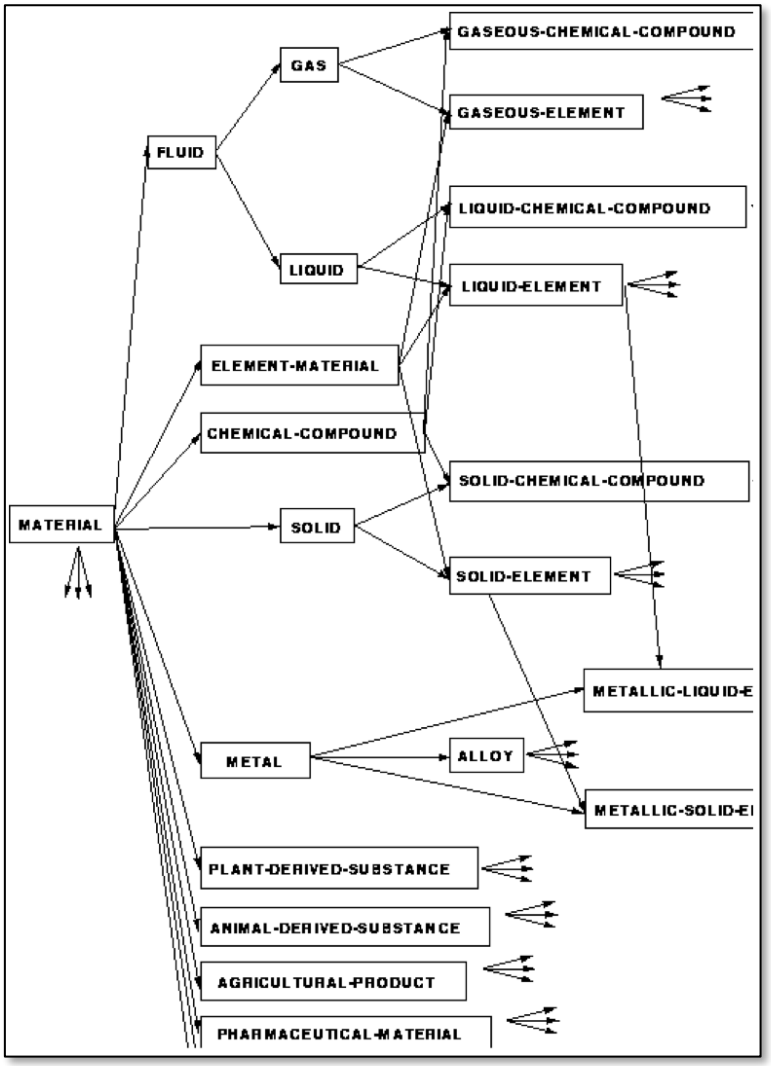


Figure 35 : l'ontologie Mikrokosmos

La logique n'est pas exempte de problème de modélisation. Ainsi, on n'offrant que le paradigme de prédicat, les différences, pourtant fondamentales, entre les notions de mortel (concept), raisonnable (caractéristique essentielle) et malade (caractéristique contingente) sont effacées pour ne faire place qu'aux seuls prédicats Humain(x), Mortel(x), Malade(x)[33].

Ainsi, faut-il choisir un langage de représentation existant et l'adapter à nos besoins ou faut-il définir un nouveau langage computationnel dédié à notre problématique, c'est-à-dire à la construction de systèmes conceptuels dédiés à la Terminologie dite « classique » ? Nous verrons que c'est sur cette dernière option que notre choix s'est porté.

Enfin, ce n'est pas parce qu'on utilise un langage dont la syntaxe et la sémantique sont clairement définies et sur lesquelles on peut s'accorder, qu'on s'accordera nécessairement sur l'ontologie construite.

[33] Ce qui ne veut pas dire qu'on ne peut pas exprimer ces différences en passant, par exemple, à une logique d'ordre supérieur avec la notion de *prédicat rigide*, prédicat vrai dans tous les mondes possibles comme Homme(x) et Raisonnable(x). Une caractéristique essentielle est donc modélisable par un prédicat rigide. D'autres solutions sont possibles comme simuler l'ordre 2 en considérant les caractéristiques essentielles comme des individus et en introduisant des prédicats spécifiques à leur gestion : Essential_Characteristic (mortel), Opposite_Essential_Characteristic (mortel, immortel), etc.

PARTIE 2 : LE TOURNANT ONTOLOGIQUE DE LA TERMINOLOGIE

Cette deuxième partie présente notre travail théorique. Elle est consacrée au tournant ontologique de la Terminologie, c'est-à-dire à l'utilisation de l'ontologie pour la représentation du système conceptuel des terminologies et à ses conséquences sur le travail terminologique.

L'utilisation de l'informatique en Terminologie n'est pas nouvelle. L'analyse automatique de corpus à des fins linguistiques (extraction de candidats termes, de collocations, recherche de relations définitoires, études de concordances, etc.) a fortement impacté la Terminologie dans ses fondements et ses principes. Elle a donné lieu à un premier tournant, le « tournant textuel » de la Terminologie où on s'est davantage intéressé aux termes qui désignent des concepts en discours qu'aux concepts eux-mêmes posés comme extra linguistiques. L'opérationnalisation de terminologies à des fins de traitement de l'information, telle que nous l'entendons ici, est une problématique différente. Elle s'appuie sur une représentation computationnelle du système notionnel, qui replace le concept au premier plan. C'est le « tournant ontologique » de la Terminologie qui remet en question certains principes et méthodes de la Terminologie dite « classique ». L'Ontologie constitue, dans un monde numérique, une des perspectives les plus prometteuses pour la Terminologie[34]. La notion « d'onto-terminologie » [Roche 2007 ; 2012], terminologie dont le système conceptuel est une ontologie formelle, en est directement issue.

On verra dans un premier temps les problèmes, tant épistémologiques que logiques, que pose la Théorie du Concept de la Terminologie telle qu'elle est définie dans les normes ISO 1087-1 et ISO 704, qui entravent une mise en œuvre informatique. De ce constat, nous proposerons une Théorie du Concept et un langage associé nommé CLT (Concept Language for Terminology) qui, tout en respectant dans la mesure du possible les principes de la Terminologie dite « classique », pourra donner lieu à des représentations computationnelles du système conceptuel. On introduira notamment une couche d'abstraction supplémentaire qui permet de distinguer les objets (instances, individus), des concepts portant sur une pluralité d'objets pouvant être éventuellement vide[35].

Dans l'approche « onto-terminologique », les *termes* et les *noms de concept* ne se confondent pas puisqu'ils relèvent de systèmes sémiotiques différents. Les *noms des concepts* en plus d'être uniques, ne sont pas limités au seul rôle *d'identifiants* pour le système formel. Ils ont une importance particulière au sens où en les lisant on comprend la place qu'occupent les concepts dans le système

---

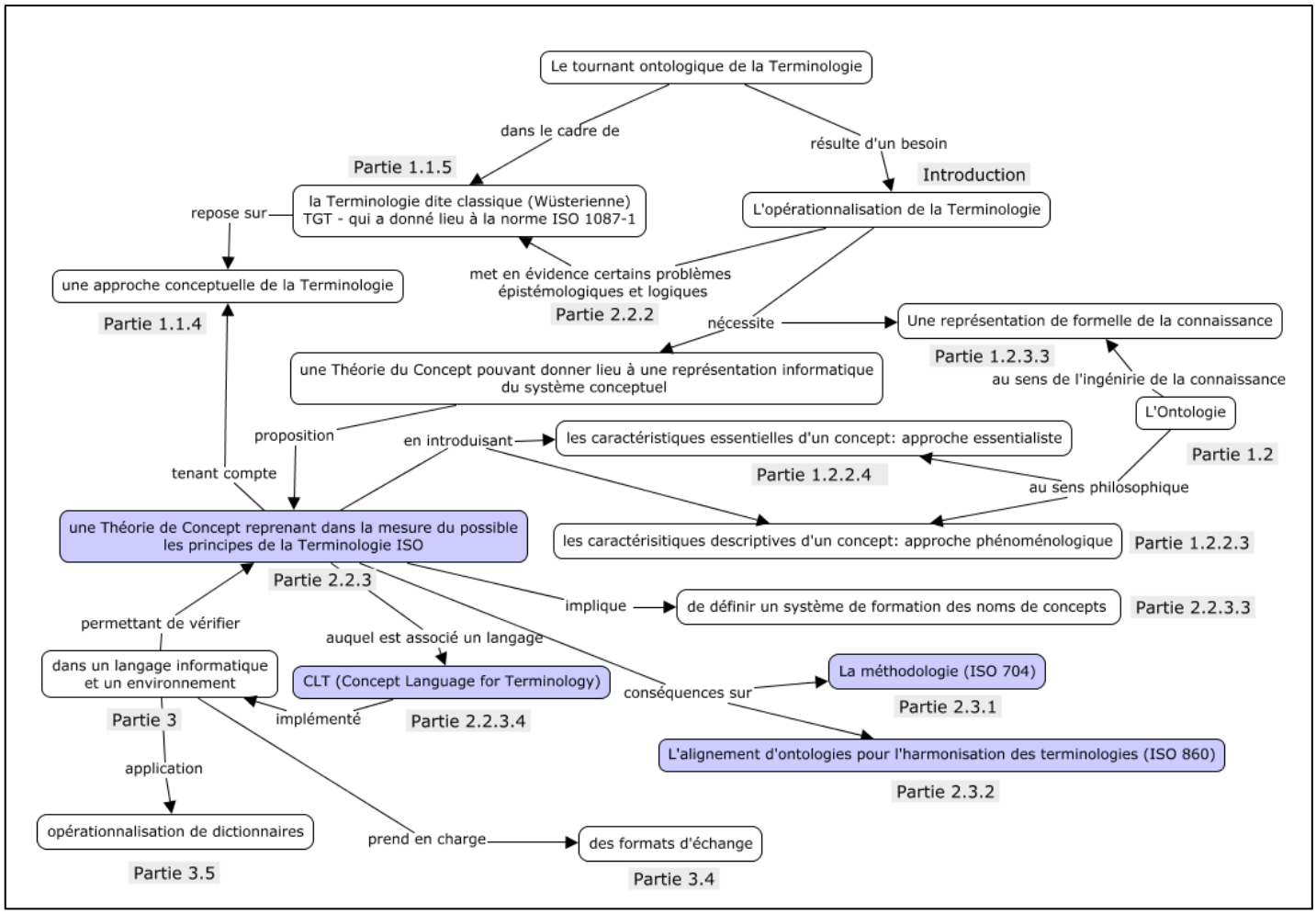
[34] Savoir si on peut construire une ontologie à partir de textes est un autre problème qui ne relève pas de notre travail, et ce d'autant plus que les ontologies construites à partir de textes ne correspondent en général pas à celles établies directement par les experts du domaine dans un langage formel [Rastier 2004], [Roche 2007].

[35] Par exemple, le concept <Satellite naturel de la lune> ne correspond à aucun objet et ne peut donc, même si le concept (idée) existe, être pris en compte par les normes ISO qui font dépendre la définition du concept selon la cardinalité de son extension.

notionnel ou la nature des objets subsumés par les concepts. La manière de nommer les concepts est donc loin d'être arbitraire et peut s'inspirer de certaines langues naturelles comme le chinois où un mot est formé par combinaison de caractères qui représentent les caractéristiques essentielles du concept dénoté par le mot. Le système qui régit la formation des mots en chinois peut être adopté pour nommer les concepts, le terme normé pourra alors se calquer avec profit sur le nom du concept. Le programme Treewords (voir Annexe C) a été développé afin d'illustrer le système de formation des termes chinois liés aux connaissances qu'ils dénotent.

Enfin, il sera question des apports méthodologiques du tournant ontologique sur la Terminologie elle-même. Expliciter le concept en mettant l'accent sur la notion de caractéristique essentielle n'est pas sans conséquence sur la façon de concevoir des terminologies, mais aussi, à travers l'alignement d'ontologies, de les harmoniser (la norme ISO 860 n'étant toujours pas révisée). Cette approche ouvre également la voie à l'opérationnalisation des terminologies comme par exemple une recherche sémantique multilingue.

Rappel du plan



2.1 INTRODUCTION

2.1.1 TOURNANT TEXTUEL

Avec l'avènement de l'ère numérique, la Terminologie a été impactée dans ses pratiques et ses méthodes pour aboutir à une opérationnalisation computationnelle. Cette opérationnalisation a commencé avec le traitement automatique de corpus, avec comme point de départ le terme. Il s'agit notamment d'extraire des termes à partir de corpus multilingues pour la traduction, de construire des réseaux de termes, de catégoriser l'information, d'extraire des relations sémantiques ou d'étudier le champ lexical des textes [Condamines & Rebeyrolle 2001]. Ce tournant textuel donne lieu à des logiciels pour la gestion de données terminologiques. L'information extraite est donc dépendante du discours. On s'éloigne alors de la Terminologie traditionnelle, une Terminologie prescriptive axée sur le concept, pour se centrer sur une terminologie descriptive axée sur le discours, on parle aussi dans ce cas de tournant textuel de la Terminologie.

Les bases de données terminologiques deviennent des bases de connaissance terminologiques (Terminological Knowledge Bases ou TKB) Le terme TKB est introduit pour la première fois en 1992 par Meyer et al. La création de ces bases de connaissance terminologiques implique l'analyse des termes utilisés dans les textes et l'extraction des relations sémantiques [Barrière, 2001]. Dans cette approche, les bases de connaissances terminologiques sont aussi nommées des ontologies [Temmerman & Kerremans, 2003], cependant la définition du concept ne peut être stable car elle est définie à partir de termes pris dans le discours. Les Terminologies prescriptive et descriptive font appel à des notions de concept différentes.

2.1.2 INTERET POUR L'ONTOLOGIE

Il existe de nombreux travaux en Terminologie s'intéressant à l'ontologie comme la termontographie, les ressources termino-ontologiques et l'ontoterminologie. Si la notion d'ontologie est importante pour chacune de ces approches, elle intervient néanmoins à des moments différents.

2.1.2.1 COGNITERM

COGNITERM est un projet établi au laboratoire d'Intelligence Artificielle de l'Université d'Ottawa depuis la fin des années 1980 sous la direction de Douglas Skuce. COGNITERM est un prototype de base de connaissance terminologique (BCT) à partir de l'outil informatique CODE[36]. Le but est de permettre à des traducteurs, rédacteurs techniques ou autres de produire plus facilement des textes de spécialité. La BCT de COGNITERM utilise une structure appelée « Concept Descriptor (CD) » pour représenter un concept. L'outil informatique possède une visualisation graphique qui se présente sous la forme d'arbres et de réseau de relations qui peuvent être hiérarchiques ou non. L'interface graphique permet une gestion des points de vue et une navigation dans les différents types

[36] Conceptually Oriented Descriptive Environment ou système global de gestion de la connaissance.

d'information comme les termes, les caractéristiques, etc. Le système donne la possibilité de construire par comparaison des définitions intensionnelles en mettant à disposition une matrice de comparaison de caractéristiques et comprend aussi un outil pour la détection d'incohérences dans les mécanismes d'héritages. La présence multiple du même concept ou de la même caractéristique est détectée automatiquement par le système. CODE aide le terminologue à construire des structures systématiques d'un domaine en lui proposant différentes possibilités de classification [Otman, 1996]. On peut noter qu'il y a donc une représentation computationnelle du concept mais sans prendre en compte les normes existantes et les conséquences sur la Terminologie elle-même.

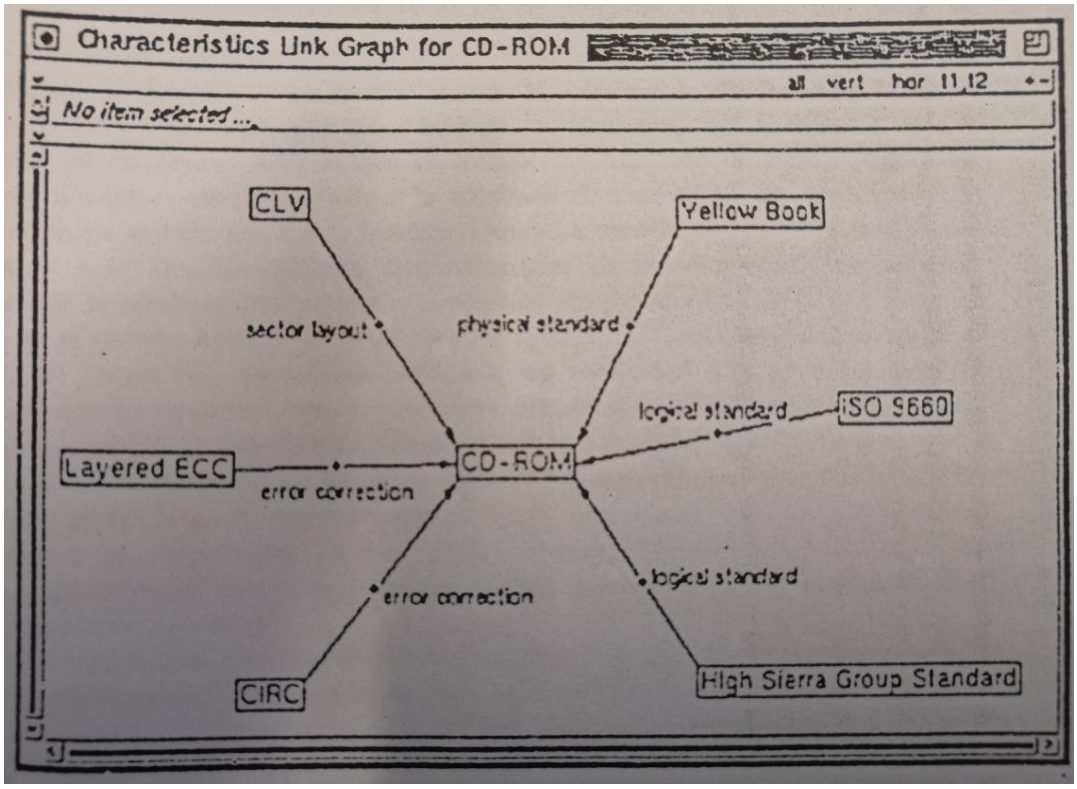


Figure 36 : Relations non hiérarchiques associées au terme CD-ROM [Otman, 1996] repris de [Meyer 94]

2.1.2.2 TERMONTOGRAPHIE DE RITA TEMMERMAN

La Termontographie se veut une approche pluridisciplinaire dans laquelle les théories et les méthodes pour l'analyse terminologique multilingue sont combinées avec l'Ontologie, car elles présentent des éléments communs. Lors de la création d'une ontologie ou de l'établissement d'une base de données terminologique, il s'agit en premier lieu d'identifier les sujets, de réduire l'étendue du domaine et d'acquérir les connaissances nécessaires à la compréhension des termes. La Termontographie propose une méthode de gestion et de représentation de la connaissance pour un domaine particulier combiné à une expertise du domaine à partir d'informations en langue naturelle. La connaissance est extraite manuellement ou de manière semi-automatique des corpus pour aboutir à une ontologie et à un dictionnaire terminologique [Temmerman, 2000].

2.1.2.3 RESSOURCES TERMINO-ONTOLOGIQUES

« Une ressource termino-ontologique (RTO) est une ressource comportant une composante conceptuelle, l'ontologie, et une composante lexicale, la terminologie. » [Tissaoui A. ,2009]. Elle permet de faire correspondre aux concepts d'un domaine les termes qui les désignent pour notamment indexer des documents. « La définition de formalismes pour représenter une RTO est l'objet de recherches récentes de sorte qu'il n'existe pas de standard à proprement parler. » [Tissaoui A. ,2009]. Reymonet propose un modèle de RTO dans lequel les termes sont reliés aux concepts par une relation de dénotation. [Reymonet et al., 2007], à partir de ce modèle, Tissaoui propose d'enrichir la composante conceptuelle, l'ontologie, par une composante lexicale, la terminologie, afin de gérer les évolutions d'ontologies. La RTO est aussi qualifiée d'ontologie à composante lexicale. Un lexique indépendant de l'ontologie est associé de manière à étiqueter les concepts. Un terme est présent dans la RTO seulement s'il est relié à au moins un concept qu'il désigne. [Tisaoui A., 2009].

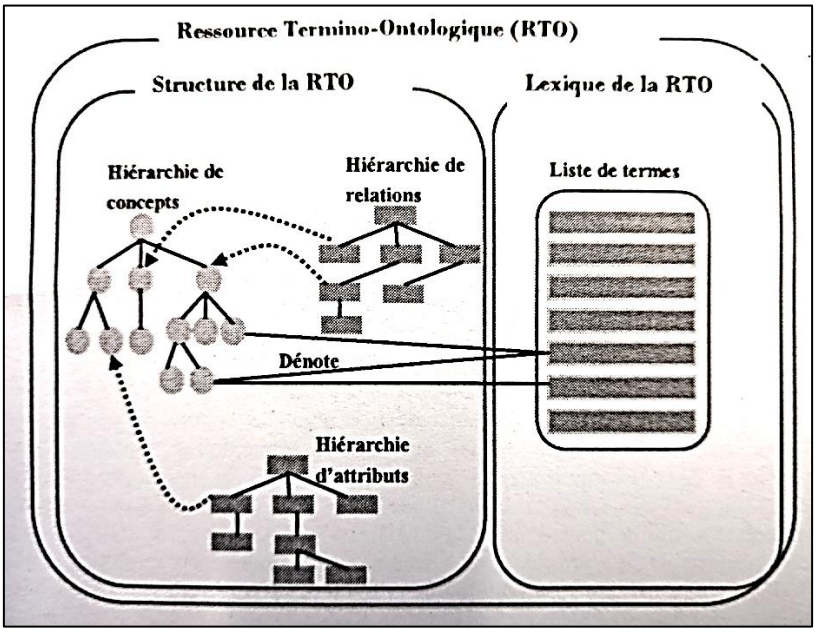


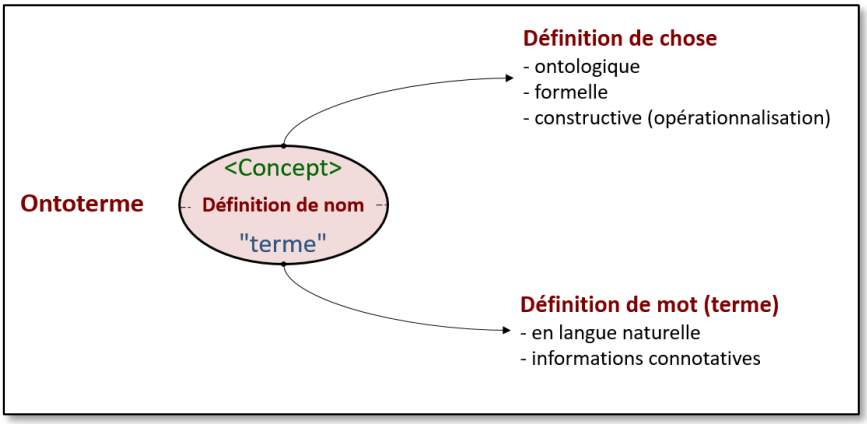
Figure 37 : Modèle de ressource termino-ontologique [Tissaoui, 2009]

2.1.2.4 L'ONTOTERMINOLOGIE

Une ontoterminologie est une terminologie dont le système conceptuel est une ontologie formelle [Roche 2007 ; 2012]. Elle rend possible une représentation explicite de la double dimension linguistique et conceptuelle de la Terminologie, les liant sans les confondre. Ainsi, à travers un double triangle sémiotique (figure 38), elle permet de ne pas confondre termes et noms de concept, définition du concept dans un langage formel et définition du terme en langue naturelle. L'ontoterminologie préserve ainsi la diversité langagière tout en normalisant la seule chose qui peut l'être à savoir les connaissances du domaine. La notion d'ontoterminologie est mise en œuvre dans

l'environnement logiciel Tedi[37] dédiée à la construction d'ontoterminologies [Papadopoulou & Roche 2018a ; 2018b].

Tout comme la ressource termino-ontologique, l'ontoterminologie repose sur la distinction et la mise en relation des dimensions linguistique et conceptuelle qui composent toute terminologie. L'ontoterminologie se distingue néanmoins de la ressource termino-ontologique par l'accent qu'elle porte sur les conséquences d'une représentation explicite et formelle du concept pour la Terminologie, tant au niveau de ses principes que de ses méthodes. Ainsi, tout en insistant sur la nécessité de pouvoir disposer d'une théorie du concept adaptée à la Terminologie, elle permet, avec l'introduction de la notion d'*ontoterme*, de prendre en compte les trois types de définition que sont la *définition de nom*, la *définition de chose* et la *définition de mot*. Elle distingue la définition du terme en langue naturelle, de nature explicative, de la définition constructive du concept écrite dans un langage formel. Enfin, l'ontoterminologie insiste également sur l'importance de la dénomination des concepts qui, au contraire des termes, n'est pas donnée, mais construite de telle sorte qu'elle traduise la nature des objets subsumés par le concept ou la place de celui-ci dans le système conceptuel[38] [Roche 2019].



La notion d' « ontoterme »

[37] http://ontoterminology.com/tedi

[38] Le nom du concept étant construit à partir de ses caractéristiques essentielles

2.2 TOURNANT ONTOLOGIQUE

2.2.1 INTRODUCTION : LORSQUE LE CONCEPT REDEVIENT LE POINT DE DEPART

Une ontologie construite à partir de textes ne correspond en général pas à celle établie par un expert du domaine dans un langage formel : « dire n'est pas concevoir » [Roche 2007]. C'est pourquoi, il est nécessaire de revenir au concept en se détachant du discours et de proposer un modèle de concept computationnel. Cette proposition de théorie du concept détaillée dans cette partie reste basée sur la norme ISO 1087-1 de la Terminologie mais adaptée à une opérationnalisation. Ce modèle permet d'envisager « un tournant ontologique » de la Terminologie et constitue une des perspectives les plus prometteuses pour la modélisation du système conceptuel d'une terminologie [Roche 2005]. La notion « d'ontoterminologie », terminologie dont le système conceptuel est une ontologie formelle[39], en est directement issue [Roche 2007].

L'opérationnalisation des terminologies n'est pas l'objectif de la norme ISO 1087-1 comme le rappelle la norme ISO 704[40], mais la théorie du concept sur laquelle repose la Terminologie doit pouvoir être formalisée d'un point de vue logique, pour garantir une cohérence et une validité du modèle. L'ontoterminologie qui explicite le concept dans un langage dédié permet de garantir une stabilité du concept, contrairement au *signifié*, construit en discours et donc variable par nature ne se confond plus avec le concept. De même le terme ne se confond plus avec le nom du concept. Si le terme est donné (issu des discours), le nom du concept est construit de telle sorte qu'il traduit ce qu'*est* la chose et puisse par sa seule lecture indiquer la place qu'occupe le concept dans le système conceptuel. La dénomination du concept contrairement au terme, relève du système formel.

Le double triangle sémiotique (Figure 38) illustre la double dimension de la Terminologie et les rapports entre les différentes notions qu'elle mobilise [41] [Roche 2007].

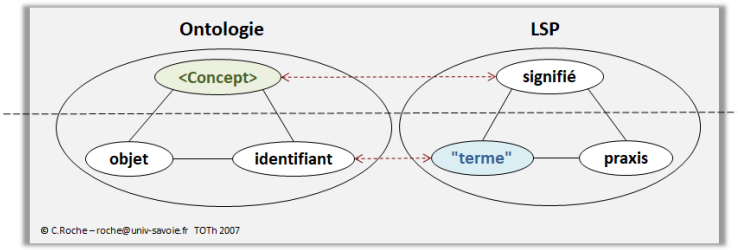


Figure 38 : Le double triangle sémiotique

[39] C'est-à-dire une définition des concepts et de leurs relations dans un langage intelligible par un ordinateur.

[40] « L'objectif du travail terminologique, tel que décrit dans le présent document, est par conséquent la clarification et la normalisation des concepts et des terminologies pour la communication entre humains. Le travail terminologique peut servir à la modélisation des informations et des données, toutefois la présente Norme internationale ne traite pas des relations avec ces domaines. » [ISO 704 :2009, p.V]

[41] Si la linguistique s'intéresse à l'étude des rapports entre signifiants et signifiés, la Terminologie porte une attention particulière entre objets et concepts.

2.2.2 Critiques de la norme ISO 1087-1

2.2.2.1 LES NORMES ISO 1087-1 :2000 ET ISO 704 :2009

La Terminologie au sens de l'ISO[42] repose sur deux normes transverses.

La norme ISO 1087-1 :2000 « Travaux terminologiques – Vocabulaire - Partie 1 : Théorie et application » a pour objectif essentiel de « fournir une description systémique des concepts appartenant au domaine de la terminologie et de clarifier l'usage des termes les désignant ». Elle « établit le vocabulaire fondamental de la théorie et de la pratique du travail terminologique ». Les définitions sont données en anglais et en français sous la forme suivante :

3.2.1
concept
unit of knowledge created by a unique combination of **characteristics** (3.2.4)

NOTE Concepts are not necessarily bound to particular languages. They are, however, influenced by the social or cultural background which often leads to different categorizations.

3.2.1
concept
notion
unité de connaissance créée par une combinaison unique de **caractères** (3.2.4)

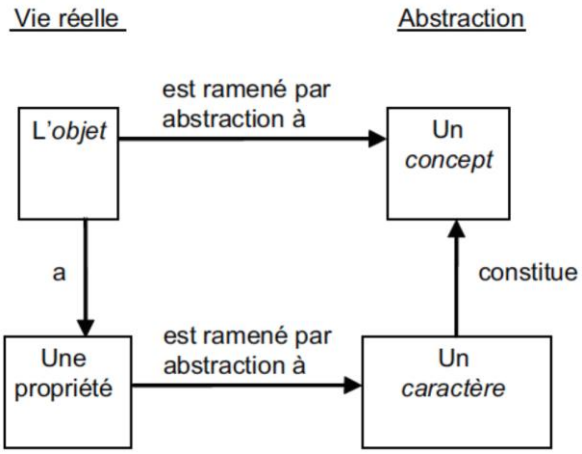
NOTE Les concepts ne sont pas nécessairement liés à des langues particulières. Ils sont cependant soumis à l'influence du contexte socioculturel qui conduit souvent à des catégorisations différentes.

La structure du document est éclairante. La notion de *concept* est au cœur de la démarche. La notion de *définition* suit celle de *concept* et précède celle de *terme*. La *définition* est une « représentation d'un concept par un énoncé descriptif permettant de le différencier des concepts », et le *terme* est une « désignation verbale d'un concept général dans un domaine spécifique ». La définition du terme est donc une *définition de chose*, c'est-à-dire une définition du concept désigné par le terme.

1 Domaine d'application 1
2 Référence normative 1
3 Vocabulaire 2
3.1 Langue et réalité 2
3.2 Concepts 2
3.3 Définitions 6
3.4 Désignations 6
3.5 Terminologie 10
3.6 Aspects du travail terminologique 10
3.7 Produits terminologiques 12
3.8 Données terminologiques 13

[42] L'ISO (Organisation internationale de normalisation) est une fédération mondiale d'organismes nationaux de normalisation (comités membres de l'ISO). L'élaboration des Normes internationales est en général confiée aux comités techniques de l'ISO. Chaque comité membre intéressé par une étude a le droit de faire partie du comité technique créé à cet effet. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'ISO, participent également aux travaux. L'ISO collabore étroitement avec la Commission électrotechnique internationale (CEI) en ce qui concerne la normalisation électrotechnique. Cf. Avant-propos [ISO 1087-1]

La norme ISO 704 :2009 « Travail terminologique - Principes et méthodes » vise à établir et à harmoniser « les principes fondamentaux et les méthodes permettant d'élaborer et de compiler des terminologies, qu'il s'agisse d'activités menées dans le cadre de la normalisation ou non, et décrit les liens établis entre les objets, les concepts et leur représentation par des terminologies ». Elle fixe également « des principes généraux qui régissent la formation des termes et appellations et la formulation des définitions ». Cette norme complète la norme ISO 1087-1 avec en illustrant et expliquant les notions définies dans la 1087-1. Ainsi la norme 704 précise pour la notion de concept : « Lorsqu'un concept décrit ou correspond à un ensemble de deux ou plusieurs objets constituant un groupe de par des propriétés communes, il est appelé concept général et, dans les langues de spécialité, sa désignation prend la forme d'un terme ». Elle présente les principes épistémologiques de la Terminologie sous la forme du schéma suivant :



2.2.2.2 PROBLEME DE COHERENCE ENTRE LES NORMES

9 ans séparent la publication des normes ISO 1087-1 et ISO 704. Il est inévitable que certaines divergences apparaissent. Cependant, il faut souligner une incohérence apparue avec la version de 2009 de la 704 concernant la notion de *caractère essentielle*. Si cette notion est primordiale pour la norme ISO 1087-1 et pour la Terminologie elle-même, la révision en cours de la 1087 confirme ce statut, ce n'est pas le cas de la norme IOS 704 :2009 qui affirme : "Terminology work is concerned with the intension and designation of concepts, and in this context necessary, sufficient, and essential characteristics are not used" ISO 704:2009 – p.7, sous prétexte que les caractéristiques essentielles sont utilisées pour déterminer l'extension d'un concept[43]. Nous pouvons espérer que dans le cadre de la prochaine révision de la norme ISO 704 on en revienne à sa version de 2000 qui précisait : « Not all characteristics are equally important. For practical purposes, the essential characteristics of the intension shall be the focal point of any analysis and may differ according to specific fields. Characteristics are considered essential if they are indispensable for the understanding of the concept in a particular field of knowledge; the absence of an essential characteristic fundamentally changes the concept. The absence of an essential characteristic in the course of an analysis will lead to poor or even erroneous understanding of the concept".

[43] "totalité des objets auxquels correspond un concept » [ISO 1087-1]

2.2.2.3 PROBLEME DE COHERENCE DANS LA MANIERE DE REPRESENTER LE MONDE (THEORIE DU CONCEPT)

La théorie du concept de la Terminologie telle qu'elle est actuellement définie dans la norme ISO 1087-1 propose une représentation du monde telle qu'elle est schématisée dans la figure ci-dessous[44].

La notion d'*objet* (individu, chose) n'existe pas en tant que telle (pas de représentation dans la norme), il sera pris en compte par l'intermédiaire d'un concept dont l'extension se réduit à un seul élément, l'objet lui-même, entraînant dans le discours une certaine confusion entre l'objet lui-même et l'ensemble réduit à ce seul objet[45]. Il n'existe dans la théorie du concept de la Terminologie dite « classique » (TGT et ISO) que des concepts que l'on différencie selon la cardinalité de leur extension : un *concept unique*[46] correspond à un seul objet quand le *concept général*[47] correspond à au moins deux objets.

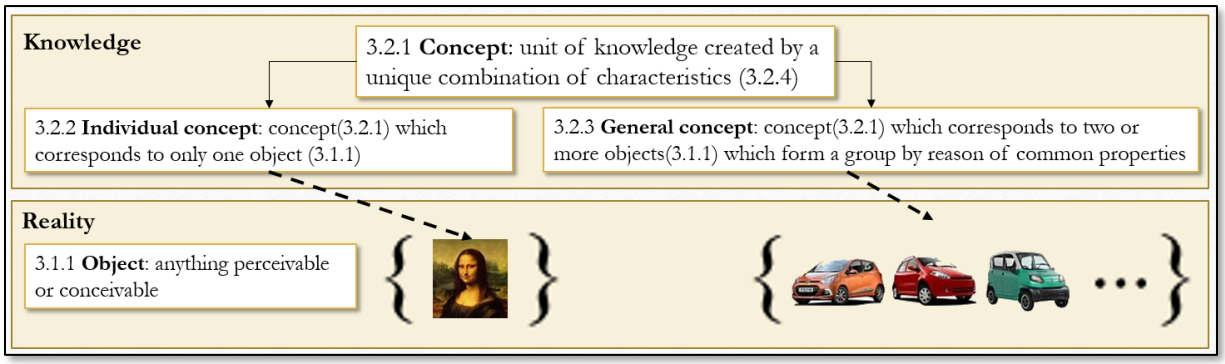


Figure 39 : Théorie du concept de la Terminologie de la norme ISO-1087

Une telle théorie du concept de la Terminologie soulève un certain nombre de problèmes dont les suivants :

- Que dire des concepts qui ne correspondent à aucun objet ou non encore connu(s) ?

---

[44] La norme ISO 1087-1 est en cours de révision sous la direction de H. Kockaert (Convenor) et C. Roche (Project Leader). La norme [ISO 1087-2:2000 Travaux terminologiques -- Vocabulaire -- Partie 2: Applications logicielles. Informations générales] a été annulée. Notre travail tient compte de la révision en cours de la norme renommée ISO 1087. Malheureusement, la notion d'objet n'est toujours pas représentée dans la norme.

[45] la proposition faite par la France d'introduire une représentation explicite de la notion d'objet dans le cadre de la révision de la norme ISO 1087-1 a été rejetée lors de la réunion du TC 37 de l'ISO de Copenhague en 2016.

[46] « concept qui ne correspond qu'à un seul objet » [ISO 1087-1]

[47] « concept qui correspond à deux objets ou plus, qui forment un groupe en raison de propriétés communes » [ISO 1087-1]

Par exemple : les concepts <Homme sur Mars> et <Satellite naturel de la Lune> ne réfèrent à aucun objet, et pour autant ils traduisent bien une idée.

- Que dire d'un concept dont l'extension se réduit à un seul élément qui correspond à une idée générale sans se confondre avec ce seul élément ?

Par exemple le concept <Satellite naturel de la Terre> qui ne réfère qu'à un seul objet désigné par l'appellation[48] 'Lune' sans pour autant se confondre avec cet objet.

- Que dire d'un concept dont l'extension change dans le temps ? Doit-il changer de nature ?

Par exemple le concept de <Satellite artificiel de la Terre> avant 1956, en 1956, en 1957

- Que dire de la relation générique entre le concept individuel et le concept général ?
  Est-ce que le concept individuel <France> est une spécialisation du concept général <Pays> et dans ce cas, quelle est sa caractéristique distinctive ? Ou en est-il une exemplification ?

  On peut également se poser un certain nombre de questions concernant les 'caractères'[49] des objets et sur la nature des connaissances qu'elles traduisent. Ainsi :

- En reprenant l'exemple des pointeurs en informatique (souris) de la norme ISO 704 : « avoir une couleur » et « muni en dessous d'une boule » traduisent-il des connaissances de même nature ? n'y-a-t-il pas confusion entre description (la couleur) et définition (muni d'une boule) ?

- Doit-on créer un nouveau concept individuel lorsqu'une de ses caractéristiques change ?
  Par exemple, si la couleur d'une voiture change, ne s'agit-il pas de la même voiture ?

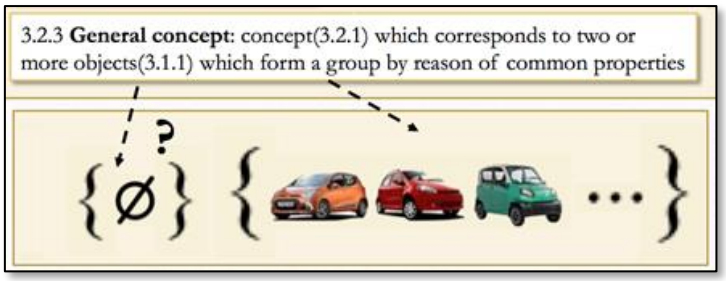


Figure 40 : ISO-1087 concept général

Ces remarques montrent non seulement un manque de spécification pour un modèle computationnel de la théorie du concept mais aussi un manque de cohérence dans la manière de représenter le monde.

---

[48] « désignation verbale d'un concept unique » [ISO 1087-1], correspond à un nom propre.

[49] « propriété abstraite d'un objet ou d'un ensemble d'objets » [ISO 1087-1]

Pour conclure, la notion d'*objet* doit pouvoir être représentée directement sans être obligé de passer par la notion de concept individuel dont l'extension se réduit à un seul élément, l'objet lui-même. De même, un concept peut exister sans qu'aucun objet ne lui corresponde et un objet pourrait très bien ne pas avoir de concept affilié. Enfin, il nécessaire d'introduire une notion de caractéristique permettant de traduire l'état dans lequel peut se trouver un objet sans pour autant en créer un nouvel à chaque changement d'état.

2.2.3 PROPOSITION D'UN MODELE CONCEPTUEL

2.2.3.1 MODELE DE CONCEPT (THEORIE DU CONCEPT) : DES PRINCIPES EPISTEMOLOGIQUES A SPECIFIER

Dans la continuité de la théorie du concept définie dans la norme ISO-1087, il s'agit de proposer une théorie du concept compatible avec une opérationnalisation de la Terminologie, c'est-à-dire une théorie du concept pouvant être définie dans un modèle computationnel. La principale différence entre les deux théories porte sur l'ajout d'une couche d'abstraction supplémentaire (Figure 41) permettant de manipuler directement les objets du réel. Dans cette proposition, la notion de concept n'est pas définie en fonction du nombre d'objets (tout ce qui peut être perçu ou conçu) que le concept représente. Ici l'objet joue le rôle d'exemplification du concept. La notion d'unité de connaissance est utilisée ici pour désigner une connaissance singulière (un objet) ou une connaissance plurielle (un concept, une classe). La connaissance est qualifiée de plurielle lorsqu'elle porte sur un ensemble d'objets, un ensemble qui peut être vide ou ne contenir qu'un seul objet. Un concept peut alors exister sans exemplification et réciproquement des objets peuvent exister sans que le concept soit pour autant défini.

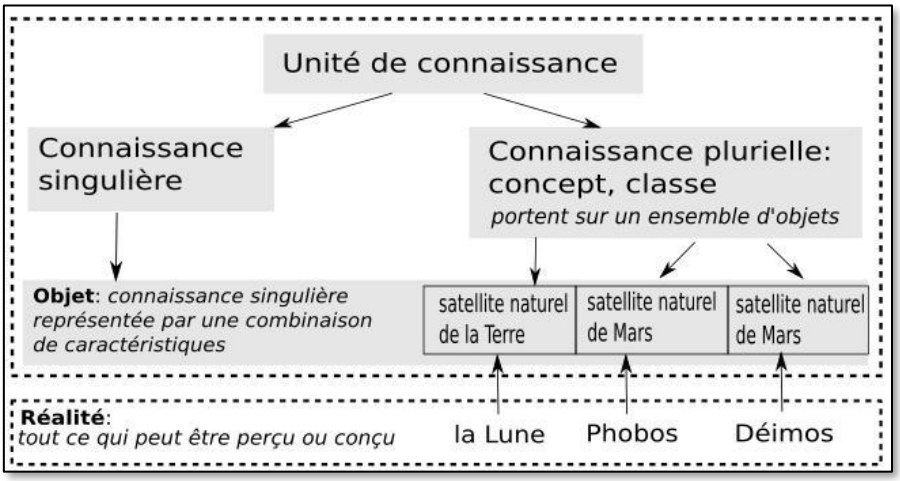


Figure 41 : Théorie du concept proposée

Les principes épistémologiques du modèle conceptuel proposé sont spécifiés comme suit :

2.3.1.1 POINT DE VUE, DOMAINE ET SOUS-DOMAINE

Une même entité de la réalité peut donner lieu à des représentations différentes, selon le point de vue sous lequel on se place dans un domaine. Les unités de connaissance sont dépendantes du point de vue d'un domaine. Le domaine, branche spécialisée de la connaissance peut être subdivisée en sous-domaines.

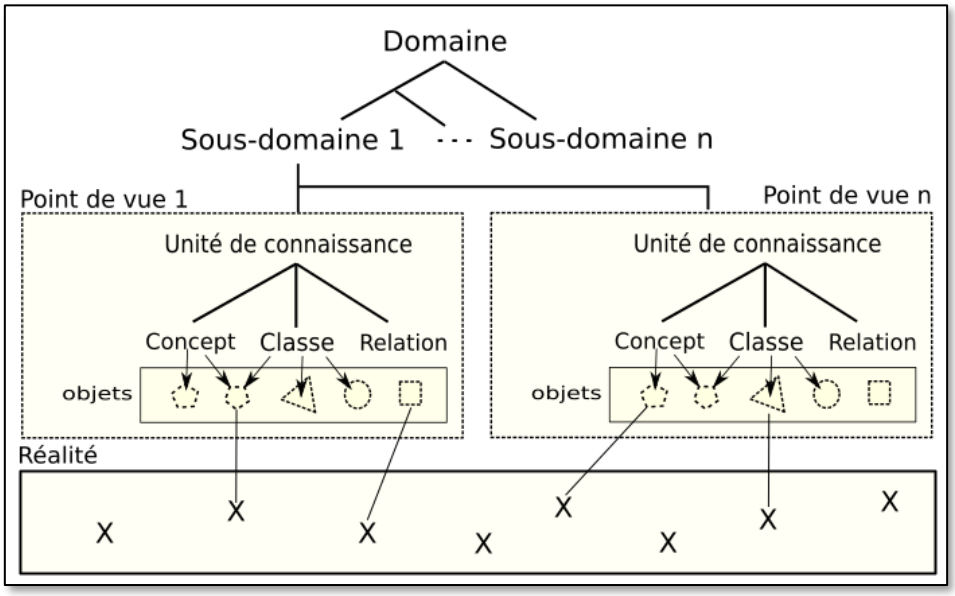


Figure 42 : Point de vue, domaine et sous-domaine

2.3.1.2 CARACTERISTIQUE

L'objet, unité de connaissance singulière, sert de support aux caractéristiques.

Caractéristique : « qui constitue le trait distinctif ou spécifique de quelque chose ou de quelqu'un. *Un fait, un signe caractéristique, une différence caractéristique ; être caractéristique de qqc* » (TLFi).

Caractère : « Trait(s) distinctif(s) d'une chose. Le caractère spécifique, dominant d'une chose ; dégager, souligner le caractère particulier d'une chose » (TLFi)

On utilisera plutôt le terme de caractéristique que de caractère (utilisé actuellement dans la version française de l'ISO) car on parlera plutôt de caractéristiques d'une machine par exemple, caractère : « manière d'être » [Rey, 1992].

Afin d'appréhender le réel et d'organiser les choses du monde, il est nécessaire de distinguer deux types de caractéristiques. « Sans un principe de sélection, la séparation entre l'essentiel et ce qui ne l'est pas perd son sens, et, sans cette séparation il n'y a pas de science. » [Théorie de la Définition p.247. H. Rickert].

Dans la norme ISO-704, si on prend l'exemple de la souris mécanique, les caractères : 'possède au moins un bouton' et 'couleur' ne sont pas des caractéristiques de même nature. C'est pourquoi on

distingue deux types de caractéristiques, les *caractéristiques essentielles* et les *caractéristiques descriptives*.

Ces deux types de caractéristiques jouent un rôle différent selon qu'on se place au niveau du concept ou de l'objet. Nous verrons qu'un concept se définit comme une combinaison unique de caractéristiques essentielles et descriptives. Les caractéristiques essentielles définissent et structurent le système conceptuel quand les caractéristiques descriptives modélisent la structure des objets et leurs états possibles (valeurs des caractéristiques descriptives).

2.3.1.2.1 CARACTERISTIQUE ESSENTIELLE

Une caractéristique est dite essentielle lorsque retranchée de la chose, la chose n'est plus ce qu'elle est. Elle exprime la nature, l'essence de la « chose » et ne peut donc être valuée. C'est une combinaison unique de caractéristiques essentielles qui établit la nature du concept ou de l'objet. La définition de l'ISO va dans le même sens : « Caractère essentiel : caractère indispensable pour comprendre un concept. » [ISO 1087-1, 3.2.6]

Les caractéristiques essentielles structurent le système conceptuel, d'où la notion de *caractère distinctif* définie comme « caractère essentiel utilisé pour distinguer un concept d'un autre » [ISO 1087-1, 3.2.7].

Nous pouvons remarquer que certaines caractéristiques essentielles peuvent être dépendantes d'autres caractéristiques. Par exemple, le concept <Véhicule à moteur électrique> est défini par les caractéristiques essentielles /avec moteur/ et /électrique/, la caractéristique /électrique/ ne pouvant être appliquée que sur des véhicules à moteur, c'est-à-dire requiert la présence de la caractéristique /avec moteur/. Cette notion de dépendance entre caractéristiques, notion importante dans le cadre de la modélisation et présente dans le Manuel de Terminologie de Felber [Felber, 1987], n'a pas été reprise par les normes ISO.

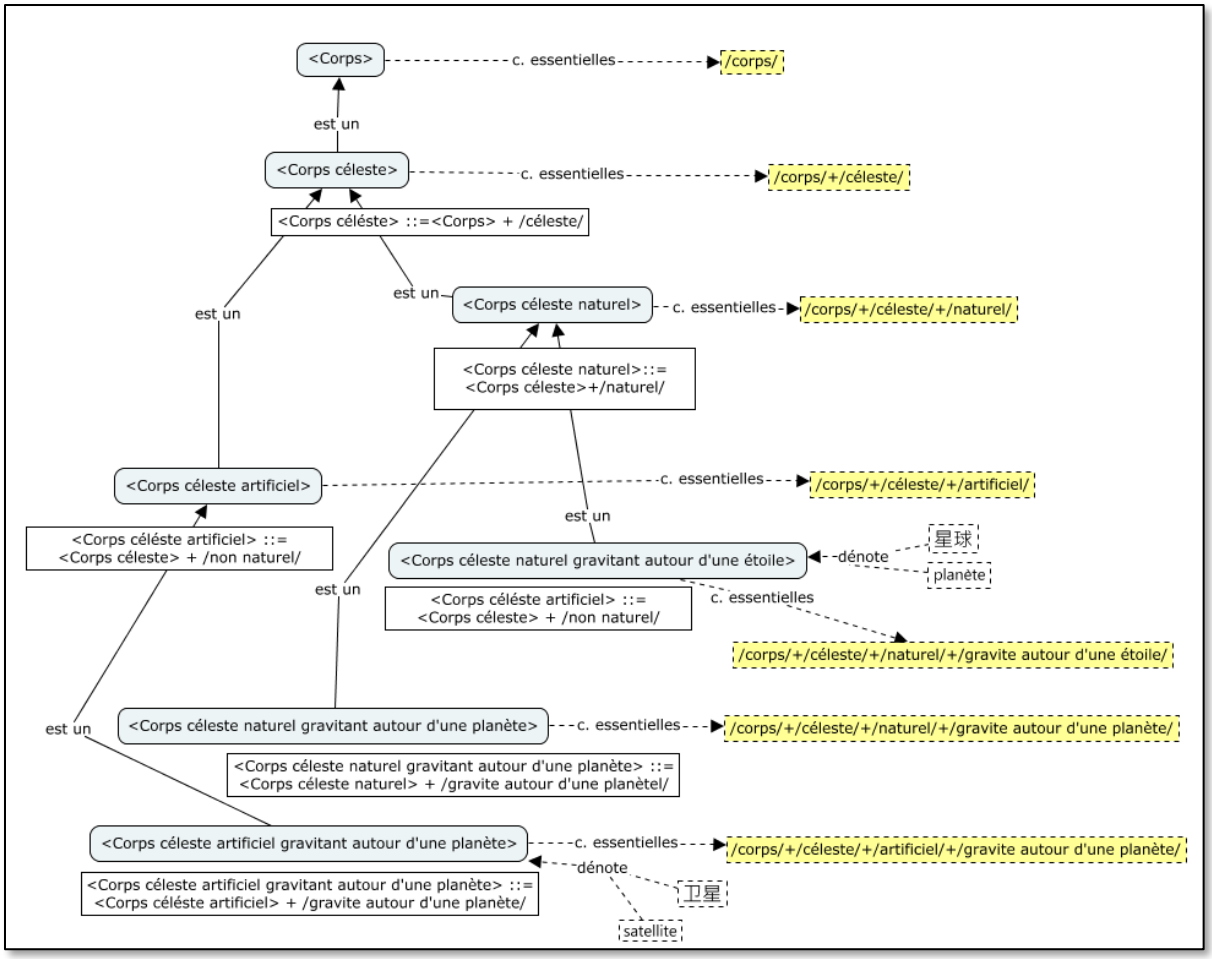
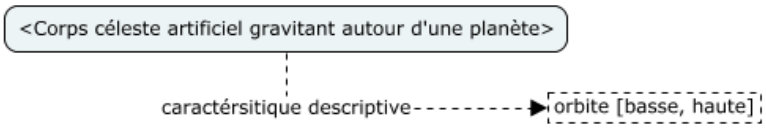


Figure 43 : Représentation des caractéristiques essentielles

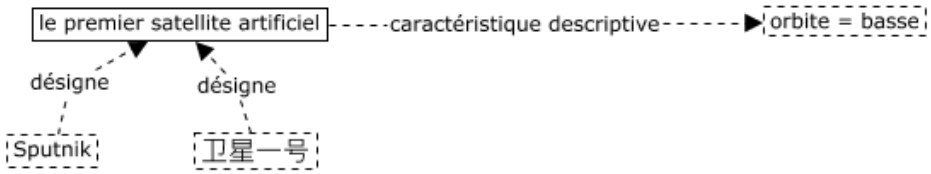
2.3.1.2.2 CARACTERISTIQUE DESCRIPTIVE

A la différence des caractéristiques essentielles, les caractéristiques descriptives ne participent pas directement à la définition du concept mais expriment la manière dont on perçoit les objets qui en relèvent. Un concept peut donc tout à fait exister sans caractéristique descriptive. Les caractéristiques descriptives sont d'une certaine façon dépendantes des caractéristiques essentielles. En effet la caractéristique descriptive « couleur » n'a de sens que si le concept en question possède une caractéristique essentielle « objet physique » par exemple. Les caractéristiques descriptives sont définies au niveau du concept en précisant l'ensemble des valeurs possibles. Elles prendront une valeur particulière au niveau de la description de l'objet, valeur qui pourra varier en fonction de l'état dans lequel peut se trouver l'objet, sans que l'objet change pour autant de nature (il demeure une instance du concept dont il relève).

- Caractéristique descriptive au niveau du concept :



➢ Caractéristique descriptive au niveau de l'objet :



Un objet est dénoté soit par un nom propre, *Spoutnik* par exemple, soit par une *description définie*, c'est-à-dire une expression dénotant un seul objet. L'étude des différents types de *désignations*, qu'elles soient verbales (termes, appellations, etc.) ou non verbales (symboles), ne rentre pas dans le cadre de cette thèse.

2.3.1.3 AXE D'ANALYSE (TYPE DE CARACTERISTIQUE)

L'axe d'analyse, originellement *type de caractère*[50] dans la norme ISO-1087, est une catégorie de caractères servant de critère de subdivision lors de l'établissement de systèmes de concepts. Le système conceptuel repose sur les caractéristiques essentielles qui définissent la structure du système. Les caractéristiques essentielles peuvent être définies de manière exclusive pour définir un même aspect du concept. Un axe d'analyse ou type de caractéristiques servira de critère de subdivision lors de la construction du système conceptuel. Par exemple on pourra avoir un axe d'analyse 'Type de corps céleste' avec les caractéristiques essentielles /naturel/, /artificiel/, un axe d'analyse qui subdivisera alors les corps célestes en deux catégories distinctes.

2.3.1.4 CONCEPT

Le concept est une connaissance plurielle portant sur une pluralité d'objets de même nature (caractéristiques essentielles) et partageant la même structure (caractéristiques descriptives). Le concept est donc défini par une combinaison unique de caractéristiques essentielles et descriptives. Porteurs des caractéristiques essentielles, les concepts participent directement à l'organisation du système conceptuel *via* les relations qu'ils entretiennent entre eux. Le concept peut tout aussi bien porter sur plusieurs objets, un seul ou aucun, c'est-à-dire qu'il n'a aucune exemplification. L'objet est considéré ici comme une exemplification du concept ou une instance de celui-ci. Le concept sera utilisé comme un modèle pour l'instanciation d'objets. Le concept sert aussi de support aux caractéristiques descriptives, c'est-à-dire que le concept peut porter les structures des

[50] "catégorie de caractère (3.2.4) servant de critère de subdivision lors de l'établissement de systèmes de concepts" [ISO 1087]

caractéristiques descriptives, par exemple, la couleur, la taille, en définissant pour chacun un champ de valeur possibles. Lors de l'exemplification d'un concept (Figure 44), ses caractéristiques descriptives sont appliquées à l'objet et une valeur est attribuée à la caractéristique comme vu précédemment dans la définition de la caractéristique descriptive.

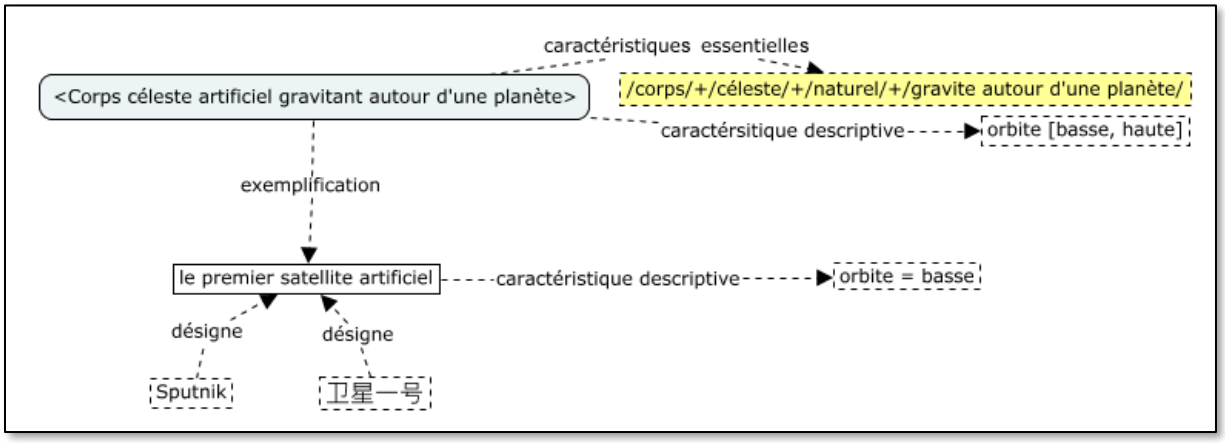


Figure 44 : Concept <Corps céleste artificiel gravitant autour d'une planète>

2.3.1.5 CLASSE

Si le concept porte sur des objets de même nature, il peut être intéressant de regrouper des objets pouvant être de nature différente mais vérifiant une même propriété portant sur leur état, c'est-à-dire sur les valeurs de leurs caractéristiques descriptives. Par exemple, on peut définir une classe qui regroupe tous les objets du concept <Corps céleste artificiel gravitant autour d'une planète>[51] vérifiant la propriété orbite='basse' :

Satellites-en-orbite-basse := { x / CorpsCélesteArtificielGravitantAutourPlanète(x) ∧ orbite(x)='basse' }.

L'ensemble des objets peut se réduire à un seul individu ou être vide.

Les objets regroupés au sein de la même classe peuvent donc être de nature différente, on peut définir la classe de tous les objets rouges (Figure 45) : objets rouges :={ x / couleur(x)=rouge }. Il est probable que cette classe contienne des objets de nature différente (une voiture rouge, un stylo rouge, etc.).

Cette notion de classe n'existe malheureusement pas dans la norme ISO 1087 et il n'a pas été possible de l'introduire, ce qui fait dire, qu'en suivant cette norme, 'enfant' est un concept spécifique au concept de 'personne' alors que ces notions sont de nature différente : Jean est une personne et le restera ce qui n'est pas le cas de son état d'enfant.

[51] A chaque concept et caractère essentiel est associé un prédicat unaire de même nom

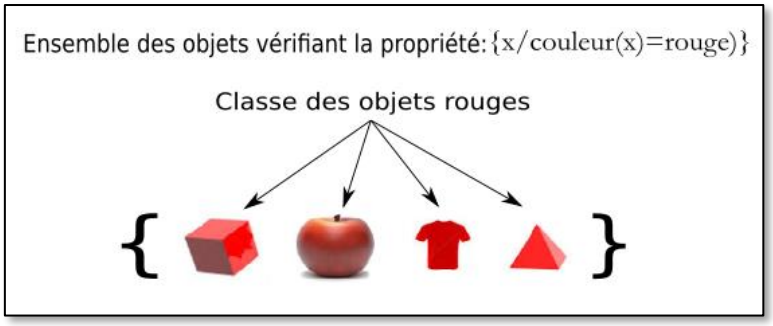


Figure 45 : Classe

2.3.1.6 L'OBJET, CONNAISSANCE SINGULIERE

L'objet, unité de connaissance singulière "indivisible", désigne une entité perçue ou conçue. L'objet représente l'idée qu'on se fait de la chose. La chose devient objet lorsqu'il est sujet à analyse, il est alors une représentation de la chose à travers les caractéristiques qu'on lui attribue. L'objet est le support d'une combinaison de caractéristiques descriptives et essentielles qui peut ne pas être unique, en effet deux objets de même nature avec les mêmes « propriétés » restent deux objets différents. Par exemple, deux pommes rouges identiques resteront deux objets distincts. Il en est de même pour les objets instanciés en programmation orientée objet. Le nom propre sert à désigner l'objet.

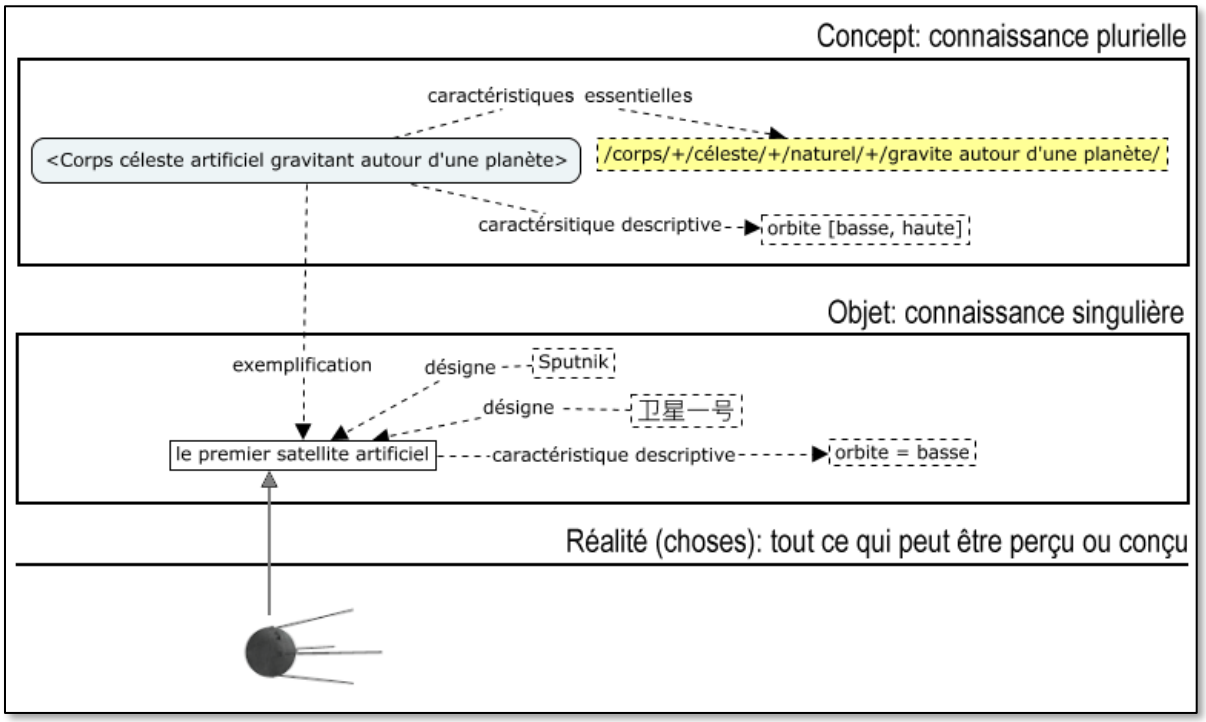


Figure 46 : L'objet, connaissance singulière

'Sputnik' et '卫星一号' (littéralement 'satellite numéro un' en français) sont ici des noms propres dénotant une instance du concept <Corps céleste artificiel gravitant autour d'une planète> dont la

valeur de la caractéristique *orbite* est 'basse'. On peut noter qu'il y a très souvent une transparence dans la formation des noms propres en chinois.

2.3.1.7 RELATION

C'est grâce aux relations entre concepts que le système conceptuel se forme. La place du concept dans le système conceptuel est tributaire des relations qu'il entretient avec les autres concepts.

« Les concepts n'existent pas en tant qu'unités de connaissance isolées mais sont toujours en relation les uns par rapport aux autres. » [ISO 704 : 2009, 5.5.1]

Toutes les relations définies ici sont binaires. Elles peuvent se retrouver à différents niveaux, au niveau des concepts, entre concepts et objets, entre classes et objets, entre les objets eux-mêmes. C'est pourquoi la nature des relations sera différente selon le niveau dans lequel on se place.

Une relation binaire est un ensemble de couples d'éléments pris dans un premier ensemble, appelé Domaine, et un deuxième ensemble, appelé Codomaine ou Range.

$R = \{ (x, y) / x \in \text{Domaine et } y \in \text{Codomaine} \}$

$R \subseteq \text{Domaine} \times \text{Codomaine}$

Par exemple, la relation générique a pour Domaine et Codomaine l'ensemble des concepts.

La relation partitive a pour Domaine et Codomaine l'ensemble des objets, mais sera spécifiée au niveau conceptuel. Ce point sera détaillé dans le chapitre dédié aux relations partitives.

2.3.1.7.1 LES RELATIONS ENTRE LES CONCEPTS

2.3.1.7.1.1 LA RELATION GENERIQUE

La relation générique ou relation de subsomption permet de lier un concept générique (concept superordonné) à un concept plus spécifique (concept subordonné). Le concept subordonné porte les caractéristiques essentielles du concept superordonné augmenté d'au moins une caractéristique essentielle. Les caractéristiques descriptives du concept générique seront également héritées et éventuellement étendues à de nouvelles caractéristiques descriptives au niveau du concept spécifique.

Nous garderons les définitions des concepts générique et spécifique de la norme ISO 1087 :

- « Concept générique : concept ayant la plus petite compréhension dans une relation générique » [ISO 1087-1, 3.2.15]
- « Concept spécifique : concept ayant la plus grande compréhension dans une relation générique » [ISO 1087-1, 3.2.16]

La relation générique porte uniquement sur des concepts : « Relation entre deux concepts dans laquelle la compréhension de l'un des concepts inclut celle de l'autre concept et au moins un

caractère distinctif supplémentaire. » [ISO 1087-1, 3.2.21] où un caractère distinctif est un « caractère essentiel utilisé pour distinguer un concept d'un autre. » [ISO 1087-1, 3.2.7]

La relation générique engendre une hiérarchie entre les concepts, elle sera qualifiée ici de relation d'ordre strict partielle.

- La relation générique est transitive[52]: si <C1> est un <C2> et C2> est un <C3> alors <C1> est un <C3>. Par exemple un <Corps céleste naturel> est un <Corps céleste> (figure 45)

- Elle est irréflexive[53] : <C1> n'est pas un <C1>, un concept n'est pas concept de lui-même. En effet, le concept subordonné doit au moins posséder une caractéristique essentielle distinctive par rapport au concept superordonné. C'est une relation pour laquelle aucun élément n'est en relation avec lui-même.

- Elle est asymétrique[54], si <C1> est un <C2>, <C2> n'est pas un <C1>. Par exemple un <Corps céleste naturel> est un <Corps céleste> mais un <Corps céleste> n'est pas une <Corps céleste naturel> (figure 47).

- Elle est partielle[55] au sens où deux concepts ne sont pas toujours comparables par la relation générique. Par exemple dans la figue 47 les concepts <Corps céleste artificiel> et <Corps céleste naturel> ne sont pas comparables.

Par conséquence, une relation générique implique l'héritage des caractéristiques essentielles et descriptives de la part du concept superordonné vers le concept subordonné. Un concept subordonné peut tout à fait avoir plus d'un concept superordonné, on parle alors « d'héritage multiple » (Figure 48). Cela est possible tant que leurs caractéristiques essentielles ne sont pas exclusives, c'est à dire que les concepts subordonnés soient définis avec des axes d'analyse indépendants.

---

[52] Définition formelle : $\forall x,y,z \in E, (x \mathrm{R} y \wedge y \mathrm{R} z) \rightarrow (x \mathrm{R} z)$. Quel que soit x, y et z appartenant à l'ensemble E, si x est lié à y et y est lié à z, alors x est lié à z par R.

[53] Définition formelle: $\forall x \in E, \rightarrow \neg (x \mathrm{R} x)$.Quel que soit x appartenant à l'ensemble E, x n'est pas lié à x par R.

[54] Définition formelle : $\forall x,y \in E, x \mathrm{R} y \rightarrow \neg (y \mathrm{R} x)$. Quel que soit x et y appartenant à l'ensemble E, si x est lié par la relation R à y alors y n'est pas lié à x.

[55] Définition formelle : $\exists x,y \in E , \neg (x \mathrm{R} y \vee y \mathrm{R} x)$. La relation R définie sur l'ensemble E est dite partielle si et seulement s'il existe un x ou un y tel que x n'est pas lié à y ou y n'est pas lié à x par R.

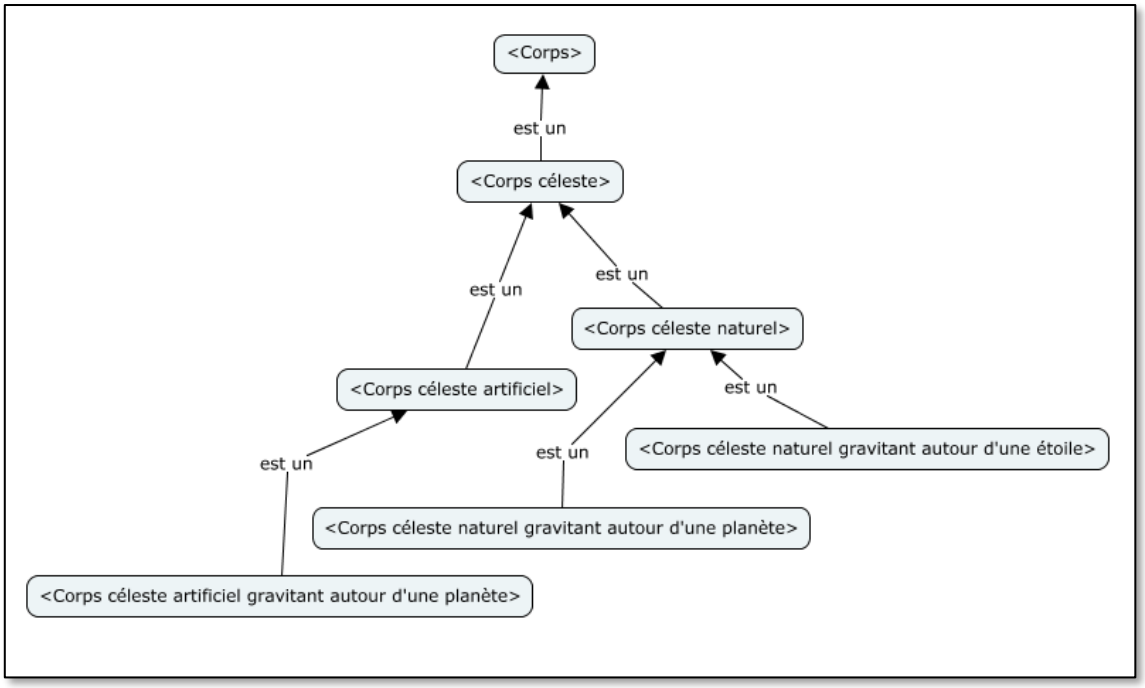


Figure 47 : Relation générique

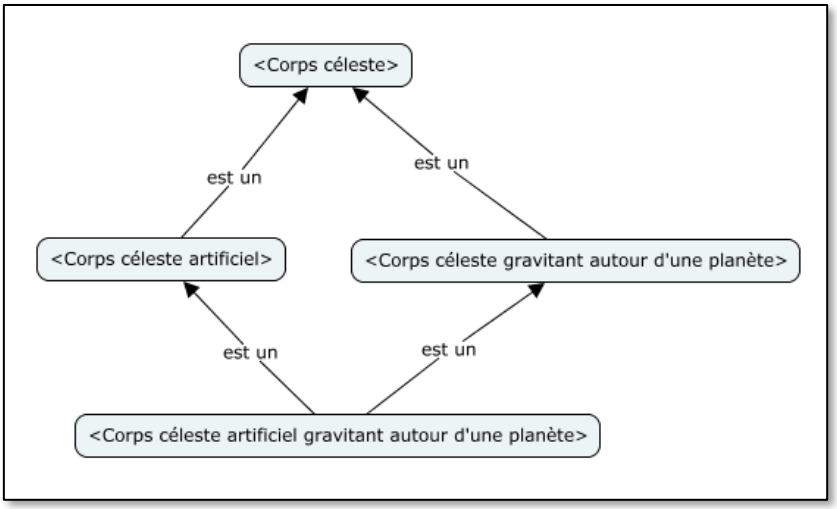


Figure 48 : Héritage multiple

En toute logique, l'instanciation[56] de concepts ne portent que sur des concepts dits « terminaux », c'est-à-dire des concepts qui ne possèdent aucun concept plus spécifique. Cependant, il est possible d'autoriser les instanciations de concepts génériques (des concepts pour lesquels il existe des concepts plus spécifiques) à la manière d'un langage de programmation orienté objet.

2.3.1.7.1.2 RELATION PARTITIVE

La relation partitive est définie (spécifiée) au niveau des concepts et est appliquée au niveau des objets. Elle décrit la manière dont vont être composés les objets du concept intégrant. Par exemple,

[56] Instanciation au sens où l'entend ici à la manière d'un langage de programmation orienté objet

‘notre système solaire’ instance d’un concept <Système solaire> sera lié par la relation partitive à ‘la Terre’[57] (Figure 49).

« Relation partitive : relation entre deux concepts dans laquelle l’un des concepts constitue le tout et l’autre concept une partie de ce tout. » [ISO 1087-1, 3.2.22]

Elle est nommée « partie de » ou « a pour partie » selon le sens de la relation. Un concept peut être une partie constituante ou un composant d’un autre. Par exemple <Système solaire> a pour partie <Corps céleste>. On parlera de « concept partitif » lorsque celui-ci est une des parties d’un autre concept appelé « concept intégrant ».

« Concept intégrant : concept dans une relation partitive considéré comme le tout. » [ISO 1087-1, 3.2.17]

« Concept partitif : concept dans une relation partitive considéré comme l’une des parties constituant le tout. » [ISO 1087-1, 3.2.18]

Dans la norme ISO 1087-1, la relation partitive est considérée au même titre que la relation générique comme une relation hiérarchique : « relation hiérarchique : relation entre deux concepts qui est soit une relation générique, soit une relation partitive » [ISO 1087-1, 3.2.20].

Même si la relation partitive sert à "hiérarchiser", la relation n’est pas forcément transitive. Suivant le sens commun, cela peut être le cas : Jean fait partie d’un laboratoire et le laboratoire fait partie de l’Université, alors Jean fait partie de l’Université. Cependant il existe de nombreux contre exemples : un océan, partie d’une planète qui elle-même est partie du système solaire, n’implique pas pour autant que l’océan fasse partie du système solaire. Donc sous la même appellation « relation partitive » existe des relations dont les propriétés peuvent varier selon la nature des objets mis en relation. Les rapports qui existent entre le tout et ce qui le constitue sont trop divers pour être réunis sous la même relation partitive. Le sujet a donné et donne toujours lieu à de nombreux travaux d’un point de vue linguistique[58]. C’est pourquoi on laissera la possibilité de définir des relations partitives avec des propriétés différentes. Par exemple on sera libre de définir une relation partitive « est membre de » qui sera transitive ou « fait de » qui ne le sera pas.

On considère ici la relation partitive comme une relation d’ordre stricte partielle. La relation partitive sera considérée « par défaut » comme une relation partitive, irréflexive, asymétrique et partielle.

En indiquant qu’une relation partitive lie un concept à un autre, cela signifie que la relation lie des instances du premier concept à des instances du deuxième concept. Lorsque le concept intégrant est instancié (exemplification du concept à travers un objet), le ou les concepts partitifs sont instanciés à la même occasion ou préalablement instanciés).

---

[57] Nom propre désignant une instance du concept <Corps céleste naturel>

[58] Les travaux de [Winston, 1987], [Lyons 1977], [Cruse 1986]

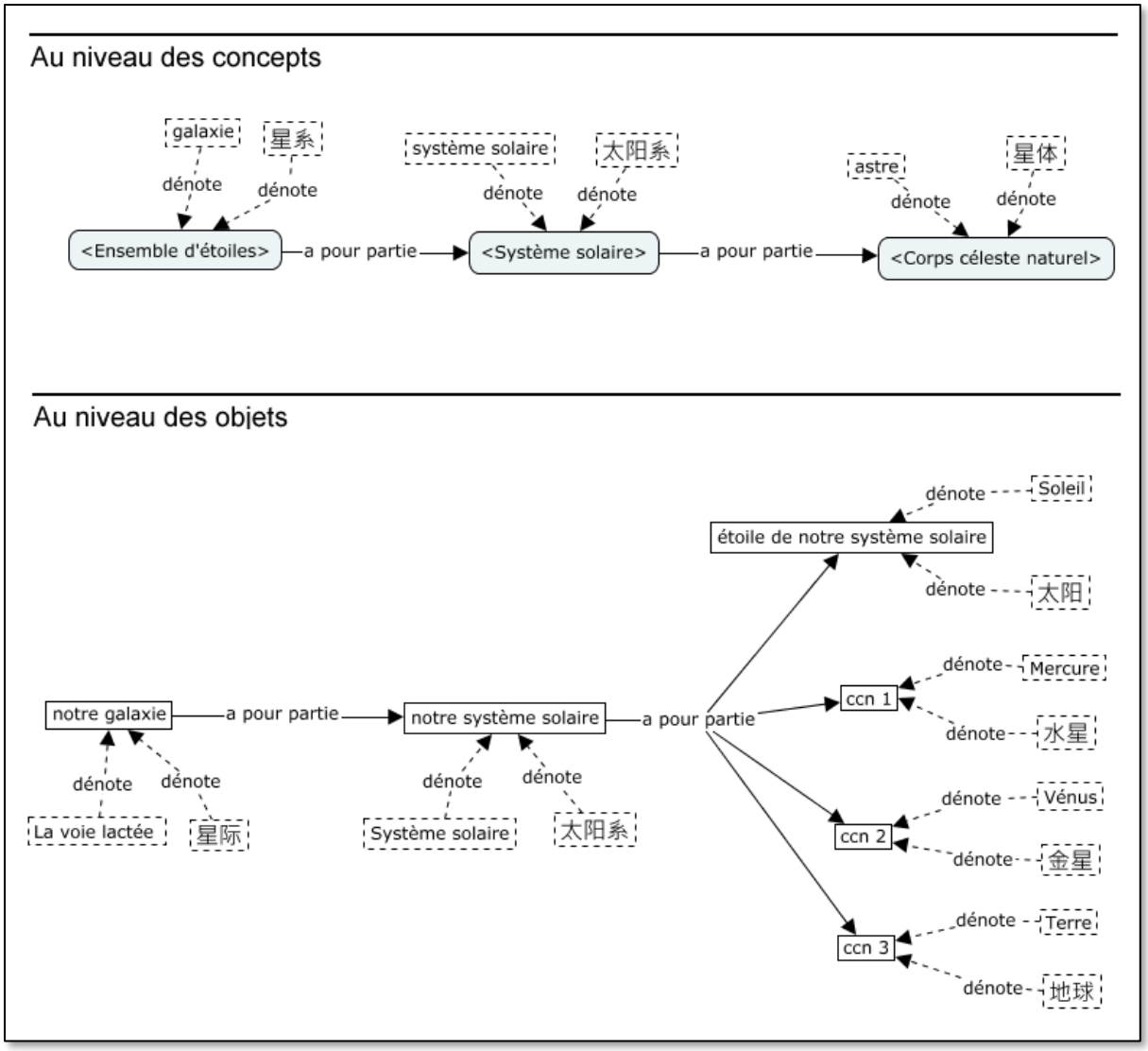


Figure 49 : Représentation de la relation partitive au niveau des concepts et des objets. A noter que le nom de l'objet (à différencier de ces dénotations) peut être généré sous la forme d'un index (incrémenté en fonction de son ordre d'instanciation) précédé de l'abréviation du nom du concept dont il est l'instance. Lorsqu'un objet n'est l'instanciation d'aucun concept, il portera simplement un index.

La relation est restreinte à un Domaine et un Codomaine particuliers. La relation partitive permet donc de lier des objets depuis un ensemble de départ (Domaine) à un ensemble d'arrivée (Range).

L'ensemble de départ peut être un ensemble de concepts : {<A>, <B>} , ou bien un ensemble de classes {classe A, classe B}.

R1= {(x,y) / C1(x) $\wedge$ C2(y)} Une relation dont « Domain » et « Range » sont des concepts.

- Domain = C1
- Range = C2

R2 = {(x,y) / classe1(x) $\wedge$ classe2(y)} Une relation dont « Domain » et « Range » sont des classes.

- Domain = classe1
- Range = classe2

2.3.1.7.1.2.1 CARDINALITES

A l'instar du langage de modélisation UML[59] (UML peut s'avérer utile pour la modélisation de concepts ISO [Kockaert H.,Antia B., 2008], [Kockaert H. et al.,2010]), on utilisera un système de notation similaire pour représenter les cardinalités des concepts reliés par une relation partitive. Cela permet d'indiquer dans le cas d'une instanciation des concepts le nombre d'objets impliqués (Figure 50).

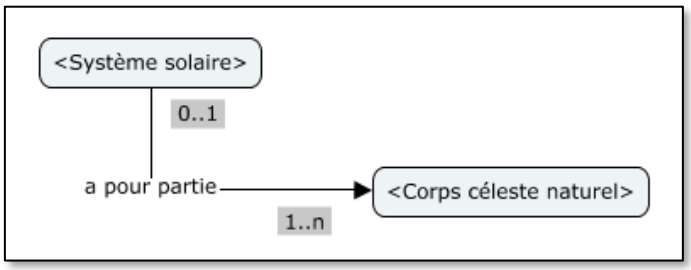


Figure 50 : Représentation des cardinalités dans une relation partitive

Dans l'exemple de la figure ci-dessus, l'ajout des cardinalités permet de savoir qu'un objet du concept <Astre> peut ne pas faire partie d'un système solaire ou au maximum un seul (cardinalité : 0..1), et un système solaire a au moins pour partie un astre (cardinalité : 0..n).

Rappelons que cette notion de cardinalité qui participe à la spécification du système conceptuel n'existe pas dans la norme ISO 1087.

2.3.1.7.1.3 RELATION ASSOCIATIVE

Les relations associatives même si elles ne participent pas directement à la définition des concepts participent à la description et à la compréhension du système conceptuel. Les objets existent indépendamment de la relation, il en est de même pour la relation partitive. On regroupera sous la même appellation « relation associative » toutes les relations autres que générique et partitive (pour les relations entre les concepts). Les relations regroupées sous cette appellation sont nombreuses, il peut s'agir de la fonction, la cause, la conséquence, la proximité, …

« Relation entre deux concepts ayant des liens thématiques non hiérarchiques fondés sur l'expérience. » [ISO 1087-1, 3.2.23]

On définira alors la relation associative comme une relation entre deux concepts ayant des liens thématiques autres que générique ou partitif. Il s'agit d'une relation d'ordre stricte partielle tout comme la relation générique et partitive.

2.3.1.7.1.3.1 CARDINALITES

Tout comme la relation partitive on pourra indiquer les cardinalités de la relation associative. Dans le cas d'une instanciation des concepts il sera possible de connaître le nombre d'objets qui pourront être impliqués.

---

[59] Acronyme de « Unified Modeling Language » est un langage graphique normalisé pour représenter la conception d'un système orienté objet.

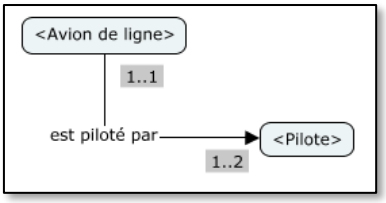


Figure 51 : Représentation des cardinalités dans la relation associative

Dans l'exemple ci-dessus, un avion de ligne peut être piloté par un ou deux pilotes au maximum. Et un pilote est associé à un avion de ligne au maximum.

2.3.1.7.2 LES RELATIONS ENTRE CONNAISSANCE PLURIELLE ET CONNAISSANCE SINGULIERE

2.3.1.7.2.1 LA RELATION D'INSTANCIATION : RELATION ENTRE UN CONCEPT ET DES OBJETS

La relation d'instanciation ici sera à l'image d'un langage de programmation objet et sera défini de la manière suivante :

Soit R la relation d'instanciation : a R b, où a est un objet et b est un concept.

- Domain = Objet
- Range = Concept feuille (le concept le plus spécialisé)

Par exemple : « Jean est instance de <Personne> »

2.3.1.7.2.2 LA RELATION ENTRE OBJETS ET CLASSES

La relation entre classe et objet est une relation d'appartenance et permet de lier des objets de nature différente à une classe.

Soit R la relation qui lie un objet à une classe : a R b, où a est un objet et b, une classe.

- Domain = Objet
- Range = Classe

Par exemple : « ma voiture rouge » est lié à la classe « objets rouges ». Cette relation permet de structurer les objets selon leur état.

2.3.1.7.3 LES RELATIONS ENTRE LES OBJETS

On retrouvera toutes les relations définies au niveau des concepts à l'exception de la relation générique, c'est-à-dire : la relation partitive et la relation associative (Figure 51).

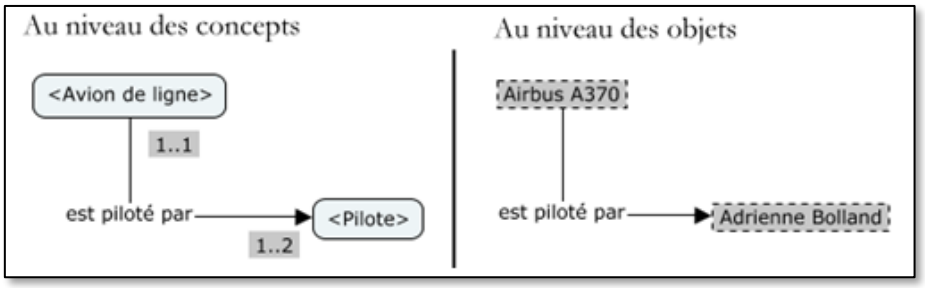


Figure 52 : Relations au niveau des objets

Remarque : Au niveau des objets, il n'y a pas lieu d'indiquer les cardinalités. Les cardinalités sont indiquées uniquement au niveau des concepts car elles permettent de donner des indications sur le nombre possible d'instanciation (un objet étant l'instanciation d'un concept).

2.3.1.8 DESIGNATION

« Désignation : représentation d'un concept par un signe qui le dénomme » [ISO 1087-1, 3.4.1].

La représentation d'un concept par un signe est à différencier du nom du concept. Contrairement aux noms des concepts, les désignations relèvent de la dimension linguistique de la Terminologie. On donnera dans ce modèle la possibilité de dénommer un concept par une désignation verbale qui est le terme.

« Terme : désignation verbale d'un concept général dans un domaine spécifique » [ISO-1087,3.4.3].

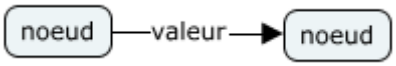
2.2.3.2 CARACTERISTIQUES ESSENTIELLES : COMMENT LES REPRESENTER ?

Notre objectif est de proposer une mise en œuvre informatique de la théorie du concept pour la Terminologie que nous venons de présenter. La question qui se pose est alors celui du choix du langage de représentation, à savoir si on peut utiliser un langage existant ou si on peut en définir un nouveau spécifiquement adapté à notre problématique.

La notion centrale de notre théorie, issue de la théorie du concept des normes ISO, est celle de caractéristique essentielle, ce qui nous amène au problème de la représentation de cette notion dans les différents langages que nous avons présentés précédemment.

2.2.3.2.1 REPRESENTATION DES CARACTERISTIQUES ESSENTIELLES AVEC CMAPTOOLS

Le modèle de CmapTools repose uniquement sur les notions de nœud et de lien orienté annoté :



Il nous faut donc, pour représenter les autres notions du modèle que nous proposons et en particulier pour celle de caractéristique essentielle s'appuyer sur la possibilité qu'offre CmapTools dans la représentation graphique des nœuds et des liens. On peut donc introduire différents codes sémiotiques pour exprimer ces différentes notions. Par exemple :

Un lien annoté « essential characteristic » suivi d'un nœud portant le nom de la caractéristique essentielle peut être ajouté au concept. Les caractéristiques descriptives sont ajoutées après le nom du concept. Il y a ainsi une distinction claire entre les caractéristiques essentielles et descriptives. Par ailleurs, le système conceptuel peut être exporté en XML sous la forme : nœud de départ, nom du lien, nœud cible. Il serait alors possible de déterminer les relations d'un concept ainsi que ses caractéristiques essentielles.

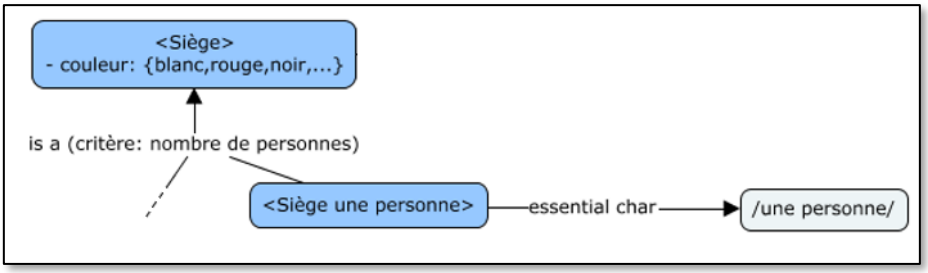


Figure 53: représentation des caractéristiques essentielles dans CmapTools

Néanmoins, cela reste un artifice graphique et l'impossibilité de définir des propriétés et de les vérifier lors de la construction de l'ontologie rend CmapTools, qui ne se veut pas un système formel, difficilement utilisable pour la construction d'ontologies formelles.

2.2.3.2.2 REPRESENTATION DES CARACTERISTIQUES ESSENTIELLES AVEC DES SYSTEMES A BASE DE SCHEMAS

La représentation des caractéristiques essentielles dans des systèmes à base de schémas tels que SRL ne pose pas de problème particulier. Une solution est de représenter une caractéristique essentielle sous la forme d'un attribut (slot) de valeur booléenne. La gestion de caractéristiques essentielles exclusives entre elles nécessitent néanmoins une représentation explicite de ces caractéristiques en tant qu'objets et non plus une représentation sous la forme d'attributs. Ces caractéristiques essentielles (objets) apparaîtront comme valeurs d'un attribut 'essential_characteristics' comme dans l'exemple ci-dessous :

```
(defun-concept 'SiegeUnePersonne
        (isa 'Siege)
        (essential-characteristic '(unePersonne)) )
```

L'utilisation de langages comme Lisp permet de (re)définir facilement la fonction de définition de concept afin de prendre en compte la gestion des caractéristiques essentielles.

2.3.3.2.3 REPRESENTATION DES CARACTERISTIQUES ESSENTIELLES AVEC PROTEGE

Protégé tient une place particulière et prépondérante dans la construction d'ontologies en Ingénierie des Connaissances. Ne pas choisir cet environnement doit être justifié, au-delà du fait que pour les experts du domaine, la maîtrise de Protégé peut s'avérer compliquée : "As the group that developed Protégé, the most widely used ontology editor, we are keenly aware of how difficult the users perceive this task to be." [Horridge et al. 2013]. En ce qui nous concerne, la raison principale est que Protégé repose sur une logique de description, qui constitue un sous ensemble de la logique du premier ordre, et qu'il n'est donc pas possible de prendre en compte directement la notion de caractéristique essentielle qui requiert une logique d'ordre supérieur[60].

Néanmoins, il est possible de représenter de différentes façons les caractéristiques essentielles en Protégé. Soit à travers une relation partitive, par exemple une chaise a comme partie un dossier, des pieds. Mais dans ce cas qu'en est-il de la caractéristique /une personne/ ? Soit en simulant l'ordre 2, c'est-à-dire en considérant les caractéristiques essentielles comme autant d'individus afin de les manipuler, c'est la solution décrite dans [Spies & Roche 2008]. L'environnement Tedi, dans l'export des ontoterminologies en OWL, adopte une solution différente qui consiste à traduire les caractéristiques d'un même axe d'analyse sous la forme de classes disjointes [Papadopoulou & Roche 2018b]. Un concept, combinaison de caractéristiques essentielles, est alors traduit sous la forme d'une classe définie comme sous-classe des classes correspondant à ses caractéristiques essentielles.

Par exemple pour définir les caractéristiques essentielles du concept :

<Siège une personne sans pied sans dossier sans bras> on aura défini au préalable les quatre classes : 'Siège', 'Siège une personne', 'Siège sans pied', 'Siège sans dossier' et 'Siège sans bras'.

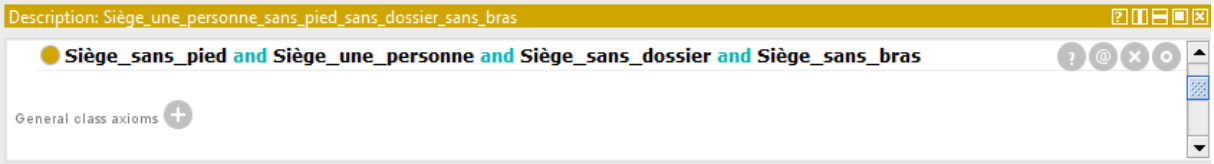


Figure 54 : Représentation des caractéristiques essentielles de <Siège_une_personne_sans_pied_sans_dossier_sans_bras>

2.2.3.3 NOMMER LES CONCEPTS

Les noms des concepts en plus d'être uniques, ont une importance particulière dans un système conceptuel. Ils doivent refléter ou pouvoir indiquer la place des concepts dans le système conceptuel,

[60] Par exemple dans une logique modale, à l'aide de prédicats « rigides », vrais dans tous les mondes possibles.

ou refléter la nature des instances des concepts[61]. Dans la mesure où les caractéristiques essentielles définissent les concepts et structurent le système conceptuel, il existe un lien direct entre le nom du concept et ses caractéristiques essentielles[62]. La manière de nommer les concepts est donc loin d'être arbitraire et peut s'inspirer de certains langages naturels comme le chinois, où un mot est formé par des caractères qui représentent les caractéristiques essentielles du mot que l'on veut introduire. La place de ces caractères (porteurs de sens) permet de situer le nouveau mot (composé de caractères existants) par rapport aux autres.

2.2.3.3.1 LE TERME CHINOIS ET SA FORMATION

Les termes chinois peuvent être relativement *transparents*[63] et peuvent dans certains cas être « identifiés » à des noms de concept En effet, le caractère « 火 » prononcé «huo» qui représente le feu et le caractère « 车 » (prononcé « che ») représentant le véhicule à roue, ont été choisis pour former le mot train, ce qui donne « 火车 » (prononcé « huoche »). D'une certaine manière, le mot « train » est formé de deux caractéristiques essentielles « feu » et « véhicule à roue », ce qui permet d'indiquer qu'il s'agit d'un véhicule à roue et de le distinguer des autres véhicules. En effet, l'ordre dans lequel sont placés ces caractères a une importance, le ou les caractères placés sur la gauche permettent de spécialiser le « concept ».

« La structure la plus courante des mots nominaux est sur le modèle : déterminant – déterminé, où le déterminé est le plus souvent un nom, tandis que le déterminant peut avoir une valeur nominale, mais aussi adjectivale ou verbale. Ainsi, véhicule à roues : *che*, train: *huoche* (littéralement «véhicule [à] feu»), automobile: *qiche* («véhicule [à] vapeur»), autobus: *gonggongqiche* (« voiture commune ») » (revue Langages 2005 n°157,118-128,Georges Métailié).

Si nous considérons qu'un terme chinois est un nom de concept bien nommé[64], c'est-à-dire que le nom du concept permet de situer la place du concept par rapport aux autres, alors d'une part, on peut rattacher les termes de différents langages (comme le français et l'anglais) à ce concept et d'autre part, on peut reconstituer un système conceptuel à partir de termes chinois. Le système

---

[61] Le nom de concept <Siège pour une personne avec pieds avec dossier et sans bras> traduit le fait que toutes ses instances sont des sièges pour une personne avec pieds, avec dossier et sans bras, indifféremment du fait qu'il puisse exister ou non un concept générique comme <Siège pour une personne>.

[62] Ainsi, l'environnement Tedi propose, lors de la création d'un concept, un nom de concept par défaut créé à partir des caractéristiques essentielles qui composent la définition. Rappelons qu'il est déconseillé de nommer les concepts avec des termes en usage au risque de confondre les différentes dimensions de la Terminologie.

[63] « A term or appellation is considered transparent when the concept it designates can be inferred, at least partially, without a definition or an explanation. In other words, the meaning of a term or appellation can be deduced from its parts. For a term to be transparent, a key characteristic – usually a delimiting characteristic –is used in the formation of the term or appellation itself. » [ISO 704 : 2009]

[64] Rappelons qu'un terme n'est pas un nom de concept, mais par « économie de la langue », nous nous autoriserons ici ce « raccourci ».

conceptuel établi sur la base de termes chinois[65] reflètera une certaine vision du monde à travers la manière dont sont formés les termes qui les désignent. Par exemple, comme vu précédemment, le terme « trainé en chinois est construit à partir de « feu » et de « véhicule à roue», on peut en déduire la vision « chinoise » du concept de train à l'époque de l'introduction de ce terme.

2.2.3.3.2 LE SYSTEME QUI REGIT LA FORMATION DES TERMES CHINOIS EST-IL ADAPTE POUR NOMMER LES CONCEPTS ?

Pour illustrer le modèle de formation des termes chinois le logiciel « Treewords » (Figure 55, se référer à l'annexe C) a été développé à cet effet dans le cadre du projet Xu Guangqi 2015.

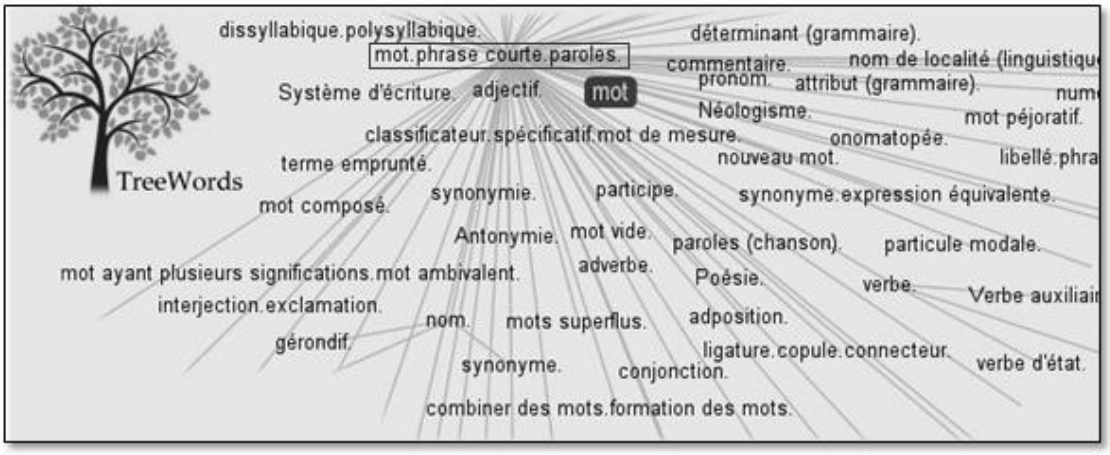


Figure 55 : Logiciel Treewords, visualisation en français du terme « 单词 » (« mot » en français) dans un « système conceptuel » chinois

Il s'agit de montrer que les règles qui régissent la formation des termes chinois permettent d'établir un système conceptuel et peuvent être adaptées pour la formation des noms des concepts. Le nom du concept est formé des noms de caractéristiques essentielles dont il est constitué. Par exemple : le concept <b a> est nommé avec les caractéristiques essentielles /b/ et /a/. L'ordre dans lequel sont placés les noms des caractéristiques essentielles permet d'indiquer la place qu'occupe le concept dans le système conceptuel. C'est ce principe que l'on retrouve dans la formation des termes chinois.

Si on cherche à visualiser le terme chinois « 宝宝推车» (« poussette» en français) par rapport aux autres termes avec le logiciel Treewords, on obtient la représentation graphique suivante (Figure 56). Les autres termes impliqués dans ce système conceptuel sont respectivement : « 手推车(brouette) », « 推车(pousser un chariot) », et « 车(véhicule à roues) ». Chaque terme chinois est composé du nom d'une caractéristique essentielle qui permet de le différencier. En effet « poussette » est composé de « 宝宝(bébé) » et de « 推车» (pousser un chariot). On peut à partir du terme « poussette » en chinois en déduire que c'est une sorte de « chariot que l'on pousse », et un « chariot que l'on pousse » est une sorte de « véhicule à roues ». Le terme brouette en chinois, pour être différencié des autres « chariots que l'on pousse » avec la caractéristique essentielle « 手(main) » (une brouette est alors une sorte de chariot que l'on pousse à la main). On dit aussi que le chinois est un langage

[65] Sur la base des connaissances désignées par les caractères chinois qui composent le terme.

transparent dans une certaine mesure [Alleton, 2001] car la simple lecture du terme permet souvent d'en déduire une définition, contrairement au français où le terme français contient dans la plupart des cas peu d'indication quant à sa place par rapport aux autres termes. « La signification d'un terme ou d'une appellation peut être déduite de ses parties » [ISO 704, 7.4.2.2].

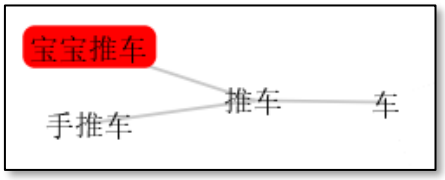


Figure 56 : La place du terme « poussette » en chinois

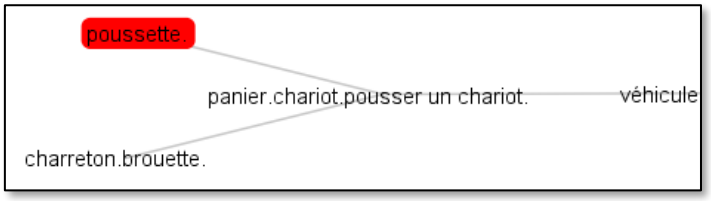


Figure 57 : « Système conceptuel » chinois et termes en français

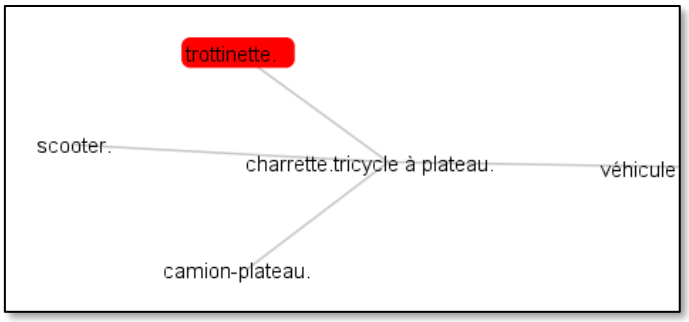


Figure 58 : La place du terme "trottinette" dans le « système conceptuel » chinois

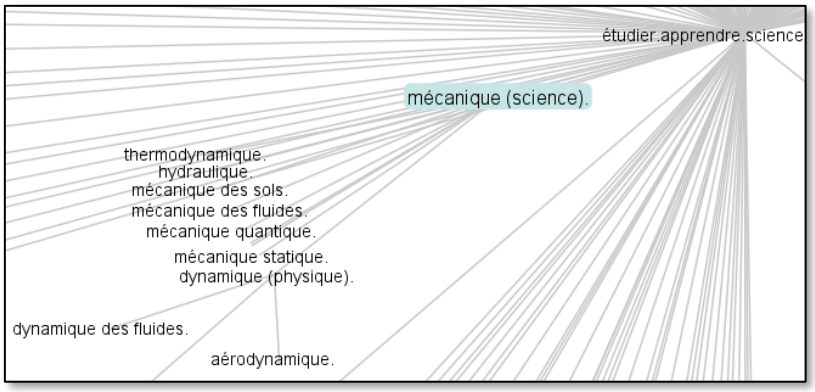


Figure 59 : la place de "mécanique" en tant que science dans le « système conceptuel » chinois

Il existe de nombreux autres exemples où il est possible de situer un terme chinois par rapport aux autres.

Le système qui régit la formation des termes chinois peut être, comme nous l'avons vu avec le logiciel Treewords transcrit dans un algorithme pour reconstituer un système conceptuel dans lequel se trouve un terme chinois. Il s'agit d'un système conceptuel subjectif qui « traduit une pensée chinoise du monde ». Par exemple, un des termes de « jalousie » en chinois : « 红眼病 », indique que c'est une sorte de maladie (la première caractéristique essentielle étant « maladie 病 », même s'il s'agit ici d'une métaphore (littéralement, « maladie des yeux rouges » en chinois).

Les règles qui régissent la formation des termes chinois semblent être adaptées à la formation des noms des concepts :

Terme chinois : « 宝宝推车 », nom du concept : </a/ + /b/ + /c/>

/a/ : « 宝宝 bébé »

/b/ : « 推 pousser »

/c/ : « 车 véhicule à roues »

Le nom du concept est alors formé des noms des caractéristiques essentielles dont il est composé, l'ordre des noms des caractéristiques doit refléter sa place dans le système conceptuel. Une « poussette » est donc d'abord « un véhicule à roues ». Sur cette base, on pourra envisager de générer des noms de concept 'normalisés' à partir d'une agrégation ordonnée de noms de caractéristiques essentielles.

2.2.3.4 NECESSITE D'UN LANGAGE FORMEL POUR MANIPULER LE SYSTEME CONCEPTUEL

La mise en ordre du réel et le choix des mots pour en parler constituent les objectifs de la Terminologie dite « classique ». Une théorie du concept sert de base à la création d'un système conceptuel permettant d'organiser les entités qui peuplent la réalité. Le système conceptuel requiert un langage pour le définir et le manipuler : « un moyen d'expression qui permette à la fois de prévenir les erreurs d'interprétation et d'empêcher les fautes de raisonnement. Les unes et les autres ont leur cause dans l'imperfection du langage. » [Frege, 2014]. C'est pourquoi nous utiliserons un langage formel tel qu'un langage de programmation pour manipuler le système conceptuel. Un langage de programmation possède une grammaire et une syntaxe formellement définie. La théorie du concept proposée (CLT) pourra être implémentée dans un langage informatique et c'est à travers celui-ci que le système conceptuel pourra être défini et manipulé, d'où l'importance d'utiliser un langage de programmation interprété et non compilé. Il sera alors possible de vérifier les spécifications de la théorie du concept et « empêcher des erreurs de raisonnement » dans la définition du système conceptuel (comme des relations génériques circulaires par exemple, ce que CmapTools[66] ne permet pas de contrôler).

Un langage interprété permet une communication « directe » avec le système en donnant la possibilité d'évaluer des instructions sans avoir été compilées (traduites dans un autre langage). C'est

[66] Outil informatique pour modéliser graphiquement un système conceptuel.

pourquoi les langages issus de l'intelligence artificielle tel que LISP[67] (par exemple SRL est écrit en LISP) sont particulièrement adaptés à cette tâche.

Enfin, la capacité d'une théorie à pouvoir donner lieu à un langage informatique est en soi une forme de « preuve constructive » de sa validité.

2.3 APPORTS POUR LA TERMINOLOGIE

2.3.1 APPORTS METHODOLOGIQUES (ISO 704)

Expliciter la dimension conceptuelle des terminologies sous la forme d'une ontologie formelle de l'Ingénierie des Connaissances sur la base d'une Théorie du Concept fondée sur la notion de caractéristique essentielle n'est pas sans conséquences sur la Terminologie elle-même et le travail terminologique. Nous n'en citerons ici que quelques-unes.

Ainsi, le tournant ontologique permet de clairement distinguer les deux dimensions de la Terminologie et les liens qu'elles peuvent entretenir. Par exemple, pour la dimension linguistique, un nom de concept se construit sur la base des caractéristiques essentielles, <Siège une personne avec pieds avec dossier et sans bras>, que dénoteront les termes dans les différentes langues, « chaise », « chair », « 椅子(yizi) ». De cette structure en découle les notions de *synonymie en terminologie*, *d'hyperonymie en terminologie*, etc. [Roche & Calberg-Challot 2010] à distinguer de la synonymie et de l'hyperonymie en linguistique (basées sur le signifié construit en discours)[68]. La définition du terme[69] en langue naturelle pourra se calquer sur la définition du concept dénoté par le terme[70]. La diversité linguistique est préservée (la bi univocité n'existe qu'au niveau conceptuel, entre le nom du concept et sa définition), l'aide à la traduction facilitée par le partage d'une ontologie commune, etc.

---

[67] LISP (« list processing ») est un langage de programmation particulièrement utilisé dans le domaine de l'intelligence artificielle, dont la structure de donnée est basée sur les expressions symboliques, les « S-expressions ». Ce langage fait l'objet d'une norme ISO intitulée ISLISP publiée en 1997 (ISO/IEC 13816:1997).

[68] On distingue la *synonymie terminologique* définie au regard du système conceptuel indépendamment de tout discours, de la *synonymie discursive* qui se construit en discours, communément appelé synonymie. Deux termes sont des *synonymes terminologiques* si et seulement si ils dénotent le même concept. Le terme t_a est un *hyperonyme terminologique* (vs *hyponyme terminologique*) du terme t_b si et seulement si le concept dénoté par le terme t_a est un concept générique (vs concept spécifique) du concept dénoté par le terme t_b. Deux termes sont des *synonymes discursifs* (synonymes) si et seulement s'ils désignent des signifiés équivalents (il reste à définir ce qu'on entend par *signifié* et par *équivalent*).

[69] Si l'on considère qu'en Terminologie « classique », la définition du terme est la définition exprimée en langue naturelle du concept dénoté par le terme.

[70] Pour exemple, l'environnement Tedi génère automatiquement des patterns de définition en langue naturelle à partir des définitions formelles du concept, à charge ensuite pour l'expert de les éditer. Par exemple, la définition suivante « Siège pour une personne avec pieds avec dossier sans bras » a été automatiquement généré pour la définition du terme « chaise ».

Si les différentes notions prises en compte dans nos travaux pour la modélisation des connaissances (objet, caractéristique essentielle, caractéristique descriptive, concept, classe, relation) conditionnent la méthode de construction de l'ontologie, elles constituent surtout une aide importante pour les experts du domaine. Le modèle conceptuel se construit sur les caractéristiques essentielles, le système vérifiant la compatibilité des caractéristiques essentielles, les caractéristiques descriptives venant ensuite compléter la structure des concepts. Les dépendances entre caractéristiques sont également un moyen d'aider les experts dans la construction de l'ontologie.

Combiner ontologie et terminologie ouvre la voie à de nouvelles approches comme la notion de définition ontologique du terme [Roche 2015] ou des méthodes conciliant termes et concepts en leur donnant une place équitable [Papadopoulou & Roche 2018]. On pourrait également s'inspirer des méthodes de gestion d'ontologies [Despres 2018].

2.3.2 ALIGNEMENT D'ONTOLOGIES POUR L'HARMONISATION DES TERMINOLOGIES (ISO 860)

« En dépit de tous les efforts qui sont faits pour coordonner les terminologies au fur et à mesure de leur évolution, il est inévitable que l'on continuera d'utiliser des terminologies qui se recouvrent et manquent d'homogénéité, car les documents et les programmes d'action s'inscrivent dans des contextes différents. ». Ainsi commence l'introduction de la norme ISO 860 :2007 Travaux terminologiques – Harmonisation des concepts et des termes[71]. L'harmonisation des concepts et des termes est un problème difficile auquel la norme de 2007 n'apporte pas de solution réellement satisfaisante : essayer de combiner deux systèmes différents en un seul ne donne pas toujours de bons résultats. Depuis 2007, date de la norme ISO 860, et l'introduction de l'ontologie formelle en Terminologie, de nombreux progrès ont été faits, en particulier dans l'alignement d'ontologies. Aujourd'hui, tout comme pour les thésaurus que nous verrons dans le chapitre suivant, une solution prometteuse réside davantage dans l'alignement que dans l' « harmonisation » au sens de l'ISO. Il restera à prendre en compte les particularités du modèle conceptuel que nous proposons dans les algorithmes d'alignement, en particulier dans gestion des caractéristiques essentielles.

2.3.3 APPLICATIONS INFORMATIQUES

Le tournant ontologique ouvre la voie à de nombreuses opérationnalisations de la terminologie telles que la gestion de l'information et de la documentation à base de thésaurus (figure 59) ou les moteurs de recherche sémantique multilingues (figure 60).

[71] Cette norme reste d'actualité, le dernier examen de cette norme datant de 2016, sans changement depuis 2007.

2.3.3.1 THESAURUS

Les thésaurus demeurent le moyen privilégié pour la gestion de l'information et de la documentation, c'est-à-dire pour leur indexation-classification et leur recherche. La norme ISO 25964-1[72] définit un thésaurus comme un « controlled and structured vocabulary in which concepts are represented by terms, organized so that relationships between concepts are made explicit, and preferred terms are accompanied by lead-in entries for synonyms or quasi-synonyms » où le concept joue un rôle important « The prime application for a thesaurus is in information retrieval, where the aim is to search for concepts », d'où un intérêt croissant pour l'ontologie de l'Ingénierie des Connaissances et la « SKOSification »[73] de thésaurus. Cependant il existe une certaine confusion entre thésaurus, terminologie, vocabulaire contrôlé et ontologie qui nuit à une bonne compréhension du domaine [Roche et al. 2014]. Par exemple quel est le statut réel d'un descripteur ? ni vraiment terme ni vraiment concept.

Le tournant ontologique apporte ici aussi une solution intéressante. Les documents sont indexés sur les noms de concepts qui jouent le rôle de descripteurs et les différents termes les dénotant comme autant de moyens d'entrées. La figure ci-dessous illustre une application de l'ontoterminologie à la gestion de thésaurus multilingues dans le cadre du projet européen AthenaPlus[74]. Elle a abouti à la réalisation de l'environnement TMP[2], pour Terminolog Management Platform.

---

[72] ISO 25964-1 « Information and documentation - Thesauri and interoperability with other vocabularies - Part 1: Thesauri for information retrieval »

[73] Opération qui consiste à traduire le thésaurus en SKOS (Simple Knowledge Organization System) : https://www.w3.org/2004/02/skos/

[74] http://www.athenaplus.eu/ « The principal objectives of the AthenaPlus project were to: I) Contribute more than 3.6 millions metadata records to Europeana; II) Improve search, retrieval and re-use of Europeana's content, bettering multilingual terminology management; III) Experiment with enriched metadata their re-use adapted for users with different needs ».

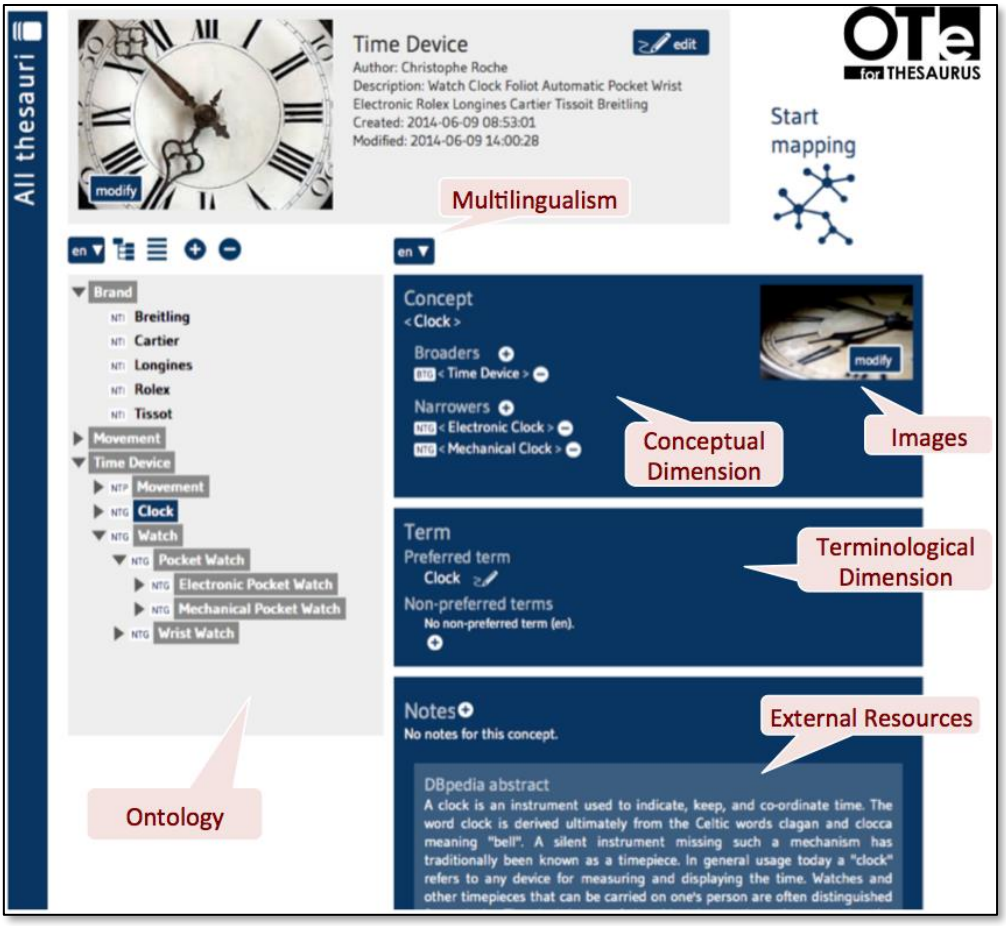


Figure 60 : L'environnement TPM2 de gestion de thésaurus multilingues.

2.3.3.2 MOTEURS DE RECHERCHE SEMANTIQUES MULTILINGUES DE DOCUMENTS

Le principe de mise en œuvre de moteurs de recherche sémantiques multilingues de documents reposant sur la base d'ontoterminologies est le suivant. Dans un premier temps, l'ontologie commune est définie ainsi que les terminologies dans chaque langue. Dans un deuxième temps, les documents, quel que soit leur langue, sont, après analyse syntaxique, classés sur les concepts qui sont référencés dans le document, c'est-à-dire pour lesquels il existe un terme les dénotant, un même document pouvant être classé sous plusieurs concepts. La recherche s'effectue soit sur un concept en navigant de manière interactive à travers l'ontologie [Tricot & Roche 2006], [Tricot et al. 2006], soit dans une langue donnée (maternelle ou autre). Le système retourne alors tous les documents correspondant aux concepts sélectionnés à l'étape précédente. A noter que la pertinence des documents est hiérarchisée en fonction des propriétés logiques des concepts définis dans l'ontologie du domaine. L'ontologie formelle utilisée comme conceptualisation du domaine permet de structurer la connaissance de manière indépendante de la langue et d'effectuer des recherches portant sur des documents écrits dans des langues naturelles différentes.

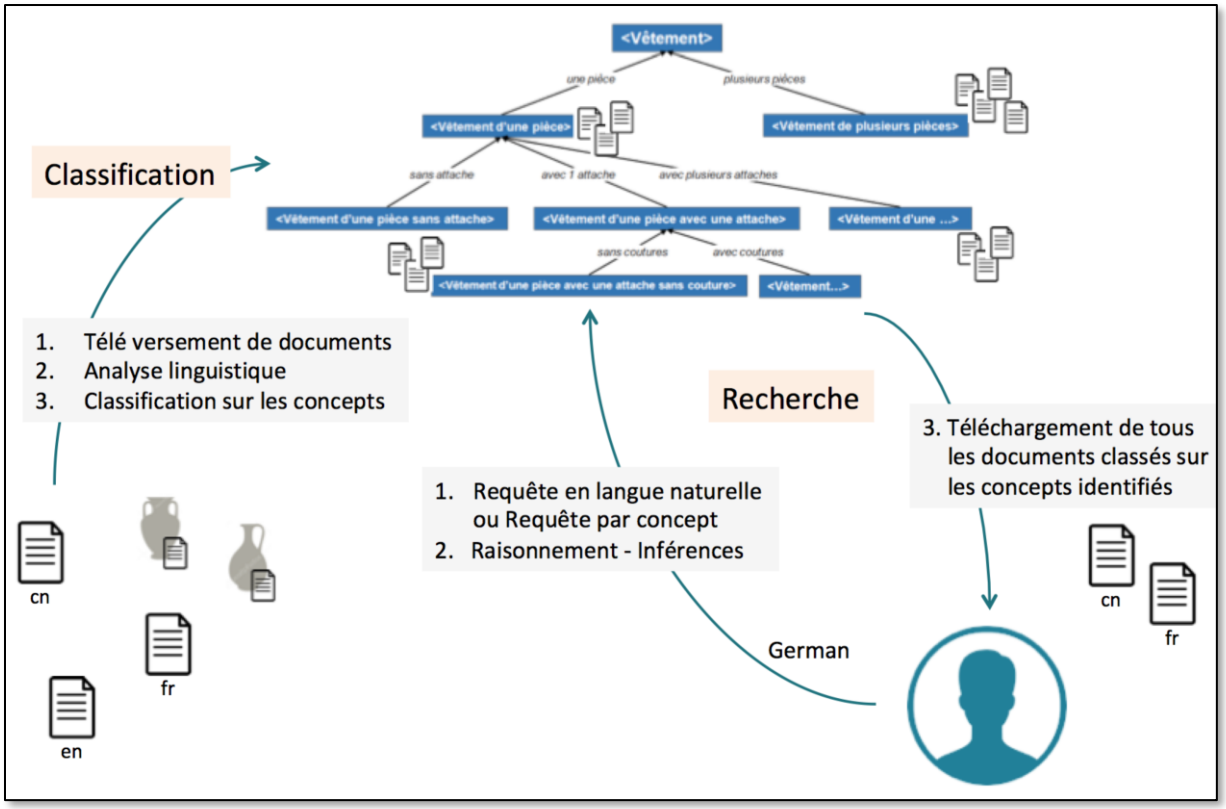


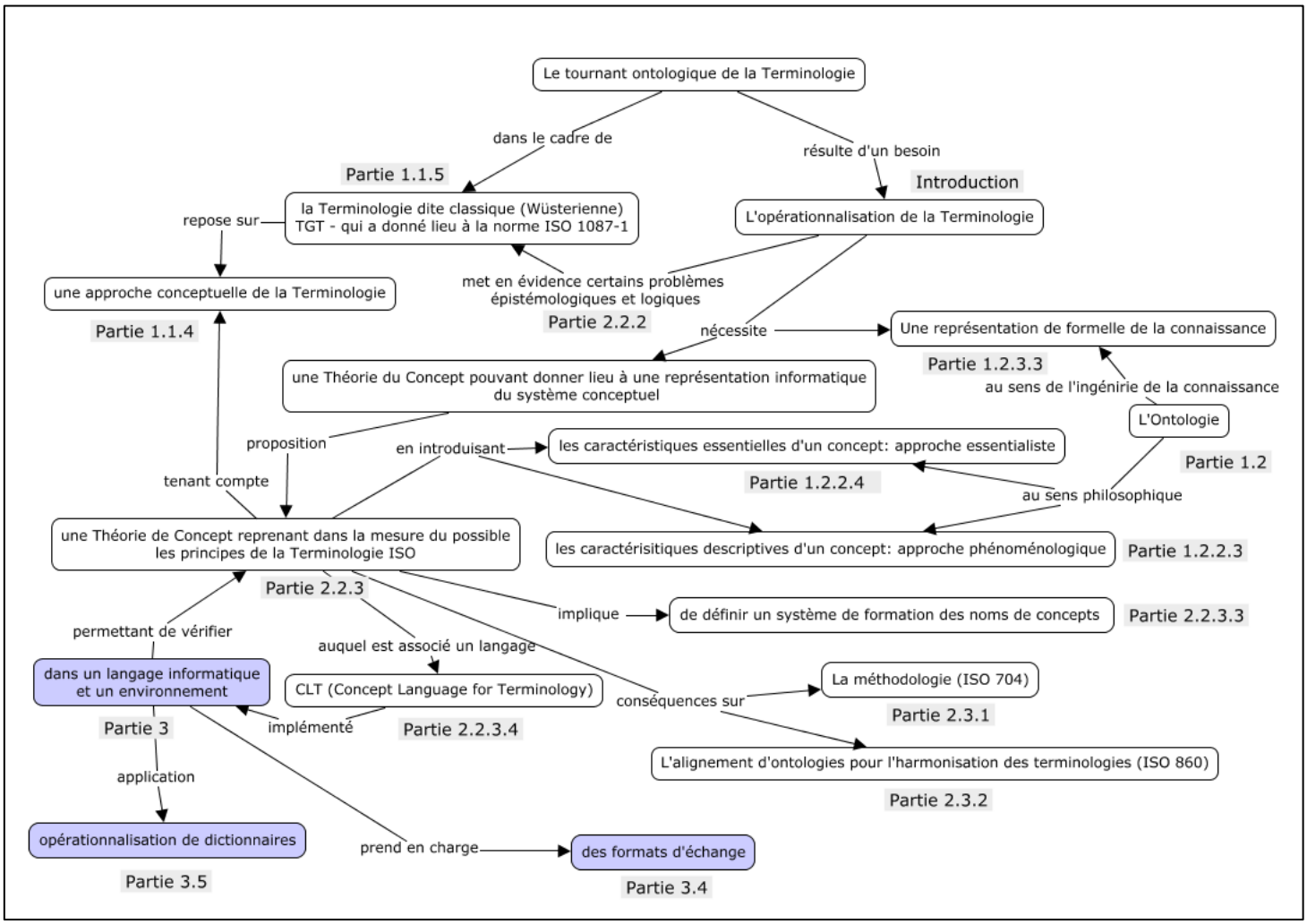
Figure 61 : Indexation et recherche sémantique multilingue de documents

PARTIE 3 : MISE EN ŒUVRE

Cette troisième et dernière partie est consacrée à la mise en œuvre des travaux théoriques**.** Nous proposons un langage informatique dédié à la CLT (Concept Language for Terminology) apte à construire un système conceptuel (définition des concepts, de leurs relations…) ce qui constitue en soi une forme de validation de la théorie. La CLT une fois implémentée, pourra être vue comme la spécialisation d'un langage de programmation pour la Terminologie. Le langage CLT sert à la fois de langage dédié à la création et la manipulation de terminologies computationnelles et de format d'échange puisque compréhensible par un ordinateur. Le choix d'une architecture Client / Serveur pour l'implémentation de la CLT permet non seulement son utilisation à travers un navigateur web, mais aussi d'envisager un aspect collaboratif dans la construction du système conceptuel. Une interface graphique est proposée afin d'établir une terminologie computationnelle sans notion de programmation. Le rôle de cette interface est de transcrire l'action de l'utilisateur en instructions CLT. Le système conceptuel réalisé à l'aide de la CLT peut être visualisé directement dans l'environnement et être exporté dans différents formats.

La mise en œuvre de la CLT est illustrée à travers un exemple d'application qui montre l'opérationnalisation d'un dictionnaire multilingue pour un domaine donné. Le système conceptuel est construit en se servant du principe de formation des termes chinois où les termes traduisent ce qu'ils désignent. Cette opérationnalisation consiste à convertir les entrées du dictionnaire en instructions du langage CLT. Ce type d'opérationnalisation basé sur le modèle de la CLT pourrait ouvrir la voie à la conversion de dictionnaires spécialisés dont les entrées sont des termes chinois vers des terminologies computationnelles.

Rappel du plan



3.1 CAHIER DES CHARGES

3.1.1 LA NECESSITE D'UN LANGAGE DE PROGRAMMATION INTERPRETE

Afin d'interpréter au niveau informatique les instructions propres à la CLT nous devons disposer d'un programme qui traduise les instructions de la CLT en langage machine et les exécute. C'est le rôle des langages de programmation, il en existe deux sortes : les langages de programmation compilés et interprétés. Les premiers nécessitent un compilateur[75] tels que le langage C[76]. Un compilateur est un programme dont le rôle est de transcrire les instructions compréhensibles par un humain en un langage machine pour être exécutées, ce qui constitue deux étapes distinctes : la compilation et l'exécution. Les langages dits interprétés tels que LISP[77] ne nécessitent pas de compilateur, les instructions sont interprétées et sont exécutées directement dans l'environnement. Il en résulte un dialogue entre l'utilisateur et le système. Quoi qu'il en soit, un langage de programmation comporte une analyse lexicale, syntaxique et sémantique des instructions. Au lieu de recréer un langage de programmation propre, nous allons en utiliser un et le compléter avec les instructions spécifiques de la CLT. Ainsi nous « transformerons » un langage de programmation général en un langage spécifique dédié à la CLT. On bénéficiera alors de l'analyse, syntaxique et sémantique du langage de programmation. On peut voir la CLT comme une spécialisation d'un langage général de programmation.

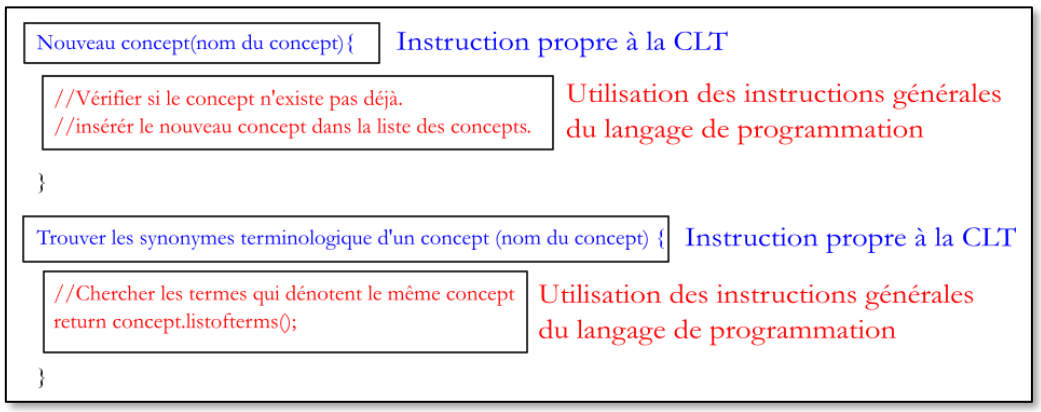


Figure 62 : Spécialisation d'un langage de programmation

---

[75] Le compilateur traduit le code source (compréhensible par l'humain) vers un langage compréhensible par une machine. Il met en œuvre une analyse lexicale, syntaxique et sémantique.

[76] Langage de programmation généraliste inventé en 1972 dans les laboratoires Bell. Chaque instruction du langage est conçue pour être compilée en un nombre d'instructions machine.

[77] LISP (« list processing ») est un langage de programmation particulièrement utilisé dans le domaine de l'intelligence artificielle, dont la structure de donnée est basée sur les expressions symboliques, les « S-expressions ». Ce langage fait l'objet d'une norme ISO intitulée ISLISP publiée en 1997 (ISO/IEC 13816:1997).

Les instructions propres à la CLT sont donc introduites comme des « fonctions » supplémentaires du langage de programmation. Ce qui constituera l'implémentation du modèle de la CLT. Les fonctions telles que la création d'un concept doivent pouvoir être ensuite appelées par l'utilisateur pour construire le système conceptuel de la terminologie. Si nous choisissons un langage de programmation compilé, il faudrait à chaque modification du système conceptuel par l'utilisateur, modifier le programme puis le recompiler pour en obtenir le résultat, ce qui est particulièrement inadapté ici. C'est pourquoi les langages de programmation interprétés disposant d'une fonction EVAL[78] seront préférés. La fonction EVAL est utilisée pour évaluer une expression pendant l'exécution du programme. Cette fonction permet d'établir un dialogue entre l'utilisateur pour construire le système conceptuel pas à pas. Par exemple lorsque l'utilisateur voudra créer un nouveau concept, il suffira d'utiliser cette fonction sur une instruction CLT de création de concept : « EVAL(newConcept(« Siège »[79])) » qui aura pour effet d'ajouter le concept « Siège » dans le système conceptuel. Pour des raisons de génie logiciel, la CLT sera implémentée d'une manière orientée objet. Les instructions du langage CLT correspondront aux méthodes[80] du modèle implémenté. Par exemple, la classe[81] « KnowledgeDatabase » contient la méthode « newConcept() ». Pour créer un nouveau concept, il faudra appeler cette méthode sur une instance de cette classe. « …knowledgeDatabase->newConcept() » . Ce type d'instruction constituera un élément du langage CLT qui utilisera la syntaxe et la grammaire d'un langage de programmation objet (appel d'une méthode sur un objet). L'instruction pourra être interprétée par le langage de programmation et faire l'objet de vérifications spécifiques à la CLT (comme vérifier si un concept n'existe pas déjà).

3.1.2 UNE ARCHITECTURE WEB CLIENT/SERVEUR

A l'ère des applications web, où l'utilisateur *via* un navigateur internet accède à une application située non pas localement mais à distance sur un serveur, l'implémentation se fera selon une architecture distribuée de type client/serveur. A l'opposition de l'informatique dite centralisée, l'ensemble des ressources ne se trouvera pas au même endroit. Comme on l'a vu précédemment, la construction du système conceptuel résulte d'un dialogue entre l'utilisateur et le système. Ce dialogue constitué d'un ensemble de requêtes envoyées par l'utilisateur et de réponses données par le système s'accorde parfaitement à une architecture client/serveur. Une instruction CLT est envoyée par un utilisateur *via* une requête http depuis un terminal (le navigateur web en l'occurrence), elle est reçue par le serveur sur lequel elle va être interprétée, puis le résultat sera renvoyé au « client » sous la forme d'une page html. Le système conceptuel sera donc situé côté serveur et l'utilisateur le manipulera côté client avec des requêtes contenant des instructions du langage CLT. Ce type d'architecture permettra d'envisager un aspect collaboratif dans la création des terminologies et de

---

[78] Est une fonction présente dans certains langages de programmation interprétés permettant d'évaluer une instruction dynamiquement. Par exemple l'évaluation de (1>0) retournera « vrai » à l'utilisateur, sans passer par une étape de compilation.

[79] les guillemets ne désignent plus ici des termes, permet de référencer des identifiants du système informatique.

[80] Les méthodes (fonctions) définies dans les classes d'un modèle orienté objet.

[81] Classe au sens des langages de programmation objet.

garder une trace des requêtes effectuées. Enfin, CLT devra pouvoir être hébergé sur un serveur web standard.

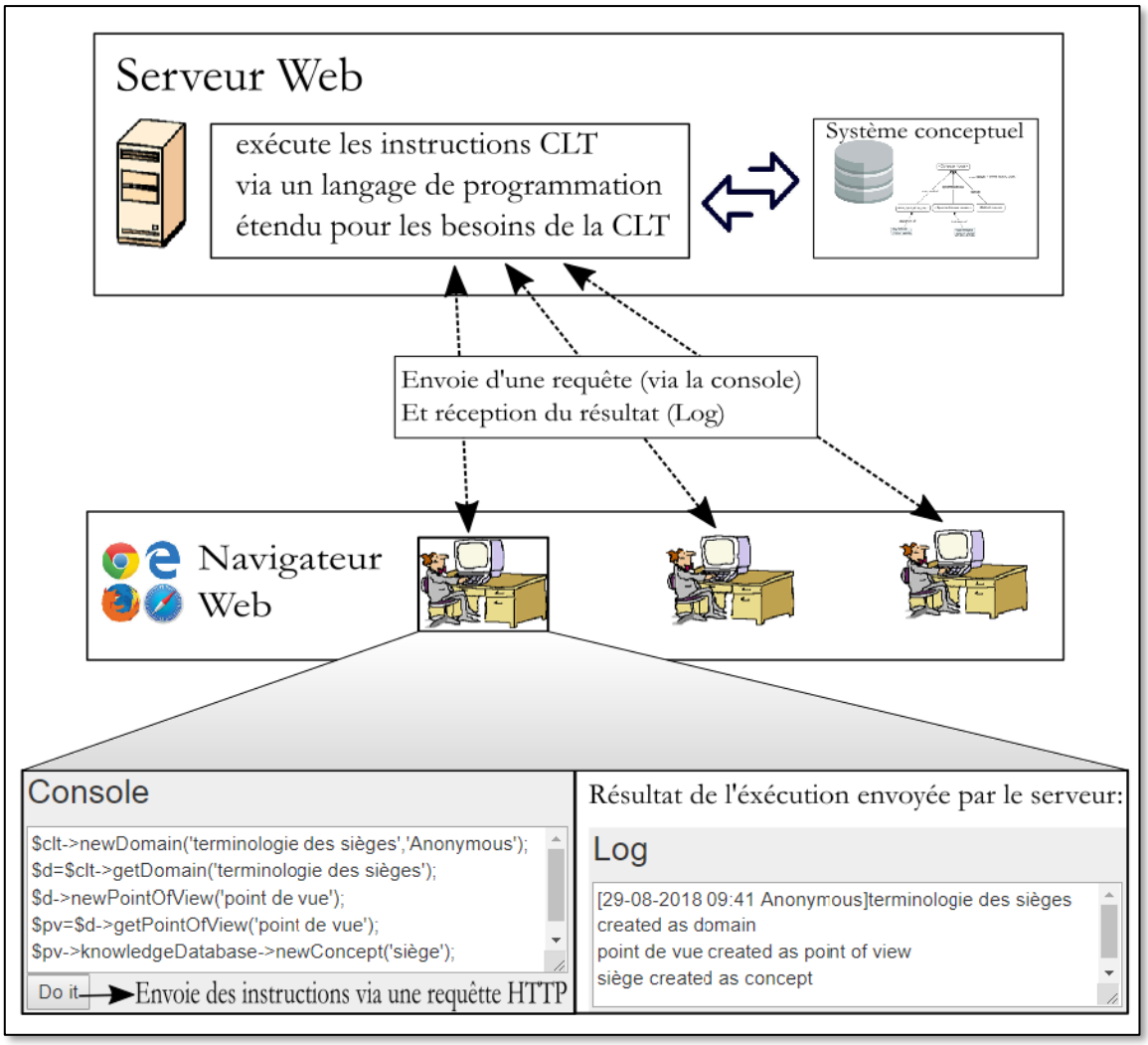


Figure 63 : architecture

3.1.3 UN MODELE MVC

MVC acronyme de « Modèle Vue Contrôleur » est un modèle de conception du génie logiciel imaginé par Trygve Reenskaug en 1978 et mis en œuvre pour la première fois dans le langage de programmation Smalltalk en 1980. Il est maintenant utilisé dans de très nombreux langages afin de séparer le modèle (la logique applicative), la vue (l'interface graphique) et le contrôleur (l'interaction entre les entrées et sorties). Cela permet de modifier le modèle sans risquer d'impacter les autres parties et réciproquement avec « la vue » et le « contrôleur ». Ici, sera appliqué ce même principe.

Le modèle sera dans un fichier séparé « CLT_kernel ». La vue et le contrôleur seront dans le même fichier html pour des raisons pratiques, mais resteront distinctes. La partie contrôleur sera représentée par des fonctions Javascript et les balises html afficheront l'interface.

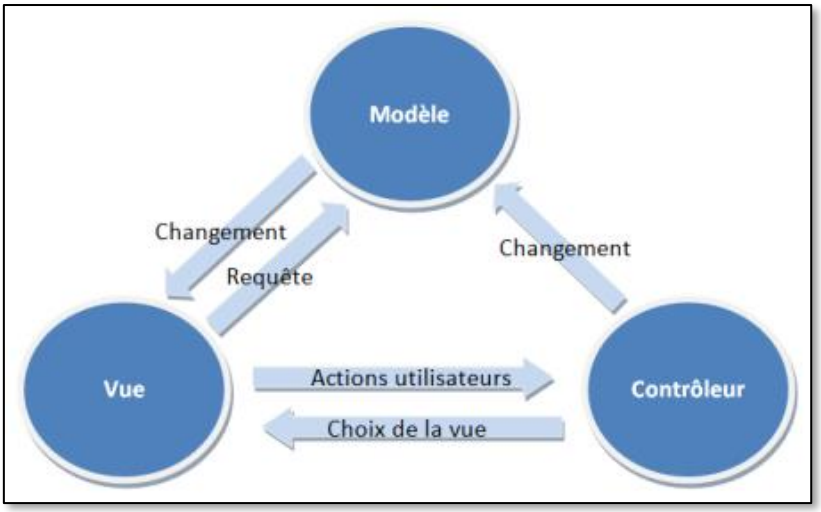


Figure 64 : Modèle MVC

3.1.4 UNE INTERFACE GRAPHIQUE FLEXIBLE

Les instructions du langage CLT en plus de pouvoir être écrites à l'aide d'une console, pourront être générées à l'aide d'une interface graphique. Il ne sera alors pas nécessaire pour l'utilisateur de connaître la syntaxe exacte du langage. L'interface graphique sera composée des éléments qui permettront la génération des instructions fondamentales du langage, telles que la création de concepts, d'objets, relations, types de caractéristiques…

L'interaction avec un élément graphique convertira l'action de l'utilisateur en instructions CLT dans la console. Il sera alors possible de voir le résultat de l'action de l'utilisateur transcrite en CLT directement dans la console. Une interface graphique sera écrite en html[82], afin de pouvoir facilement la modifier. Les actions de l'utilisateur retranscrites dans la console seront faites en utilisant des fonctions Javascript. Une fonction Javascript[83] contiendra l'équivalent de l'instruction CLT à retranscrire et sera appelée à chaque action de l'utilisateur sur l'interface graphique. Les fonctions Javascript serviront donc de lien entre l'action de l'utilisateur et la génération de code CLT dans la console.

[82] Hyper Text Markup Language, un langage à « balises » ou « tags » conçu pour représenter les pages web.

[83] Instructions du langage de programmation Javascript exécutées dans un navigateur web.

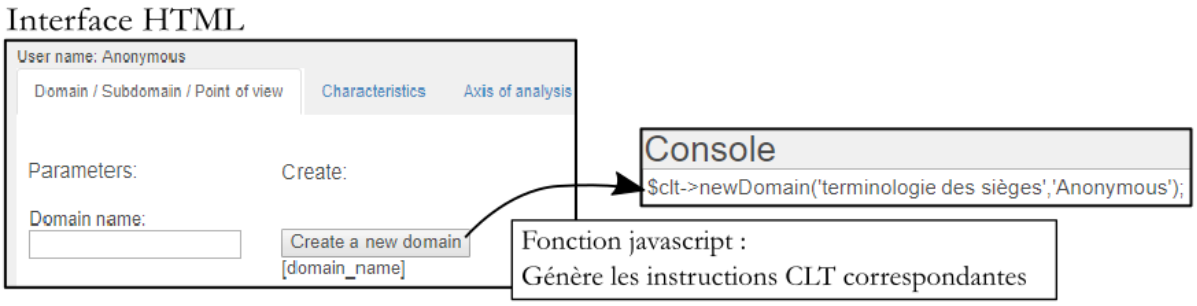


Figure 65 : Interface graphique HTML

Les utilisateurs ne feront pas tous la même utilisation du langage. Les interfaces pourront avoir des degrés de complexité différents. L'interface graphique écrite en HTML pourra donc facilement être adaptée en fonction des besoins de l'utilisateur et être visualisable sur support mobile dans un navigateur web.

3.1.5 FORMATS D'ECHANGE

Le système devra être capable d'importer et d'exporter des terminologies computationnelles. Le langage CLT servira aussi de format d'échange, car le système est en capacité d'interpréter des instructions CLT. Il faudra pour tout format de données à importer, un programme pour retranscrire les données à importer en instructions CLT. Il devra être possible aussi d'exporter une terminologie CLT dans différents formats tels que CmapTools(CSV), RDF, RDFS et OWL.

3.1.6 REPRESENTATION GRAPHIQUE

L'environnement disposera d'une fonction de représentation graphique donnant la possibilité d'afficher le système conceptuel (Figure 66) ou bien le réseau de termes correspondant. L'affichage se fera dans une page html.

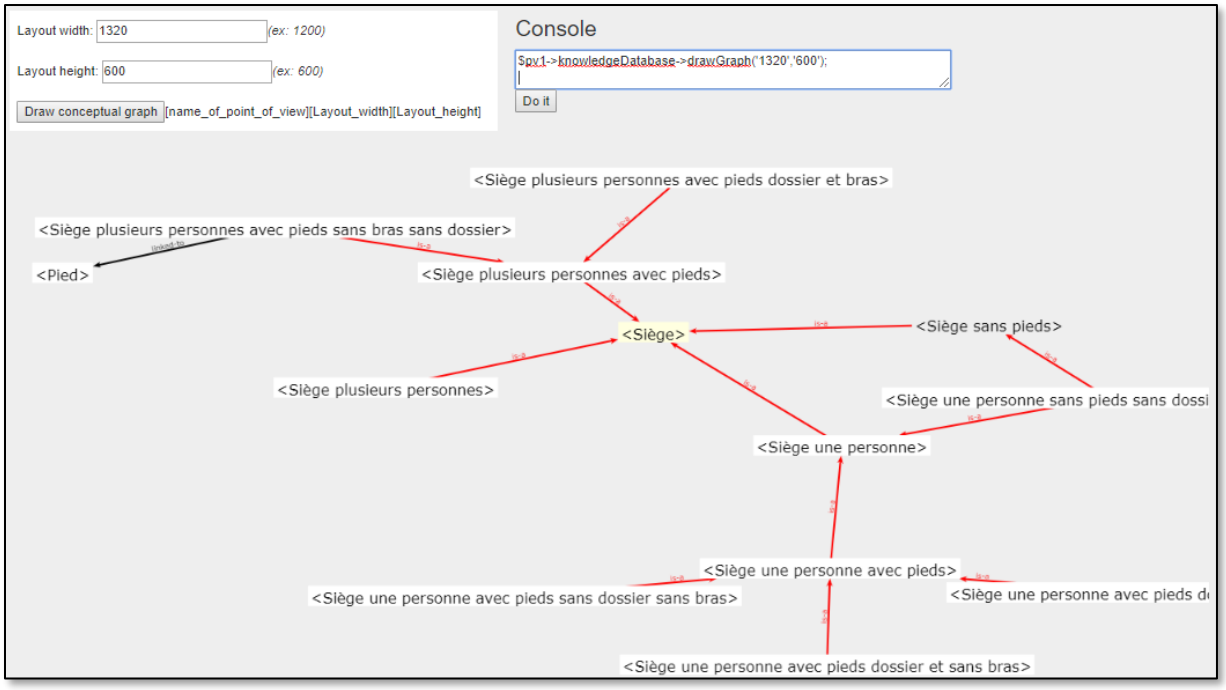


Figure 66 : Affichage du système conceptuel de la terminologie des sièges

3.2 SPECIFICATION UML

Il s'agit de représenter ici la théorie du concept proposée sous la forme d'un digramme UML[84], pour servir de spécification du modèle avant l'implémentation. Le langage de représentation graphique UML est fréquemment utilisé dans la conception d'un modèle informatique orienté objet. Le modèle du concept sera donc pensé d'une manière objet afin de faciliter l'implémentation dans un langage de programmation orienté objet. Un modèle UML permet d'avoir une idée du système sans connaissance en informatique et donne une vue d'ensemble du système. Il existe différentes manières de représenter un système informatique en UML, une vue statique : le diagramme de classe (le plus couramment utilisé), une vue dynamique (un exemple d'instanciation du modèle) et les cas d'utilisation (Use cases) pour illustrer les scénarios d'utilisation du système. On détaillera ici le système de manière statique (avec le diagramme de classe) et un exemple d'instanciation du modèle (une représentation dynamique du système). On retrouvera dans la spécification UML du système les principales notions explicitées dans la théorie du concept revisitée.

[84] Acronyme de « Unified Modeling Language » est un langage graphique normalisé pour représenter la conception d'un système orienté objet.

3.2.1 REPRESENTATION STATIQUE DU SYSTEME : LE DIAGRAMME DE CLASSE UML

La modélisation UML est réalisée ici à l'aide de l'outil ArgoUML[85], logiciel libre distribué sous licence EPL1.0. Il permet entre autre la création de diagramme de classes et de cas d'utilisation.

UML dispose de nombreuses notions pour spécifier un système informatique, parmi les plus courantes on retrouvera la classe, la classe abstraite, l'interface, la relation générique, d'association, d'agrégation et de composition.

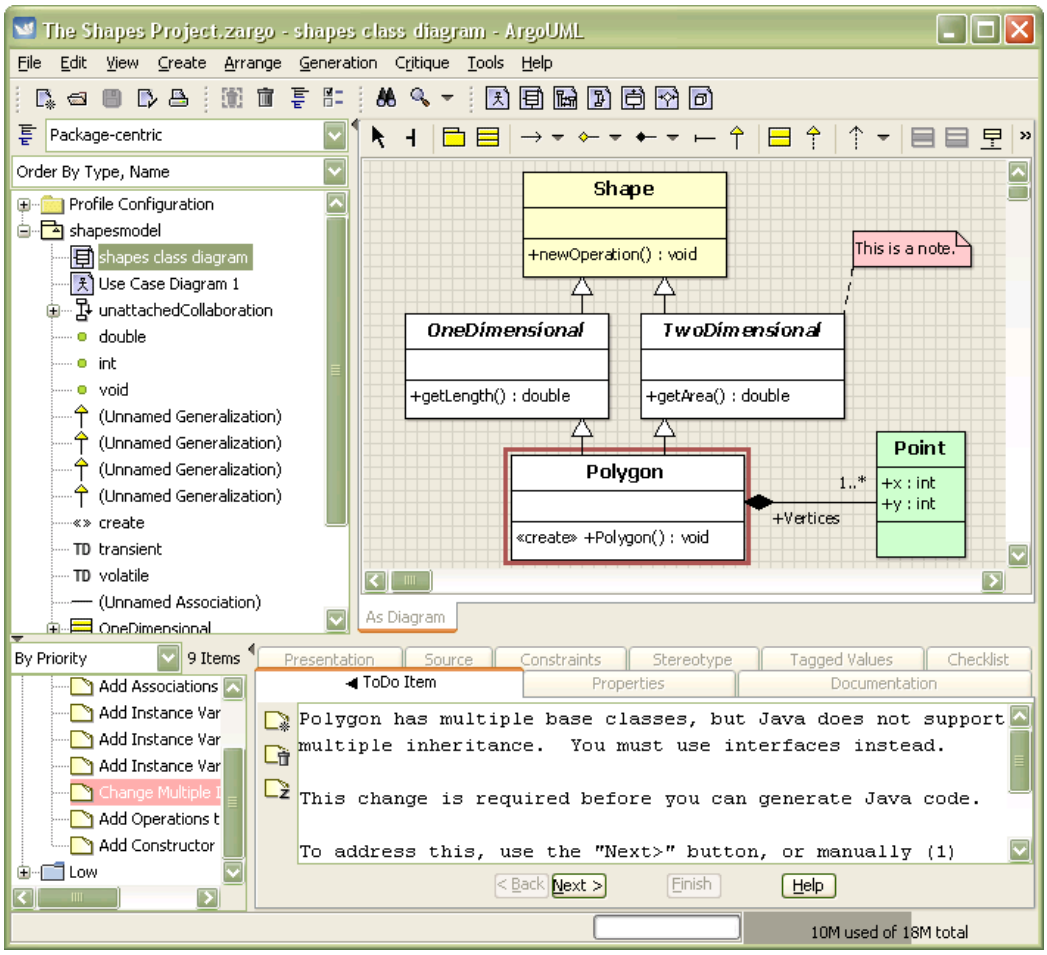


Figure 67 : Le logiciel ArgoUML

[85] http://argouml.tigris.org/ (consulté le 30/08/2018)

3.2.1.1 LA CLASSE UML

La classe est l'élément principal d'un diagramme de classe et son nom commence par une majuscule. La classe intervient dans une représentation statique du système. La classe se compose de deux parties : les variables d'instance et les méthodes qui traduisent son comportement. Dans la conception du modèle, il s'agit d'identifier les classes mis en jeu et ensuite de déterminer leurs relations.

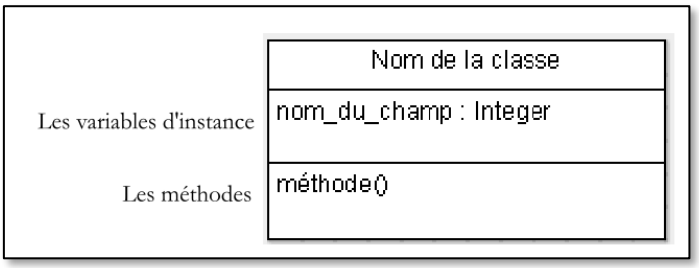


Figure 68 : Une classe UML représentée avec le logiciel ArgoUML

3.1.1.2 LES TYPES DE RELATIONS

Figure 69 : Les relations UML telles qu'elles sont représentées dans le logiciel ArgoUML

3.1.2.1 LA RELATION GENERIQUE OU D'HERITAGE

Elle permet de décrire une relation entre une classe générale (classe « parent ») et une classe spécialisée (sous-classe) (Figure 70). La sous-classe comporte des informations supplémentaires comme des attributs ou des méthodes, on parle de notion d'héritage. Même si ArgoUML autorise l'héritage multiple (une sous-classe héritant de plusieurs super classes) nous ne l'utiliserons pas. De nombreux langages de programmation orientés objets n'autorisent pas directement l'héritage multiple en raison de certains problèmes que cela peut impliquer[86].

[86] Problème du diamant ou en losange : un conflit peut exister lorsque certaines classes impliquées portent des méthodes ou des champs de même nom.

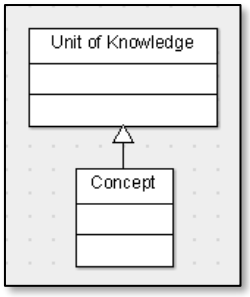


Figure 70 : Exemple de relation générique avec ArgoUML

3.1.2.2 LA RELATION D'ASSOCIATION

Une relation d'association implique deux classes du modèle et les met en relation. L'association est mise en œuvre par une variable de classe représentant chacune des classes. Par exemple une classe « voiture » possèdera une variable d'instance « conducteur » et la classe « Conducteur » une variable d'instance faisant référence à la classe « Voiture » (Figure 71). Il n'est pas nécessaire de représenter ces variables d'instance (conducteur et voiture de part et d'autre) dans le schéma UML car la relation d'association implique leur présence.

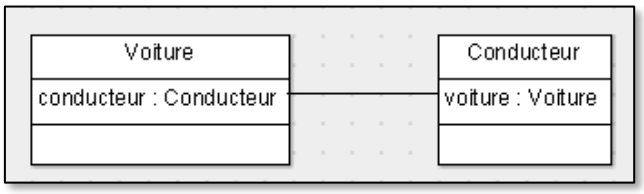


Figure 71 : Représentation de la relation d'association avec ArgoUML

3.1.2.3 LA RELATION D'AGREGATION

L'agrégation est une forme particulière de l'association et indique qu'une classe est partie d'une autre. La mise en œuvre de l'agrégation se traduit par la présence d'une instance de variable que dans une classe. Par exemple si la classe « Page » est partie de la classe « Livre », la classe « Livre » possèdera une ou plusieurs variables d'instance de type « Page ». Cependant la classe « Page » ne contiendra aucune référence à la classe « Livre », c'est ce qui la différencie de l'association (Figure 72).

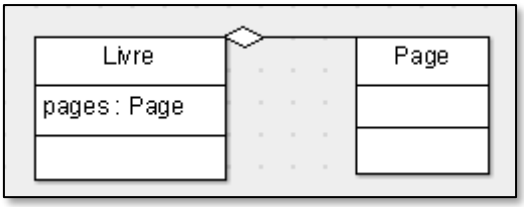


Figure 72 : Exemple de relation d'agrégation avec ArgoUML

3.1.2.4 LA RELATION DE COMPOSITION

La relation de composition est une forme plus forte de l'agrégation. Elle implique une dépendance d'existence entre la partie et le contenant, si le contenant est supprimé, la partie l'est aussi. Cela amène à se demander si la classe servant de composant peut exister sans son contenant et si cela a un sens. Par exemple la classe « Université » peut être liée à la classe « Etudiant » par une relation de composition (Figure 73). Une classe « Etudiant » n'a plus vraiment de sens sans la classe « Université ».

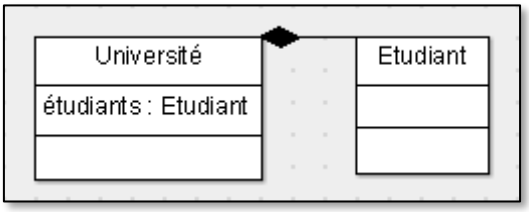


Figure 73 : Exemple de la relation de composition avec ArgoUML

3.2.2 LA REPRESENTATION UML DE LA CLT

Avec la notation UML, nous pouvons représenter le modèle de la CLT (Figure 74) sous sa forme statique à l'aide d'un diagramme d'une classe. Toutes les méthodes des classes ne sont pas représentées dans ce schéma, mais cela donne un aperçu des classes qui vont être impliquées ainsi que leurs relations. Ici le modèle est représenté en utilisant les classes, les relations génériques, d'association et d'agrégation.

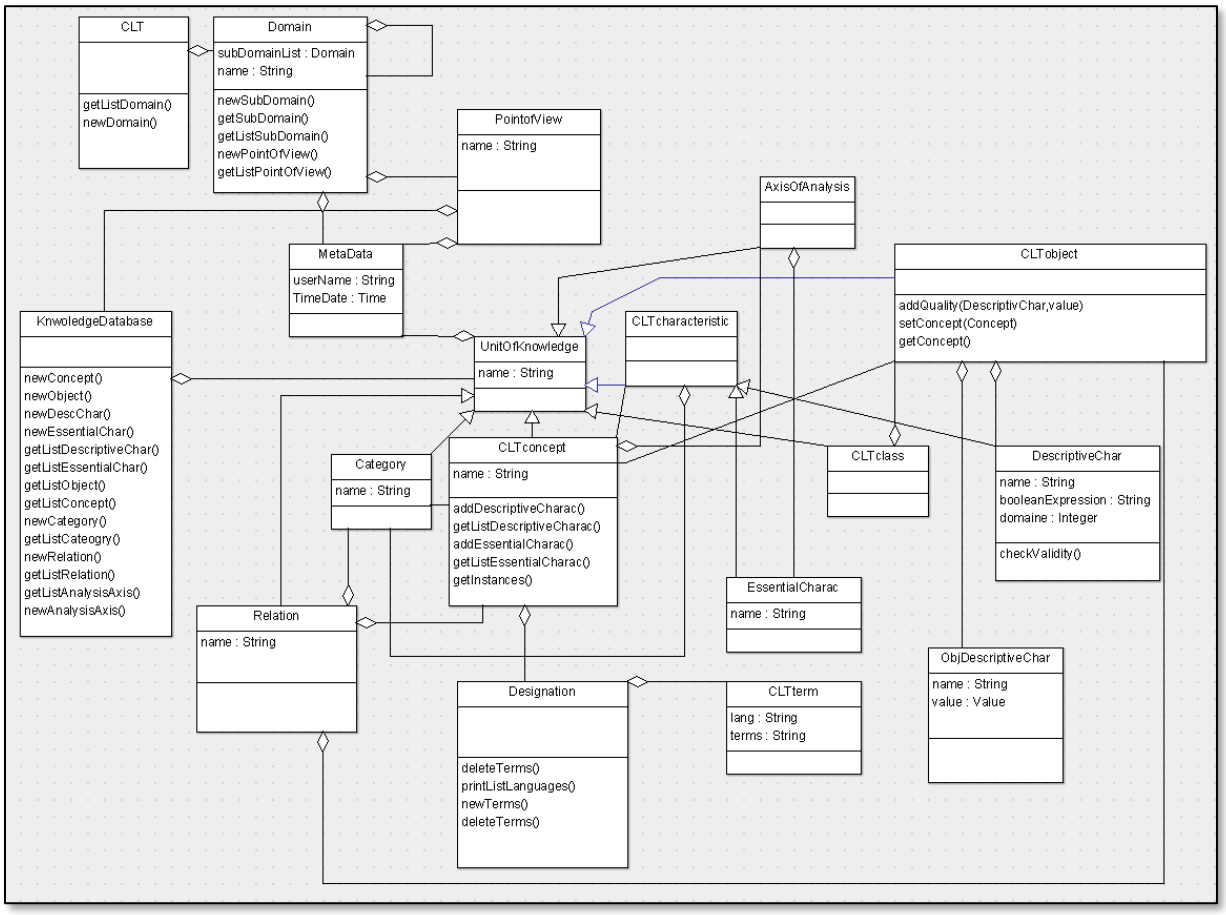


Figure 74 : Modèle UML de la théorie du concept proposée

3.2.2.1 LA CLASSE « CLT »

La classe CLT (Figure 75) est la classe « racine » servant de point de départ pour la création du système conceptuel. Elle contient la classe Domaine. C'est à partir d'une instance de la classe CLT que les domaines seront manipulés pour permettre par exemple la création d'un nouveau domaine ou de le supprimer.

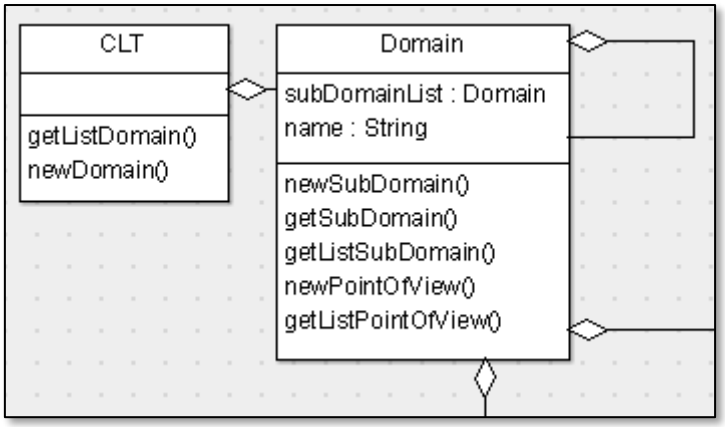


Figure 75 : La classe CLT

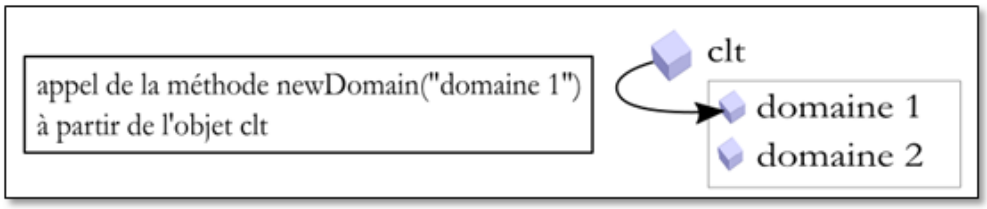


Figure 76 : Exemple d'instanciation de la classe CLT

Les méthodes accessibles à partir de la classe CLT sont les suivantes :

- newDomain(nom du domaine) : méthode de création d'un domaine
- deleteDomain(nom du domaine) : permet de supprimer un domaine
- getDomain(nom du domaine) : retourne un objet « domaine »
- printDomainList() : affiche la liste de tous les domaines

Par exemple, l'exécution de l'instruction « $clt->newDomain('domaine 1'); » créera une nouvelle instance de la classe « Domain » dans l'instance « clt », cette nouvelle instance de domaine aura comme nom « domaine 1 ».

3.2.2.2 LA CLASSE « DOMAIN »

La classe « Domain » (Figure 77) permet d'accéder aux sous-domaines d'un domaine ainsi qu'aux points de vue qui y sont rattachés et ainsi donner la possibilité de créer ou la supprimer des sous-domaines et des points de vue. Les domaines, sous-domaines tout comme les points de vue possèdent des « méta données », qui contiendront les informations comme le nom de l'utilisateur, l'heure et la date de création ou de modification. Ces informations sont mises à jour lors de modifications liées aux domaines et aux points de vue. Par exemple, lors de la création d'un point de vue, les informations concernant le nom de l'utilisateur et la date seront contenues dans le point de vue.

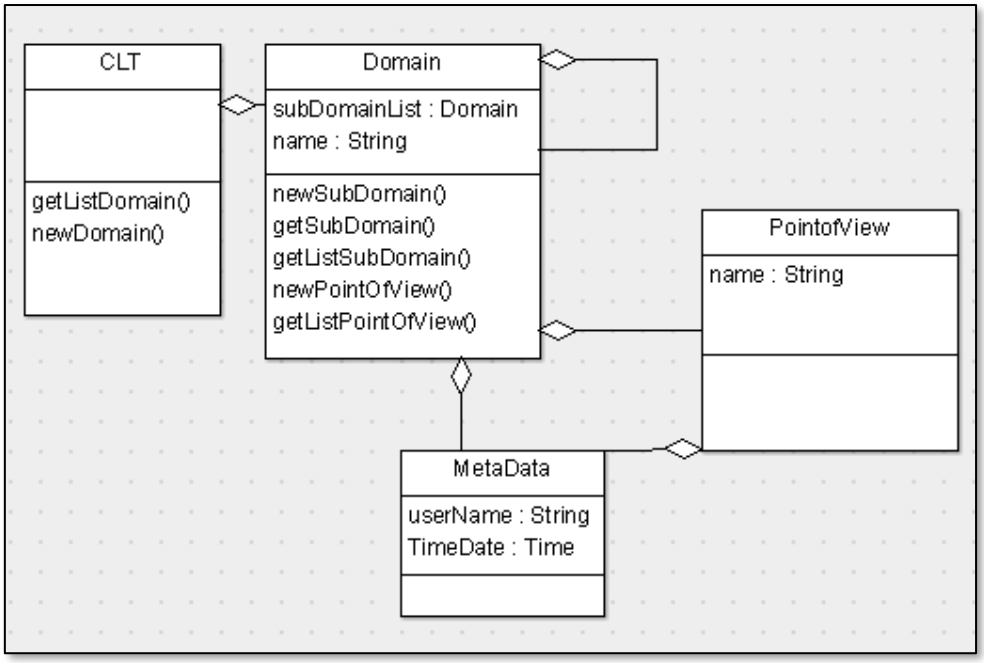


Figure 77 : Classe "Domain" du modèle CLT

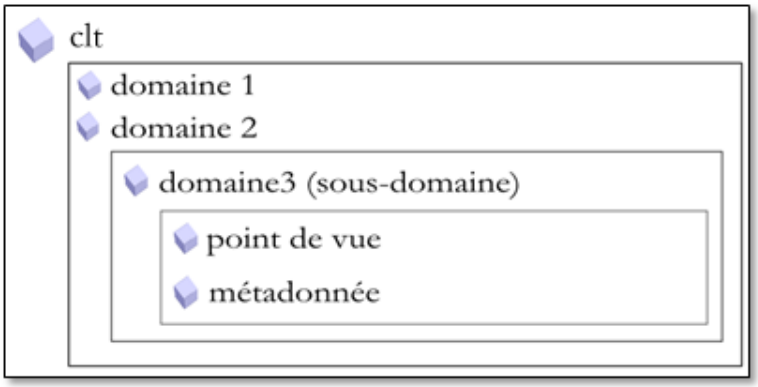


Figure 78 : Exemple d'instanciation de la classe "Domain"

Les méthodes accessibles à partir de la classe « Domain » sont les suivantes :

- printName() : affiche de le nom du domaine à partir de son instance
- printSubDomainList() : affiche tous les sous-domaines directs d'un domaine
- printOfViewList() : affiche tous les points de vue liés du domaine
- newSubDomain(nom du domaine) : permet de créer un sous-domaine (un sous-domaine a la même structure qu'un domaine)
- deleteSubDomain(nom du domaine) : supprime un sous-domaine donné
- newPointOfView(nom du point de vue) : permet de créer un point de vue dans un domaine ou sous-domaine.
- deletePointOfView(nom du point de vue) : supprime un point de vue donné par nom
- getSubDomain(nom du sous-domaine) : retourne une instance de la classe « Domain »
- getPointOfView(nom du point de vue) : retourne une instance « Point of View »

Par exemple, l'instruction « $d->newPointOfView('pv1'); » a pour effet de créer un nouveau point de vue nommé « pv1 » grâce à la méthode « newPointOfview » appelée à partir d'une instance « domain » ici représentée par la variable $d.

3.2.2.3 LA CLASSE « POINT OF VIEW »

Un point de vue peut être créé à partir d'un domaine ou d'un sous-domaine. La classe « Point of view » (Figure 79) est composée de la classe « Knowledge database » qui servira à contenir et à manipuler les unités de connaissances du système conceptuel. Chaque terminologie pourra être accédée à partir d'un point de vue spécifique. Chaque point de vue différent impliquera une terminologie différente pour un domaine donné.

La classe « Point of View » ne contient que la méthode « printName() », le but de cette classe est de contenir et pouvoir accéder à la « base de connaissance » représentée par une instance « knowledge database ».

Par exemple l'exécution de l'instruction «$pv->printName() » qui contient l'appel de la méthode « printName() » sur une instance de point de vue « $pv » affichera le nom du point de vue.

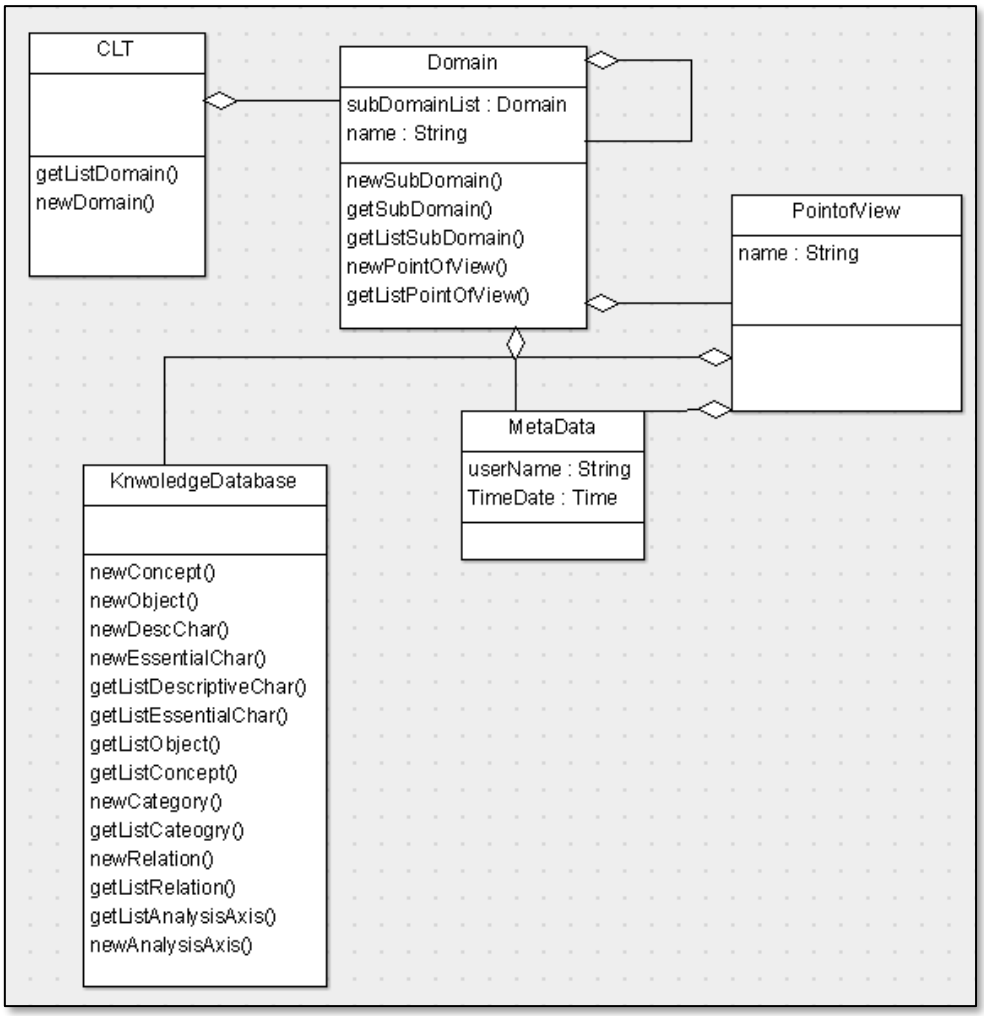


Figure 79 : La classe "Point de vue" du modèle CLT

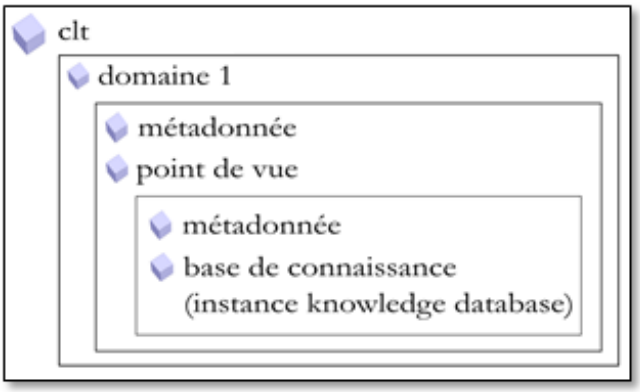


Figure 80 : Exemple d'instanciation d'un point de vue

3.2.2.4 LA CLASSE « KNOWLEDGE DATABASE »

La classe “Knowledge database” (Figure 81) est la classe clef du modèle et contient toutes les méthodes de manipulation des unités de connaissance. C’est à travers celle-ci que la terminologie est construite et manipulée. Une terminologie pour un point de vue donné sera donc représentée par une instance de la classe « KnowledgeDatabase », conteneur direct des unités de connaissance de la terminologie. Les appels aux méthodes pour la création des différentes unités de connaissance se feront à partir de celle-ci, comme la création d’un nouveau concept par exemple. C’est une classe qui sera utilisée lorsque nous voudrons effectuer des opérations sur la terminologie pour effectuer des requêtes spécifiques d’information sur un concept ou générer sa définition d’un point de vue logique.

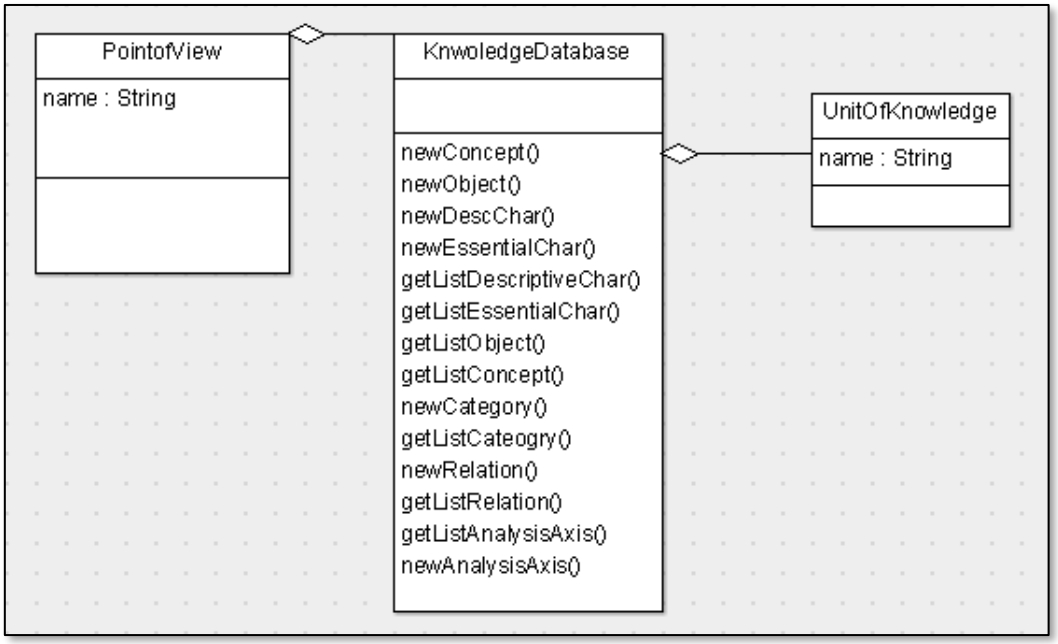


Figure 81 : La classe "Knowledge database" du modèle CLT

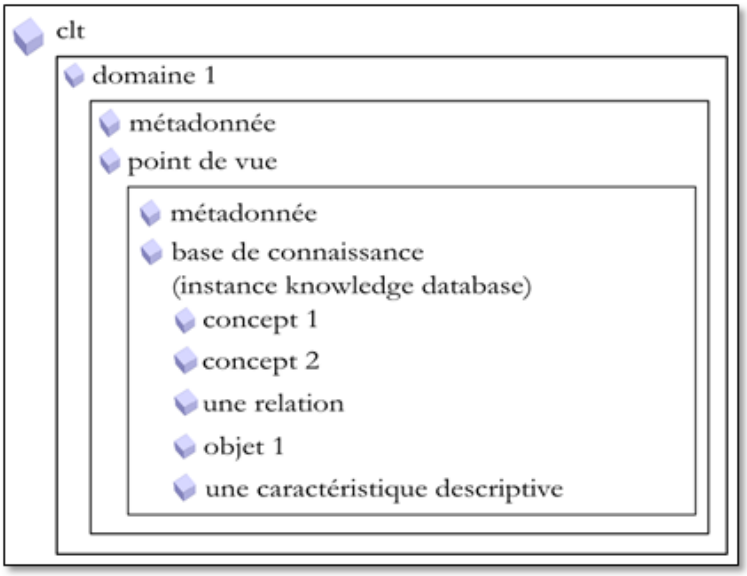


Figure 82 : Exemple d'instanciation de la classe "Knowledge database"

Les méthodes accessibles à partir de la classe « Knowledge database » sont les suivantes :

- drawGraph(largeur, hauteur) : représente graphiquement le système conceptuel en affichant les noms des concepts et les relations entre les concepts
- drawGraphTerms(largeur, hauteur, langue) : représente graphiquement le système conceptuel en affichant les relations entre concepts et les termes dans une langue donnée dénotant chaque concept
- printHTMLtermsList(langue) : affiche un tableau HTML de tous les termes du point de vue dans une langue donnée
- printListAnalysisAxis() : affiche tous les axes d'analyse d'un point de vue
- printListDescriptiveChar() : affiche l'ensemble des caractéristiques descriptives d'un point de vue, qu'elles soient affiliées ou non à un concept
- printListEssentialChar() : affiche l'ensemble des caractéristiques descriptives d'un point de vue, qu'elles soient affiliées ou non à un concept ou présentes dans un axe d'analyse
- printObjectList() : affiche l'ensemble des objets d'un point de vue
- printConceptList() : affiche l'ensemble des concepts d'un point de vue
- printRelationList() : affiche l'ensemble des relations d'un point de vue
- testCyclesIsA() : une méthode qui permet de vérifier s'il existe un ou plusieurs cycles dans un graphe de concepts liés par une relation générique. Elle est appelée automatiquement par le système à chaque ajout de relations génériques.
- printCategoryList() : affiche l'ensemble des catégories d'un domaine
- getCategory(nom de la catégorie) : retourne une instance « catégorie » à partir de son nom
- getAnalysisAxis(nom de l'axe d'analyse) : retourne une instance « analysis axis » à partir de son nom
- getObject(nom de l'objet) : retourne une instance « object » à partir de son nom
- getConcept(nom du concept) : retourne une instance « concept » à partir de son nom
- getRelation(nom de la relation) : retourne une instance « relation » à partir de son nom

- getDescriptiveChar(nom de la caractéristique descriptive) : retourne une instance « descriptive characteristic » à partir de son nom
- getEssentialChar(nom de la caractéristique essentielle) : retourne une instance « essential characteristic » à partir de son nom
- newDescriptiveCharWithConstraint(nom de la caractéristique, expression booléenne) : ajoute une caractéristique descriptive en spécifiant un champ de valeurs possibles, par exemple « ($age>0 && $age<100) » Cette expression booléenne permettra de vérifier si la valeur de la caractéristique descriptive « âge » est comprise strictement entre 0 et 100. Cette vérification aura lieu lors de l'affectation de cette caractéristique à un objet.
- newDescriptiveChar(nom de la caractéristique) : ajoute une caractéristique descriptive dans le système sans contrainte particulière
- newEssentialChar(nom de la caractéristique) : ajoute dans le système une caractéristique essentielle
- newRelation(concept 1, type de la relation, concept 2) : ajoute une relation dans le système en spécifiant le concept « de départ », le concept « d'arrivée » et le type de relation
- newObject(nom de l'objet) : ajoute un nouvel objet dans le système
- newConcept(nom du concept) : ajoute un nouveau concept dans le système
- newCategory(nom de la catégorie) : permet de créer une catégorie qui contiendra une liste de concepts.
- newCLTclass(nom de la classe) : permet de créer une classe CLT afin de regrouper des objets en fonction d'une expression donnée, par exemple : tous les objets de couleur rouge
- deleteRelation(nom de la relation) : supprime une relation donnée par son nom. Le nom de la relation est composé du nom de concept « de départ », du concept « d'arrivée » et du type de relation
- deleteEssentialChar(nom de la caractéristique) : supprime une caractéristique essentielle du système en donnant son nom
- deleteDescriptiveChar(nom de la caratéristique) : supprime une caractéristique descriptive du système en donnant son nom
- deleteAnalysisAxis(nom de l'axe d'analyse) : supprime un axe d'analyse du système à partir de son nom
- deleteObject(nom de l'objet) : supprime un objet du système en donnant son nom
- deleteConcept(nom du concept) : supprime un concept en donnant son nom
- deleteClass(nom de la classe) : supprime une classe CLT du système
- generateFormalDefinition(instance d'un concept) : affiche la définition formelle d'un concept, c'est-à-dire ses caractéristiques essentielles, descriptives et ses relations directes avec d'autres concepts si elles existent
- Les méthodes d'export au format CSV, RDF, RDFS et OWL

Par exemple, l'instruction suivante :

« $pv1->knowledgeDatabase->newConcept('Siège'); » a pour effet de créer une nouveau concept dans l'instance « knowledgeDatabase » rattaché au point de vue « $pv1 ».

3.2.2.5 LA CLASSE « UNIT OF KNOWLEDGE »

La classe « Unit Of knowledge » (Figure 83) ou “unité de connaissance” est la classe “parent” ou superordonnée des classes subordonnées « CLTconcept, CLTcharacteristic, CLTclass, CLTobject, Relation, Category, AxisOfanalysis ». Cette classe ne sera pas instanciée directement et pourrait être représentée sous la forme d’une classe abstraite. Ce sont ses classes subordonnées (spécialisées) tels que CLTconcept qui seront instanciées. Elles représentent les éléments fondamentaux de la théorie du concept. La classe « Unit of Knowledge » se voit rattachée une classe « Meta data » dans le même but que précédemment, afin de savoir par exemple qui est l’auteur d’un concept créé et sa date de création.

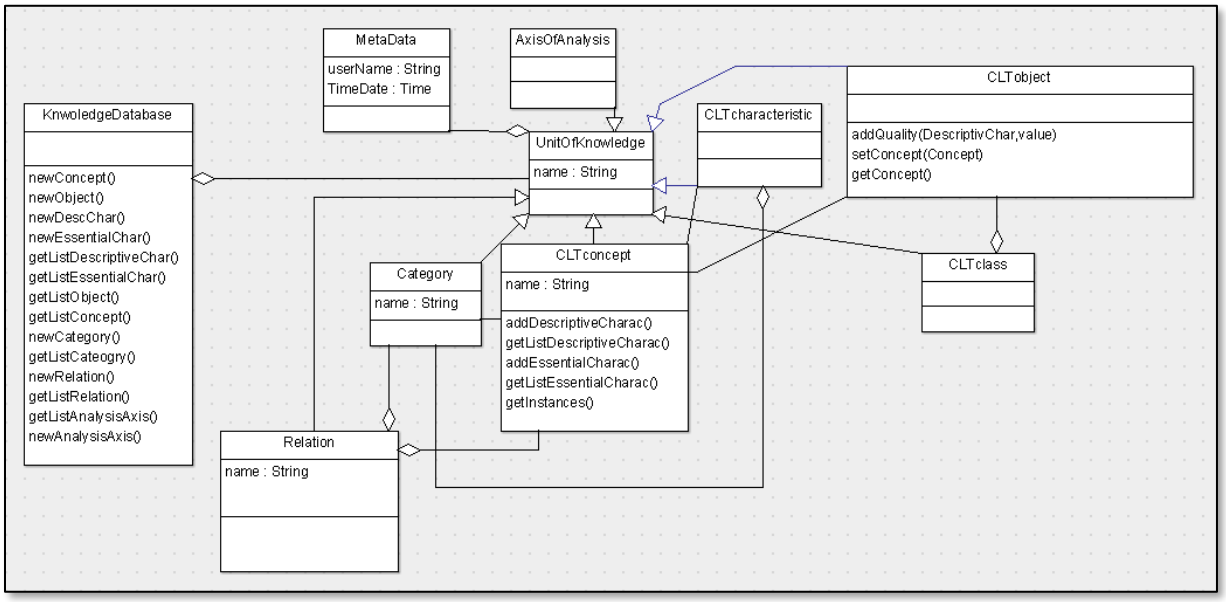


Figure 83 : La classe "Unit of Knowledge" et ses relations

La classe « Unit of Knowledge » ne possède pas de méthodes propres.

3.2.2.6 LA CLASSE « CLT CONCEPT »

La classe « CLTconcept » (Figure 84), classe subordonnée à la classe « Unité de connaissance » est utilisée pour représenter le concept de la CLT. Les instances de la classe « CLT concept » seront créées à l’aide de la méthode « newConcept(nom du concept) » de la classe « Knowledge Database ». La représentation du concept telle qu’elle est faite ici *via* la classe « CLTconcept » permettra d’affecter au concept des caractéristiques aussi bien descriptives qu’essentielles. Des axes d’analyse peuvent être affectés au concept, c’est pourquoi un lien d’agrégation existe entre la classe « CLT concept » et « Axis of Analysis ».

La classe « CLT concept » entretient un lien d’association avec la classe « CLTobject » et contient une méthode pour l’instanciation d’objets en utilisant la classe « CLTobject ». Lorsque cette méthode est appelée, elle aura pour effet d’appeler la méthode « newRelationInstanceOf(concept, objet) » avec pour paramètre le nom du concept et de l’objet. Il sera aussi possible de connaître les objets dont le concept est le modèle. La classe « CLTconcept » a pour partie la classe « Désignation » afin de

pouvoir désigner un concept par des termes. Enfin elle se retrouve aussi utilisée par la classe « Relation » car une relation dans le système conceptuel est composée de concepts.

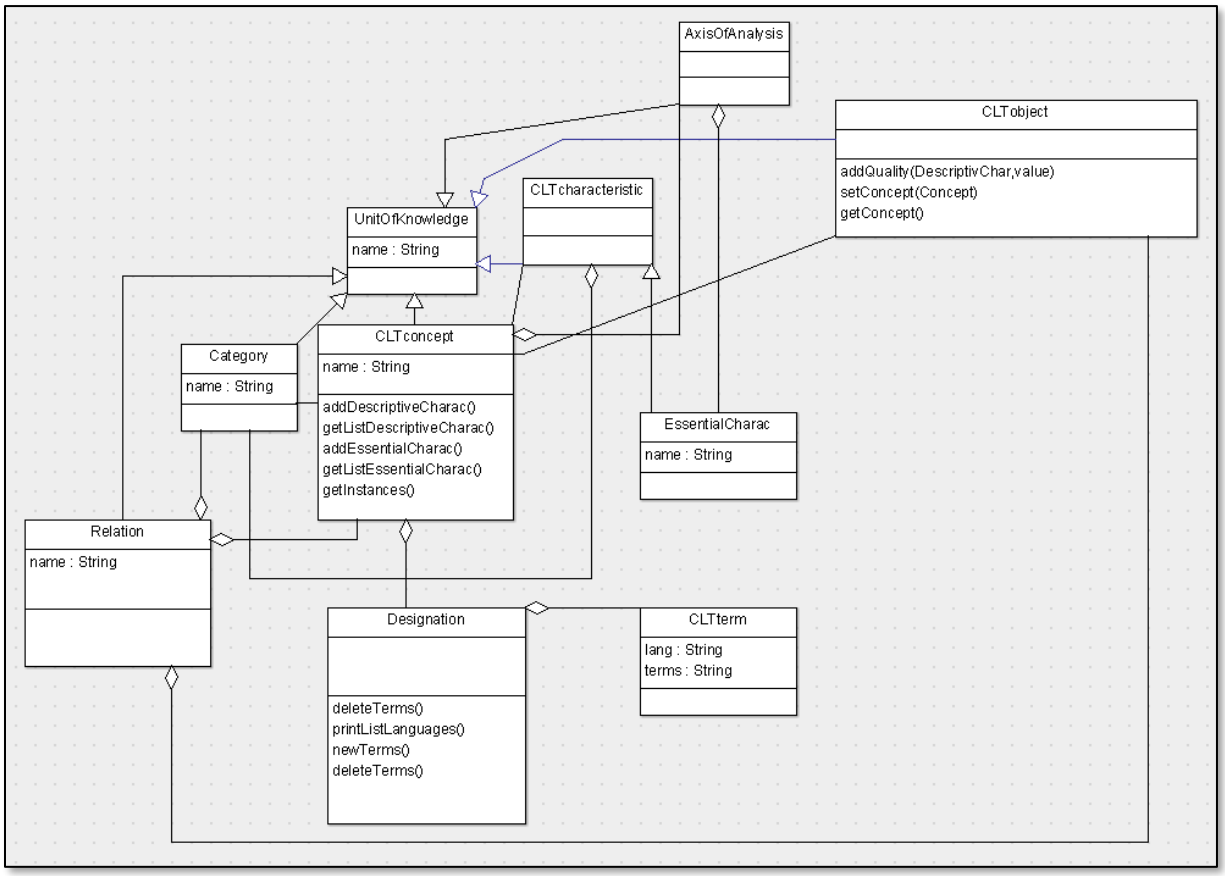


Figure 84 : La classe "CLTconcept" et ses relations

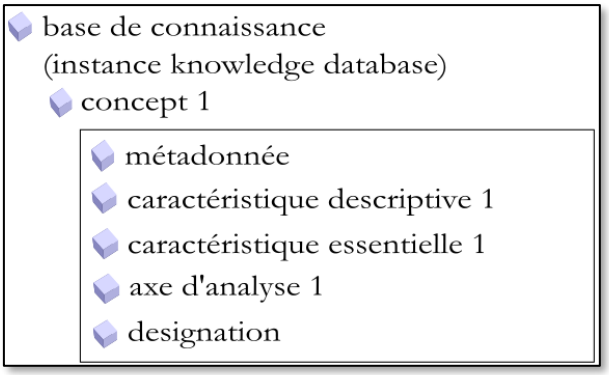


Figure 85 : Exemple d'instanciation de la classe "CLTconcept"

Les méthodes accessibles à partir de la classe « CLT concept » sont les suivantes :

- printName() : affiche le nom du concept
- printDescriptiveCharList() : affiche la liste des caractéristiques descriptives du concept en question
- printEssentialCharList() : affiche la liste des caractéristiques essentielles du concept en question

- addDescriptiveChar(instance de la classe caractéristique descriptive) : permet d'affecter une instance de la classe caractéristique descriptive au concept
- addEssentialChar(instance de la classe caractéristique essentielle) : permet d'affecter une instance de la classe caractéristique essentielle au concept
- addAnalysisAxis(instance de la classe axe d'analyse) : permet d'affecter un axe d'analyse au concept
- removeEssentialChar(instance de la classe caractéristique essentielle) : permet d'enlever le lien qui existe entre le concept et une caractéristique essentielle mais ne la supprime pas du système
- removeDescriptiveChar(instance de la classe caractéristique descriptive) : permet d'enlever le lien qui existe entre le concept et une caractéristique descriptive mais ne la supprime pas du système
- removeAnalysisAxis (instance de la classe axe d'analyse) : permet d'enlever le lien qui existe entre le concept et un axe d'analyse descriptive mais ne le supprime pas du système

Par exemple, l'instruction « $c->addEssentialChar($ec); » a pour effet « d'affecter » au concept une instance de la classe « Caractéristique essentielle » à un concept « $c ».

3.2.2.7 LA CLASSE « CLT RELATION »

La classe « CLT relation » (Figure 86) est une « unité de connaissance » et sert de modèle à la création des relations dans le système conceptuel. Ces instances sont créées grâce à la méthode « newRelation(concept 1, nom de la relation, concept 2) » ou bien avec la méthode newRelationInstanceOf(concept, objet) de la classe « Knowledge Database ». Lorsqu'il s'agit de lier des concepts entre eux, c'est la première méthode qui est utilisée et prend comme paramètre deux concepts. Le « range » et le « domain » (l'ensemble de départ et d'arrivée) sont spécifiés à l'aide de la classe catégorie qui contient une liste de concepts. C'est pourquoi une relation aura pour partie deux catégories si l'on veut définir l'ensemble de départ et d'arrivée de la relation. Les concepts passés en paramètre devront appartenir aux catégories respectives pour que la relation soit valide. Si l'on souhaite créer une relation d'instanciation dans le système conceptuel, c'est donc la deuxième méthode qui est utilisée, celle qui permet de lier un concept à un objet par une relation d'instanciation. C'est pourquoi la classe « CLT relation » possède aussi une relation d'agrégation avec la classe « CLT object ». Dans les relations établies entre les concepts, le nom de la relation aura son importance car il servira à détecter des relations spécifiques telle que la relation de « généricité » ou « partie de » et engendreront un comportement différent du système en fonction des types de relation. Les relations portant un autre nom seront considérées comme des relations d'association.

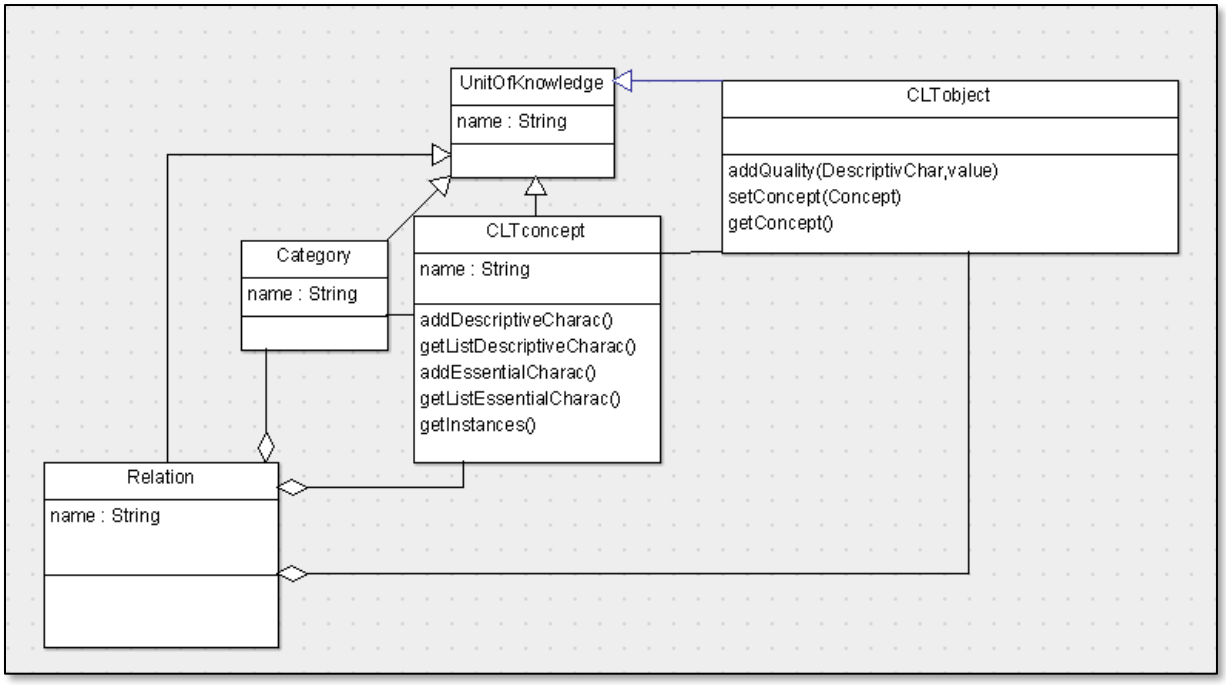


Figure 86 : La classe "Relation" dans le modèle CLT

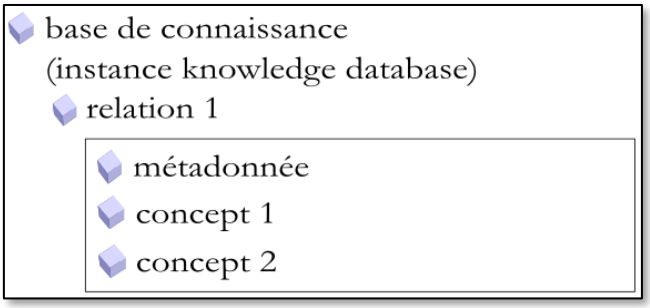


Figure 87 : Exemple d'instanciation d'une relation

La classe « Relation » ne contient qu'une méthode « printName() » qui affichera le nom de la relation, on pourra cependant accéder aux variables d'instance de la relation, par exemple aux concepts impliqués dans la relation. L'instruction « $r1->printName() » permet d'afficher le nom de la relation de l'instance « r1 ».

3.2.2.8 LA CLASSE « CLT OBJECT »

La classe « CLT object » (Figure 88) est une « unité de connaissance » servant de modèle à la création d'objets dans le système conceptuel. Elle est liée à la classe « CLT concept » par une relation d'association. En effet, même si un objet peut exister dans le système sans être affilié à un concept, il devra être possible de connaître le concept dont il peut être l'instance si le concept existe, ou bien si une relation d'instanciation existe entre le concept et l'objet en question (la classe « CLT object » est liée aussi à la classe Relation). L'objet tel qu'il est défini ne possède pas directement de caractéristiques essentielles, on pourra accéder aux caractéristiques essentielles de l'objet *via* le concept dont il est l'instance. L'objet possède directement ses propres caractéristiques descriptives.

En effet, elle a pour partie la classe « Object descriptive characteristic». Un objet pourra être créé de manière autonome en utilisant une méthode de la classe « Knowledge Database » ou bien à partir de la classe « CLT concept » à partir de la méthode newObject().

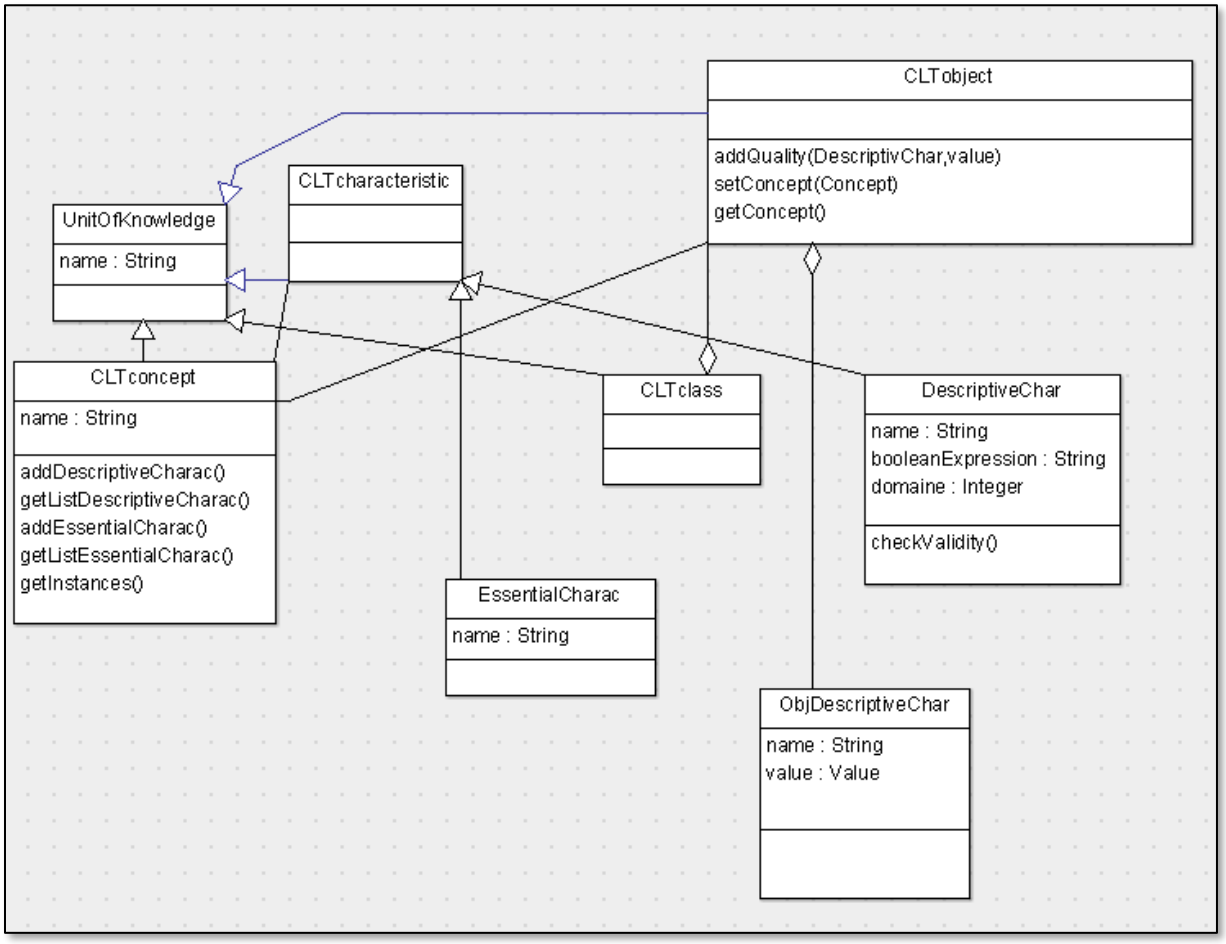


Figure 88 : La classe "CLT object" et ses relations

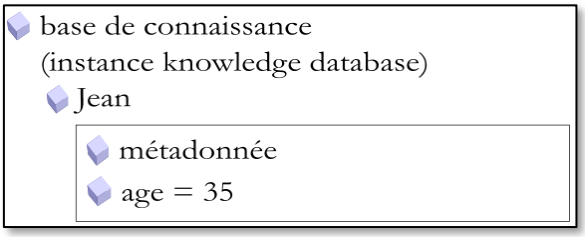


Figure 89 : Exemple d'instanciation d'un objet

Les méthodes accessibles à partir de la classe « CLT object » sont les suivantes :

- printName() : affiche le nom de l'objet
- printObjDescriptiveCarList() : affiche la liste des caractéristiques descriptives de l'objet
- printInstanceOf() : retourne le nom du concept dont il est une exemplification, à noter qu'un objet peut exister sans concept.

- addObjDescriptiveChar(instance de la caractéristique descriptive, valeur) : affecte une caractéristique descriptive à l'objet avec une valeur donnée. La valeur attribuée à la caractéristique doit être comprise dans le champ de valeur préalablement défini
- getObjDescripitiveChar(nom de la caractéristique descriptive) : retourne une instance de la classe « Object descriptive characteristic »

Par exemple l'exécution de l'instruction « $o-> addObjDescriptiveChar($age,'35') ; » permettra d'ajouter la caractéristique « âge » avec la valeur « 35 » à l'objet « $o ».

3.2.2.9 LA CLASSE « CLT CHARACTERISTIC »

La classe « CLT characteristic » (Figure 90) subordonnée à la classe « Unit Of Knowledge », est la classe superordonnée aux classes « Essential Characteristic » et « Descriptive Characteristic ». Elle entretient une relation d'association avec la classe « CLT concept » car il peut être utile de connaitre à quel concept appartient une caractéristique descriptive ou essentielle. Les caractéristiques sont instanciées à l'aide des classes spécialisées « Essential Characteristic » ou « Descriptive Characteristic » au niveau de la classe « Knowledge Database » et pourront ensuite être rattachées directement à un concept. La classe « CLT characteristic » est liée à la classe « Category » par une relation d'agrégation afin de pouvoir restreindre une caractéristique essentielle ou descriptive à une catégorie de concepts donnée.

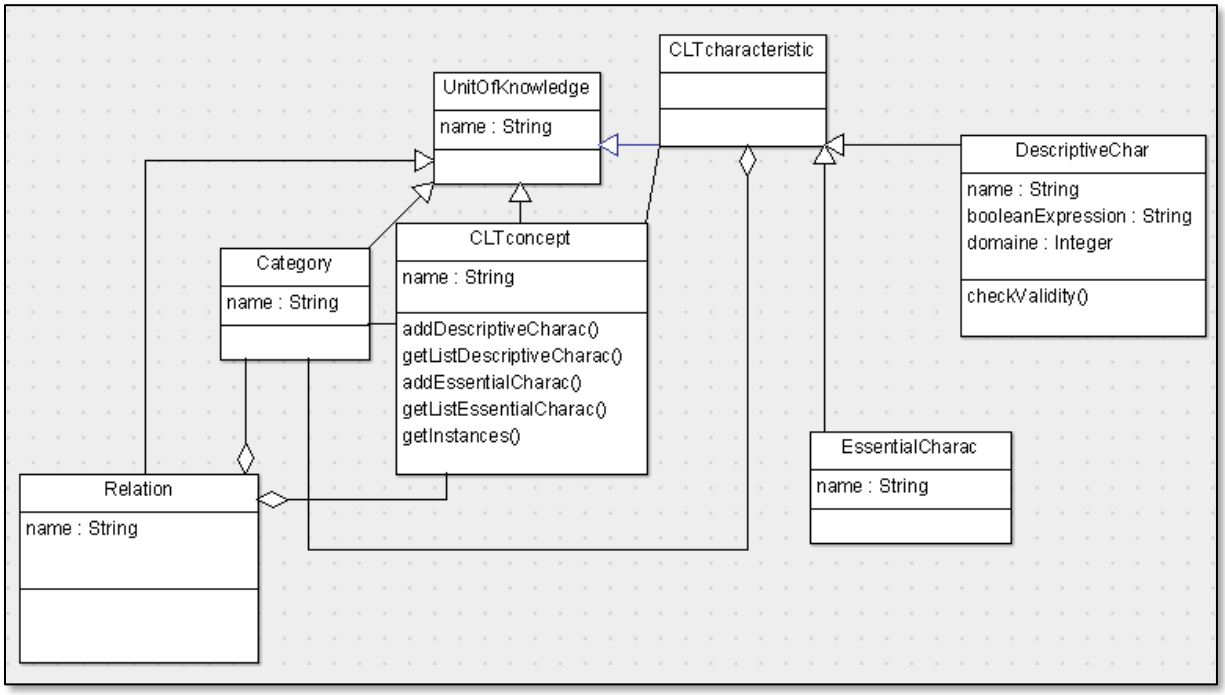


Figure 90 : La classe "CLT characteristic" et ses relations

La classe « CLT characteristic » n'a pas pour vocation à être instanciée et pourrait être représentée sous la forme d'une classe abstraite.

3.2.2.10 LA CLASSE « ESSENTIAL CHARACTERISTIC »

La classe « Essential Characteristic » (Figure 91) est une classe subordonnée à la classe « CLT characteristic ». Elle contient simplement un champ pour le nom de la caractéristique essentielle. Une instance de cette classe servira à représenter la caractéristique essentielle d'un concept. Un concept peut se voir attribuer une ou plusieurs caractéristiques essentielles de manière directe (La classe « CLT concept » contient une méthode d'affectation de caractéristiques essentielles ») ou bien *via* un axe d'analyse. C'est pourquoi la classe « Essential caracteristic » est une partie de la classe « Axis of Analysis », un axe d'analyse est défini à partir d'une liste de caractéristiques essentielles. A noter qu'elle peut être créée indépendamment d'un axe d'analyse *via* une méthode de la classe « Knowledge database ».

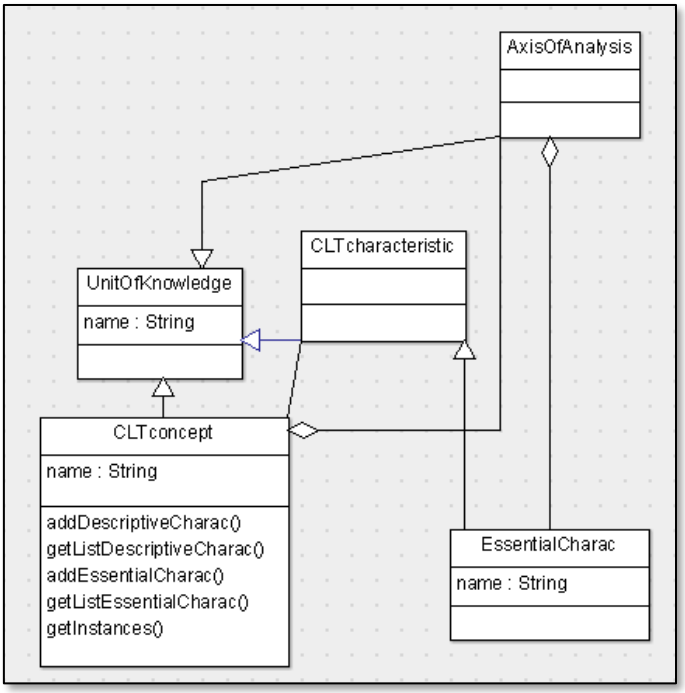


Figure 91 : la classe "Essential characteristic" et ses relations

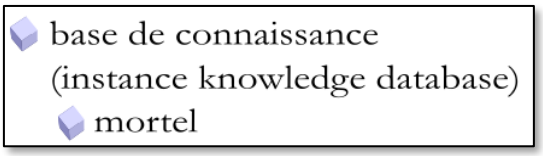


Figure 92 : Exemple d'instanciation d'une caractéristique essentielle

La classe « Caractéristique essentielle » ne contient qu'une méthode permettant d'afficher le nom de la caractéristique essentielle : « printName() », elle pourra être utilisée de la manière suivante : « $ec->printName() » ; « $ec » étant une instance de la classe « Caractéristique essentielle ».

3.2.2.11 LES CLASSES «DESCRIPTIVE CHARACTERISTIC» ET «OBJECT DESCRIPTIVE CHARACTERISTIC»

La classe « Descriptive Characteristic » (Figure 93) hérite de la classe « CLT characteristic », elle permet de décrire la chose à travers un couple, nom de l'attribut / valeur. Par exemple : « âge = 35 ». Une caractéristique descriptive est créée à partir de la classe « Knowledge Database » grâce à la méthode « newDescritiveChar(nom de la caractéristique descriptive, expression booléenne) ». Le premier paramètre de la méthode sert à nommer la caractéristique, le deuxième permet d'ajouter une contrainte sur la valeur qui doit être attribuée au niveau de l'objet. Par exemple on peut spécifier que l'âge est un nombre strictement positif « (age > 0) ». Cette expression sera évaluée lorsqu'une caractéristique descriptive sera affectée à l'objet sous la forme d'une instance de la classe « Object Descriptive Characteristic ». La classe « Descriptive Characteristic » et « Object Descriptive Characteristic » n'ont pas la même structure. L'une sert à représenter la caractéristique descriptive au niveau du concept et l'autre, au niveau de l'objet. La caractéristique descriptive au niveau de l'objet contient une valeur dont le champ de validité est défini dans une instance de la classe « Descriptive Characteristic » à travers l'attribut « expression booléenne ». On peut voir la classe « Object Descriptive Characteristic » comme une réalisation de la caractéristique descriptive. Les caractéristiques descriptives de l'objet sont créées grâce à la méthode « addObjDescripitiveChar(caractéristique descriptive, valeur) » de la classe « CLT object ».

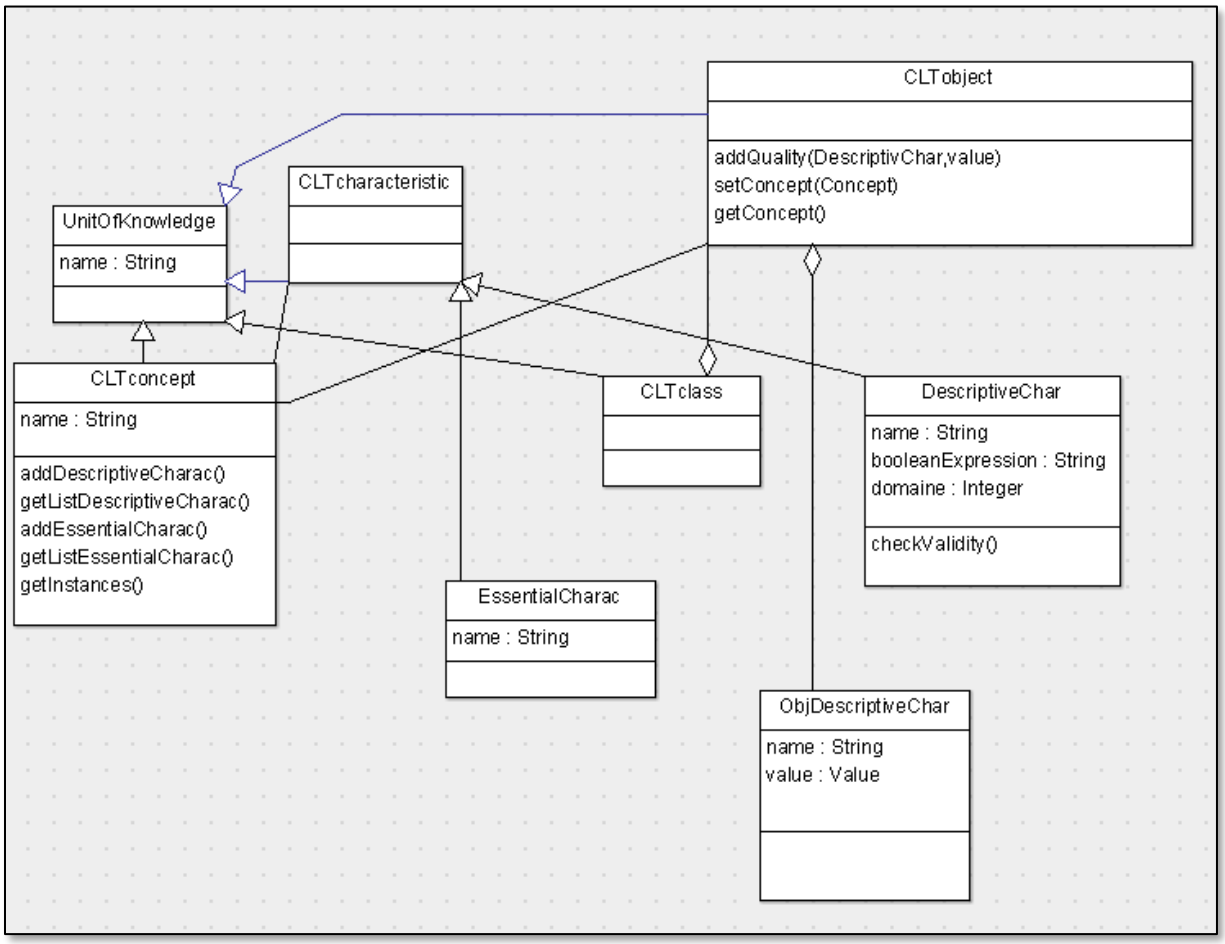


Figure 93 : Les classes « Descriptive characteristic » et « Object descriptive characteristic » dans le modèle

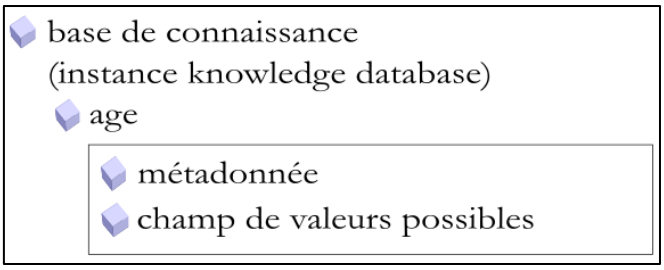


Figure 94 : Exemple d'instanciation d'une caractéristique descriptive

Les méthodes accessibles à partir de la classe « Caractéristique descriptive » sont les suivantes :

- printName() : affiche le nom de l'objet
- printRange() : affiche l'expression booléenne utilisée pour définir le champ des valeurs possibles
- checkExpression() : permet d'évaluer l'expression booléenne, cette méthode est appelée automatiquement à chaque affectation d'une nouvelle caractéristique au niveau de l'objet

Par exemple, l'utilisation de l'instruction « $cd->printRange() ; » affichera l'expression qui définira le champ de validité de la valeur de la caractéristique.

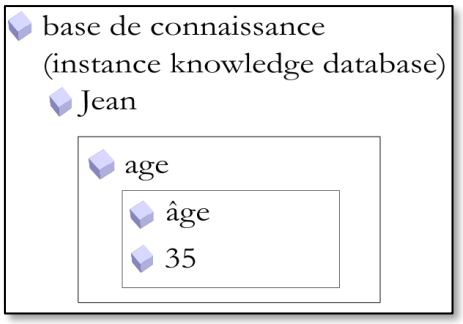


Figure 95 : Exemple d'instanciation de la classe "Object descriptive characteristic"

Les méthodes accessibles à partir de la classe « Object descriptive characteristic » sont les suivantes :

- printName() : affiche le nom de la caractéristique descriptive
- printValue() : affiche la valeur courante de la caractéristique descriptive affectée à l'objet

Par exemple, l'exécution de l'instruction « $odc->printValue() » retournera la valeur 35 dans l'exemple ci-dessus.

3.2.2.12 LA CLASSE « AXIS OF ANALYSIS »

La classe « Axis of analysis » (Figure 96) est « une unité de connaissance » et permet de représenter un type de caractéristique. Elle a pour partie la classe « Essential Characteristic » et pourra contenir une liste de caractéristique essentielles qui seront définies comme exclusives, par exemple : « mortel vs immortel ». Un concept ne pourra posséder alors qu'une seule des caractéristiques de l'axe d'analyse. La classe « CLT concept » est liée à la classe « Axis of analysis » par une relation

d'agrégation. Un concept peut se voir affecter un ou plusieurs axes d'analyse qui devront être compatibles entre eux.

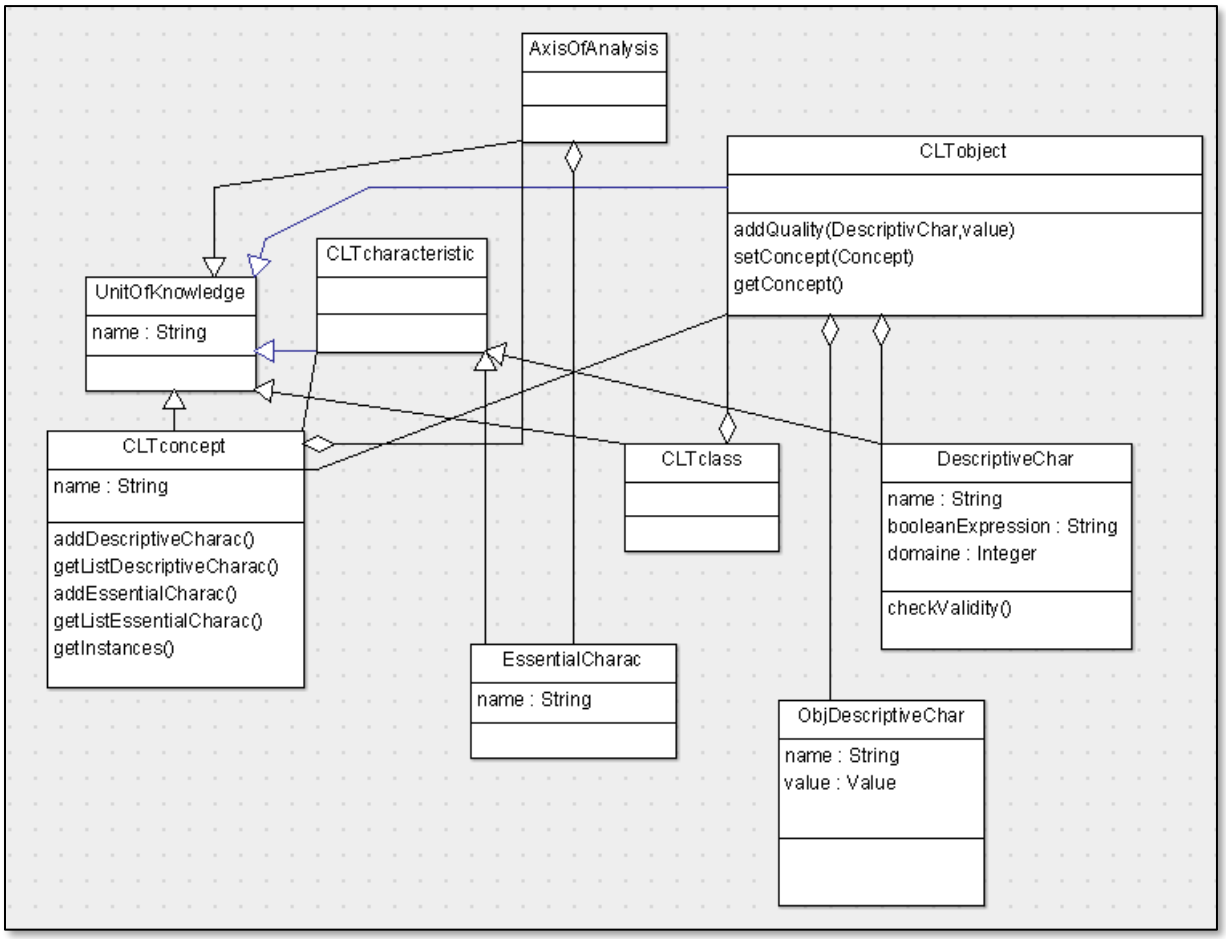


Figure 96 : La classe "Axis of analysis" ou type de caractéristique

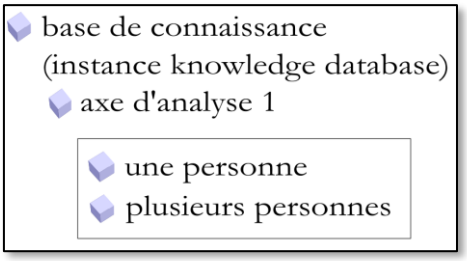


Figure 97 : Exemple d'instanciation d'un axe d'analyse

La classe « Axe d'analyse » contient les méthodes suivantes :

- printName() : affiche le nom de l'axe d'analyse
- printEssentialCharList() : affiche les caractéristiques rattachées à l'axe d'analyse
- addEssentialChar(nom de la caractéristique essentielle) : ajoute une caractéristique essentielle dans l'axe d'analyse
- removeEssentialChar(nom de la caractéristique essentielle) : enlève une caractéristique essentielle de l'axe d'analyse

L'exécution de l'instruction « $ax->addEssentialChar('une personne') » aura pour effet d'ajouter la caractéristique essentielle « une personne » à l'instance $ax qui représente un axe d'analyse.

3.2.2.13 LA CLASSE « CATEGORY »

La classe « Category » (Figure 98) est une une unité de connaissance. Elle est instanciée au niveau de la classe « Knowledge database » grâce à la méthode « newCategory(nom de la catégorie) », des concepts peuvent être ensuite ajoutés ou retirés de la catégorie. Elle entretient une relation d'association avec la classe « CLT concept » car pour une catégorie nous avons besoin de savoir quels concepts sont liés à celle-ci et pour un concept nous pourrons connaître à quelle(s) catégorie(s) il est rattaché. Elle est aussi liée à la classe « Relation » et « CLT characteristic » par une relation d'agrégation.

La classe « Category » sert ici simplement à définir un ensemble de concepts qui pourront intervenir dans la définition du « Range » et du « Domain » dans une relation. Elle intervient aussi dans la définition des caractéristiques, où il sera possible attribuer un ensemble de concepts compatible avec la caractéristique en question. Par exemple l'affectation de la caractéristique « vitesse dans l'air » pourra être restreinte à un certain type d'objet, les objets dont le concept appartient à la catégorie des « véhicules aériens ».

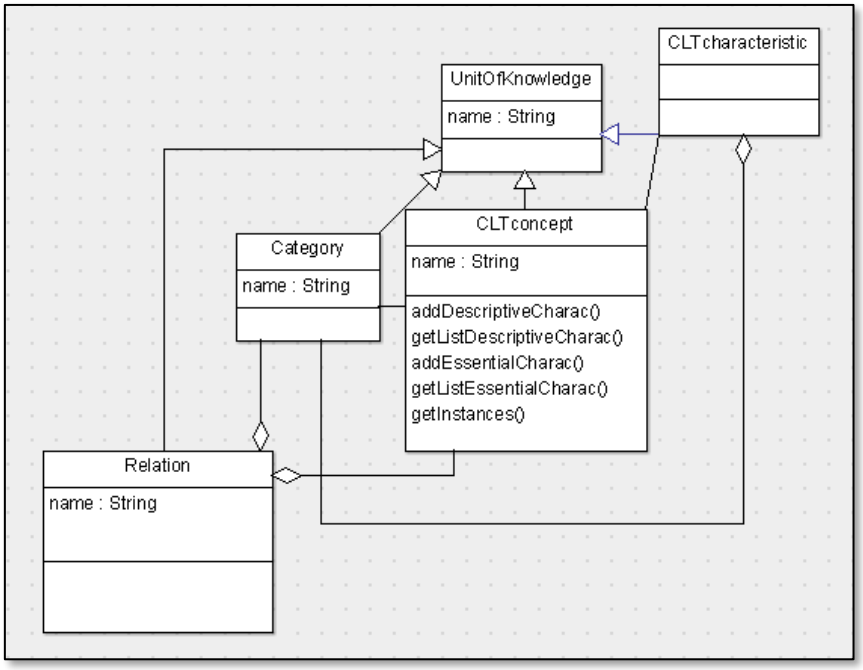


Figure 98 : La classe "Category" dans le modèle de la CLT

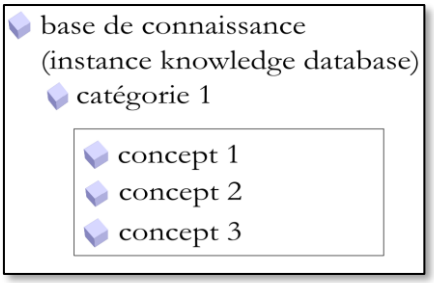


Figure 99 : Exemple d'instanciation d'une catégorie

Les méthodes accessibles depuis la classe « Category » sont les suivantes :

- printName() : affiche le nom de la catégorie
- printConceptList() : affiche la liste des concepts affectée à la catégorie
- addAconcept(instance d'un concept) : ajoute une instance de la classe « CLT concept » à la catégorie
- getConcept(nom du concept) : retourne une instance de concept présent dans la catégorie à partir du nom du concept
- removeConcept(instance d'un concept) : enlève un concept de la catégorie

Par exemple, l'utilsation de l'instruction « $categoerie1->printConceptList() » affichera la liste des concepts qui figurent dans la catégorie « $categorie1 ».

3.2.2.13 LA CLASSE « CLT CLASS »

La classe « CLT class » (Figure 100), unité de connaissance permet de regrouper des objets de tout type en fonction de leurs propriétés (par exemple tous les objets dont la caractéristique « couleur » possède la valeur « rouge ». Une instance de la classe « CLT class » est une liste d'objets qui est créée à partir de la classe « Knowledge database ». Elle entretient donc une relation d'agrégation avec la classe « CLT object ».

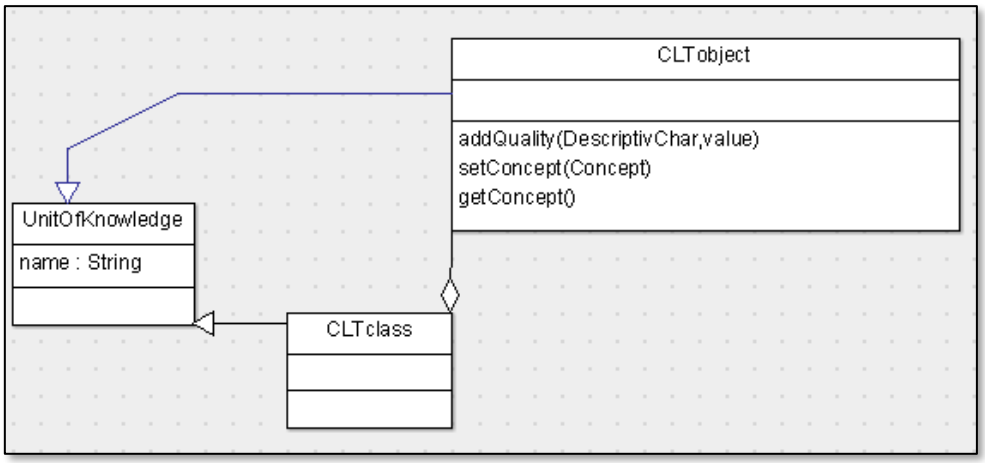


Figure 100 : La classe "CLT class"

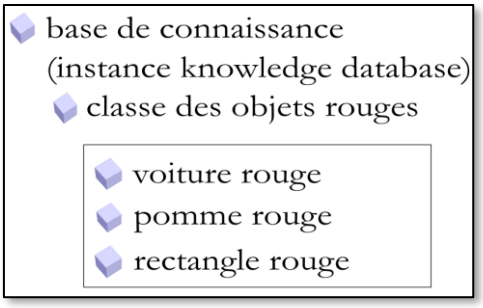


Figure 101 : Exemple d'instanciation de la classe "CLT classe"

Les méthodes accessibles à partir de cette classe sont les suivantes :

- printName() : affiche le nom de la classe
- addObject(instance d'une classe « Objet ») : ajoute un objet spécifique dans la classe
- getObject(nom de l'objet) : retourne une instance de la classe « Objet » à partir de son nom
- removeObject(instance d'une classe « Objet ») : enlève de la classe un objet spécifique
- addObjects(Expression booléenne) : ajoute tous les objets qui vérifient l'expression passée en paramètre dans la classe

L'instruction « $classe1->addObject($o) » permettra d'ajouter l'instance $o de la classe « Objet » dans l'instance « $classe1 » qui représente une classe CLT.

3.2.2.14 LA CLASSE « DESIGNATION »

La classe « Designation » (Figure 102) est utilisée ici pour les désignations verbales d'un concept. Elle est représentée comme une partie de la classe « CLT concept », l'accès à la désignation se fera donc à travers le concept. Elle est liée à la classe « CLT term » par une relation d'agrégation, une désignation pourra alors se faire affecter une liste d'instances de la classe « CLT term ». La classe « Designation » servira à manipuler cette liste de termes qui dénotent le concept en question. Pour chaque terme nous aurons une chaîne de caractères et une information concernant la langue utilisée. Il sera alors possible de connaître dans différentes langues les termes qui désignent un concept.

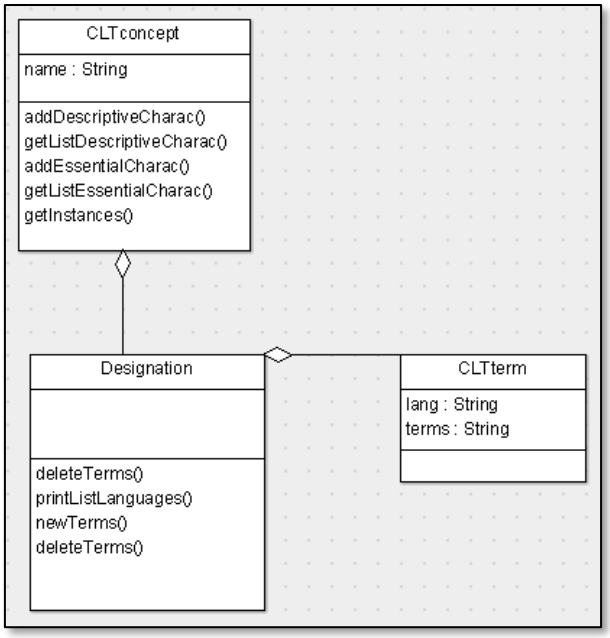


Figure 102 : La classe "Designation" dans le modèle CLT

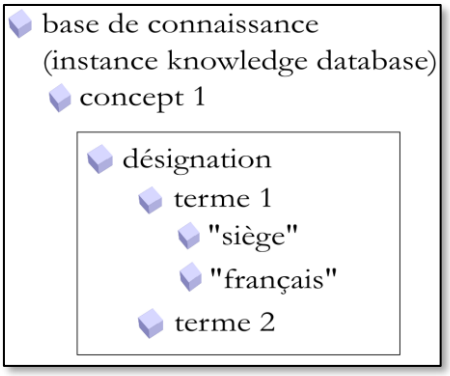


Figure 103 : Exemple d'instanciation de la classe "Désignation"

Les méthodes disponibles depuis la classe « Désignation » sont les suivantes :

- printLanguages() : affiche la liste des langues disponibles pour une désignation
- printTerms(langage) : affiche tous les termes d'une désignation pour une langue donnée
- printHtmlTermsListLang(langage) : retourne un tableau html contenant la liste de tous les termes qui dénotent un concept dans une langue donnée
- printListHtmlTerms() : retourne dans un tableau HTML tous les termes d'une désignation de concept indépendamment de la langue
- newTerm(langage, terme) : ajoute pour une désignation un terme dans une langue donnée
- deleteTerm(langage, terme) : supprime de la désignation un terme dans une langue donnée

3.2.2.15 LA CLASSE « METADATA »

La classe « Meta data » (Figure 104) a pour simple vocation de garder des informations concernant la construction du système conceptuel, c'est pourquoi à chaque unité de connaissance est associée une « méta donnée », l'auteur et la date. Cette information est aussi présente dans les domaines et les points de vue. Par exemple, on pourra connaître l'auteur et la date de création de chaque concept dans le système conceptuel.

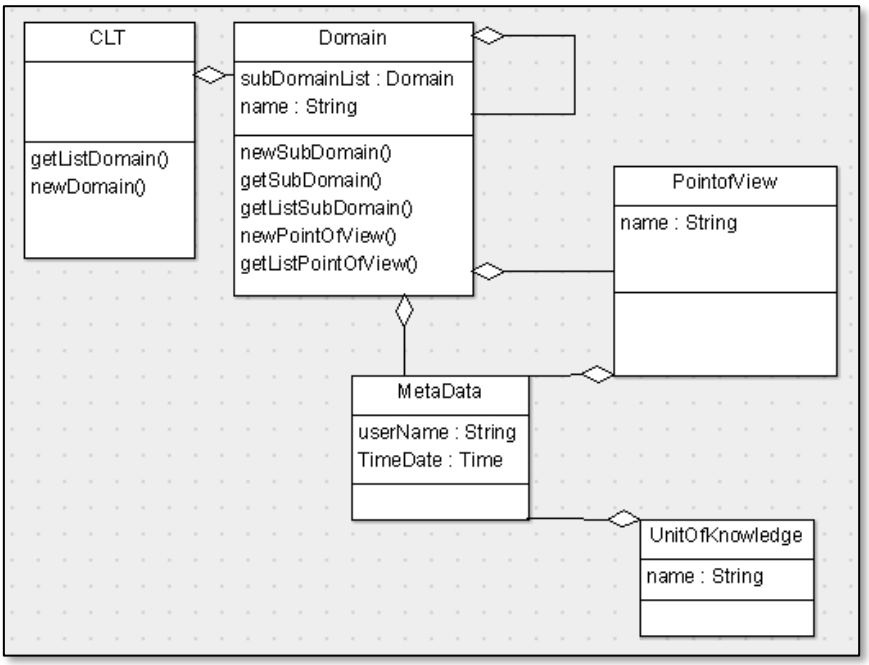


Figure 104 : La classe "Meta Data" dans le modèle CLT

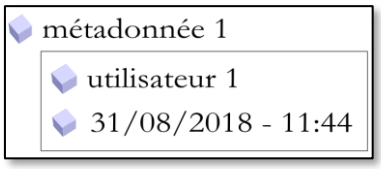


Figure 105 : Exemple d'instanciation d'une métadonnée

La classe « Metadata » contient une méthode qui retourne le nom de l'utilisateur « getUserName() » et une méthode retournant la date et l'heure de création de cette métadonnée : « getDateTime () ».

3.2.2.16 REPRESENTATION DYNAMIQUE DU SYSTEME : UN EXEMPLE D'INSTANCIATION

Dans la partie précédente, nous avons vu la représentation statique de la théorie du concept (CLT) sous la forme d'un diagramme de classe en UML. Il s'agit ici d'instancier à titre d'exemple le modèle CLT, c'est-à-dire qu'au lieu de représenter les classes de la CLT, nous proposerons une possible instanciation de ces classes, on parle alors de représentation dynamique du système. Nous présentons ici un exemple d'instanciation de la CLT mettant en œuvre ses principales classes (Figure 106). Un objet de la classe CLT contient toutes les instances créées à partir de celle-ci. Le modèle

orienté objet choisi pour cette implémentation fait apparaître la notion d'encapsulation : les variables d'instances et les méthodes sont regroupées dans les classes et une variable d'instance peut faire référence à une autre classe. La modification d'une classe n'impactera pas les autres, c'est tout l'intérêt de la notion d'encapsulation en programmation objet. Cette encapsulation se retrouve dans la structure des objets lorsqu'ils sont instanciés. La structure des objets et la manière dont ils sont agencés sont déduites de leur classe UML. Par exemple, un concept sera créé en appelant une méthode de création de concept sur une instance de la classe « Knowledge database ». Une modification relative à un concept, comme lui ajouter une caractéristique descriptive sera faite depuis le concept lui-même, en appelant sa méthode « addDescritiveChar(nom de la caractéristique) ». Ce type d'organisation permettra de garder une structure du modèle en vue de son évolution.

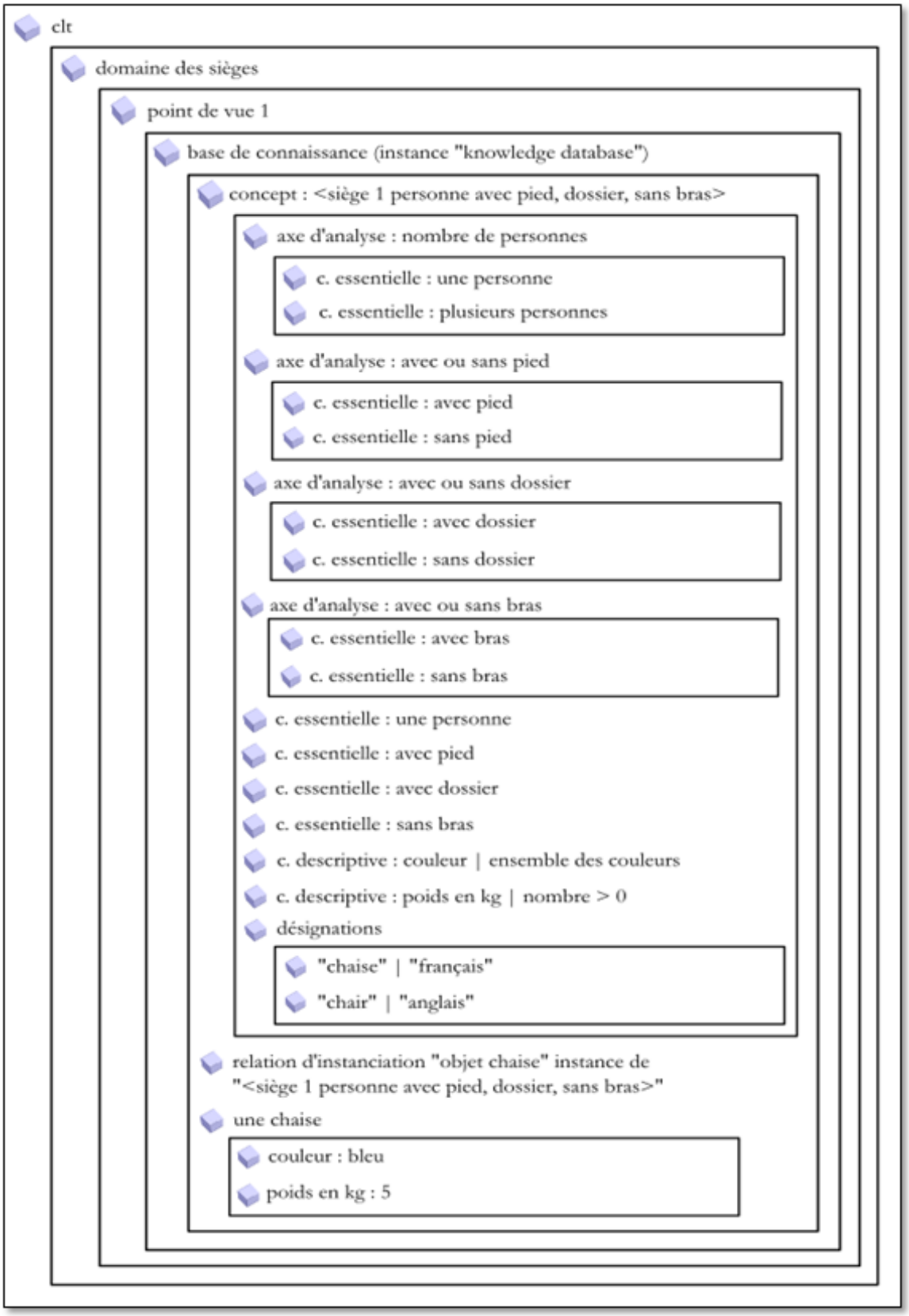


Figure 106 : Exemple d'instanciation de la classe CLT

3.3 IMPLEMENTATION EN PHP5

Le langage PHP5 est un langage de programmation orienté objet interprété et exécuté côté serveur, il dispose d'une fonction « eval » afin d'évaluer une expression, un programme. C'est aussi un langage qui est exécuté sur la plupart des serveurs web. Un langage de concept pour la terminologie, implémenté en PHP aura donc une grande portabilité puisqu'il pourra être exécuté sur la plupart des serveurs web existant actuellement. Le modèle de concept étant spécifié en UML, il est ainsi aisé de le transposer en PHP dont la version 5 est orientée objet. Les classes définies en UML seront les mêmes mais écrites en PHP. Elles sont contenues dans un fichier « clt_kernel.php » pour représenter notre modèle de concept. Il nous suffira ensuite d'instancier ce modèle et d'utliser le langage CLT pour créer un système conceptuel. La syntaxe du langage PHP sera utilisée pour manipuler le système conceptuel. On aura donc des méthodes spécifiques à notre modèle (le langage CLT) pour la manipulation des terminologies computationnelles. Un autre fichier « clt_console.php » contient une interface graphique pour utiliser les éléments du langage CLT sans en avoir à écrire les instructions manuellement. C'est à travers une console que les instructions du langage de concept pour la Terminologie seront exécutées.

Figure 107 : Console pour l'interprétation des commandes CLT

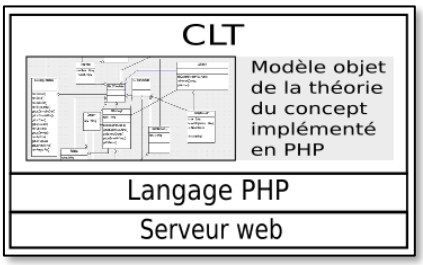


Figure 108 : couche CLT reposant sur PHP

3.3.1 ARCHITECTURE

L'utilisateur dispose d'un navigateur web pour l'affichage de fichiers html et envoyer des requêtes http à un serveur web. Le serveur web sur lequel est exécuté un programme PHP, interprète la requête émise par le client et renvoie une réponse sous la forme d'une page html.

Les instructions propres à la CLT sont placées dans un fichier nommé « clt_kernel.php » qui est un programme PHP. Il contient le modèle implémenté, c'est-à-dire les classes définies dans la spécification UML. Le modèle est ensuite instancié grâce à un objet « clt ». Les méthodes appelées à partir de cette instance permettront de créer et de manipuler des systèmes conceptuels. Les appels

aux méthodes de la CLT sont faits par l'utilisateur, côté client à l'aide d'une interface graphique en html qui contient une console. Les instructions CLT sont envoyées au programme « clt_kernel.php » pour être interprétées grâce la fonction PHP « eval() », le programme renvoie au client le résultat de l'instruction dans une page html où figure une fenêtre de « log ». Les « logs » informent l'utilisateur du traitement des instructions CLT au format texte. Le système conceptuel constitué d'instructions CLT est enregistré séparément du programme « clt_kernel.php » dans un fichier texte « workspace.txt » côté serveur, il s'agit du recueil des requêtes effectuées par l'utilisateur.

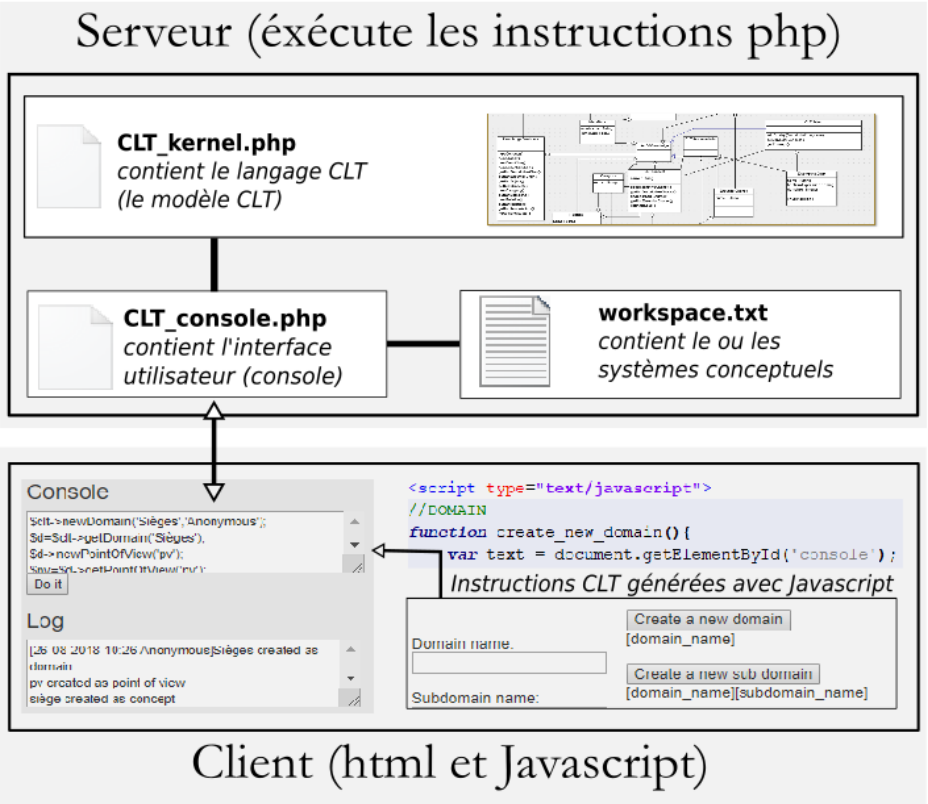


Figure 109 : Architecture du système

Les méthodes appelées à partir de l'instance « clt » permettent de créer et de manipuler des systèmes conceptuels qui sont eux-mêmes enregistrés côté serveur dans un fichier texte « workspace.txt ». Le « workspace » contient les instructions du langage CLT (utilisant la syntaxe PHP) qui ont servi à établir le ou les systèmes conceptuels.

3.3.2 UNE IMPLEMENTATION DANS UNE PERSPECTIVE D'EXTENSION DU SYSTEME

Nous avons vu dans la spécification UML que les classes impliquées et leurs méthodes sont parfois nombreuses (création de concepts, de caractéristiques, d'axes d'analyse…). Pour permettre à l'utilisateur du langage CLT de connaître les méthodes auxquelles il peut utiliser pour chaque classe, une méthode listMethodsOf(nom de la classe) est à sa disposition. Cette méthode a pour fonction de lister toutes les méthodes d'une classe donnée. Conçu dans une perspective d'évolution, de nouveaux éléments du langage CLT peuvent apparaitre sous la forme de méthodes. Il sera alors utile pour l'utilisateur, côté client, de connaître l'ensemble des méthodes d'une classe de la CLT en vue de les utiliser. Une implémentation de la CLT côté serveur aura l'avantage de permettre à l'utilisateur

d'accéder aux nouveaux éléments du langage CLT sans mise à jour ou installation nécessaire de sa part.

```php
<?php
class CLT {
    /* Member variables */
    //List of domains
    var $listDomain = array();
    function printListDomain(){
        global $txt_output;
        ...
    }
    ...
    //Introspection
    function listMethodsOf($className){
        global $txt_output;
        $f = new ReflectionClass($className);
        $methods = array();
        foreach ($f->getMethods() as $m) {
            if ($m->class == $className) {
                $methods[] = $m->name;
                $txt_output.= "$m->name\n";
            }
        }
    }
}
$clt= new CLT;
?>
```

Figure 110 : Un extrait de la classe "CLT" implémentée en PHP

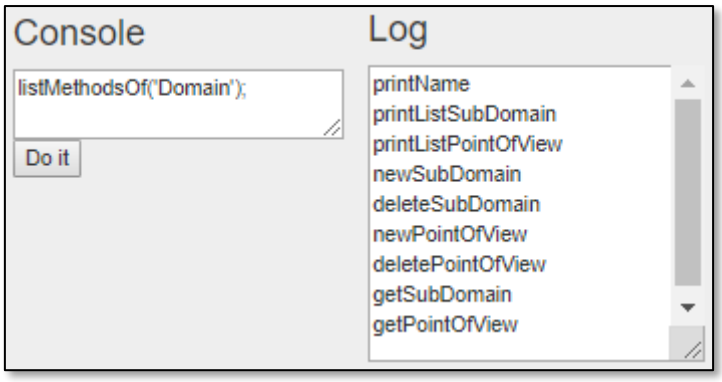


Figure 111 : La possibilité pour l'utilisateur de lister les éléments du langage de la CLT

3.3.3 LA REPRESENTATION D'UN SYSTEME CONCEPTUEL

Le système conceptuel d'une terminologie est construit à l'aide des instructions CLT en utilisant la syntaxe PHP. [Roche, 2018]. La première étape est de créer un domaine et un point de vue :

« $clt->newDomain('terminologie des sièges'); »

La méthode de création de domaine est appelée à partir de l'objet « clt » qui est une variable php et commence par le caractère « $ ». Le symbole « -> » permet d'appeler une méthode de l'objet. La

méthode « newDomain » prend en paramètre un nom de domaine qui doit être unique. Le résultat de l'appel de méthode est un nouvel objet domaine dans une instance clt.

« $d=$clt->getDomain('terminologie des sièges'); »

Pour manipuler un domaine il faut accéder au domaine en utilisant son nom, c'est le but de la méthode « getDomain » qui prend pour paramètre un nom de domaine, cette méthode retourne un objet « domaine « qui peut être placé dans une variable php, ici « $d ».

« $d->newPointOfView('point de vue'); »

Depuis un objet « domaine » est créé un point de vue, ici nommé simplement « point de vue ».

« $pv=$d->getPointOfView('point de vue') ;»

Selon le même principe, on accède au point de vue, cette fois en utilisant la méthode « getPointOfview() » d'un objet « domaine », avec le nom du point de vue.

« $pv->knowledgeDatabase->newConcept('siège'); »

Un point de vue contient un objet « knowledgeDatabase », il contient les méthodes de construction d'un système conceptuel comme la méthode « newConcept() ». A partir d'un point de vue, on accède à l'objet knowledgeDatabase en utilisant « -> » puis à la méthode de création de concept qui a ici pour paramètre le nom du concept « siège ». Une instance de la classe « Concept » est alors ajoutée au niveau de l'objet « knowledgeDatabase », c'est ce même objet qui contient tout le système conceptuel, car il est le conteneur de toutes les « unités de connaissance ».

A chaque exécution d'instruction résulte une réponse du système à travers la fenêtre « log » :

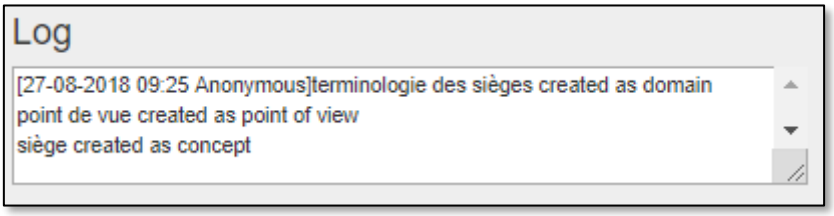


Figure 112 : réponse du système aux instructions exécutées

L'ensemble des instructions CLT exécutées se retrouve sauvegardé côté serveur dans le fichier « workspace.txt », il contient l'ensemble des domaines, points de vue et les systèmes conceptuels affiliés.

```
$clt->newDomain('terminologie des sièges','Anonymous');
$d=$clt->getDomain('terminologie des sièges');
$d->newPointOfView('point de vue');
$pv=$d->getPointOfView('point de vue');
$pv->knowledgeDatabase->newConcept('siège');
```

Figure 113 : Exemple de contenu du fichier "workspace.txt"

3.3.4 L'IMPLEMENTATION DES RELATIONS, EXEMPLE D'IMPLEMENTATION DE LA RELATION GENERIQUE

Une relation est représentée dans le modèle par la classe « CLTrelation ». C'est une instanciation de cette classe qui représente les relations du système conceptuel. Une nouvelle relation peut être créée à partir de l'objet « knowledgeDatabase » en appelant la méthode « newRelation (concept 1, type de la relation, concept 2) ». La méthode «newRelation » implémentée dans la classe « KnowledgeDatabase » a pour but de lier des concepts entre eux en fonction du type de relation. Certains noms de relation sont réservés par le système tels que la relation « is a », « part of », « instance of ». Si le nom de la relation est inconnu, elle sera considérée comme une relation d'association et n'engendrera pas un comportement particulier du système. Par exemple, la relation générique « is a » est implémentée de la manière suivante :

Lors de l'instanciation d'une relation, nous commençons par vérifier s'il s'agit d'une relation générique :

« if ($relation->type_name=="is-a") », si c'est le cas, la relation générique est ajoutée dans le système en vérifiant que cette nouvelle relation ne créera pas un cycle, c'est le rôle de la méthode testCycleIsA(). Cette méthode utilise l'algorithme de Tarjan qui porte le nom de son inventeur et permet dans la théorie des graphes de déterminer les composantes fortement connexes d'un graphe orienté. [Depth-first search and linear graph algorithms, SIAM journal on Computing, vol.1, 1972, p.146-160].

Après s'être assuré que la nouvelle relation ne créera pas de cycles, les axes d'analyse, les caractéristiques et les relations du concept superordonné sont « copiés » dans le concept subordonné. Le concept subordonné héritera alors des axes d'analyse, des caractéristiques et des relations de son concept superordonné.

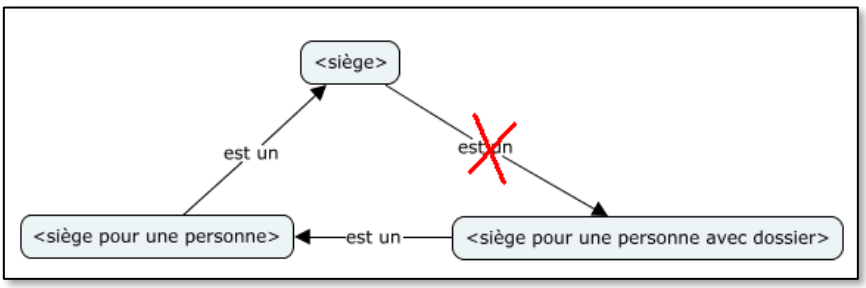


Figure 114 : Détection d'un cycle en utilisant l'algorithme de Tarjan

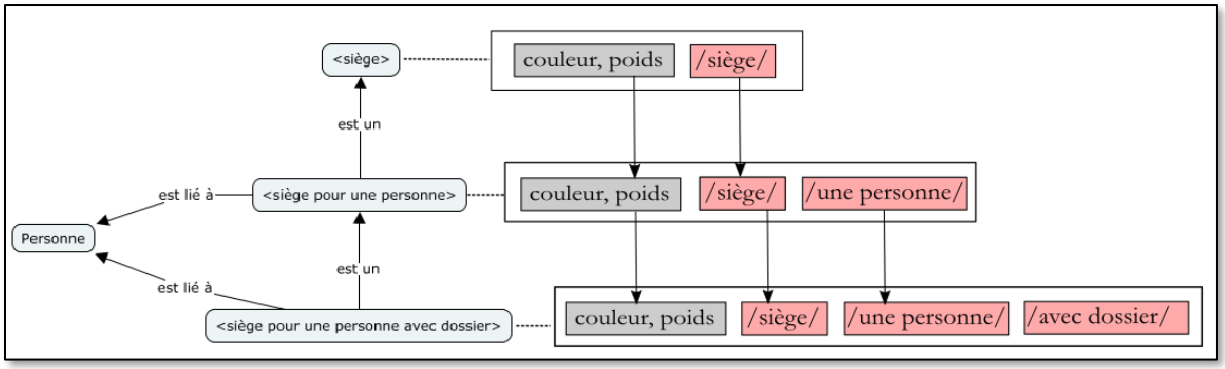


Figure 115 : Héritage des caractéristiques et des relations du concept superordonné

3.3.5 LA GESTION DES ERREURS

On peut distinguer deux types d'erreur :

- Les erreurs propres au langage PHP, comme une erreur de syntaxe dans l'appel d'une méthode « $d->>newPointOfView('pv'); » au lieu de « $d->newPointOfView('pv'); ».
  Le langage PHP générera l'erreur suivante :
  Parse error: syntax error, unexpected '>', expecting identifier (T_STRING) or variable (T_VARIABLE) or '{' or '$' in **console.php(343) : eval()'d code** on line **3"**
  Cela nous indique une erreur de syntaxe : l'utilisation du caractère '>' est inappropriée ici et nous permet de savoir que l'erreur se situe à la 3ème ligne.

```
1  $clt->newDomain('館','Anonymous');
2  $d=$clt->getDomain('館');
3  $d->>newPointOfView('pv');
```

Figure 116 : Exemple d'erreur de syntaxe dans l'appel d'une méthode

- Les erreurs spécifiques à la CLT, qui seront des messages d'erreur liés à l'établissement du système conceptuel. A chaque manipulation du système conceptuel, des vérifications sont faites et permet d'assurer une certaine cohérence du système. Par exemple, si un lien générique entre deux concepts engendre un cycle dans le graphe alors un message d'erreur sera retourné à l'utilisateur et la relation ne pourra être ajoutée dans le système. L'affichage des erreurs propres à la CLT est effectué dans la fenêtre de « log ».

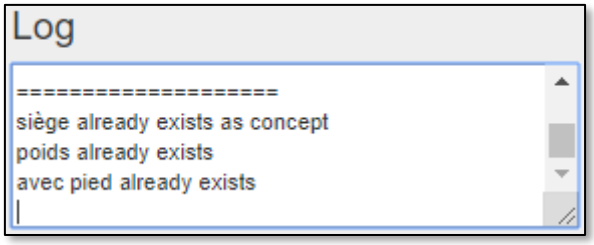


Figure 117 : Exemple de messages d'erreur propres à la CLT

3.3.6 L'IMPLEMENTATION DE LA VUE (L'INTERFACE UTILISATEUR)

L'interface utilisateur comprend deux éléments fondamentaux :

- Une console affichée sous la forme d'une zone de texte permettant d'entrer des commandes de la CLT en utilisant la syntaxe du langage PHP. Celle-ci est accompagnée d'un bouton « submit » ou « d'exécution » qui enverra dans les lignes de commande contenues dans la console au serveur sous la forme d'une requête HTTP[87] pour y être interprétée. La console est implémentée simplement dans un « formulaire HTML ».

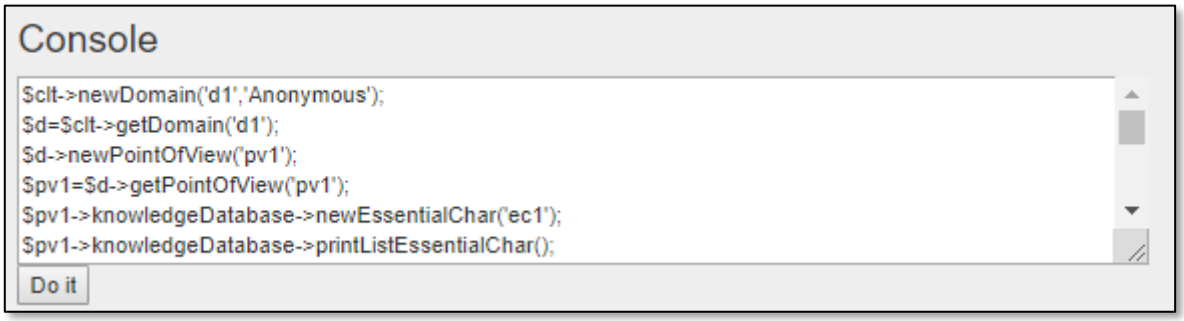


Figure 118 : La console, permettant l'entrée des instructions de la CLT

```
=========================================================================================//CONSOLE
"<h3>Console</h3>
 <textarea id=\"console\" style=\"width:100%;height:25%;\" name=\"console\">$workspace</textarea><br>
 <input type=\"submit\" name=\"submit\" value=\"Do it\"  />
 ";
```

Figure 119 : Implémentation HTML de la console

- La fenêtre de « Log » est utilisée pour afficher le résultat des instructions exécutées par le serveur dans une simple zone de texte HTML.

Figure 120 : Résultat des instructions exécutées

[87] HyperText Transfer Protocol est un protocole de communication client-serveur développé pour le World Wide Web et est utilisé pour échanger des données entre un client http (navigateur internet) et un serveur http (serveur web).

```
<h3>Log</h3>
<textarea id=\"output\" style=\"width:100%;height:25%;\" name=\"output\">$txt_output</textarea>
<br><br>
```

Figure 121 : Implémentation HTML de la fenêtre de "Log"

Ces deux éléments de l'interface (la console et la fenêtre de « log ») sont à la base du dialogue entre l'utilisateur et le système dans l'utilisation de la CLT. La construction d'un système conceptuel résulte de cette interaction, qui implique une entrée des instructions de la CLT *via* la console et une visualisation des résultats *via* la fenêtre des logs. Lorsqu'il s'agit de résultats non textuels, tel qu'un graphe représentant un système conceptuel, l'affichage se fera au-dessus de la fenêtre de log dans la même page html ou séparément.

Afin de rendre plus aisée la construction du système conceptuel et sa manipulation, l'interface utilisateur peut être dotée d'une interface graphique dont les éléments correspondront aux notions essentielles de la CLT : la création de domaines, de points de vue, de concepts, de relations. On retrouvera à travers cette interface la structure générale de la Théorie. Les instructions CLT correspondantes seront générées à partir des éléments de l'interface de l'interface graphique. L'utilisation du bouton « create new domain » avec le champ « com de domaine contenant « d1 » produira l'écriture dans la console la ligne suivante : « $clt->newDomain('d1','Anonymous'); »

Comme nous l'avons vu, l'interface graphique est réalisée simplement à l'aide du langage html. Il sera alors aisé de personnaliser l'interface à l'aide de différents fichiers de style (CSS[88]) en fonction des besoins de l'utilisateur. L'utilisation du langage de description HTML pour la conception de l'interface graphique permet de la modifier facilement.

[88] Cascading Style Sheets, est un langage pour décrire le style des documents HTML. Les standards définissant **CSS** sont publiés par le World Wide Web Consortium (W3C)

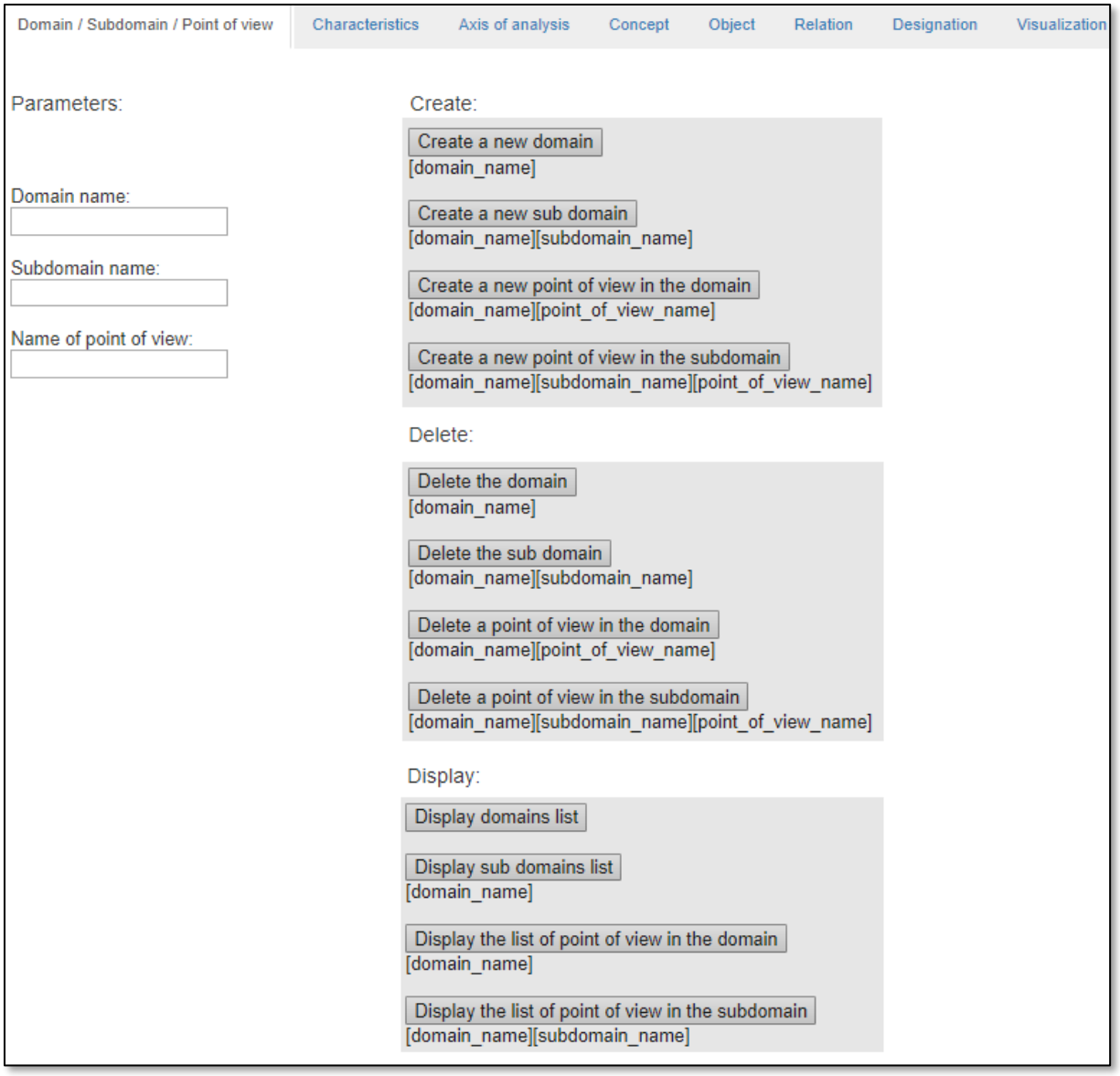
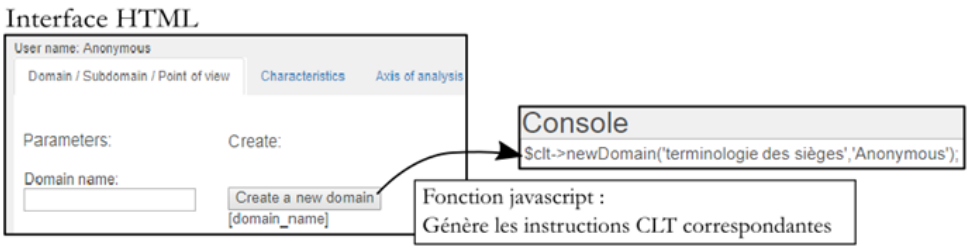


Figure 122 : Exemple d'interface graphique de la CLT en HTML

3.3.7 L'IMPLEMENTATION DU CONTROLEUR (LA GESTION DES EVENEMENTS)

La gestion des évènements liés à l'interface graphique est implémentée en Javascript qui est un langage de programmation orienté « événement » exécuté dans un navigateur web. A chaque action de l'utilisateur correspondra une fonction Javascript qui aura pour rôle de la transcrire en instruction CLT et de l'afficher dans la console.




```
<script type="text/javascript">
//DOMAIN
function create_new_domain(){
    var text = document.getElementById('console');
    var text_tf_domain_name = document.getElementById('tf_domain_name');
    var text_tf_userName = document.getElementById('userName');
    text.value +="\n$clt->newDomain("+"'"+text_tf_domain_name.value+"','"+text_tf_userName.value+"');";
}
```

Figure 123 : La gestion des évènements *via* des fonctions en Javascript

Dans le fichier html contenant l'interface graphique, on retrouvera les fonctions Javascript qui traduiront les actions de l'utilisateur. Dans l'exemple ci-dessus, on concatène à la variable « text » l'instruction CLT correspondant à la création d'un domaine. La fonction de création de domaine nécessite ici deux paramètres : le premier est complété à l'aide d'un champ du formulaire html « nom du domaine », le deuxième est le nom de l'utilisateur qui est entré lors de sa connexion à l'application.

3.3.8 L'IMPLEMENTATION DE FONCTIONS D'ANALYSE DU SYSTEME CONCEPTUEL

A partir de la console utilisateur de la CLT, il est possible de définir des fonctions d'analyse du système conceptuel propres aux besoins de l'utilisateur. Cela peut être une fonction qui affiche les concepts qui n'ont aucune désignation dans un langage donné ou l'implémentation de fonctions de recherche sur un terme ou un nom de concept. A noter que l'implémentation de ce type de fonctions nécessite la connaissance de la syntaxe du langage PHP et la structure du modèle de la CLT, car il s'agit de manipuler les objets de la CLT avec le langage PHP. Le code correspondant à la fonctionnalité voulue peut être intégré à l'interface graphique HTML afin de rendre son utilisation plus facile.

Console

```
$clt->printHtml("<br><b>Liste des concepts n'ayant aucune désignation en français:</b><br>");
foreach( $pv1->knowledgeDatabase->listCLTconcept as $value ){
        $terms=$value->designation->getTerms('français');
        if (strlen($terms)==0) $clt->printHtml("$value->name <br>");
}

$c_siege=$pv1->knowledgeDatabase->getConcept('siège');
if ($c_siege!=null){
      $clt->printHtml("<br><b>Terme en anglais désignant le concept de siège:</b><br>");
      $t=$c_siege->designation->getTerms('anglais');

      $clt->printHtml("$t");
}
```

Do it

Liste des concepts n'ayant aucune désignation en français:
siège 1 personne
siège plusieurs personnes

Terme en anglais désignant le concept de siège:
seat

Figure 124 : Exemples d'implémentation de fonctions d'analyse du système conceptuel

3.3.9 L'IMPLEMENTATION DES FONCTIONS DE VISUALISATION

CLT propose différentes fonctions de visualisation d'une terminologie computationnelle en fonction des besoins de l'utilisateur. Cette visualisation peut prendre la forme d'un glossaire en utilisant la fonction « printListHTMLterms(choix du langage) » d'une instance « knowledge database » qui permet d'afficher la liste des concepts ainsi que les termes associés dans une langue donnée. L'implémentation de ce type de fonction est faite à l'aide de balises HTML pour la représentation des tableaux HTML.

Console

```
//$c->designation->printListHtmlTerms();
$clt->printHtml('<h1 align="center">Glossary</h1>');
$pv->knowledgeDatabase->printListHtmlTerms('English');
```

Do it

127.0.0.1/CLT/user_

Glossary

<馆>	house.establishment.
<报馆>	newspaper office.
<报>	to announce.to inform.report.newspaper.recompense.revenge.
<宾馆>	guesthouse.
<宾>	visitor.guest.
<殡仪馆>	the undertaker s.funeral parlor.
<博物馆>	museum.
<博物>	natural science.
<餐馆>	restaurant.
<餐>	eat meal.

29

Figure 125 : Visualisation sous la forme d'un glossaire dans une langue donnée

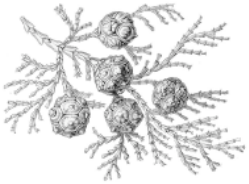
Il peut être aussi utile d'avoir une visualisation sous forme de graphe du système conceptuel, pour cela deux fonctions (méthodes de la classe « Knowledge database ») ont été implémentées : « drawGraph(largeur, hauteur) » et « drawGraphTerms(largeur, hauteur, choix du langage) ». La première méthode permet d'afficher le système conceptuel présent dans l'instance « knowledge database » en affichant les noms des concepts et leurs relations. La deuxième est similaire mais affiche un réseau de termes qui peut s'avérer être différent de la représentation du système conceptuel, les termes dénotant les concepts seront affichés dans un langage donné. L'implémentation de ces fonctions d'affichage est faite à l'aide de la librairie Javascript « Springy.js[89]» qui propose une collection de fonctions pour le tracé de graphes orientés[90] dans une page html. L'algorithme à la base de « Springy.js » est de type « force directed » ce qui signifie qu'il utilise des lois de la physique pour obtenir un rendu esthétique du graphe.

[89] Springy est un « framework » Javascript sous licence MIT pour représenter et afficher des graphes orientés. http://getspringy.com/, consulté le 03/09/2018.

[90] Dans la théorie des graphes, un graphe est orienté lorsque ses arêtes sont définies par leur origine et leur extrémité.

Springy.js

A force directed graph layout algorithm in JavaScript.



Demo | Download | Getting started | GitHub | Contact

What is Springy?

Springy is a force directed graph layout algorithm.

So what does this "force directed" stuff mean anyway? *Excellent question!*

It means that springy uses some real world physics to try and figure out how to show a network graph in a way that looks good.

Here's an example:

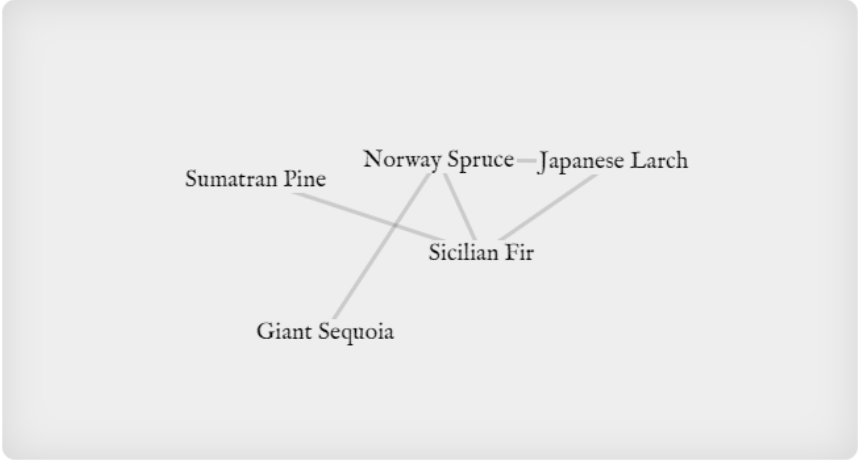


Figure 126 : http://getspringy.com/ consulté le 03/09/2018

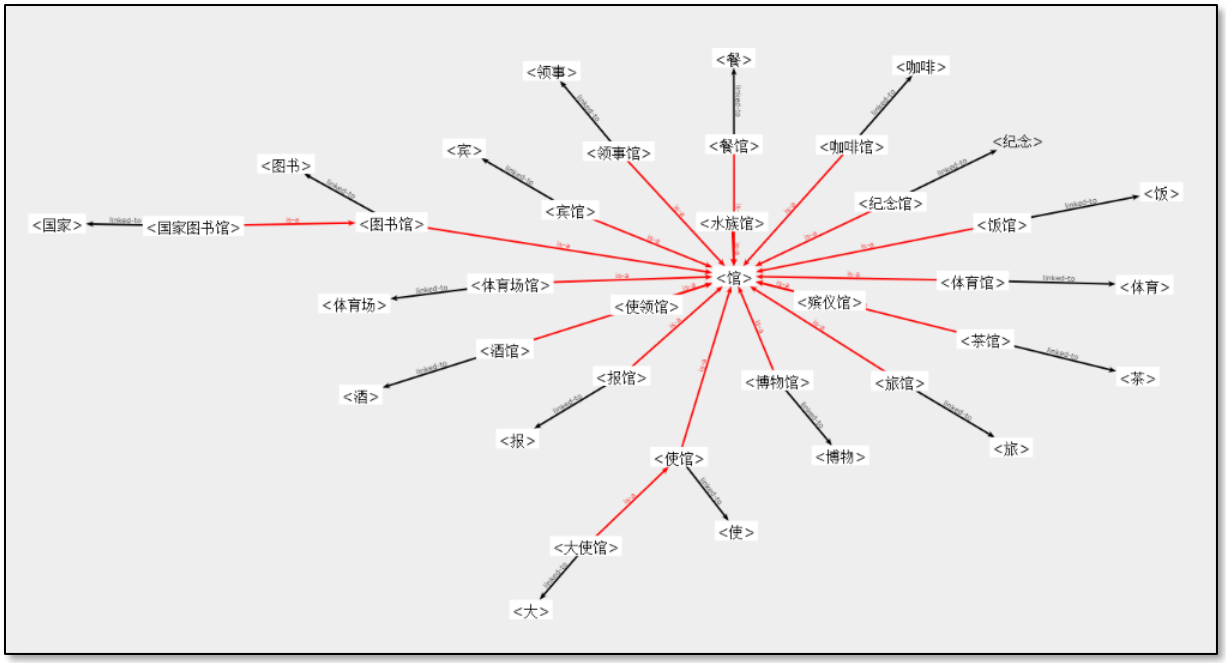


Figure 127 : La visualisation d'un système conceptuel réalisé avec la CLT affiché avec « Springy.js »

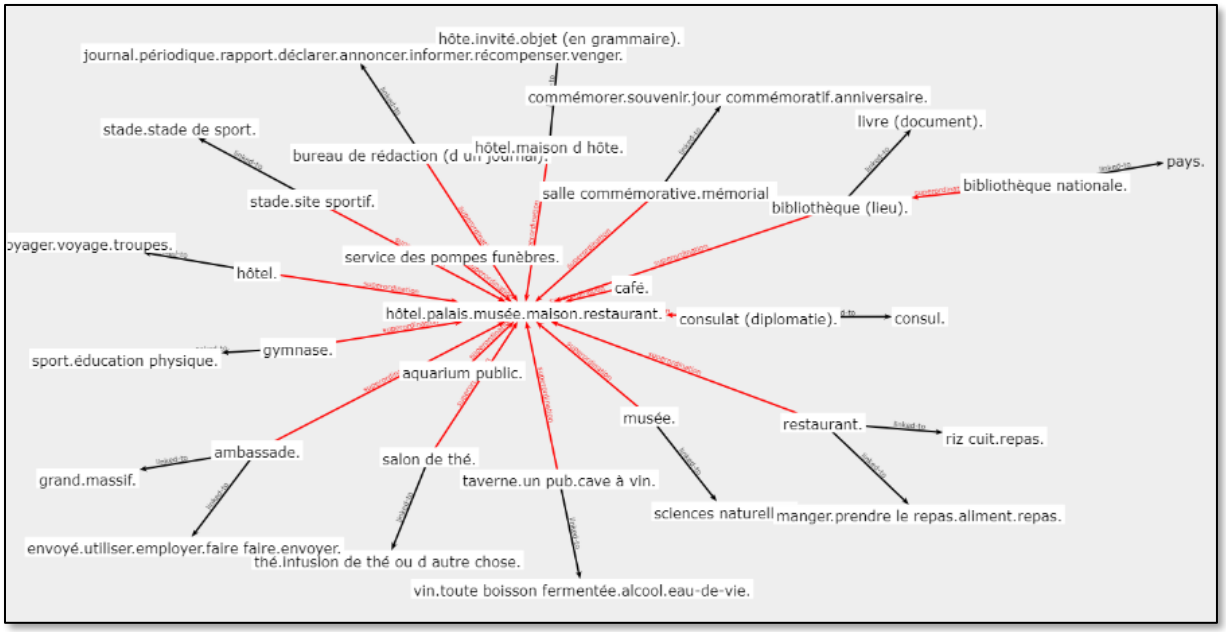


Figure 128 : Un réseau de termes en français de la CLT affiché avec « Springy.js »

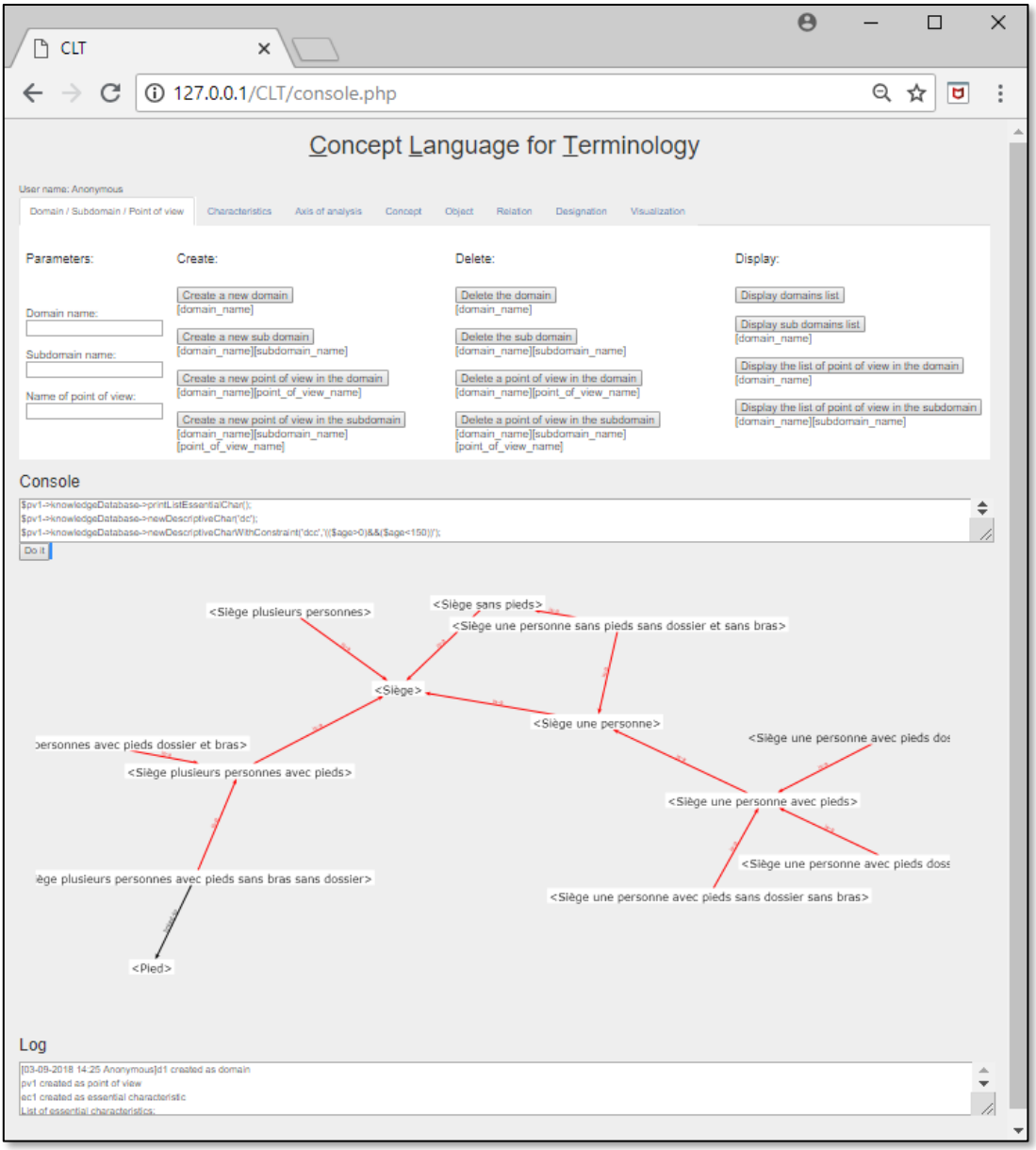


Figure 129 : L'intégration de "Springy.js" pour visualiser un système conceptuel dans l'environnement de la CLT

L'utilisateur a la possibilité de visualiser le résultat de ses requêtes en dehors de l'environnement de la CLT, dans une page html indépendante qui ne contiendra que ce qu'il souhaite afficher : par exemple un glossaire, la représentation du système conceptuel...

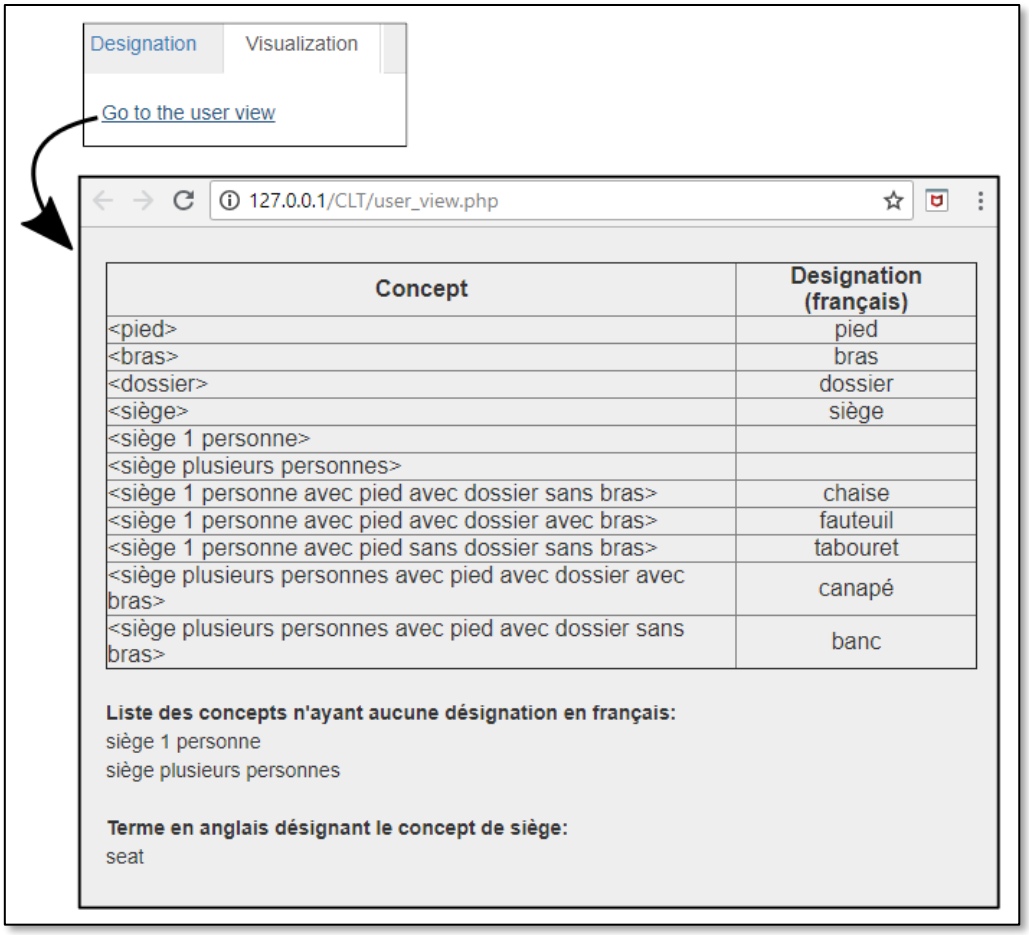


Figure 130 : Une visualisation des résultats indépendante de l'environnement de la CLT

3.3.10 LES PROBLEMES DE SECURITE LIES A L'UTILISATION DE LA FONCTION PHP « EVAL() »

Les instructions CLT reposant sur le langage de programmation PHP sont exécutées grâce à la fonction « eval() ». Cette fonction permet d'interpréter tout programme PHP, l'utilisateur peut potentiellement utiliser toutes les fonctions PHP du langage, ce qui peut engendrer des problèmes de sécurité côté serveur. Il est possible par exemple d'exécuter des fonctions de manipulation de fichiers. Afin de prévenir ces problèmes de sécurité, il est possible de restreindre l'ensemble des fonctions PHP utilisables en exécutant seulement les fonctions jugées utiles à la CLT.

3.4 FORMATS D'ECHANGE

Même si le langage CLT peut être utilisé comme format d'échange et être interprété par le langage PHP disposant des classes de la CLT, le système propose des fonctions d'export du système conceptuel vers d'autres formats tels que CmapTools(CSV), RDF, RDFS et OWL. Certains formats ne permettent pas de représenter toutes les notions de la CLT, par exemple les formats utilisés par CmapTools et RDF ne possèdent pas la notion de relations génériques, mis à part le nom de la relation aucune distinction ne sera faite par rapport aux autres relations. Il faudra ainsi, selon le format d'échange choisi s'attendre à des pertes d'information car chaque format repose sur un modèle différent.

3.4.1 FORMAT D'EXPORT POUR CMAPTOOLS

Un des formats de fichier utilisé par CmapTools a la structure suivante : « Nom du concept de départ, nom du lien, nom du concept d'arrivée », le système conceptuel est donc exporté avec la même structure en faisant apparaître les concepts et leurs relations. Comme on l'a vu dans la partie représentation des caractéristiques essentielles dans CmapTools (partie 2.2.3.2.1), il est possible de représenter par des nœuds et des liens supplémentaires les notions de caractéristiques descriptives et essentielles pour chaque concept mais CmapTools ne permettra pas d'exploiter ce type d'information. A noter que Cmaptools a simplement pour vocation de représenter graphiquement une carte de concepts. Quant aux désignations des concepts, elles peuvent être ajoutées en créant un lien « désignation » entre les concepts et des « nœuds » où seraient affichés les termes. On fera le choix ici d'exporter uniquement les noms des concepts et leurs relations.

```
==============================================================
Export to CmapTools
==============================================================
<siège 1 personne>        is-a        <siège>
<siège 1 personne>        is-a        <siège>
<siège plusieurs personnes>  is-a        <siège>
<siège plusieurs personnes>  is-a        <siège>
<siège 1 personne avec pied avec dossier sans bras>  is-a        <siège 1 personne>
<siège 1 personne avec pied avec dossier sans bras>  is-a        <siège 1 personne>
<pied>     is-part-of       <siège 1 personne avec pied avec dossier sans bras>
<pied>     is-part-of       <siège 1 personne avec pied avec dossier sans bras>
<dossier>        is-part-of       <siège 1 personne avec pied avec dossier sans bras>
<dossier>        is-part-of       <siège 1 personne avec pied avec dossier sans bras>
<siège 1 personne avec pied avec dossier avec bras>  is-a        <siège 1 personne>
<siège 1 personne avec pied avec dossier avec bras>  is-a        <siège 1 personne>
<pied>     is-part-of       <siège 1 personne avec pied avec dossier avec bras>
<pied>     is-part-of       <siège 1 personne avec pied avec dossier avec bras>
<dossier>        is-part-of       <siège 1 personne avec pied avec dossier avec bras>
<dossier>        is-part-of       <siège 1 personne avec pied avec dossier avec bras>
```

Figure 131 : Le système conceptuel des sièges exporté au format CmapTools

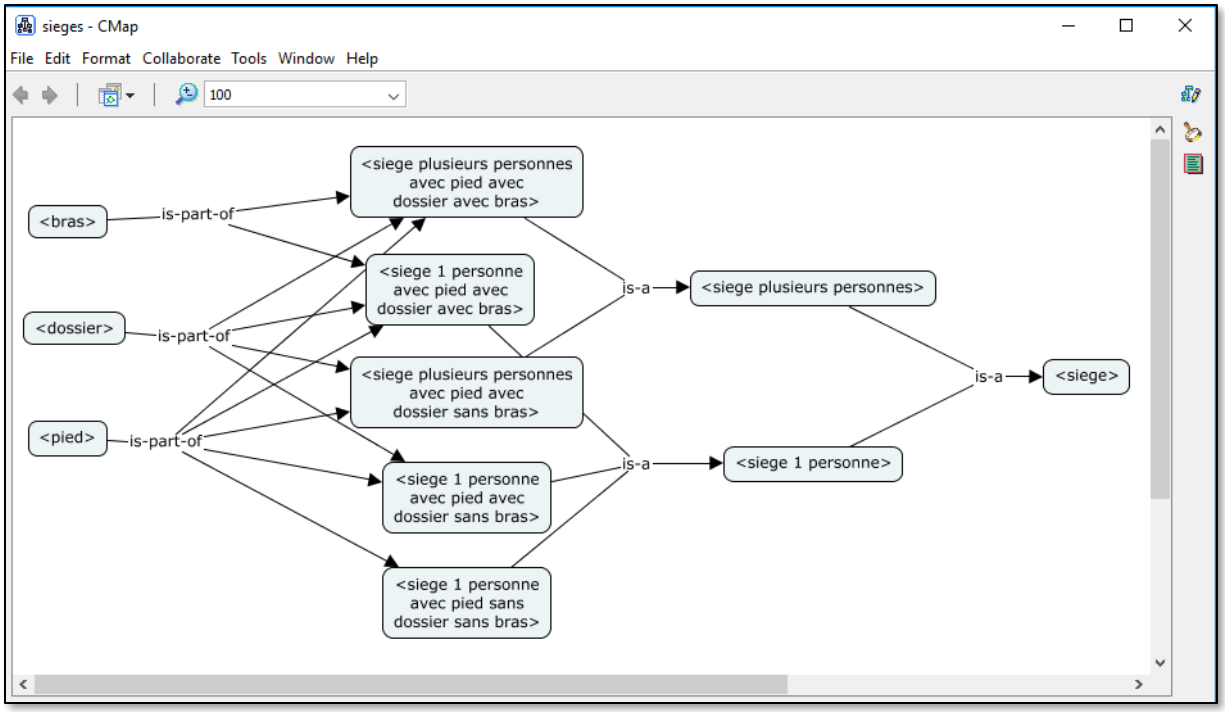


Figure 132 : Import du système conceptuel des sièges dans CampTools

3.4.2 FORMAT D'EXPORT POUR RDF

RDF possède la même structure que le format de CmapTools, il s'agit d'une structure de représentation d'un graphe orienté où les nœuds sont des concepts et les liens portent les noms des relations entre les concepts. La principale différence réside dans le fait que RDF repose sur XML et possède des identificateurs de ressources. Si dans notre système conceptuel nous avons : le concept <siege 1 personne> est une sorte de <siege>, cette information serait exportée de la manière suivante en RDF :

```
<rdf:Description rdf:about="http://rdf/clt/siege.1.personne"/>
  <ctl:is-a rdf:resource="http://rdf/clt/siege"/>
</rdf:Description>
```

Le format RDF est représenté par des triplets (sujet, prédicat, objet), le « sujet » porte sur la ressource à décrire, ici : le concept <siege 1 personne>, le « prédicat » représente la propriété applicable à cette ressource, dans notre cas : la relation « est un » et enfin « l'objet » qui représente une donnée ou une autre ressource, ici : le concept <siege>. RDF aura tout de même l'avantage par rapport au format CmapTools de pouvoir être interrogé par un langage de requête tel que SPARQL[91].

[91] Acronyme de **P**rotocol **a**nd **R**DF **Q**uery **L**anguage, est un langage de requête et de protocole qui permet de rechercher ou de manipuler des données RDF. Standard du W3C, il est considéré comme une technologie clef du web sémantique.

3.4.3 FORMAT D'EXPORT POUR RDFS

RDFS garde la même structure que RDF mais est enrichie d'informations sémantiques supplémentaires, ce qui nous permettra de conserver les liens génériques et de définir le domaine de départ et d'arrivée d'une relation. Avec ce format, les concepts de la CLT seront représentés comme des classes RDFS et l'on pourra conserver une hiérarchie entre les classes grâce à la notion « rdfs :subClassOf ». Par exemple « <Siège 1 personne> est un <siège> » sera exportée en RDFS de la manière suivante :

```
<rdfs:Class rdf:about="http://rdfs/clt/siege.1.personne">
   <rdfs:subClassOf>
     <rdfs:Class rdf:about="http://rdfs/clt/siege"/>
   </rdfs:subClassOf>
 </rdfs:Class>
```

Chaque caractéristique essentielle sera représentée par une classe, comme on l'a vu dans Protégé dans la représentation des caractéristiques essentielles.

3.4.4 FORMAT D'EXPORT POUR OWL

De façon similaire à RFDS, il est possible d'exporter les concepts sous la même structure. Ci-dessous les caractéristiques essentielles du concept <Siège_une_personne_sans_pieds_sans_dossier_et_sans_bras> sont représentées par les classes "Siège_une_personne", "Siège_sans_pieds" , "sans_bras" , "sans_dossier " .

```
<owl:Class rdf:about="Siège_une_personne_sans_pieds_sans_dossier_et_sans_bras">

      <rdfs:subClassOf rdf:resource="Siège_une_personne"/>
      <rdfs:subClassOf rdf:resource="Siège_sans_pieds"/>
      <rdfs:subClassOf rdf:resource="sans_bras"/>

      <rdfs:subClassOf rdf:resource="sans_dossier"/>

</owl:Class>
```

Contrairement à RDFS, OWL offre aussi la possibilité de définir une classe en utilisant des opérateurs logiques, c'est-à-dire que le concept <Siège_une_personne_sans_pieds_sans_dossier_et_sans_bras> pourrait aussi être exporté de la manière suivante tel que le fait Protégé :

```
<Class IRI="#Siège_une_personne_sans_pied_sans_dossier_sans_bras"/>

      <ObjectIntersectionOf>
         <Class IRI="#Siège_sans_bras"/>
         <Class IRI="#Siège_sans_dossier"/>
         <Class IRI="#Siège_sans_pied"/>
         <Class IRI="#Siège_une_personne"/>
      </ObjectIntersectionOf
```

3.5 APPLICATION : OPERATIONNALISATION DE DICTIONNAIRES

Il sera question ici d'un exemple d'application mettant en œuvre la CLT à travers son environnement portant sur l'opérationnalisation d'un dictionnaire multilingue pour un domaine donné. Le système conceptuel est construit selon le principe de formation des termes chinois détaillé dans la partie « C.2.3.4.1 Le terme chinois et sa formation ». Cette opérationnalisation consiste à convertir les données du dictionnaire en instructions du langage CLT.

Ici, nous prendrons pour exemple l'opérationnalisation des dictionnaires CEDICT[92] (chinois / anglais) et CFDICT[93] (chinois / français). Il s'agit de dictionnaires dits généraux et nous allons nous intéresser qu'à un domaine particulier : les établissements (hôtel, bibliothèque, ambassade etc.). En se basant sur le principe de formation des termes chinois détaillé dans la partie « C.2.3.4.1 Le terme chinois et sa formation », nous allons sélectionner les termes chinois du dictionnaire finissant par le caractère désignant les établissements, soit le caractère « 馆 ». A noter que les termes chinois constituent les entrées du dictionnaire, pour chaque entrée nous aurons la désignation en anglais contenue dans le dictionnaire CEDICT et la désignation en français dans le dictionnaire CFDICT. Une entrée du dictionnaire en chinois sera considérée ici comme un nom de concept et aura des désignations en chinois, chinois traditionnel, pinyin, français et anglais.

```
1 # CEDICT 25 August 2007; Copyright 2007; -*- coding: utf-8-dos -*-; Entries: 44782
2 呵 呵 [a1] /(phonetic particle)/
3 啊 啊 [a1] /(interj.)/ah/
4 錒 锕 [a1] /actinium/
```

Figure 133 : Extrait du dictionnaire CEDICT

3.5.1 CONVERSION DES DICTIONNAIRES CEDICT ET CFDICT EN INSTRUCTIONS CLT

Nous avons vu qu'un terme chinois était suffisamment transparent[94] pour en extraire les caractéristiques essentielles du concept qu'il dénote. Pour établir le système conceptuel, les instructions CLT seront générées à partir des termes chinois à l'aide d'un programme Java (Figure 135). En entrée, le programme prendra un fichier contenant les informations des dictionnaires CEDICT et CFDICT, on retrouvera pour chaque ligne du fichier un terme chinois et ses désignations. Pour chaque ligne, un nom de concept a été ajouté, il correspond ici au terme chinois. C'est à partir du nom de concept que des liens « épistémologiques »[95] seront extraits pour établir le système conceptuel du domaine des établissements. Par exemple le nom du concept <宾馆> est formé par les

[92] CEDICT est un projet initié par Paul Denisowski en 1997 qui a pour but de proposer un dictionnaire complet du chinois vers l'anglais.

[93] CFDICT est l'équivalent de CEDICT à la différence qu'il s'agit d'un dictionnaire du chinois vers le français.

[94] Répétons à nouveau qu'un terme n'est pas un nom de concept, mais qu'un terme, afin d'être le plus transparent possible (au sens de l'ISO), peut se calquer avec profit sur le nom du concept qu'il dénote.

[95] ici en relation avec les caractéristiques essentielles

deux caractères chinois : « 宾 » (qui désigne un invité) et « 馆 » (qui désigne un établissement) qui sont ici à la fois des noms de concept et des caractéristiques essentielles du concept « <宾馆> ». Le programme, génèrera pour le concept « <宾馆> » les instructions CLT de définition du concept, ses caractéristiques essentielles « /宾/,/馆/ » et les relations qu'il entretient avec les concepts « <宾> » et « <馆> ». Le même procédé est appliqué sur tous les concepts du fichier d'entrée. Le programme en question est basé sur un algorithme développé pour le logiciel « Treewords » (se référer à la partie : « 2.2.3.3.2 Le système qui régit la formation des termes chinois est-il adapté pour nommer les concepts ? ») qui permet de segmenter un terme chinois en deux parties : « déterminant – déterminé ».

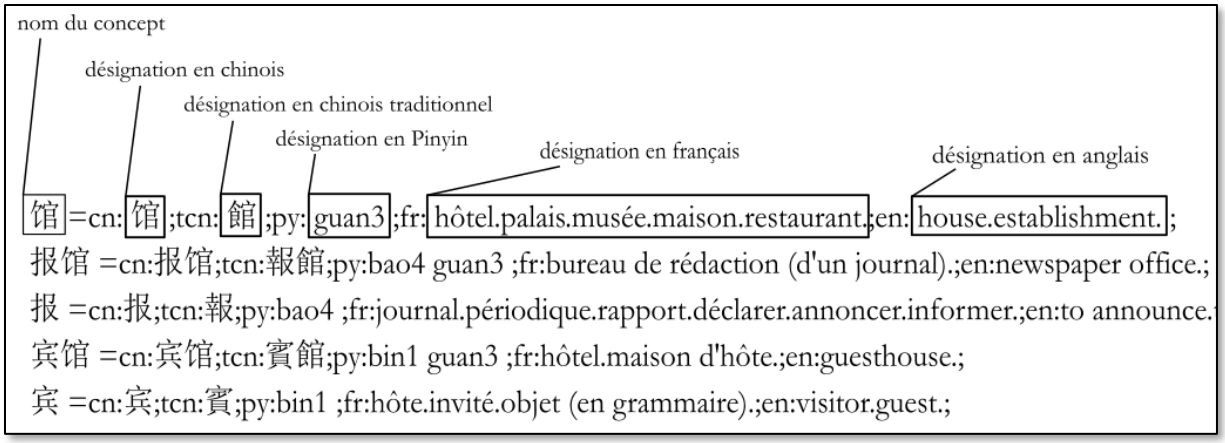


Figure 134 : Extrait du fichier d'entrée du programme contenant les noms des concepts et leurs désignations

```
$pv->knowledgeDatabase->newConcept('宾馆');
$c=$pv->knowledgeDatabase->getConcept('宾馆');
$c->designation->newTerms('Chinese','宾馆');
$c->designation->newTerms('French','hôtel.maison d hôte.');
$c->designation->newTerms('English','guesthouse.');
$c->designation->newTerms('Trad.Chinese','賓館');
$pv->knowledgeDatabase->newEssentialChar('宾');
$ec=$pv->knowledgeDatabase->getEssentialChar('宾');
$c->addEssentialChar($ec);
$pv->knowledgeDatabase->newEssentialChar('馆');
$ec=$pv->knowledgeDatabase->getEssentialChar('馆');
$c->addEssentialChar($ec);
$pv->knowledgeDatabase->newRelation($c1,'is-a',$c2);
$c1=$pv->knowledgeDatabase->getConcept('宾馆');
$c2=$pv->knowledgeDatabase->getConcept('馆');
$pv->knowledgeDatabase->newRelation($c1,'linked-to',$c2);
$c1=$pv->knowledgeDatabase->getConcept('宾馆');
$c2=$pv->knowledgeDatabase->getConcept('宾');
```

Figure 135 : Extrait du fichier sortie, génération des instructions CLT pour le concept "<宾馆>" (hôtel)

Les instructions CLT générées seront ensuite exécutées pour établir le système conceptuel « des établissements » avec un point de vue « chinois » puisque les noms des concepts sont formés en utilisant des termes chinois. Le programme mis au point ici, fonctionne comme une application d'export de dictionnaires dont les entrées sont des termes chinois vers une terminologie

computationnelle en utilisant le modèle de la CLT. Il sera possible alors de générer une définition formelle pour chaque concept et visualiser le système conceptuel des établissements.

3.5.2 VISUALISATION DE LA TERMINOLOGIE DES ETABLISSEMENTS

3.5.2.1 AFFICHAGE DE LA DEFINITION FORMELLE D'UN CONCEPT

```
Formal definition of <宾馆>: <hôtel>
==============================================================
Essential characteristics:
/宾/ /hôte/
/馆/ /établissement/
--------------------------------------------------------------
Descriptive characteristics:
--------------------------------------------------------------
Relations:
<宾馆> is-a <馆> <hôtel> is-a <établissement>
<宾馆> linked-to <宾> <hôtel> est lié à <hôte>
==============================================================
```

Figure 136 : Définition formelle du concept <宾馆>

3.5.2.2 GENERATION D'UN GLOSSAIRE EN ANGLAIS

Concept	Designation (English)
<馆>	house.establishment.
<报馆>	newspaper office.
<报>	to announce.to inform.report.newspaper.recompense.revenge.
<宾馆>	guesthouse.
<宾>	visitor.guest.
<殡仪馆>	the undertaker s.funeral parlor.
<博物馆>	museum.
<博物>	natural science.
<餐馆>	restaurant.
<餐>	eat.meal.
<茶馆>	teahouse.
<茶>	tea.tea plant.
<大使馆>	embassy.
<大>	big.huge.large.major.great.wide.deep.oldest.eldest.
<饭馆>	restaurant.
<饭>	food.cuisine.cooked rice.meal.
<国家图书馆>	national library.
<国家>	country.nation.
<纪念馆>	memorial hall.commemorative museum.
<纪念>	to commemorate.to remember.
<酒馆>	(n) barroom.pub.public house.wineshop.
<酒>	wine.liquor.spirits.
<咖啡馆>	cafe.coffee shop.
<咖啡>	coffee.
<领事馆>	consulate.
<领事>	consul.
<旅馆>	hotel.
<旅>	trip.travel.
<体育场馆>	stadium.
<体育场>	stadium.
<体育馆>	gym.gymnasium.stadium.
<体育>	physical culture.physical training.sports.
<使领馆>	embassy and consulate.
<使馆>	consulate.diplomatic mission.
<使>	to make.to cause.to enable.to use.to employ.messenger.
<图书馆>	library.
<图书>	books (in a library or bookstore).
<水族馆>	aquarium.

Figure 137 : Génération d'un glossaire en anglais

3.5.2.3 AFFICHAGE DU SYSTEME CONCEPTUEL

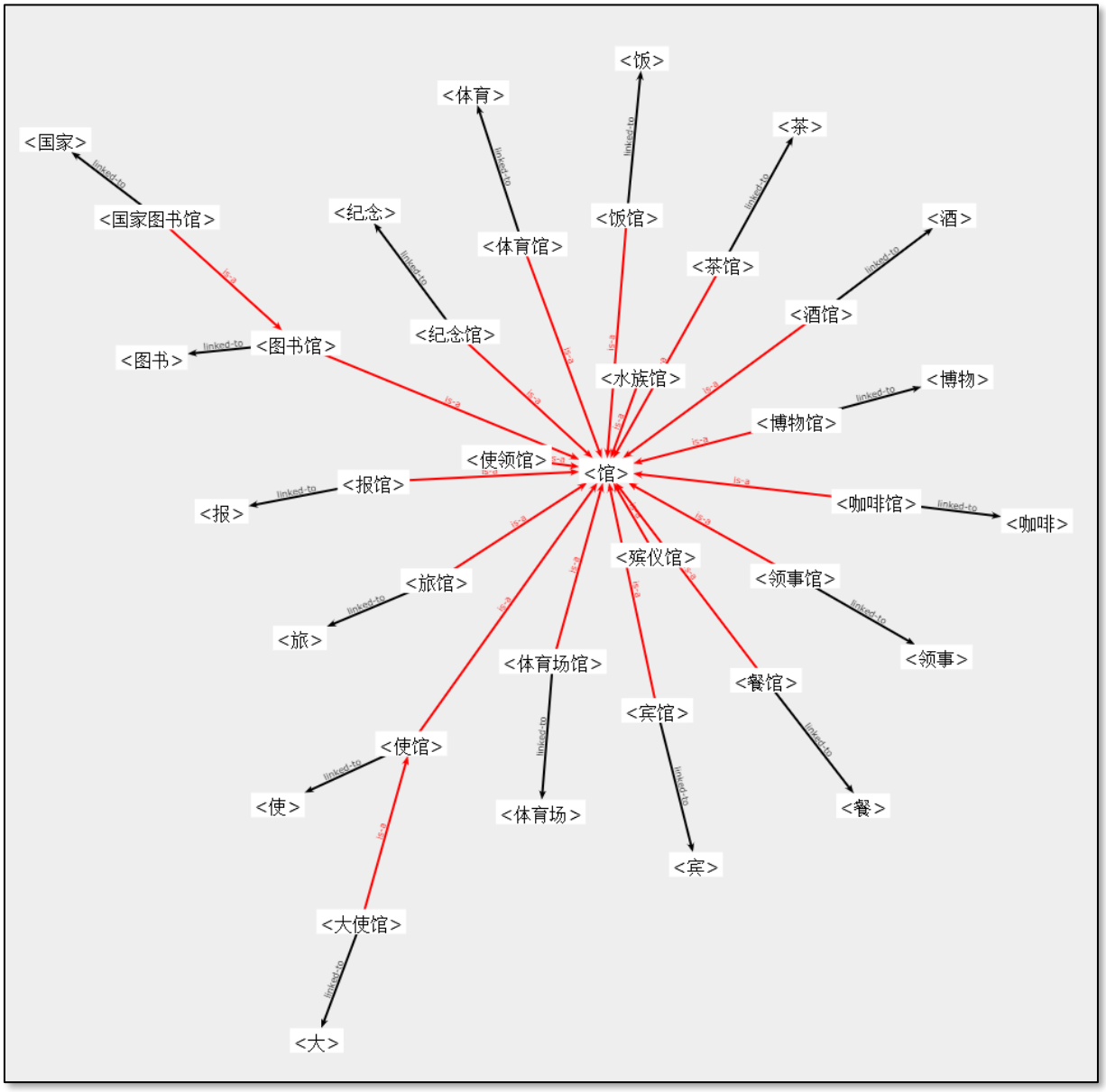


Figure 138 : Représentation du système conceptuel des établissements

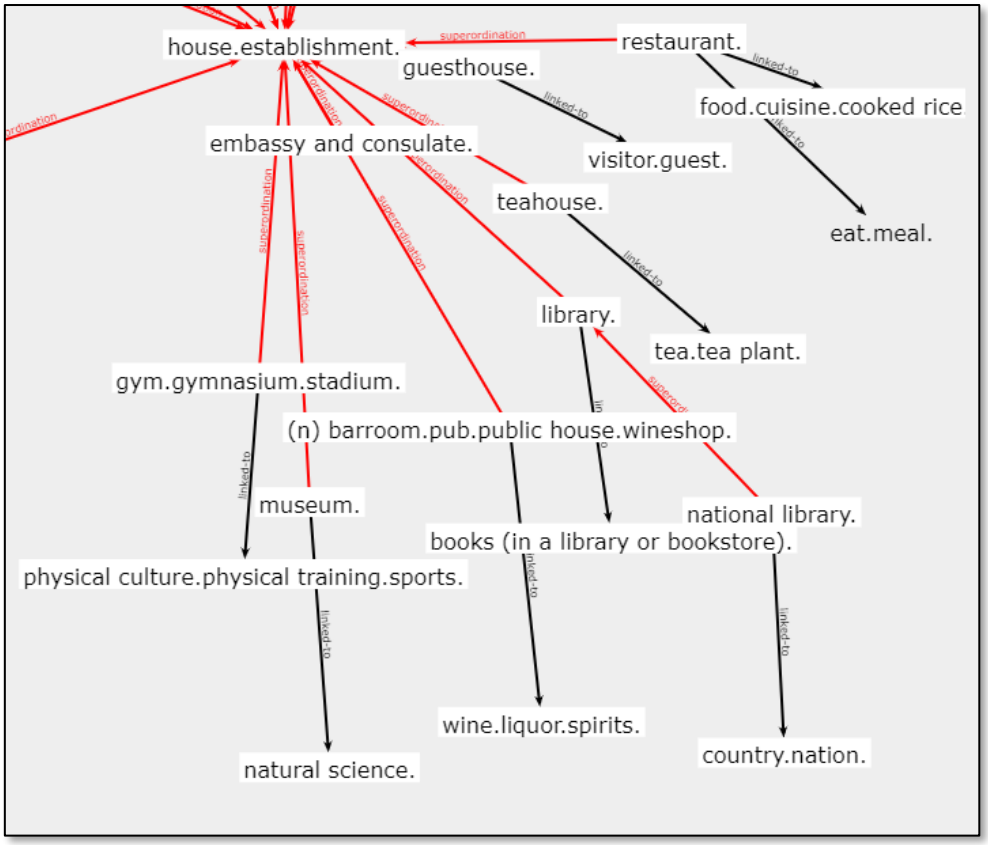


Figure 139 : Réseau de termes en anglais généré à partir du système conceptuel du domaine des établissements

3.5.3 CONCLUSION

En considérant les termes chinois comme étant une traduction en langue des noms bien formés de concept, on a pu, à partir des noms de concept, reconstituer un système conceptuel en générant les instructions CLT correspondantes. En effet, le nom du concept permet ici de le situer par rapport aux autres concepts dans la mesure où nous considérons que le terme chinois est composé de caractères (un ou plusieurs) désignant des caractéristiques essentielles qui définissent le concept. Cette considération est d'autant plus pertinente lorsqu'il s'agit d'un domaine spécialisé ou technique.

Ce type d'opérationnalisation basée sur la CLT pourrait ouvrir la voie à la conversion de dictionnaires spécialisés dont les entrées sont des termes chinois vers des terminologies computationnelles. Cela permettra entre autres de vérifier la cohérence de la définition d'un terme ou de s'assurer du bien-fondé du choix d'un terme.

CONCLUSION

Ce travail se situe dans le cadre de la Terminologie dite « classique » telle que l'ISO la définit principalement à partir de ses deux normes transverses ISO 1087-1 et ISO 704. Même si cette Terminologie peut être sujette à critiques, en particulier du point de vue d'une Terminologie descriptive, elle demeure importante dans sa démarche prescriptive pour de nombreux domaines (en fait, les deux approches sont importantes et doivent être prises en compte). De plus, elle correspond à la Terminologie telle que la pratiquent de nombreux terminologues pour lesquels la définition du terme est avant tout une définition de chose (c'est-à-dire une définition du concept dénoté par le terme) : un terme est une désignation verbale du concept.

A cela nous devons prendre en compte la nécessaire opérationnalisation des terminologies, aujourd'hui devenue incontournable, à des fins de traitement de l'information comme les moteurs de recherche sémantiques multilingues, la gestion des connaissances, les dictionnaires (encyclopédies) de spécialité, etc. Cette opérationnalisation repose sur une représentation informatique du système conceptuel qui compose toute terminologie et pour laquelle les ontologies au sens de l'Ingénierie des Connaissances constituent une des perspectives les plus prometteuses.

Même si les normes ISO 1087-1 et ISO 704 n'ont pas pour vocation de définir une théorie du concept qui soit informatisable – le principal objectif de la Terminologie au sens de l'ISO demeure la communication entre humains – les problèmes soulevés pour en proposer un modèle computationnel a permis de mettre en évidence un certain nombre « d'incohérences » du point de vue logique, mais aussi épistémologique.

A cela s'ajoute le fait que les théories sur lesquelles reposent les ontologies au sens de l'Ingénierie des Connaissances (nous pensons particulièrement à l'environnement Protégé) sont difficilement directement applicables à la Terminologie de l'ISO. Indépendamment du fait qu'il est difficile de demander à un expert d'un domaine de définir ses concepts comme restrictions de rôles, la théorie du concept sur laquelle reposent ces ontologies demeure trop éloignée de celle utilisée en Terminologie classique.

C'est à partir de ce constat qu'a été proposée une Théorie du Concept et un langage associé nommé CLT (Concept Language for Terminology) qui, tout en respectant dans la mesure du possible les principes de la Terminologie, pourra donner lieu à des représentations computationnelles du système conceptuel. Ce modèle vise à répondre aux critiques émises à l'encontre de la Théorie du Concept de l'ISO quant à son opérationnalisation. Ainsi, la notion d'individu est prise en compte afin d'introduire la nécessaire distinction entre objet (sujet) et concept (prédicat). Le concept pouvant porter sur un ensemble quelconque d'objets, que cet ensemble comporte plusieurs objets (satellites artificiels de la Terre), un seul sans confondre cet ensemble constitué d'un seul objet avec l'objet lui-même (satellite naturel de la Terre), ou aucun (satellite naturel de la Lune). Ce modèle implémente également la notion de caractéristique essentielle, notion fondamentale s'il en est, de la Terminologie, ainsi que celle de caractéristique descriptive afin de distinguer ce qui essentiel de ce qui est contingent. A cela s'ajoute la possibilité d'associer aux relations des propriétés (réflexivité, symétrie, transitivité) ainsi qu'une « signature » précisant la nature des objets liés par les relations.

Cette représentation du concept ouvre de nouvelles perspectives pour le travail terminologique lui-même. En explicitant le concept, elle permet de distinguer sans les confondre ce qui relève de la connaissance de ce qui relève des discours sur la connaissance. Elle permet également d'aborder le problème de l'harmonisation de terminologies, sujet de la norme ISO 860 dont la révision reste encore à faire, au travers de l'alignement de leurs ontologies respectives. On peut ainsi définir une relation de mise en correspondance entre termes sur la base d'une mise en correspondance des concepts désignés par ces termes.

Le modèle proposé, tout en conservant certains principes parmi les plus importants de la Terminologie dite « classique » a été validée par une implémentation du langage de la CLT dans un langage informatique interprété permettant une construction incrémentale du système conceptuel. Par ailleurs, L'architecture du système sur lequel l'implémentation de la CLT a été réalisée permet d'envisager la création d'onto-terminologies (des terminologies dont le système conceptuel repose sur un langage formel) de manière collaborative tout en conservant une trace de chaque instruction donnée par les utilisateurs du langage.

Si le principal objectif de ce travail a été de montrer la nécessité et la faisabilité de disposer d'une théorie du concept dédiée à la Terminologie dite « classique » qui puisse donner lieu à un modèle computationnel, il ouvre également la voie à de nouvelles perspectives et de nouveaux travaux de recherche. Citons, sans ordre de priorité, celui d'une méthodologie outillée adaptée à la construction de systèmes conceptuels centrés sur la notion de caractéristique essentielle, celui d'interfaces spécifiques (combinant par exemple le langage CLT avec une interface conforme à la norme ISO 24156). C'est aussi, nous l'avons mentionné, l'étude de l'harmonisation des termes et des concepts sur la base de l'alignement des ontologies prenant en compte les spécificités de notre théorie du concept et en particulier de la mise en correspondance d'ensemble de caractéristiques essentielles. Enfin, à une époque où la notion de travail collaboratif prend une place importante, c'est la perspective de la mise en place, la CLT en donne les moyens en associant à chaque opération et à chaque information des métadonnées (auteur et dates), d'un environnement collaboratif de construction de systèmes conceptuels dédiés aux experts.

ANNEXES

ANNEXE A : TABLEAUX DES CORRESPONDANCES ENTRE LA NORME ISO 1087-1 ET LA NOTATION ADOPTEE DANS LA NORME ISO 24156-1

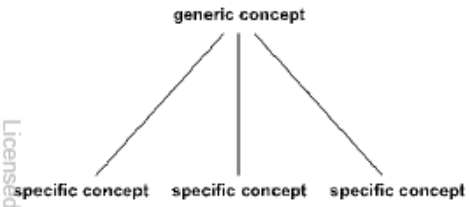
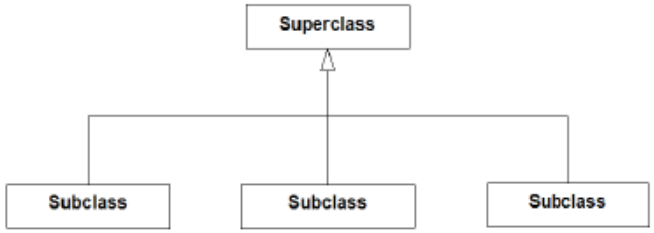
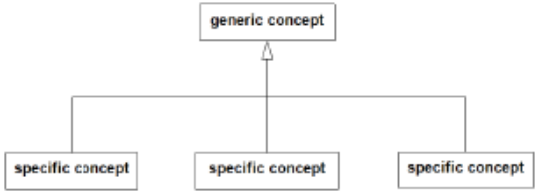
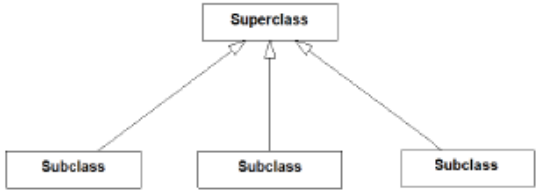
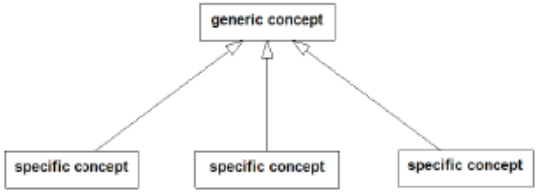
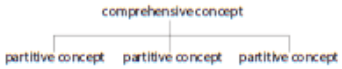
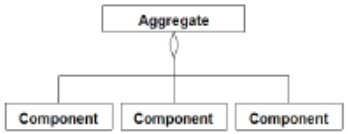
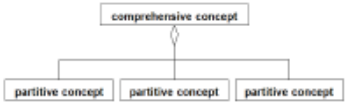
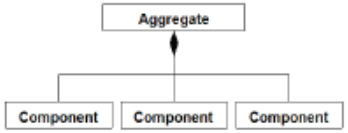
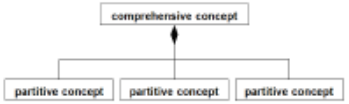
Table A.1 — Table of correspondence between ISO 1087-1 concepts and their adopted symbols in the ISO 24156-1 user-defined UML profile

ISO 1087-1	UML (ISO/IEC 19505-2)	ISO 24156-1 user-defined UML profile
associative relation concept 1 ⟷ concept 2	association Class 1 — name ▶ — Class 2 Class 1 — name ▶ — Class 2	associative relation concept 1 — name ▶ — concept 2 concept 1 — name ▶ — concept 2
	multiple association Class 1 — name ▶ ◀ name — Class 2	multiple associative relation concept 1 — name ▶ ◀ name — concept 2
concept and characteristics	Class	Concept
Note: General and individual concepts are represented in the same way.	attribute 1 = value 1.1 attribute 2 = value 2.1	attribute 1 = value 1.1 attribute 2 = value 2.1
designation	class name ClassName	designation designation
generic relation (single) generic concept — specific concept	generalization Superclass ◁ Subclass	generic relation generic concept ◁ specific concept

Licensed to: Lookkaert, Hendrik Dr
Downloaded:2015-11-19
Single user licence only, copying and networking prohibited.

© ISO 2014 – All rights reserved

Table A.1 *(continued)*

ISO 1087-1	UML (ISO/IEC 19505-2)	ISO 24156-1 **user-defined UML profile**
generic relation (multiple) 	generalization (shared target style) 	generic relation (shared target style) 
	generalization (separate target style) 	generic relation (separate target style) 
partitive relation 	aggregation 	partitive relation (aggregation) 
	composition 	partitive relation (composition) 

Licensed to:Kockaert, Hendrik Dr
Downloaded:2015-11-19
Single user licence only, copying and networking prohibited

© ISO 2014 – All rights reserved

ANNEXE B : CAS D'UTILISATION DE LA CLT : CONSTRUCTION DE LA TERMINOLOGIE DES SIEGES

Cette annexe décrit l'exemple classique de la terminologie des sièges.

Afin d'établir la terminologie des sièges avec la CLT, on dispose tout d'abord sous la forme d'un tableau, des concepts impliqués, leurs caractéristiques essentielles organisées en axe d'analyse, leurs caractéristiques descriptives et les désignations des concepts.

Concept	Axe d'analyse		Axe d'analyse		Axe d'analyse		A d'analyse		Désignation en anglais	Désignation en français
	une personne	plusieurs personnes	avec pied	sans pied	avec dossier	sans dossier	avec bras	sans bras		
<Siège>									« seat »	« siège »
<Siège 1 personne >[96]	x									
<Siège plusieurs personnes >		x								
<Siège 1 personne avec pied avec dossier sans bras>	x		x		x			x	« chair »	« chaise »
<Siège 1 personne avec pied avec dossier avec bras>	x		x		x		x		« armchair »	« fauteuil »
<Siège 1 personne avec pied sans dossier sans	x		x			x		x	« stool »	« tabouret »

[96] L'accès en discours à des concepts sans désignation se fait à l'aide d'unités lexicales descriptives (paraphrases), par exemple « siège pour une personne » pour le concept « <siège 1 personne> ».

bras>										
<Siège plusieurs personnes avec pied avec dossier avec bras>		x	x		x		x		« couch »	« canapé »
<Siège plusieurs personnes avec pied sans dossier sans bras>		x	x			x		x	« bench »	« banc »

LES ETAPES DE CONSTRUCTION DE LA TERMINOLOGIE DES SIEGES AVEC LA CLT

La construction de la Terminologie avec la CLT fera intervenir les étapes suivantes :

LA DEFINITION D'UN DOMAINE

Console

```
$clt->newDomain('domaine des sièges','Anonymous');
```

Do it

Log

```
[03-09-2018 16:52 Anonymous]domaine des sièges created as domain
```

LA DEFINITION D'UN POINT DE VUE

```
$clt->newDomain('domaine des sièges','Anonymous');
$d=$clt->getDomain('domaine des sièges');
$d->newPointOfView('point de vue 1');
```

```
[03-09-2018 16:55 Anonymous]domaine des sièges created as domain
point de vue 1 created as point of view
```

LA DEFINITION DES AXES D'ANALYSE

```
$pv1->knowledgeDatabase->newAnalysisAxis('nombre de personnes');
$a1=$pv1->knowledgeDatabase->getAnalysisAxis('nombre de personnes');
$pv1->knowledgeDatabase->newEssentialChar('1 personne');
$a1->newEssentialChar('1 personne');
$pv1->knowledgeDatabase->newEssentialChar('plusieurs personnes');
$a1->newEssentialChar('plusieurs personnes');
```

```
nombre de personnes created as analysis axis
1 personne created as essential characteristic
1 personne created as essential characteristic in nombre de personnes
plusieurs personnes created as essential characteristic
plusieurs personnes created as essential characteristic in nombre de personnes
```

LA DEFINITION DES CARACTERISTIQUES DESCRIPTIVES

```
$pv1->knowledgeDatabase->newDescriptiveChar('couleur');
$dc_couleur=$pv1->knowledgeDatabase->getDescriptiveChar('couleur');
$pv1->knowledgeDatabase->newDescriptiveChar('poids');
$dc_poids=$pv1->knowledgeDatabase->getDescriptiveChar('poids','($poids>0)');
```

```
couleur created as descriptive characteristic
poids created as descriptive characteristic
```

LA DEFINITION DES CONCEPTS

```
$pv1->knowledgeDatabase->newConcept('siège');
$c_siege=$pv1->knowledgeDatabase->getConcept('siège');
$pv1->knowledgeDatabase->newConcept('siège 1 personne');
$c_siege_1_personne=$pv1->knowledgeDatabase->getConcept('siège 1 personne');
$pv1->knowledgeDatabase->newConcept('siège plusieurs personnes');
$c_siege_plusieurs_personnes=$pv1->knowledgeDatabase->getConcept('siège plusieurs
personnes');
```

```
siège created as concept
siège 1 personne created as concept
siège plusieurs personnes created as concept
siège 1 personne avec pied avec dossier sans bras created as concept
siège 1 personne avec pied avec dossier avec bras created as concept
siège 1 personne avec pied sans dossier sans bras created as concept
```

L'AFFECTATION DES CARACTERISTIQUES ESSENTIELLES ET DESCRIPTIVES AU NIVEAU DES CONCEPTS

```
$c_siege->addEssentialChar($ec_siege);
$c_siege->addDescripitiveChar($dc_couleur);
$c_siege->addDescripitiveChar($dc_poids);
$c_siege_1_personne->addEssentialChar($ec_siege);
$c_siege_1_personne->addEssentialChar($ec_1personne);
```

```
couleur  added as a descriptive characteristic of the concept siège
poids  added as a descriptive characteristic of the concept siège
siège  added as an essential characteristic of the concept siège 1
personne
1 personne  added as an essential characteristic of the concept siège 1
```

LA MISE EN RELATION DES CONCEPTS

```
$pv1->knowledgeDatabase-
>newRelation($c_siege_plusieurs_personnes_avec_pied_avec_d
ossier_avec_bras,'is-a',$c_siege_plusieurs_personnes);
$pv1->knowledgeDatabase->newRelation($c_pied,'is-part-
of',$c_siege_plusieurs_personnes_avec_pied_avec_dossier_avec
```

```
siège 1 personne_is-a_siège created as relation
siège plusieurs personnes_is-a_siège created as relation
siège 1 personne avec pied avec dossier sans bras_is-a_siège 1
personne created as relation
pied_is-part-of_siège 1 personne avec pied avec dossier sans
bras created as relation
```

L'AFFECTATION DES DESIGNATIONS AU NIVEAU DES CONCEPTS

```
$c_siege->designation->newTerms('français','siège');
$c_siege->designation->newTerms('anglais','seat');
$c_siege_1_personne_avec_pied_avec_dossier_sans_bras-
>designation->newTerms('français','chaise');
$c_siege_1_personne_avec_pied_avec_dossier_sans_bras-
```

```
siège added for français
seat added for anglais
chaise added for français
chair added for anglais
fauteuil added for français
armchair added for anglais
tabouret added for français
stool added for anglais
canapé added for français
```

AFFICHAGE DE LA DEFINITION FORMELLE D'UN CONCEPT

L'instruction « generateFormalDefinition(instance d'un concept) » permet d'afficher la définition formelle d'un concept.

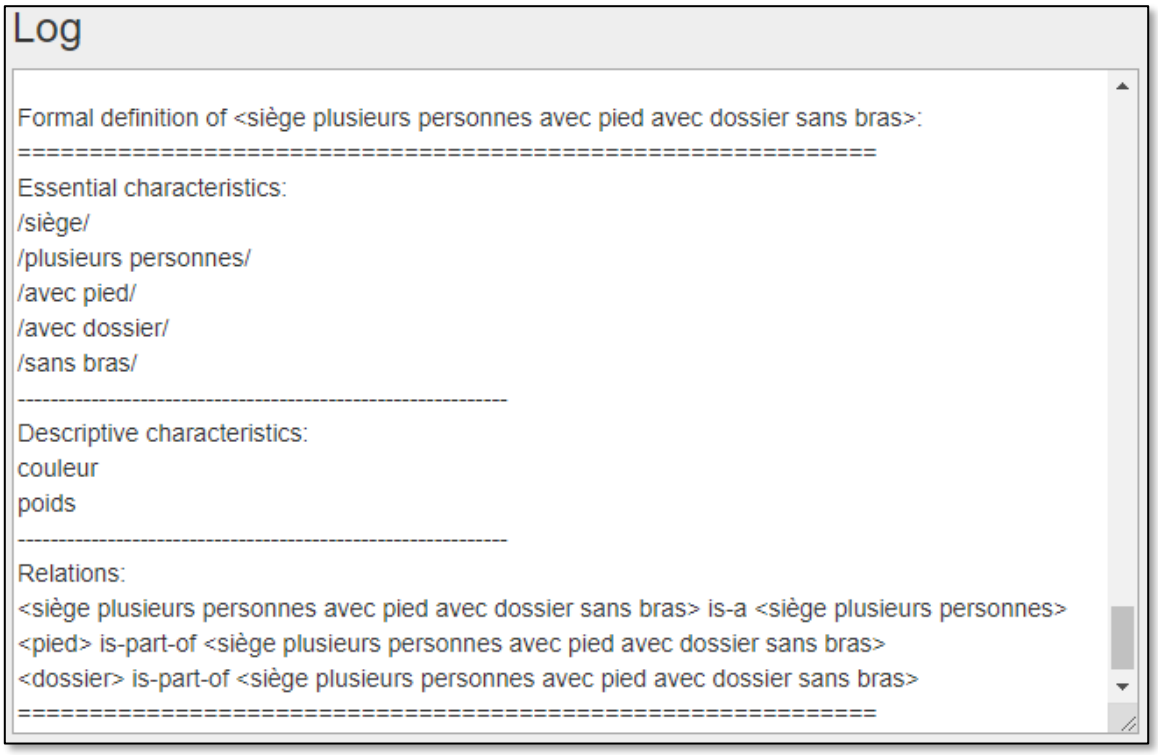


Figure 140 : Définition formelle du concept « <siège plusieurs personnes avec pied avec dossier sans bras> »

AFFICHAGE DE GLOSSAIRES EN FONCTION D'UNE LANGUE DONNEE

A l'aide d'une fonction de la CLT, il est possible dans une langue donnée, d'afficher les termes désignant chaque concept du système conceptuel sous la forme d'un glossaire

```
$pv1->knowledgeDatabase->printListHtmlTerms('anglais');
$clt->printHtml('<br>');
$pv1->knowledgeDatabase->printListHtmlTerms('français');
```

Do it

Concept	Designation (anglais)
<pied>	feet
<bras>	arm
<dossier>	back
<siège>	seat
<siège 1 personne>	
<siège plusieurs personnes>	
<siège 1 personne avec pied avec dossier sans bras>	chair
<siège 1 personne avec pied avec dossier avec bras>	armchair
<siège 1 personne avec pied sans dossier sans bras>	stool
<siège plusieurs personnes avec pied avec dossier avec bras>	couch
<siège plusieurs personnes avec pied avec dossier sans bras>	bench

Concept	Designation (français)
<pied>	pied
<bras>	bras
<dossier>	dossier
<siège>	siège
<siège 1 personne>	
<siège plusieurs personnes>	
<siège 1 personne avec pied avec dossier sans bras>	chaise
<siège 1 personne avec pied avec dossier avec bras>	fauteuil
<siège 1 personne avec pied sans dossier sans bras>	tabouret
<siège plusieurs personnes avec pied avec dossier avec bras>	canapé
<siège plusieurs personnes avec pied avec dossier sans bras>	banc

Figure 141 : Affichage des désignations des concepts dans une langue donnée

AFFICHAGE DU SYSTEME CONCEPTUEL

Comme on l'a vu l'environnement de la CLT propose une fonction d'affichage du système conceptuel, ce qui donne pour la terminologie des sièges le résultat suivant :

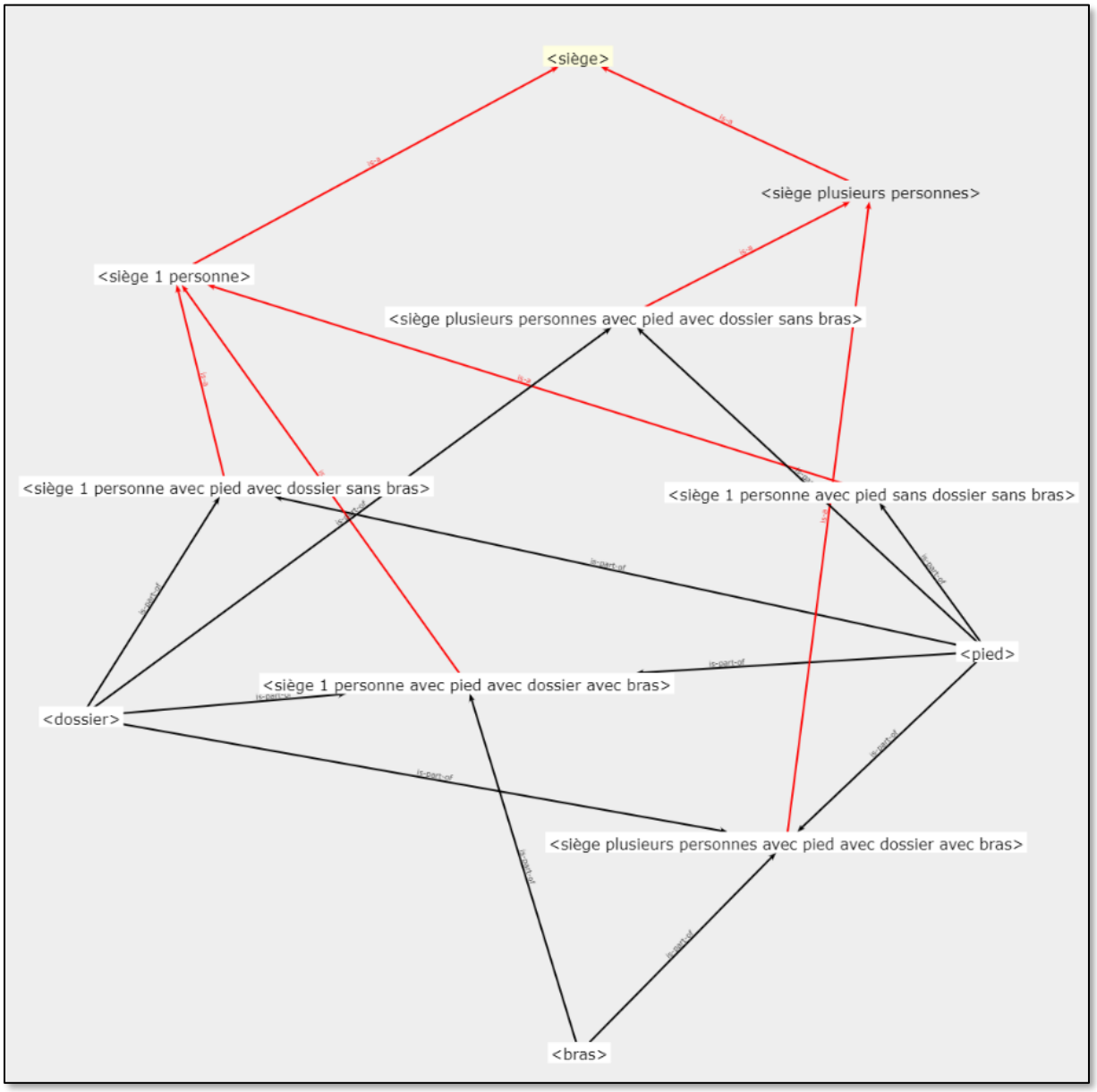
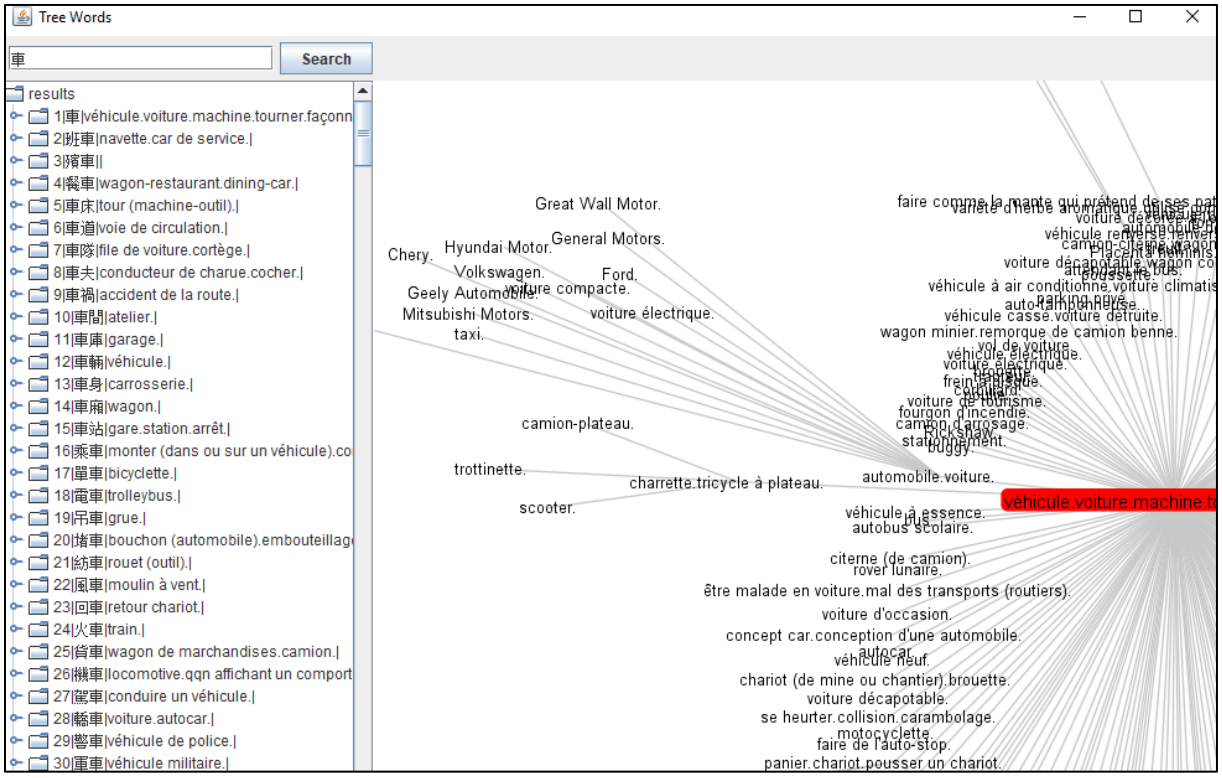


Figure 142 : Affichage du système conceptuel de la terminologie des sièges

Les noms des concepts ainsi que les noms des relations sont affichés. Les relations génériques apparaissent en rouge. Ce type de visualisation nous permet d'avoir une vue d'ensemble du système conceptuel directement dans l'environnement de la CLT.

ANNEXE C : LE LOGICIEL TREEWORDS

Le logiciel TreeWords, intègre deux dictionnaires : chinois/anglais (CEDICT) et chinois/français (CFDICT). En utilisant le principe de formation des mots chinois, il est possible de générer un réseau de mots multilingues se basant sur les caractéristiques qui composent le mot chinois. Par exemple la sélection du mot chinois 车 désignant principalement « véhicule à roues », généra un réseau de termes désignant les véhicules à roues. Ces termes peuvent être affichés en chinois, en anglais ou en français à la condition qu'ils existent. Grâce à la transparence des mots chinois, on a l'indication de la vision chinoise de la conceptualisation d'un domaine. On peut voir ci-dessous, qu'une trottinette est une sorte de tricycle à plateau qui est lui-même un type de véhicule à roues.



BIBLIOGRAPHIE

[Alleton V., 1988]	**Alleton, V. (1988). Présentation: Classifications chinoises ou les dangers du réductionnisme. Extrême-Orient Extrême-Occident, 7-12.**
[Alleton V., 2001]	Viviane Alleton. (2001). Chinese Terminologies: On Preconceptions. New Terms for New Ideas: Western Knowledge and Lexical Change in Late Imperial China, pp.15-34
[Alleton V., 2002]	Alleton Viviane.(2002). L'écriture chinoise. Paris: Presses Universitaires de France, 128p.
[Arbib, 1992]	Arbib, M. A. (1992). Schema theory. The Encyclopedia of Artificial Intelligence, 2, 1427-1443.
[Aussenac-Gilles et al. ,2006]	Aussenac-Gilles N., Condamines A., Sedes F. (2006). Evolution et maintenance des ressources termino-ontologiques : une question à approfondir. Information – Interaction – Intelligence, Cépaduès Editions, Numéro spécial Ressources onto-terminologiques, Vol. Hors-série, p.7-14,2006.
[Baader et al., 2003]	Baader F., Horrocks, I., & Sattler, U. (2003). Description logics as ontology languages for the semantic web. In Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence (pp. 228–248). : Springer-Verlag.
[Baader & Nutt, 2002]	Baader F. & Nutt, W. (2002). the Description Logic Handbook, chapter Basic Description Logics, (pp. 47–100). Cambridge University Press.
[Baader et al., 2003]	Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., & Nardi, D. (Eds.). (2003). The description logic handbook: Theory, implementation and applications. Cambridge university press.
[Barrière, 2001]	Barrière C., "Investigating the causal relation in informative texts." *Terminology* 7(2):135–154. DOI: 10.1075/term.7.2.02bar
[**Berners-Lee, 2000]**	Berners-Lee, T. (2000). Semantic Web Stack. Presentation at XML.
[Berrueta et al., 2007]	Berrueta, D., Brickley, D., Decker, S., Fernández, S., Görn, C., Harth, A., ... & Passant, A. (2007). Sioc core ontology specification. *W3c member submission, W3C*.
[Bourigaut, Slodzian, 1999]	Bourigault, D., & Slodzian M. (1999). Pour une terminologie textuelle. Terminologies nouvelles, 19(1999), 29-32.
[Bowker & Eck, 1992]	Bowker, L., & Eck, K. COGNITERM: An Experiment in Building a Terminological Knowledge Base, Euralex 92
[**Brachman & Levesque, 1992]**	Brachman, R. J., Levesque, H. J., & Reiter, R. (Eds.). (1992). Knowledge representation. MIT press.
[Buitelaar et al., 2005]	Buitelaar, P., Cimiano, P., & Magnini, B. (Eds.). (2005). Ontology learning from text: methods, evaluation and applications (Vol. 123). IOS press.

[Cabré, 1998]	Cabré (M.T.),1998 : La terminologie. Théorie, méthode et applications, traduit du catalan, adapté et mis à jour par Cormier (M.) et Humbley (J.), Ottawa, Presses de l'Université d'Ottawa (Regards sur la traduction) et Paris, Armand Colin (U-Linguistique).
[Campo, 2013]	Campo, A. (2013). The reception of Eugen Wüster's work and the development of terminology.
[Cañas et al., 2004]	Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Gómez, G., ... & Carvajal, R. (2004). CmapTools: A knowledge modeling and sharing environment.
[Candel, 2004]	« Wüster par lui-même », dans Cortès (C.), éd., Des fondements théoriques de la terminologie, Cahiers du CIEL, 2004, p. 15-31
[Charlet et al., 2003]	Charlet, J., Laublet, P., & Reynaud, C. (2003). Le web sémantique. Cépaduès-Ed.
[Condamines & Rebeyrolle, 2001]	Condamines, A., & Rebeyrolle, J. (2001). Searching for and identifying conceptual relationships via acorpus-based approach to a Terminological Knowledge Base (CTKB). In D. Bourigault, M. C. L'Homme, & C. Jacquemin (Eds.), Recent Advances in Computational Terminology, (pp. 127–148). Amsterdam/New York: John Benjamins.
[Condamines, 2005]	Condamines, A. (2005). Linguistique de corpus et terminologie. Langages, (1), 36-47.
[Costa, 2013]	Costa, R. (2013). Terminology and Specialised Lexicography: two complementary domains. Lexicographica, 29(1), 29–42.
[Cruse 1986]	Cruse, D. A (1986). Lexical semantics. Cambridge University Press.
[Després & Szulman, 2008]	Després, Sylvie et Szulman, Sylvie (2008) : Réseau terminologique versus Ontologie. In : Actes de la deuxième conférence TOTh (Annecy, 5-6 juin 2008). Annecy : Institut Porphyre, 17-34.
[Després, 2018]	Despres, S. (2018, July). oogo: Ontologie des Outils utiles à la Gestion d'Ontologies. In 29es Journées Francophones d'Ingénierie des Connaissances, IC 2018 (pp. 163-178).
[Euzenat, Shvaiko, 2007]	Euzenat, J., & Shvaiko, P. (2007). Ontology matching (Vol. 18). Heidelberg: Springer.
[Felber, 1980]	Felber, H. (1980). In memory of Eugen Wüster, founder of the general theory of terminology. International Journal of the Sociology of Language, 1980(23), 7-14.
[Felber, 1987]	Felbert (H.), 1987: Manuel de terminologie, Paris, Unesco.
[Frege, 2014]	Frege, G. (2014). Écrits logiques et philosophiques. Le Seuil.
[Fox & Adam, 1985]	Fox, M. S., Wright, J. M., & Adam, D. (1985). Experiences with SRL: An analysis of a frame-based knowledge representation.
[Giaretta & Guarino, 1995]	Giaretta, P., & Guarino, N. (1995). Ontologies and knowledge bases towards a terminological clarification. Towards very large knowledge bases: knowledge building & knowledge sharing, 25(32), 307-317
[Guarino, 1994]	Guarino, N., Carrara, M., & Giaretta, P. (1994). An Ontology of Meta-Level Categories. KR, 94, 270-280.
[Guarino, 1998]	Guarino, N. (Ed.). (1998). Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8,

	Trento, Italy (Vol. 46). IOS press.
[Gennari et al., 2003]	Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., ... & Tu, S. W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-computer studies, 58(1), 89-123.
[Gruber, 1995]	Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, 43(5-6), 907-928.
[Horridge, 2011]	Horridge, M. (2011). A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. Retrieved March 10, 2013
[Horridge, 2013]	Horridge M., Tudorache T., Vendetti J., Nyulas C., Musen M., Noy N. (2013). Simplified OWL ontology editing for the web: is WebProtégé enough? The Semantic Web - ISWC 2013, Proceedings part I - 12th International Semantic Web Conference, Sydney New South Wales, Australia, p. 200-215.
[Humbley, 2004]	Humbley (J.), 2004: "La réception de l'œuvre d'Eugène Wüster dans les pays de la langue française », dans Cortès (C.), éd., Des fondements théoriques de la terminologie, Cahiers du CIEL, 2004, p. 33-51.
[Infoterm, 2004]	Infoterm, 30 years of Infoterm, 2004-06-21 NI+GA+DE
[ISO 704:2000]	"Terminology work - Principles and methods". International Organization for Standardization (2000)
[ISO 704:2009]	"Terminology work - Principles and methods". International Organization for Standardization (2009)
[ISO 860:2007]	Harmonisation des concepts et des termes". International Organization for Standardization (2007)
[ISO 1087-1:2000]	"Terminology work-Vocabulary-Part 1: Theory and application". International Organization for Standardization (2000)
[ISO 24156-1:2014]	"Graphic notations for concept modelling in terminology work and its relationship with UML". International Organization for Standardization (2014)
[ISO 25964-1:2011]	ISO 25964-1 "Information and documentation - Thesauri and interoperability with other vocabularies - Part 1: Thesauri for information retrieval" (2011)
[Kockaert H.,Antia B., 2008]	Kockaert H., Antia B. (2008). Comment modéliser des concepts en rapprochant un langage orienté objet et deux normes terminologiques orientées concept ?. In Roche, C. (Ed.), Terminologie & Ontologie :Théories et Applications. Actes de la deuxième Conférence TOTh 2008. TOTh. Annecy, France, 5-6 June2008 (pp. 103-123). Annecy: Institut Porphyre. Savoir et Connaissance.
[Kockaert H. et al.,2010]	Kockaert H., Steurs F., Antia B. (2010). Filling the gaps between the object-oriented UML modeling andconcept-oriented terminological modeling in ISO standards: Application of ISO/DIS 704 and ISO 1087-1in ISO/TR 24156 on the basis of UML in terminological concept modeling. In Bhreathnach, .

	(Ed.), deBarra Cusack, F. (Ed.), Proceedings TKE 2010: Presenting terminology and knowledge engineeringresources online: Models and challenges (on cd-rom). Presenting Terminology and KnowledgeEngineering Resources Online: Models and Challenges. Fiontar, Dublin City University, Ireland, 12-13August 2010 (pp. 437-456). Dublin: Fiontar, Dublin City University.
[Kocourek, 1991]	La langue française de la technique et de la science. Vers une linguistique de la langue savante, 2ème édition, Wiesbaden, Oscar Brandstetter Verlag & co.
[Krötzsch et al., 2012]	Krötzsch, M., Simancik, F., & Horrocks, I. (2012). A description logic primer. arXiv preprint arXiv:1201.4089.
[Lassila & Swick, 1998]	Lassila, O., & Swick, R. R. (1998). Resource description framework (RDF) model and syntax specification.
[L'Homme, 2004]	L'Homme (M.-Cl.), 2004: La terminologie: principes et techniques, Montréal, Les presses de l'Université de Montréal (Paramètres).
[Löckinger et al.]	Löckinger, G., Kockaert, H. J., & Budin, G. (2015). Intensional definitions. *Handbook of Terminology*, *1*, 60.
[Lyons, 1997]	Lyons, J. (1977). Semantics. Volume I. Cambridge UP, Cambridge.
[Mascardi et al., 2007]	Mascardi, V., Cordì, V., & Rosso, P. (2007, September). A Comparison of Upper Ontologies. In Woa (Vol. 2007, pp. 55-64)
[McBride, 2004]	McBride, B. (2004). The resource description framework (rdf) and its vocabulary description language rdfs. In Handbook on Ontologies (pp. 51–66).
[McGuinness & Van Harmelen]	McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. W3C recommendation, 10(10), 2004.
[Métailié, 2005]	Métailié G., Terminologie et approche interculturelle : terminologies scientifiques du chinois. Armand Colin | Langages2005/1 - n° 157 pages 118 à 128
[Miles et al., 2005]	Miles, A., Matthews, B., Wilson, M., & Brickley, D. (2005, September). SKOS core: simple knowledge organisation for the web. In International Conference on Dublin Core and Metadata Applications (pp. 3-10).
[Miller, 1998]	Miller, G. (1998). WordNet: An electronic lexical database. MIT press.
[Minsky, Marvin, 1974]	Minsky, Marvin "A Framework for Representing Knowledge" MIT AI Lab Memo AIM-306 June 1974
[Otman, 1996]	Otman, G. (1996). Les représentations sémantiques en terminologie.
[Papadopoulou & Roche, 2018a]	M. Papadopoulou, C. Roche. SEMAPRO 2018: The Twelfth International Conference on Advances in Semantic Processing. 18-22 November 2018, Athens, Greece. AIDAinformazioni Journal, number 1-2/2018, volume XXXVI.
[Papadopoulou & Roche, 2018b]	M. Papadopoulou, C. Roche. Ontologization of Terminology. A worked example from the domain of ancient Greek dress. AIDAinformazioni Journal, number 1-2/2018, volume XXXVI

[Pavel & Nolet, 2001]	Pavel, S., & Nolet, D. (2001). Handbook of Terminology. Minister of Public Works and Government Services Canada 2001, Catalogue No. S53-28.
[Rastier, 2004]	Rastier F. (2004). Ontologie(s). Revue d'Intelligence Artificielle vol. 18 n°1 2004. pp. 15-40.
[Rey, 1979]	Rey, A. (1979). La terminologie: noms et notions (No. 1780). Presses Univ. de France.
[Rey, 1992]	Rey, A. (1992). Dictionnaire historique de la langue française. Le robert.
[Roche, 2003]	Roche C., Ontology: a survey, 8th Symposium on Automated Systems Based on Human Skill and Knowledge – IFAC, September 22-24 2003, Göteborg, Sweden
[Roche, 2005]	C. Roche., Terminologie et ontologie. In : Langages, 39e année, n°157, 2005. La terminologie : nature et enjeux. pp. 48-62.
[Roche, 2007]	Roche, C. (2007, June). Saying is not modelling. In 4th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2007) (pp. 47-56).
[Roche et al., 2009]	C. Roche, M. Calberg-Challot, L. Damas, P. Rouard., Ontoterminology: A new paradigm for terminology, KEOD 2009 (International Conference on Knowledge Engineering and Ontology Development), 5-8 October, Madeira (Portugal)
[Roche & Calberg-Challot, 2010]	Roche, C., & Calberg-Challot, M. (2010, October). Synonymy in Terminology: The Contribution of Ontoterminology. In Symposium on Re-thinking synonymy.
[Roche, 2011]	Roche, C. (2011, May). L'Isagoge de Porphyre. In Terminology & Ontology: Theories and applications (TOTH 2011) (Vol. 2011). Institut Porphyre, Savoir et Connaissance.
[Roche, 2012]	Roche C., Should Terminology Principles be re-examined?, TKE 2012, 10th Terminology and Knowledge Engineering Conference , Madrid (Spain), 19-22 June 2012, pp.17-32
[Roche et al., 2014]	Roche, C., Damas, L., & Roche, J. (2014, September). Multilingual Thesaurus: The Ontoterminology Approach. In CIDOC 2014-Access and Understanding–Networking in the Digital Era.
[Roche, 2015]	C.Roche., Ontological definition, Handbook of Terminology, Volume 1, John Benjamins Publishing, 2015, pp.128-152
[Roche & Zhang 2016]	J. Roche, Z. Zhang. Neologism in Chinese: borrowing western terms using a clustering approach. Workshop TOth 2016, Neology in Terminology, Nov 2016, Paris, France. 〈hal-01851368〉
[Roche et al., 2018]	J. Roche, C. Roche, Z. Zhang, Y. Jia, T. Wei. Proposition d'un modèle computationnel du concept pour la Terminologie. Conférence TOTh 2018, Terminology & Ontology: Theories and applications, Jun 2018, Chambéry, France. 〈hal-01851353〉
[Roche, 2019]	"Une Théorie du Concept pour la Terminologie". Collection Terminologica, Presses Universitaires Savoie Mont-Blanc, à paraître 2019.
[Roussey et al., 2010]	Roussey, C., Scharffe, F., Corcho, O., & Zamazal, O. (2010, June). Une méthode de débogage d'ontologies OWL basées sur la détection d'anti-

	patrons. In 21èmes Journées Francophones d'Ingénierie des Connaissances (pp. 43-54). Ecole des Mines d'Alès.
[Sager, 1990]	Sager, Juan C. Practical course in terminology processing. John Benjamins Publishing, 1990.
[Slodzian, 1995]	Slodzian (M.), 1995 : « Comment revisiter la doctrine terminologique aujourd'hui », dans la banque des mots, numéro spécial 7,p. 11-18.
[Slodzian, 2000]	Slodzian, M. (2000). L'émergence d'une terminologie textuelle et le retour du sens. Le sens en terminologie, 61-85.
[Sowa, 2000]	Sowa, J. F. (2000). Knowledge representation: logical, philosophical, and computational foundations (Vol. 13). Pacific Grove, CA: Brooks/Cole.
[Spies & Roche, 2008]	Spies M., Roche C. (2008). Aristotelian Ontologies and OWL Modeling. Handbook of Ontologies for Business Interaction. Information Science Reference, Hershey, New York, p. 21-33.
[Staab & Studer, 2010]	Staab, S., & Studer, R. (Eds.). (2010). Handbook on ontologies. Springer Science & Business Media.
[Tellier, 2010]	Tellier, I. (2010). Introduction au TALN et à l'ingénierie linguistique. Polycopié de cours: Université de Lille, 3.
[Temmerman, 2000]	Temmerman (R.), 2000 : Towards New Ways of Terminology Description.The Sociocognitive Approach, Amsterdam & Philadlphia, John Benjamins Publishing (Terminology and Lexicography Research and Practice).
[Temmerman & Kerremans, 2003]	Temmerman, R., & Kerremans, K. (2003). Termontography: Ontology building and the sociocognitive approach to terminology description. Proceedings of CIL17, 1-10.
[Tissaoui ,2009]	Tissaoui A. (2009). Typologie de changements et leurs effets sur l'évolution de Ressources Termino-Ontologiques. Actes de la conférence Ingénierie des Connaissances (IC 2009).
[Tricot & Roche, 2006]	C. Tricot & C. Roche. « Visualisation of Ontology: a focus and context approach ». InSciT2006, Mérida, October 25-28th 2006, Spain
[Tricot & al.,2006]	C. Tricot, C. Roche, C. Foveau, S. Reguigui.« Cartographie sémantique de fonds numériques scientifiques et techniques ». Document Numérique « Visualisation pour les bibliothèques numériques », pp.13-36, Vol. 9 – n°2/2006
[Uschold & Gruninger, 1996]	Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. The knowledge engineering review, 11(2), 93-136.
[Van Campenhoudt, 2006]	Van Campenhoudt, M. (2006, February). Que nous reste-t-il d'Eugen Wüster?. In Proceedings of the Colloque international Eugen Wüster et la terminologie de l'Ecole de Vienne (pp. 3-4).
[Weisfeld, 2008]	Weisfeld, M. (2008). The object-oriented thought process. Pearson Education.
[Winston, 1987]	Winston, M. E., Chaffin, R., & Herrmann, D. (1987). A taxonomy of part-whole relations. Cognitive science, 11(4), 417-444.
[Wright & Fox, 1983]	Wright, J. M., & Fox, M. S. (1983). SRL/1.5 user manual. Carnegie-Mellon University.

[Wüster, 1968]	Wüster (E.), 1968 : Dictionnaire multilingue de la machine-outil. Notions fondamentales, définies et illustrées, présentées dans l'ordre systématique et l'ordre alphabétique. Volume de base anglais-français : The Machine Tool. An interlingual Dictionary of Basic Concepts comprising an Alphabetical Dictionary and a Classified Vocabulary with Definitions and Illustrations. English-French Master Volume, London, Technical Press.
[Zhang et al., 2018]	Z. Zhang, C. Roche, Y. Jia, J. Roche, T.Wei. Term formation method for Chinese neoterm based on concept model and Chinese attribute-head construction. TOTh 2018: Terminology & Ontology: Theories and applications, Jun 2018, Chambéry, France.

EXTENDED ABSTRACT

INTRODUCTION

1. NEEDS AND ISSUES: THE COMPUTATIONAL REPRESENTATION OF TERMINOLOGY

With the exponential growth of information in digital society, new needs and applications emerge. Knowledge management, specialized dictionaries, semantic information retrieval in a multi-lingual context are some examples. All of these require a computational representation of Terminology, i.e. a computational representation of the conceptual system of terminology.

Nevertheless, the current definition of Terminology in ISO standards (ISO 1087-1) cannot be directly implemented in a computational model. Although ISO standards on Terminology do not aim to operationalize terminologies but to communication between humans, they should be used to model information and data [ISO 704: 2009]. The results of disciplines such as knowledge engineering highlighted the need for a theory of concept that could lead to a computational representation. In such context, ontology in the sense of Knowledge Engineering, constitutes one of the most interesting perspectives to model the conceptual system of Terminology, bringing new approaches and methods to Terminology, e.g. alignment of ontologies for Terminology harmonization (ISO 860), verifying properties, reasoning, etc. It is the Ontological turn of Terminology.

2. OBJECTIVES AND CONTEXT

The objectives of this thesis are threefold. The first objective is to propose a theory of the concept that is consistent from an epistemological and logical point of view, the second is to propose a computational model, the third is to study the consequences on the Terminology itself, as well as in its principles and methods. The study concludes with an example of a software application developed specifically in the context of this thesis and some further research perspectives.

This thesis is situated within the framework of the General Theory of Terminology and the International Standard Organization (ISO) for Terminology. Terminology is governed by two transversal standards: ISO 1087-1 (Terminology work - vocabulary - part1: Theory and application) and ISO 704 (Terminology work - principles and methods). These two standards, as their title indicates, describe the vocabulary, principles and methods of so-called "classical" Terminology rooted in the General Theory of Terminology (Felber 1984). Our study takes into account the ongoing revision of the ISO 1087-1 standard and the work carried out within the AFNOR / X03A "Terminology: principles and coordination" committee of AFNOR.

This thesis is the result of an international collaboration between the research team Condillac "Ontology and Terminology" LISTIC laboratory of the University Savoie Mont Blanc and the Research

Center in Knowledge Engineering and Terminology of the University of Liaocheng (Shandong Province, China).

3. THESIS STRUCTURE

The first chapter is dedicated to the state-of-the-art on Terminology and Ontology. We present and discuss the definition of the discipline of Terminology, its dual dimension (linguistic and conceptual), its different approaches (textual and conceptual) as introduced in the General Theory of Terminology and in ISO-1087-1 and 704 standards. Emphasis is placed on the presentation of the principles that govern the conceptual model of Terminology.

Ontology, in the sense of Knowledge Engineering, is the definition of concepts and their relationships in a formal language. Ontology is one of the most promising ways for making Terminology computable. After having presented what Ontology (in the philosophical point of view) can bring for knowledge modelization, we detail the different types of ontologies resulting from Artificial Intelligence, as well as the languages for knowledge representation according to their degree of formality.

The second chapter is devoted to the ontological turn of Terminology. The use of Information Technology in Terminology is not new. The automatic analysis of corpus for linguistic purposes (extraction of candidate terms, collocations, search for defining relations, concordance studies, etc.) has strongly impacted Terminology in its foundations and its principles. It gave rise to a first turning point, the "textual turn" of Terminology. The operationalization of terminologies for information processing purposes is a different issue. It requires a computational representation of the notional system, which places the concept in the foreground. It is the "ontological turn" of Terminology that challenges certain principles and methods of Terminology. Ontology is one of the most promising prospects for Terminology[97]. in a digital world. The notion of "ontoterminology", a terminology whose conceptual system is a formal ontology, emerges directly from it.

We first probe into the problems, both epistemological and logical, posed by the Theory of Concept of Terminology, as defined in the ISO 1087-1 and ISO 704 standards, which hamper a computer implementation. Based on this observation, we propose a Concept Theory and an associated language named CLT (Concept Language for Terminology) which, while respecting as far as possible Terminology principles, leads to a computational representation of the system. In particular, we introduce an additional abstraction layer that allows us to manipulate objects, as such, and concepts, independently of the cardinality of their extension[98].

It should be noted that in the "ontoterminological" approach, terms and concept names are not confused, since they come from different semiotic systems. In addition to being unique, concept names are not limited to a role of identifiers in a formal system. They have a particular importance in the conceptual system because they must reflect or be able to indicate the place of the concepts that they represent. Far from being arbitrary, the way of naming concepts can therefore be inspired by

---

[97] Whether an ontology can be built from texts is out of the scope of our work, especially since ontologies built automatically from texts do not generally correspond to those established directly by experts in the field with a formal language.

[98] For example, the concept <Natural satellite of moon> does not correspond to any object.

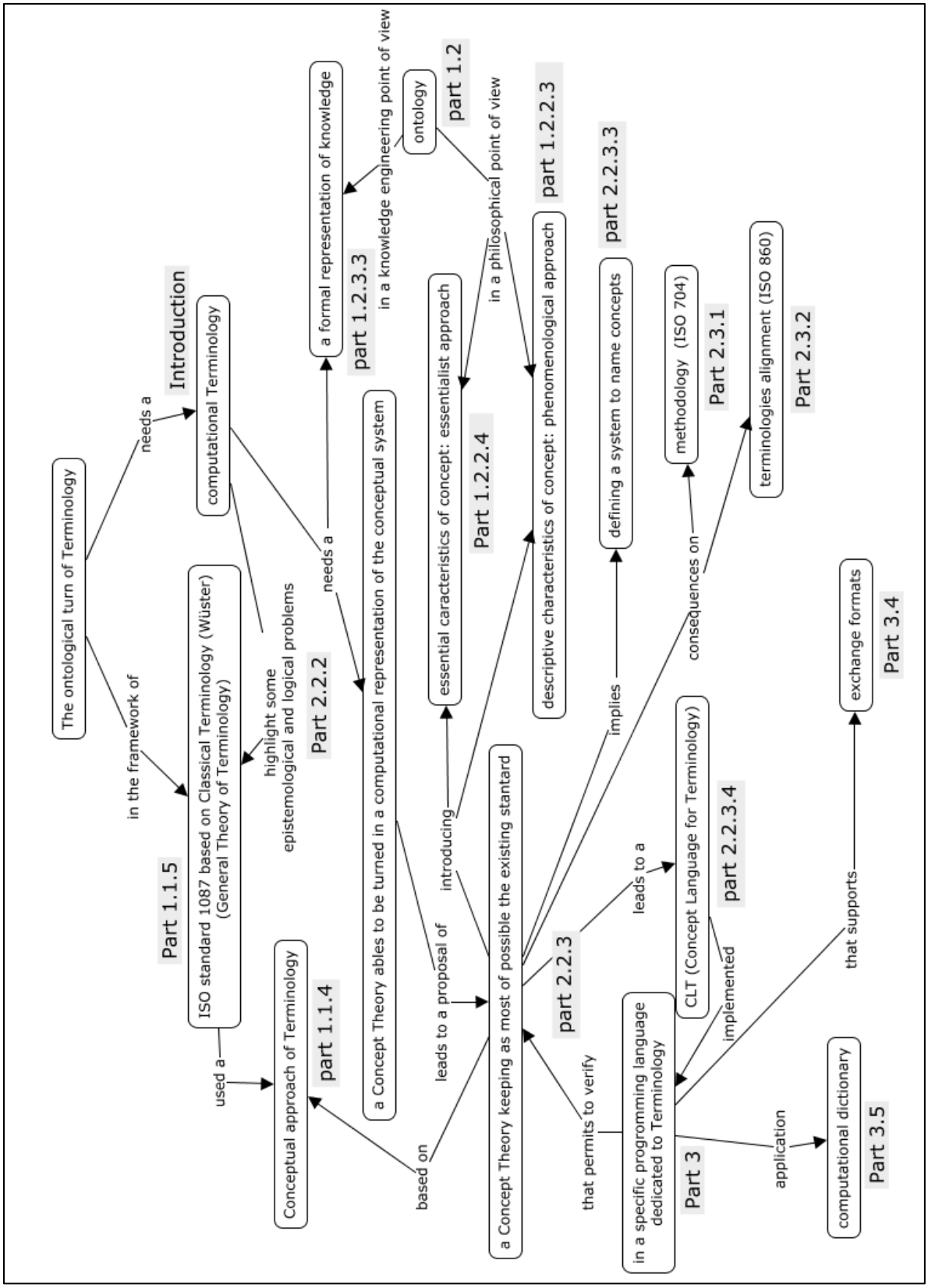
certain natural languages such as Chinese, where a word is formed by characters that represent the essential characteristics of the word that one wants to introduce. The place of these characters (bearers of meaning) makes it possible to locate the new word (composed of existing characters) related to the others. We consider whether the system that governs the formation of Chinese terms can be adapted to name the concepts. The Treewords program has been developed to illustrate the Chinese terms formation system.

Finally, we discuss the methodological contributions of Terminology taking into account the distinction between the linguistic system (the terms) and the Ontology-based conceptual system, by integrating such as the essential characteristics of concepts. We analyze how the representation of the conceptual system in the form of an ontology permits to align terminologies. This approach opens the way to the operationalization of terminologies such as multilingual semantic search.

In a last chapter we propose a computer language dedicated to the CLT (Concept Language for Terminology) able to construct a conceptual system (to define concepts and their relations) which in itself constitutes a validation of the theory. The CLT once implemented, can be seen as a specialization of a programming language for Terminology. CLT can serve both as a language dedicated to the creation and manipulation of computational terminologies and as an interchange format, since it is understandable by a computer. The choice of a Client / Server architecture for the implementation of the CLT allows to consider a collaborative aspect in the conceptual system building phase. Integrating a graphical user will permit to establish a computational terminology without notion of programming. The role of this interface will be to transcribe the action of the user into CLT instructions. The conceptual system produced using the CLT can be visualized directly in the environment and exported in different formats.

The implementation of the CLT will be illustrated through an application, the Treewords program that aims to operationalize a multilingual dictionary for a given domain. The conceptual system is built using the principles of Chinese terms formation. This operationalization convert entries of the dictionary into instructions of the CLT language. This kind of operationalization based on the CLT model could be used to convert specialized dictionaries that contains Chinese entries to computational terminologies. In this way, it gives the possibility to check the coherence of the term definition and/or make sure that the term is well chosen.

THESIS PLAN:


The ontological turn of Terminology
needs a
Introduction
in the framework of
Part 1.1.5
ISO standard 1087 based on Classical Terminology (Wüster)
(General Theory of Terminology)
used a
Conceptual approach of Terminology
part 1.1.4
computational Terminology
highlight some
epistemological and logical problems
Part 2.2.2
needs a
a formal representation of knowledge
part 1.2.3.3
in a knowledge engineering point of view
ontology
part 1.2
a Concept Theory ables to be turned in a computational representation of the conceptual system
based on
leads to a proposal of
introducing
essential caracteristics of concept: essentialist approach
Part 1.2.2.4
in a philosophical point of view
descriptive characteristics of concept: phenomenological approach
part 1.2.2.3
a Concept Theory keeping as most of possible the existing stantard
part 2.2.3
implies
defining a system to name concepts
part 2.2.3.3
that permits to verify
leads to a
in a specific programming language
dedicated to Terminology
Part 3
CLT (Concept Language for Terminology)
part 2.2.3.4
implemented
consequences on
methodology (ISO 704)
Part 2.3.1
terminologies alignment (ISO 860)
Part 2.3.2
application
computational dictionary
Part 3.5
that supports
exchange formats
Part 3.4

Titre de la thèse en français : Le Tournant Ontologique de la Terminologie

Mots clés en français : terminologie, ontologie, ontoterminologie, ingénierie des connaissances, ISO-1087, ISO-704, ISO-860

Résumé de la thèse en français : L'opérationnalisation des terminologies à des fins de traitement de l'information requiert une représentation computationnelle du système conceptuel. La théorie du concept sur laquelle se fonde la Terminologie ISO ne permet pas aujourd'hui une telle représentation informatique [Roche, 2012]. Même si les normes ISO sur la Terminologie n'ont pas pour objectif l'opérationnalisation de terminologies mais la communication entre humains, elles doivent pouvoir – elles le précisent – servir à la modélisation des informations et des données [ISO 704 :2009]. Les résultats de disciplines telles que l'ingénierie des connaissances ont permis de mettre en évidence la nécessité de disposer d'une théorie du concept qui puisse donner lieu à une représentation informatique. Dans ce cadre, les ontologies, issues de l'ingénierie des connaissances, constituent une des perspectives les plus intéressantes pour la modélisation du système conceptuel d'une terminologie [Roche, 2015].

Titre de la thèse en anglais : The Ontological Turn in Terminology

Mots clés de la thèse en anglais : terminology, ontology, ontoterminology, knowledge engineering, ISO-1087, ISO-704, ISO-860

Résumé de la thèse en anglais : The operationalization of terminologies for information processing purposes requires a computational representation of the conceptual system. The theory of concept currently defined in ISO Terminology does not allow a computational representation [Roche, 2012]. Although ISO standards on Terminology do not aim to operationalize terminologies but communication between humans, they should be used to model information and data [ISO 704: 2009]. The results of disciplines such as knowledge engineering highlighted the need for a theory of concept that could lead to a computational representation. In this context, ontologies, from knowledge engineering, is one of the most interesting perspectives to model the conceptual system of Terminology [Roche, 2015].

Titre de la thèse en chinois : 术语的本体表示

Mots clés en chinois : 术语学, 本体论, 知识工程, ISO-1087, ISO-704, ISO-860

Résumé de la thèse en chinois : 概念系统的可计算表示可以为信息处理中术语的操作提供支持，但目前，在 ISO 标准中的概念理论并没有提供这种术语的可计算表示[Roche, 2012]。虽然 ISO 中术语相关标准是为方便交流而不是以术语的可操作化为目的，但它必须能够为信息和数据建模提供服务[ISO 704:2009]。知识工程等学科的快速发展凸显了对概念理论进行可操作化表示的巨大需求。在此背景下，知识工程中的本体理论和技术是对术语的概念系统进行建模最有效的方法之一[Roche,2015]。