

Représentation, iconisation et visualisation des connaissances : Principes et applications à l'aide à la décision médicale

Jean-Baptiste Lamy

▶ **To cite this version:**

Jean-Baptiste Lamy. Représentation, iconisation et visualisation des connaissances : Principes et applications à l'aide à la décision médicale. Intelligence artificielle [cs.AI]. Normandie Université, Université de Rouen, 2017. tel-01849832

HAL Id: tel-01849832
https://hal.science/tel-01849832

Submitted on 26 Jul 2018

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

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







Habilitation à Diriger les Recherches

Spécialité : Informatique

Normandie Université, Université de Rouen
UFR Sciences et Techniques

Représentation, iconisation et visualisation des connaissances : Principes et applications à l'aide à la décision médicale

Par
Jean-Baptiste LAMY

Soutenance publique le 21 novembre 2017 devant le jury composé de :

M. Gilles Venturini, professeur à l'université de Tour, rapporteur
Mme Sandra Bringay, professeure à l'université de Montpellier 3, rapportrice
M. Nhan le Thanh, professeur à l'université de Nice, rapporteur
M. Thierry Lecroq, professeur, université de Rouen, examinateur
Mme Marie-Christine Jaulent, directrice de recherches INSERM à Paris, examinatrice
Mme Lina F Soualmia, maître de conférences à l'université de Rouen, examinatrice

Remerciements

Je remercie Marie-Christine Jaulent pour avoir essayé d'organiser visuellement le laboratoire qu'elle dirige.
Je remercie Lina Soualmia et Thierry Lecroq pour leur soutien et leur invitation à Rouen.
Je remercie Brigitte Séroussi et Jacques Bouaud pour avoir pensé à moi même en Espagne.
Je remercie Karima Sedki et Fadi Badra pour les longues discussions passées à discuter nos discussions.
Je remercie Rosy Tsopra pour avoir été une collègue de bureau (presque ;-) supportable.
Je remercie Gaoussou Camara pour m'avoir permis d'enseigner Python au pays des pythons.
Je remercie Appoh Kouame pour m'avoir contraint, bien malgré lui, de raccourcir mon manuscrit.
Je remercie Sylvie Despres pour ne se formaliser qu'informatiquement.
Je remercie Alain Venot pour m'avoir appris que la rigueur ne s'apprend pas nécessairement avec rigueur.
Je remercie Catherine Duclos pour m'avoir inoculé le virus alors qu'elle travaille sur l'antibiothérapie.
Je remercie Vincent Rialle pour m'avoir appris que la curiosité est un vilain défaut mais un beau métier.
Je remercie Mobin Yasini, Massoud Toussi et Vahid Ebrahiminia pour la poésie iranienne.
Je remercie Gilles Venturini, Sandra Bringay et Nhan le Thanh pour avoir accepté de me rapporter.
Je remercie Candice pour l'intérêt qu'elle porte à des langages qu'elle ne comprend pas toujours (ou pas encore !).
Je remercie Antoine, Patricia et Estelle pour d'autres relectures.
Je remercie également tous ceux que j'ai oublié, pour m'avoir appris que l'oubli est nécessaire à l'apprentissage.

Table des matières

1 Introduction **8**
- 1.1 Démarche générale . . . 12
- 1.2 Projets . . . 12
  - 1.2.1 Projets L3IM et SiFaDO . . . 12
  - 1.2.2 Projet VIIIP . . . 14
  - 1.2.3 Recette iconique en médecine traditionnelle africaine . . . 14
- 1.3 Plan . . . 14

2 Représentation des connaissances **17**
- 2.1 Introduction et contexte . . . 17
  - 2.1.1 Définitions . . . 17
  - 2.1.2 Les connaissances médicales . . . 18
  - 2.1.3 Formalismes de représentation des connaissances . . . 19
  - 2.1.4 Spécificités des ontologies dans le domaine biomédical . . . 21
  - 2.1.5 Méthodologies pour la modélisation du domaine . . . 22
    - 2.1.5.1 Méthodologies collaboratives . . . 22
    - 2.1.5.2 Méthodes automatiques et semi-automatiques . . . 23
    - 2.1.5.3 Méthodologies dans le domaine biomédical . . . 23
  - 2.1.6 Opérationnalisation des connaissances . . . 24
    - 2.1.6.1 Les langages de requêtes : SPARQL . . . 24
    - 2.1.6.2 API classiques . . . 24
    - 2.1.6.3 Programmation orientée ontologie . . . 26
- 2.2 Contribution : Modélisation du domaine en médecine . . . 27
  - 2.2.1 Méthodologie . . . 27
  - 2.2.2 Modélisation conceptuelle du domaine pour l'aide à la décision en biologie . . . 30
  - 2.2.3 Construction d'une ontologie comparative des médicaments (projet VIIIP) . . . 34
  - 2.2.4 Discussion . . . 37
- 2.3 Contribution : Module pour la programmation orientée ontologie . . . 38
  - 2.3.1 Architecture générale . . . 38
  - 2.3.2 Comparaison du modèle objet et des ontologies formelles . . . 38
  - 2.3.3 Implémentation . . . 40
  - 2.3.4 Syntaxe . . . 41
  - 2.3.5 Exemple . . . 44
  - 2.3.6 Raisonnement automatique et monde fermé local . . . 44
  - 2.3.7 Cas d'utilisation : raisonnement automatique sur les contre-indications des médicaments 45
  - 2.3.8 Discussion . . . 48
    - 2.3.8.1 Comparaison à la littérature . . . 49
    - 2.3.8.2 Raisonnement en monde fermé local . . . 49
    - 2.3.8.3 Conclusion . . . 50

3 Visualisation iconique des connaissances **51**
3.1 Introduction et contexte . . . 51
3.1.1 Les icônes simples . . . 51
3.1.2 Les langages iconiques . . . 52
3.1.2.1 La signalisation routière . . . 53
3.1.2.2 Le langage VCM (Visualisation des Connaissances Médicales) . . . 53
3.1.3 Les langages diagrammatiques . . . 56
3.1.4 La formalisation des langages graphiques . . . 56
3.2 Contribution : Création d'un langage graphique pour représenter les plantes médicinales, les recettes et les remèdes . . . 58
3.2.1 Représentation iconique des plantes . . . 58
3.2.1.1 Extraction d'un jeu de données . . . 59
3.2.1.2 Détermination des critères . . . 59
3.2.1.3 Sélection des critères . . . 59
3.2.1.4 Représentation graphique des valeurs des critères . . . 61
3.2.1.5 Assemblage des icônes . . . 61
3.2.2 Représentation graphique des recettes et des remèdes . . . 61
3.2.2.1 Icônes de parties de plantes, de modes de préparation et de modes d'administration . . . 62
3.2.2.2 Diagrammes . . . 62
3.2.3 Discussion . . . 62
3.3 Contribution : Formalisation des langages iconiques . . . 64
3.3.1 Méthode proposée pour la formalisation d'un langage iconique . . . 64
3.3.2 Applications à la signalisation routière . . . 67
3.3.3 Applications à VCM . . . 71
3.3.3.1 L'ontologie VCM . . . 71
3.3.3.2 Vérification de la consistance des icônes . . . 73
3.3.3.3 Autres applications . . . 74
3.3.4 Discussion . . . 74
3.3.4.1 Comparaison à la littérature . . . 74
3.3.4.2 Grammaire ou sémantique ? . . . 75
3.3.4.3 Perspectives . . . 75

4 Visualisation structurelle des connaissances **76**
4.1 Introduction et contexte . . . 76
4.1.1 Visualisation d'information . . . 76
4.1.1.1 Graphes . . . 76
4.1.1.2 Tableaux et représentations tabulaires . . . 77
4.1.1.3 Positions . . . 78
4.1.1.4 Facettes . . . 78
4.1.2 Visualisation d'ensembles non disjoints . . . 80
4.2 Contribution : Les boîtes arc-en-ciel . . . 82
4.2.1 Principe général des boîtes arc-en-ciel . . . 82
4.2.2 Méthodologie pour la mise au point des boîtes arc-en-ciel . . . 84
4.2.3 Optimisation de l'ordre des colonnes . . . 84
4.2.4 Implémentation . . . 85
4.2.5 Application à la comparaison des propriétés des médicaments . . . 85
4.2.5.1 Comparaison des contre-indications . . . 85
4.2.5.2 Comparaison des effets indésirables . . . 87
4.2.6 Application à la visualisation d'annotations de gènes extraites de Gene Ontology . . . 88
4.2.7 Visualisation du raisonnement d'un perceptron et application en antibiothérapie . . . 90
4.2.8 Expérimentations utilisateur . . . 92

4.2.9 Discussion . . . 92
4.2.9.1 Boîtes arc-en-ciel . . . 92
4.2.9.2 Optimisation de l'ordre des colonnes . . . 93
4.2.9.3 Application à la comparaison des propriétés des médicaments . . . 93
4.2.9.4 Application à la visualisation d'annotations de gène . . . 94
4.2.9.5 Application à la visualisation du raisonnement d'un perceptron . . . 94

5 Applications médicales de la visualisation des connaissances 95
5.1 Contribution : Nouvelles applications intégrant VCM . . . 96
5.1.1 Méthode générale pour intégrer VCM dans les applications médicales . . . 96
5.1.2 Intégration dans un dossier patient de cabinet : éO Médecin . . . 97
5.1.3 Intégration dans un dossier patient hospitalier : Crossway Hôpital . . . 99
5.1.4 Intégration dans un moteur de recherche médical : CISMeF . . . 100
5.1.5 Intégration dans les guides de bonnes pratiques : Vidal reco . . . 102
5.1.6 Discussion . . . 103
5.2 Contribution : Le site VIIIP pour la comparaison des médicaments . . . 105
5.2.1 Prototype . . . 105
5.2.2 Présentation du site web . . . 106
5.2.3 Évaluation du site web . . . 107
5.2.3.1 Recrutement des médecins . . . 107
5.2.3.2 Protocole . . . 107
5.2.3.3 Résultats : opinion générale des médecins sur le site web . . . 112
5.2.3.4 Résultats : impact sur le choix de prescrire . . . 112
5.2.3.5 Résultats : discussion générale . . . 113
5.2.4 Discussion . . . 115

6 Discussion et conclusion 117
6.1 De la visualisation des connaissances à la connaissance des visualisations . . . 117
6.1.1 État de l'art . . . 117
6.1.2 Positionnement par rapport aux autres disciplines . . . 121
6.2 Évaluation . . . 122
6.3 Multidisciplinarité . . . 123
6.4 Éthique . . . 124
6.5 Conclusion . . . 125

7 Projet de recherches : vers la visualisation du raisonnement 126

8 CV détaillé 131
8.1 Parcours de recherche . . . 134
8.2 Publications . . . 147
8.3 Enseignements . . . 152

9 Bibliographie 154

Chapitre 1

Introduction

— J'aime les images [...].

— Aujourd'hui elles habillent des produits de consommation, enjolivent des objets, rendent séduisantes des idées douteuses... Mais cette profusion ne nous modifie pas... Elle ne nous apprend rien...

— Ça ne durera pas. Nous ne pourrons pas nous passer de l'essentiel bien longtemps.

(Le combat ordinaire, tome 3 : ce qui est précieux, Manu Larcenet, éditions Dargaud, 2006)

Et si justement les images nous apprenaient quelque chose ?

Les connaissances sont de plus en plus nombreuses et le temps est souvent limité pour se les approprier. Elles évoluent rapidement : en milieu professionnel, cela rend nécessaire la formation continue tout au long de la vie. Dans le domaine médical, ce problème est particulièrement important : les connaissances médicales sont très volumineuses (puisqu'il faut au moins 9 ans d'étude après le bac pour devenir médecin), elles évoluent en permanence, et les erreurs peuvent avoir des conséquences dramatiques pour les patients.

Le format textuel reste la référence pour décrire des connaissances de manière précise et exhaustive, cependant le texte n'est pas nécessairement le format le plus adapté pour donner une vue d'ensemble des connaissances ou pour faciliter la prise de décision. Lors de la lecture d'un article scientifique, qui n'a pas jeté un œil aux figures avant même d'attaquer le texte, afin de se faire rapidement une idée de la thématique et des résultats obtenus ?

La présentation visuelle pourrait donc être une solution à l'explosion des connaissances, elle nécessite néanmoins la mise au point d'approches spécifiques.

Historiquement, la présentation graphique des connaissances est un sujet de recherche très ancien [128]. Une première catégorie de travaux a cherché à utiliser l'analogie pour rendre les connaissances visuelles. Par exemple, des dessins de flacons ont été utilisés dans des traités de médecine pour représenter la couleur des urines au cours de diverses maladies (voir Figure 1.1). Il s'agit de l'uroscopie (examen des urines à des fins diagnostics), science qui préfigure l'urologie.

Une seconde catégorie de travaux a cherché à organiser et à structurer les connaissances de manière visuelle. Les premiers tableaux structurés sont attestés en Mésopotamie dès l'an -1850, pour tenir des comptes. Des tableaux plus conceptuels (par opposition aux tableaux numériques) ont été réalisés par Eusèbe de Césarée au IV$^{\text{ème}}$ siècle pour relier les passages communs des 4 évangiles (selon Matthieu, Marc, Luc et Jean). Eusèbe met au point 10 tables (ou canons), chacune listant les passages communs à un sous-ensemble des évangiles (par exemple Matthieu + Marc). Ce sont les « canons de concordance », qui resteront en usage durant tout le moyen-âge (voir Figure 1.2).

De nos jours, l'image est utilisée pour faire passer des messages simples ou pour présenter une vue d'ensemble de connaissances complexes. Les exemples sont nombreux dans la vie quotidienne : d'une part, les pictogrammes dans les lieux publics, les panneaux routiers et les icônes dans les logiciels informatiques, d'autre part, les tableaux (de plus en plus sophistiqués, tel que le tableau périodique des éléments) ou les graphes (par exemple les cartes de connaissances).

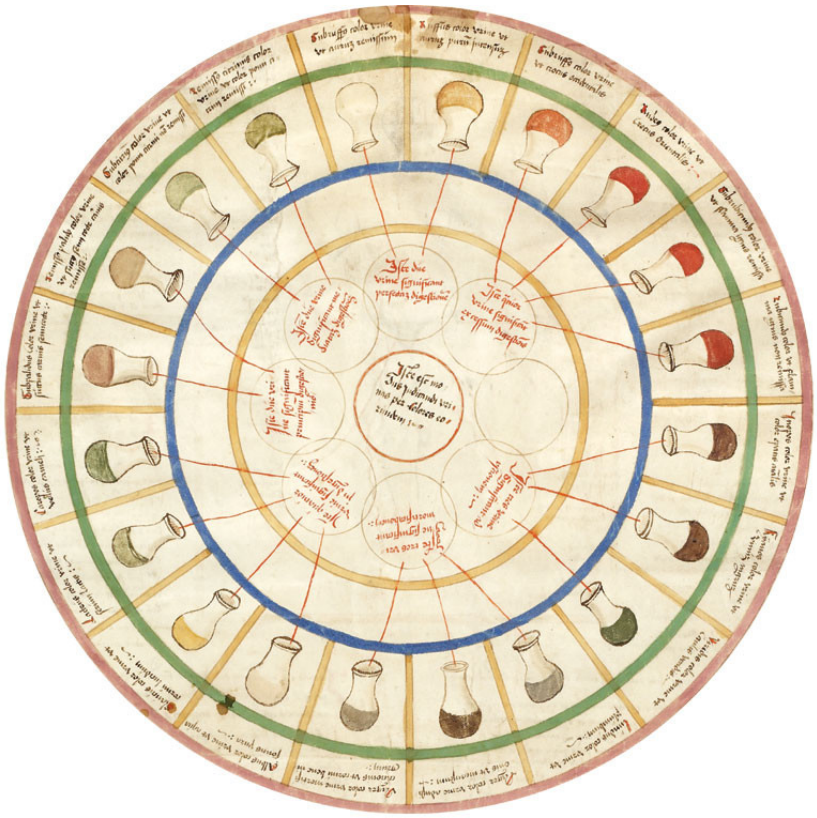


FIGURE 1.1 – Roue des urines, *Epiphanie Medicorum* par Ullrich Pinder, 1506.

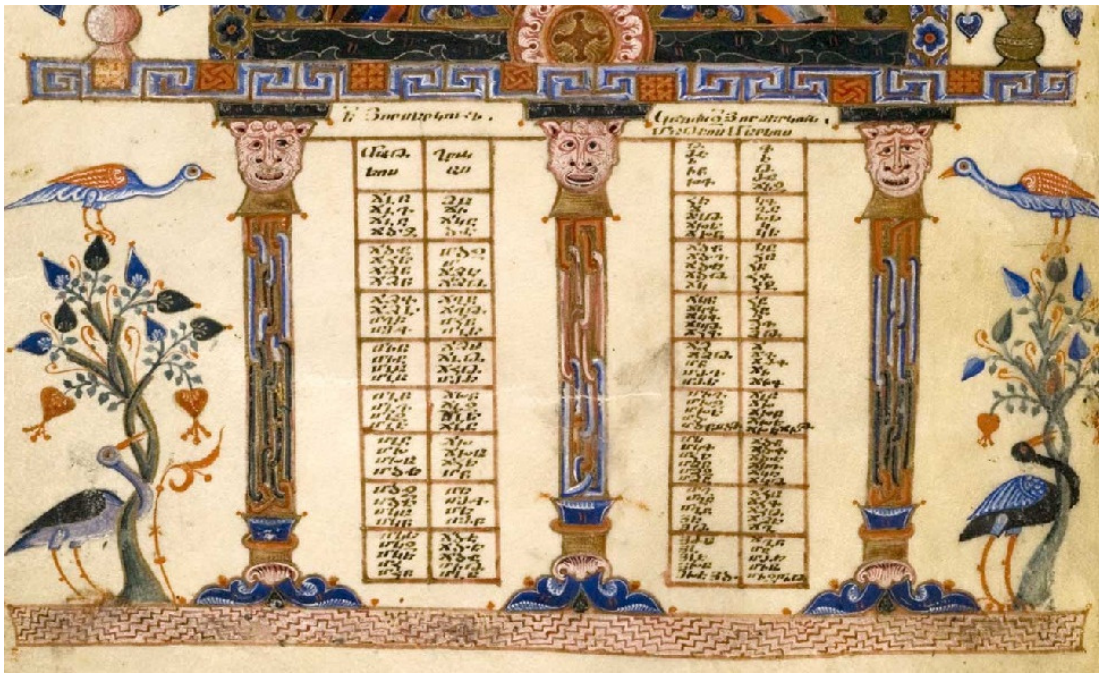


FIGURE 1.2 – Canon de concordance de l'évangéliaire de Lwow, XII^{ème} siècle.

Un exemple très simple est la représentation d'un itinéraire pour aller d'un lieu à un autre (Figure 1.3). L'itinéraire peut être présenté comme une suite d'instructions textuelles, ou visuellement à l'aide d'un trajet sur une carte. Cependant, l'approche visuelle présente plusieurs avantages. Tout d'abord, elle donne une vue d'ensemble, accessible en un seul coup d'oeil. De plus, l'utilisateur reste « maître » et peut modifier le trajet à sa guise, en s'appuyant sur la carte mais en s'écartant momentanément du trajet proposé. Au contraire, dans la version textuelle, l'utilisateur est perdu dès qu'il s'écarte du trajet qui lui est imposé. Enfin, l'aspect visuel de la carte favorise l'apprentissage des lieux et de leurs localisations relatives.

Nous pouvons distinguer deux types d'approches visuelles :

— **Les langages iconiques**, qui consistent à présenter visuellement la nature des connaissances à l'aide d'icônes ou de pictogrammes, en s'appuyant sur l'analogie, c'est-à-dire la ressemblance avec l'objet représenté, ou les conventions existantes.

— **Les techniques de visualisation**, qui consistent à présenter visuellement la structure des connaissances, en visualisant les relations entre les différents éléments de connaissance par exemple à l'aide de graphes, de tableaux ou de cartes.

Ces deux approches ne sont pas exclusives : par exemple une carte (seconde approche) peut très bien faire figurer des pictogrammes (première approche).

Dans le domaine médical, les systèmes d'aide à la décision [38, 97] ont souvent été présentés comme une solution aux problèmes soulevés par le volume et la complexité des connaissances présentes dans les guides de bonnes pratiques cliniques. Les systèmes d'aide à la décision mettent généralement en œuvre des méthodes d'intelligence artificielle afin d'automatiser un raisonnement médical tel que le diagnostic ou le choix du traitement. Ils font ensuite des propositions aux médecins en s'appuyant sur les recommandations des guides (approche « à la demande » ou guidée), ou bien ils déclenchent des alertes lorsque le médecin s'écarte des recommandations (approche automatique ou critique). Le premier système de ce type, Mycin [166], remonte aux années 70 ; un exemple plus récent est le projet ASTI (Aide à la Stratégie Thérapeutique Informatisée) auquel j'ai contribué [173, 107, 174]. Cependant, ces systèmes n'ont pas abouti au succès initialement escompté en termes d'impact clinique [25], et l'acceptation par les médecins reste faible [138] voire mauvaise [176]. Lors du projet ASTI, le mode critique n'a pas pu être déployé en clinique, et le mode guidé a conduit un taux d'adhérence aux recommandations plutôt faible à cause de nombreuses erreurs lors de la saisie des données par le médecin [175]. En revanche, très peu d'approches visuelles ont été proposées pour présenter les connaissances médicales.

Dans ce document, nous proposons une démarche multidisciplinaire pour la visualisation des connaissances, en intégrant les apports de plusieurs branches de l'informatique : la représentation des connaissances, la visualisation d'information et l'aide à la décision. Notre champ d'application sera le domaine médical et l'aide à la décision clinique, avec pour objectif de faciliter et d'accélérer l'accès aux connaissances, de laisser l'utilisateur maître de ses décisions, et de faciliter l'apprentissage (nous retrouvons les trois avantages des approches visuelles déjà mentionnés dans l'exemple de la recherche d'itinéraire).

Après une formation initiale de pharmacie à l'université Grenoble 1, j'ai poursuivi des études en informatique médicale (Diplôme d'Étude Approfondi, puis thèse de science) avant d'obtenir un poste de maître de conférence en informatique (section 27). Je présente ici des travaux de recherche que j'ai réalisés après ma thèse, en tant que post-doc puis après ma nomination comme enseignant-chercheur en 2009 au laboratoire LIM&BIO (Laboratoire d'Informatique Médicale et de BIOinformatique, université Paris 13), devenu par la suite LIMICS en 2014 (Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, université Paris 13, université Paris 6 et UMR_S 1142 INSERM).

Le LIM&BIO puis le LIMICS sont des laboratoires pluridisciplinaires rassemblant des chercheurs d'horizons variés (informaticiens, médecins, pharmaciens, hospitaliers) ayant pour objectif à la fois la mise au point de nouvelles méthodes et l'application de méthodes existantes au domaine de la santé. Le LIMICS possède deux axes de recherche :

1. l'axe « systèmes décisionnels pour la recherche médicale et la prise en charge des patients »,

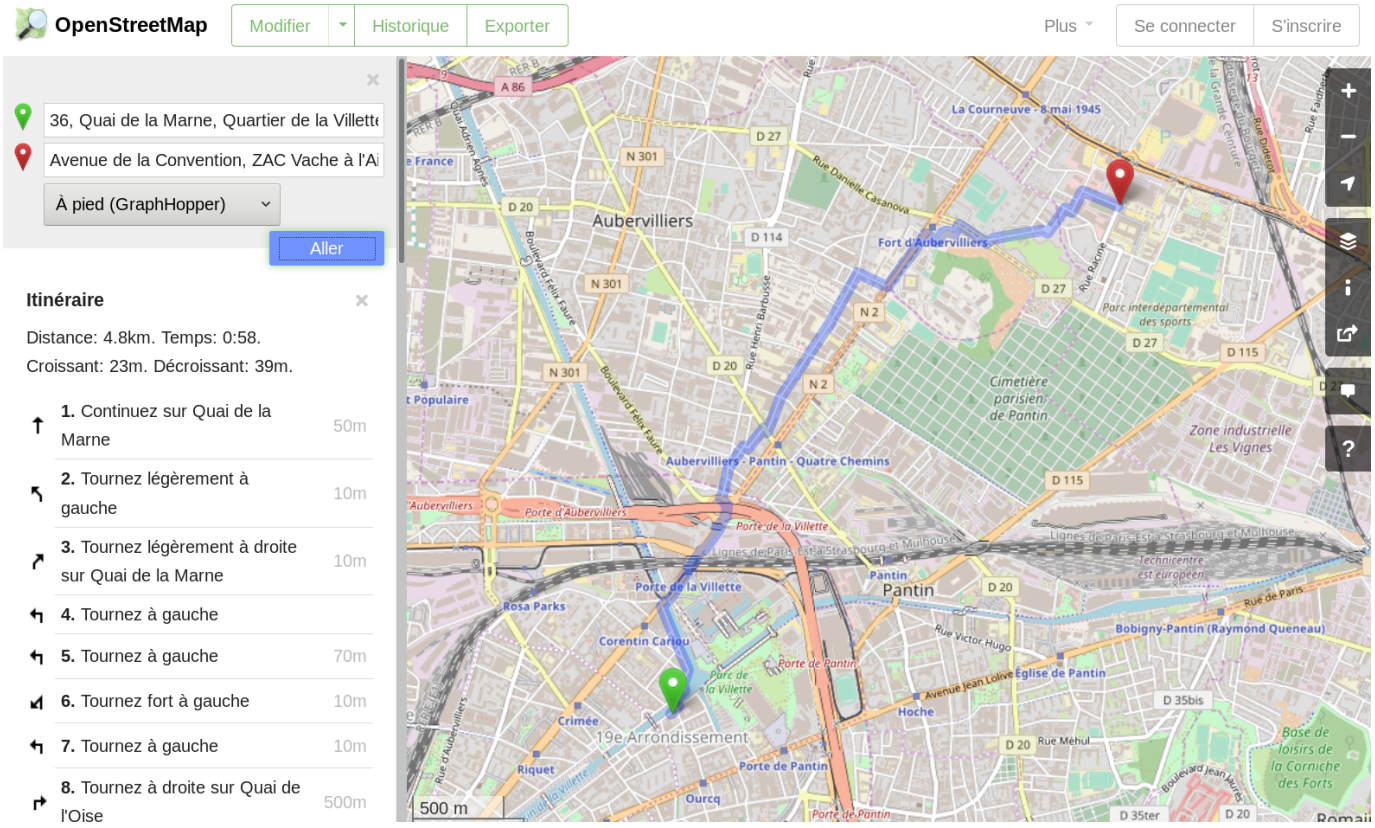


FIGURE 1.3 – Recherche d'itinéraire avec OpenStreetMap : à gauche l'itinéraire est présenté de manière textuelle, sous forme d'une suite d'instructions ; à droite l'itinéraire est présenté de manière visuelle, sous forme d'un plan donnant une vue d'ensemble.

2. l'axe « ingénierie de l'information en santé : méthodes et outils pour l'acquisition, la modélisation et la formalisation des connaissances pour la e-santé ».

Ma thèse a porté sur la conception et l'évaluation d'un langage iconique pour représenter les propriétés des médicaments, appelé VCM (Visualisation des Connaissances en Médecine) et qui a été breveté par la suite. Ces travaux m'ont convaincu du potentiel de la visualisation et de son application dans le domaine médical. En tant que postdoc, j'ai travaillé sur les systèmes « non-visuels » d'aide à la décision clinique lors du projet ASTI et j'ai constaté l'échec relatif que représente cette approche. Par la suite, j'ai été coéditeur de la section « aide à la décision » du *Yearbook* d'informatique médicale de l'IMIA (*International Medical Informatics Association*), et cela m'a conforté dans cette opinion. Lorsque j'ai été nommé enseignant-chercheur, je me suis de nouveau intéressé à la visualisation, d'une part en étendant les utilisations du langage VCM lors du projet L3IM (Langage Iconique et Interfaces Interactives pour la Médecine, financement ANR, Agence Nationale de la Recherche), et d'autre part en étudiant la visualisation d'information pour la comparaison de plusieurs médicaments, lors du projet VIIIP (Visualisation Intégrée de l'Information sur l'Innovation Pharmaceutique, financement ANSM, Agence Nationale de Sécurité du Médicament et des produits de santé), que j'ai coordonné. J'ai également cherché à développer davantage les aspects méthodologiques liés à mes travaux, notamment en formalisant la sémantique du langage VCM.

Au fil de mes recherches, initialement plutôt orientées « informatique médicale », j'ai rencontré plusieurs difficultés récurrentes :

— **pour construire et utiliser des ontologies formelles dans le domaine biomédicale**, à cause de deux spécificités : *la médecine fondée sur les preuves* et l'omniprésence des relations de subsomption, qui compliquent le raisonnement et la manipulation des ontologies dans ce domaine.

— **pour définir la sémantique d'un langage iconique** comme VCM, le besoin d'une sémantique formelle se faisant de plus en plus pressant au fil des années (pour vérifier la cohérence, aligner les icônes avec d'autres ressources, générer un lexique et des libellés).

— **pour présenter et comparer visuellement les propriétés de plusieurs alternatives** afin d'aider un clinicien à prendre une décision et à choisir l'une des alternatives (par exemple un traitement ou un médicament, ou bien un diagnostic possible dans une situation de diagnostic différentiel).

Ces problèmes m'ont amené à mettre au point des méthodes nouvelles qui vont au-delà de l'informatique médicale et relèvent de l'intelligence artificielle et de la visualisation.

Dans le reste de cette introduction, nous présenterons tout d'abord la démarche globale que nous proposons pour la visualisation des connaissances. Nous décrirons ensuite les 3 projets au cours desquels nous avons mis en œuvre cette démarche. Enfin, nous donnerons le plan général de ce document.

1.1 Démarche générale

La Figure 1.4 présente la démarche que nous proposons. Celle-ci comprend quatre grandes étapes (en colonne sur la figure) :

1. **La représentation des connaissances** consiste à modéliser et à structurer les connaissances que l'on souhaite visualiser, par exemple à l'aide d'un modèle conceptuel en UML (*Unified Modeling Language*) ou d'une ontologie formelle, puis à opérationnaliser le modèle.
2. **La visualisation iconique des connaissances** consiste à traduire les connaissances sous forme d'icônes, à l'aide d'un langage iconique comprenant un lexique de pictogramme et une grammaire.
3. **La visualisation structurelle des connaissances** consiste à représenter graphiquement la structure des connaissances, à l'aide de techniques de visualisation.
4. **La conception et l'évaluation d'applications** est l'étape finale, elle consiste à intégrer les méthodes de visualisation des connaissances dans des applications logicielles ou des sites web, et à évaluer les outils résultants.

L'étape 1 est un préalable nécessaire : la structuration des connaissances permet d'effectuer des traitements automatiques sur les connaissances et notamment d'automatiser la génération des visualisations. Cette étape relève de l'intelligence artificielle et plus spécifiquement de la représentation des connaissances. Les étapes 2 et 3 relèvent de la visualisation. Enfin, l'étape 4 relève du domaine d'application, en l'occurrence de l'informatique médicale pour mes travaux.

L'évaluation ne constitue pas une étape isolée. Au contraire, elle doit être intégrée à chaque étape : par exemple, le modèle conceptuel des connaissances doit faire l'objet d'une évaluation, les techniques de visualisation doivent être évaluées, de même que les applications finales. Sur la Figure 1.4, les étiquettes « éval » indiquent pour chaque projet les étapes qui ont fait l'objet d'une évaluation.

1.2 Projets

La démarche que je propose sera illustrée tout au long de ce document au travers de 3 projets de recherche auxquels j'ai participé. La Figure 1.4 montre les 3 projets et leurs différentes étapes ; j'ai fait le choix de ne pas retenir chaque étape pour chaque projet, soit parce que l'étape en question n'était que peu ou pas développée pour ce projet, soit parce que je n'ai pas participé personnellement à cette étape dans le projet. Les sections suivantes décrivent les différents projets.

1.2.1 Projets L3IM et SiFaDO

Les médecins réalisent plusieurs tâches informatiques fastidieuses :

1. des tâches de recherche d'information, comprenant :
   (a) la recherche de documents d'information à l'aide d'un moteur de recherche généraliste ou spécifique tel que CISMeF (Catalogue et Index des Sites Médicaux Francophones, http://cismef.org/)
   (b) la consultation de guides de bonnes pratiques cliniques,

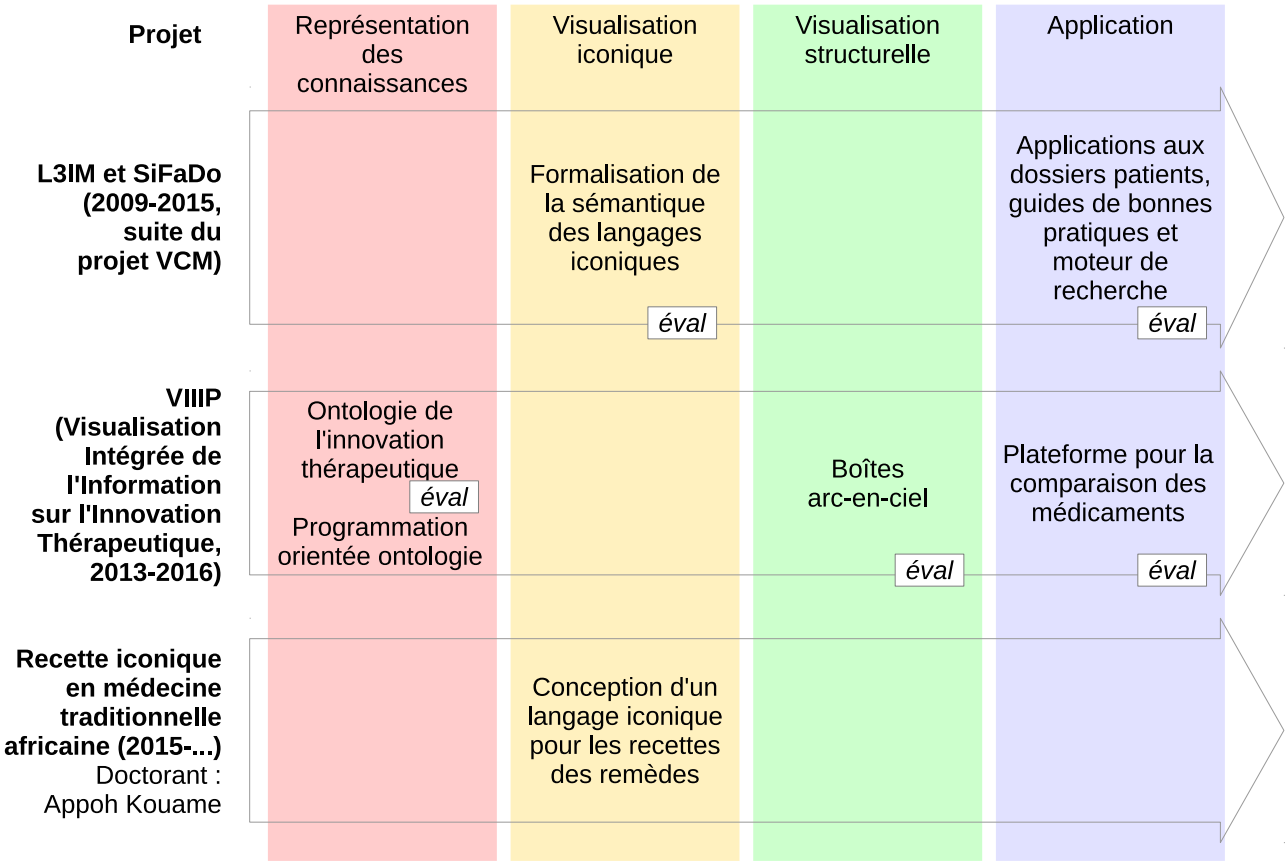


FIGURE 1.4 – Démarche proposée pour la conception d'application à base de visualisation des connaissances. La démarche comporte 4 étapes (en colonne), qui seront illustrées au travers de 3 exemples (en ligne).

(c) la vérification des propriétés des médicaments qu'ils souhaitent prescrire,

(d) la lecture des caractéristiques et des antécédents des patients dans les dossiers patients informatisés,

2. des tâches pour la saisie des données de leurs patients, et notamment le codage des pathologies du patient à l'aide d'une terminologie médicale normalisée telle que la CIM10 (Classification Internationale des Maladies, 10ème révision).

Ces tâches sont longues, complexes et demandent une charge cognitive élevée ; de plus, beaucoup de médecins considèrent qu'elles ne font pas réellement partie du coeur de leur métier. Par conséquent, ces tâches sont souvent négligées, voire ignorées : par exemple il a été montré qu'un médecin consacre au maximum 2 minutes [46] à chercher la réponse à une question qu'il se pose en consultation, après quoi il abandonne la recherche et se débrouille sans la réponse. De même, une part importante des dossiers patients ne sont pas codés, par exemple concernant des dossiers hospitaliers de patientes avec un cancer ovarien, 90% des dossiers mentionnent ce diagnostic en texte mais il n'est codé que dans 45% des dossiers [177].

Lors de mes travaux de thèse [118], j'avais proposé un langage iconique appelé VCM (à l'époque, Visualisation des Connaissances sur le Médicament) pour faciliter la tâche 1.(c) concernant les propriétés des médicaments. Les icônes de ce langage permettaient de représenter graphiquement les propriétés des médicaments (contre-indications, effets indésirables, interactions,...). Ces icônes pouvaient ensuite être insérées en marge du texte pour faciliter la recherche d'une propriété au sein du texte [104]. Les icônes pouvaient aussi être rassemblées sur un schéma anatomique interactif appelé « Monsieur VCM » ; il était alors possible d'accéder au texte correspondant en cliquant sur les icônes. L'évaluation de cette interface dans des conditions contrôlées a montré qu'elle permettait au médecin de répondre à une question deux fois plus rapidement, tout en faisant significativement moins d'erreurs [116].

Le projet L3IM (Langage Iconique et Interfaces Interactives en Médecine) est une suite de mes travaux de thèse. Le projet visait à étendre l'utilisation du langage VCM (devenu Visualisation des Concepts en Médecine) à d'autres connaissances et données médicales : les moteurs de recherche, les guides de bonnes

pratiques et les dossiers patients à l'hôpital, au cabinet et dans les réseaux de soins. Il a été financé par l'ANR TecSan 2008 à hauteur de 947 288 E et regroupait 9 partenaires publics et privés.

Le projet SiFaDO (Saisie Facile de Données médicales) est une suite du projet L3IM portant sur l'aide au codage des dossiers patients. Ce projet a été financé par l'ANR TecSan 2011 à hauteur de 956 878 E et regroupait 9 partenaires publics et privés.

1.2.2 Projet VIIIP

L'innovation pharmaceutique et la mise sur le marché de nouveaux médicaments peuvent conduire à des progrès majeurs en santé. Cependant, les nouveaux médicaments doivent être prescrits en prenant en compte les informations les plus récentes. La principale source d'information des médecins sur les nouveaux médicaments est la visite médicale [130]. Mais elle n'est pas indépendante des laboratoires pharmaceutiques et l'information est souvent incomplète [171].

Les journaux indépendants sont une meilleure source, mais pas toujours à jour, de plus ils présentent trop souvent des avis d'experts « prémâchés », ce qui soulève divers problèmes :

- les avis peuvent diverger d'un expert à l'autre,
- l'indépendance des experts est souvent problématique car leurs recherches sont parfois financées par l'industrie,
- certains médecins sont méfiants vis-à-vis des experts, à cause des liens entre experts et industrie,
- les avis d'experts ne prennent pas en compte la patientèle du médecin.

Le projet VIIIP (Visualisation Intégrée de l'Information sur l'Innovation Pharmaceutique) a pour objectif la conception d'une plate-forme web pour collecter et présenter une information fiable, indépendante et comparative sur les nouveaux médicaments. Cette plate-forme doit permettre aux médecins de se forger leur propre opinion sur le nouveau médicament, à partir de données « brutes » objectives présentées de manière visuelle et d'une comparaison entre le nouveau médicament et les médicaments existants.

Ce projet a été financé par l'ANSM (Agence Nationale de Sécurité du Médicament et des produits de santé, 2013-2016) à hauteur de 190 625 E. J'en ai assuré le montage, la rédaction puis la coordination.

1.2.3 Recette iconique en médecine traditionnelle africaine

En Afrique de l'Ouest, la médecine traditionnelle reste très présente en parallèle de la médecine moderne conventionnelle, et les deux formes de médecine sont utilisées de manière complémentaire par les patients. Cette médecine traditionnelle s'appuie sur des remèdes, la plupart étant à base de plantes. Des recettes simples permettent la fabrication de ces remèdes par les tradi-praticiens, par exemple en sélectionnant la bonne plante, en effectuant une décoction puis en la faisant boire au patient.

Ces recettes sont transmises de manière orale entre tradi-praticiens, cependant la disparition d'un tradi-praticien entraîne souvent la perte de ses connaissances. Le besoin de conserver et de fixer les connaissances se fait donc sentir, mais l'écriture des recettes est difficile car beaucoup de tradi-praticiens sont illettrés. Pour résoudre ce problème, le projet propose de concevoir un wiki sémantique pour la faciliter la collaboration entre tradi-praticiens, ce wiki intégrant notamment un langage iconique pour représenter graphiquement les recettes et leur utilisation. De cette manière, les tradi-praticiens analphabètes pourront consulter les recettes ou en rédiger.

Ce projet correspond aux travaux réalisés par Appoh Kouame durant sa thèse d'informatique (2013-..., taux d'encadrement 50%).

1.3 Plan

Afin de mettre en valeur les aspects méthodologiques, nous avons choisi d'organiser notre plan en fonction des 4 grandes étapes méthodologiques plutôt qu'en fonction des projets.

Le chapitre 2, Représentation des connaissances, abordera la structuration, la formalisation et l'opérationnalisation des connaissances. Je décrirai la méthodologie mise au point au LIMICS pour la modélisation conceptuelle du domaine en santé, et je donnerai deux exemples d'application de cette méthodologie, sur les

recommandations de bonnes pratiques cliniques pour les examens de biologie et sur les connaissances sur le médicament lors du projet VIIIP. Dans une seconde partie, je décrirai la mise au point d'un module pour la programmation orientée ontologie en langage Python, et son utilisation lors du projet VIIIP et lors du projet de recette iconique pour la médecine traditionnelle.

Le chapitre 3, Visualisation iconique des connaissances, abordera l'iconisation des connaissances. Je décrirai tout d'abord la conception d'un langage iconique et diagrammatique pour représenter les recettes de remèdes en médecine traditionnelle d'Afrique de l'Ouest (thèse d'Appoh Kouame). Je proposerai ensuite une méthode pour formaliser la sémantique d'un langage iconique, en prenant pour exemple les panneaux routiers et le langage VCM. Je décrirai aussi les applications rendues possibles par cette formalisation.

Le chapitre 4, Visualisation structurelle des connaissances, abordera les techniques de visualisation d'information. Je décrirai les boîtes arc-en-ciel, une nouvelle technique que j'ai développée pour visualiser des ensembles non disjoints, ainsi que deux applications : la comparaison des propriétés des médicaments dans le cadre de VIIIP et la visualisation d'extraits de Gene Ontology. Je donnerai également les résultats de plusieurs évaluations portant sur cette technique de visualisation.

Le chapitre 5, Applications médicales de la visualisation des connaissances, présentera deux exemples de visualisation des connaissances en santé. Le premier porte sur l'intégration du langage iconique VCM au sein de 4 nouvelles applications : le dossier patient en cabinet, le dossier patient hospitalier, les moteurs de recherche et les guides de bonnes pratiques cliniques. Le second est le site web VIIIP comparant les propriétés des nouveaux médicaments à celles des médicaments existants.

Le chapitre 6, Discussion et conclusion, abordera plusieurs points de discussion, notamment sur l'évaluation, sur les questions de la multidisciplinarité et de l'éthique. Je présenterai également un état de l'art sur la visualisation des connaissances.

Le chapitre 7, Projet de recherches : vers la visualisation du raisonnement, présentera le projet de recherche que je me donne pour les années à venir.

Le chapitre 8, CV détaillé, présentera l'ensemble de mon parcours et la liste de mes publications.

Chapitre 2

Représentation des connaissances

« Lorsqu'il dit, par exemple, qu'un personnage a été empoisonné avec deux grains et demi de cyanure, cela ne signifie pas qu'il est nécessairement mort, raison pour laquelle on le voit réapparaître à la fin du roman. »
(Le réveil d'Ymir, Nicolas Bouchard, éditions Mnémos, 2001)

2.1 Introduction et contexte

La représentation des connaissances est un champ de recherche qui vise à représenter, à structurer, à organiser et à formaliser la connaissance.

Dans ce chapitre, nous définirons tout d'abord la connaissance, puis nous présenterons les connaissances médicales au travers de deux exemples. Nous décrirons ensuite rapidement les formalismes existants pour représenter les connaissances, les méthodologies proposées pour la modélisation d'un domaine de connaissance et les outils pour l'opérationnalisation des connaissances. Nous détaillerons davantage deux difficultés que nous avons rencontrées dans le domaine biomédicale : le raisonnement en monde fermé local et la manipulation des classes.

Enfin, nous présenterons deux contributions : la première portera sur une méthodologie de modélisation spécifiquement adaptée au domaine biomédical, et la seconde sur un module pour la programmation orientée ontologie en Python, qui est à même de résoudre les deux difficultés mentionnées.

2.1.1 Définitions

Les notions de données, d'information et de connaissance sont sujettes à débats. Les définitions que nous donnerons ici serviront à clarifier la suite de l'exposé mais n'ont pas vocation à être universelles. F. Nake [143] propose une définition de ces 3 termes à partir de la sémiotique et de la triade peircienne (dans laquelle R est la Représentation, O l'Objet représenté et I l'Interprétation).

La ***donnée*** correspond à la dimension *syntaxique*. Une donnée est un *signe*, par exemple « la glycémie du patient A est de 1,5g/l ». La représentation (R) est interprétée comme étant elle même (dans l'exemple, R = O = I = glycémie de 1,5g/l).

L'***information*** correspond à la dimension *sémantique*. Une information est une donnée à laquelle on associe un sens, par exemple « la glycémie du patient A est de 1,5g/l donc le patient A est diabétique » [1]. Elle est composée d'une représentation et d'un objet (R - O) ; la représentation représente l'objet qui est interprété comme étant lui-même (dans l'exemple, R = glycémie de 1,5g/l, I = O = le patient A est diabétique).

La ***connaissance*** correspond à la dimension *pragmatique*. Une connaissance est une information réutilisable dans un autre contexte ; cette notion est subjective et dépend donc de l'utilisateur. Par exemple « si la glycémie est de 1,5g/l, le patient est diabétique » ; cette règle est générale et réutilisable sur n'importe quel patient, un exemple d'interprétation subjective est « donc le patient B hyperglycémique qui est dans mon

1. En pratique médicale, plusieurs mesures de la glycémie sont nécessaires pour réaliser le diagnostic du diabète.

service est aussi diabétique ». En conséquence, puisque la connaissance est réutilisable, il est intéressant de l'apprendre et de la retenir. La connaissance est une triade complète ((R - O) - I) : R représente O qui est interprété pour former une interprétation I, différente de O (dans l'exemple, R = glycémie de 1,5g/l, O = A est diabétique, I = B est diabétique).

Certains auteurs [3] ajoutent un quatrième niveau, la ***sagesse***, correspondant à la compréhension des principes sous-jacents à la connaissance ou à l'application des connaissances.

Dans certains domaines, l'usage fait que l'on utilise un mot plutôt qu'un autre. Par exemple l'usage veut que l'on parle d' « information sur le médicament » alors que ces informations sont en réalité des connaissances : en effet les indications, contre-indications, effets indésirables,... d'un médicament s'appliquent aux différents patients et les professionnels de santé doivent apprendre les principales propriétés des grandes classes de médicaments. Dans la suite de ce document, nous nous efforcerons de nous tenir aux définitions données dans cette section, et nous parlerons donc plutôt de « connaissances sur le médicament ».

2.1.2 Les connaissances médicales

Dans le domaine médical, de nombreuses connaissances sont disponibles. Celles-ci ne se limitent pas aux connaissances transmises lors de la formation initiale des professionnels de santé, car ces connaissances évoluent et la formation continue est une obligation pour les professionnels de santé. Les connaissances sont très majoritairement diffusées sous forme de textes. Nous donnerons deux exemples :

— **Les guides de bonnes pratiques cliniques (GBP)** : Ces guides décrivent les prises en charge médicales recommandées pour une catégorie de patient ou pour un traitement particulier [50]. Il existe notamment des guides pour la prise en charge d'une maladie, avec des recommandations pour la prévention de la maladie, pour le diagnostic, pour le traitement et pour le suivi. D'autres guides font le point sur un traitement ou une méthode diagnostique.
Par exemple, le guide européen pour la prise en charge de l'hypertension [127] recommande de prescrire en première intention chez les patients hypertendus diabétiques un médicament appartenant à l'une des deux classes suivantes : les IEC (Inhibiteurs de l'Enzyme de Conversion) ou les ARA2 (Antagonistes des Récepteurs de l'Angiotensine 2).
Les GBP sont publiés à l'initiative soit d'une agence gouvernementale telle que, en France, la Haute Autorité de Santé (HAS) ou l'Agence Nationale pour la Sécurité du Médicament et des produits de santé (ANSM), soit d'une société savante telle que l'*European Society of Hypertension* (ESH). Ils sont rédigés par des groupes d'experts réunissant des personnes ayant des intérêts divers voire conflictuels, et qui fonctionnent sur le mode du consensus.
L'impact des GBP est réel mais limité dans leur format textuel [63], les médecins n'ayant que rarement le temps de les consulter. L'intégration des GBP au sein de systèmes d'aide à la décision permet d'en augmenter l'impact [53].

— **Les Résumés des Caractéristiques Produit (RCP) et les notices des médicaments** : ces deux textes existent pour chacun des médicaments disponibles sur le marché. Le RCP décrit l'ensemble des propriétés cliniques du médicament (notamment les indications, les contre-indications, les interactions, les effets indésirables, les précautions d'emploi et les excipients à effet notoire), les propriétés pharmacologiques et pharmacocinétiques. Le RCP est à destination des professionnels de santé. La notice patient correspond au document papier inséré dans l'emballage du médicament, elle reprend un résumé des propriétés du médicament dans un langage simplifié.
Les connaissances contenues dans les RCP sont ensuite rassemblées dans des bases médicaments (en France, quatre bases existent : Vidal, la Banque Claude Bernard BCB, Thériaque et Thésorimed), cependant la qualité des données de ces bases est très variable, d'où la nécessité de se référer au texte du RCP.
Le domaine du médicament est particulièrement complexe car il fait intervenir de nombreux acteurs [17] et en particulier l'industrie pharmaceutique, pour laquelle la connaissance est associée à des enjeux économiques très importants. Les industriels n'hésitent pas à diffuser des connaissances partielles voire biaisées pour favoriser leurs médicaments, notamment *via* les visiteurs médicaux. Ces visiteurs sont payés par l'industrie pour présenter aux médecins les nouveaux médicaments des laboratoires et ils

représentent la première source des médecins sur l'innovation pharmaceutique [130, 153]. Cependant, de nombreuses études ont montré que les connaissances médicales des visiteurs étaient limitées [121], que les connaissances délivrées étaient incomplètes [122] et que la visite médicale diminue la qualité des prescriptions des médecins [140].

Les connaissances médicales ont plusieurs spécificités. Tout d'abord, en médecine, la connaissance est autant que possible fondée sur les preuves (*evidence-based medicine*, EBM) : c'est-à-dire que la connaissance ne provient pas d'un expert ou d'un groupe d'expert, mais d'une preuve établie à l'aide d'un test statistique lors d'un *essai clinique*. Un exemple type d'essai clinique consiste à prendre deux groupes de patients, chaque groupe recevant un traitement différent (médicament A ou B) en aveugle (*i.e.* sans savoir lequel). À la fin de l'étude, on évalue l'état du patient à l'aide d'indicateurs de la maladie (marqueur biologique, niveau des symptômes, *etc*). Si la différence est statistiquement significative entre les deux groupes de patient, l'un des deux médicaments est meilleur que l'autre et cela produit une nouvelle connaissance médicale.

Une autre particularité de la médecine est de faire intervenir la notion de *rapport bénéfice risque* : le rapport entre les bénéfices attendus d'une prise en charge médicale et les risques qu'elle fait courir. En médecine, il est possible de prendre une décision associée à un risque élevé, du moment que le bénéfice attendu le justifie. Au contraire, dans la plupart des autres domaines, le risque doit être minimisé et ne peut pas être « négocié » en échange d'un bénéfice plus grand (par exemple accepter un risque d'accident de 10% en échange d'une réduction du prix de son billet d'avion n'est pas une option envisageable).

Une troisième particularité est de faire intervenir des *niveaux de granularité* multiples et nombreux. À titre d'illustration, les maladies peuvent être exprimées à des niveaux de granularité variables, par exemple maladie inflammatoire de l'intestin, maladie de Crohn, maladie de Crohn sévère avec manifestation cutanée, *etc*. Les antécédents et les pathologies d'un patient sont souvent décrits à un niveau fin, par exemple « M X a des antécédents de maladie de Crohn », tandis que les connaissances médicales (contre-indications des médicaments, profils patients des guides de bonnes pratiques) sont décrites à des niveaux plus généraux, par exemple « ce médicament est contre-indiqué avec les maladies inflammatoires de l'intestin » (ce qui inclut donc la maladie de Crohn). Les médicaments peuvent aussi être exprimés à plusieurs niveaux de granularité : antalgique, anti-inflammatoire non stéroïdien (AINS), aspirine, aspirine dans l'indication antiagrégant plaquettaire, aspirine 1g en comprimé effervescent, *etc*. Cette particularité complique la représentation des connaissances médicales, car lorsque l'on considère un concept médical, il est quasiment toujours possible de le préciser davantage pour arriver à un concept plus fin.

Les connaissances médicales sont intéressantes à plusieurs titres. Premièrement, les enjeux sociétaux de ces connaissances sont très importants et touchent l'ensemble de la population : tout le monde a un jour rencontré des problèmes de santé, a été amené à prendre des médicaments ou à effectuer des actes de prévention. Une conséquence de l'importance de ces enjeux est que, d'un point de vue très terre-à-terre, il est relativement facile d'obtenir des financements pour des recherches sur ces thématiques.

De plus, d'un point de vue scientifique, ces connaissances sont parmi les plus complexes qui existent à ce jour : en effet, les études de médecine sont les plus longues (9 ans d'étude après le bac ou plus). Par conséquent, nous pouvons raisonnablement penser que les méthodes mises au point sur les connaissances médicales pourront fonctionner sur les connaissances d'autres domaines (selon le vieil adage « qui peut le plus peut le moins »).

2.1.3 Formalismes de représentation des connaissances

De nombreux formalismes ont été proposés pour la représentation des connaissances. Dans le domaine de la modélisation objet, les différentes méthodes et représentations ont convergé pour aboutir **au langage UML** (*Unified Modeling Language*) à la fin des années 90. En particulier, les diagrammes de classe UML permettent de représenter les classes, attributs, relations et méthodes issus de la conceptualisation.

Les logiques de description [13] (*Description Logics*, DL) permettent de formaliser la connaissance d'un domaine par la description des différents concepts de ce domaine et de la sémantique associée. La sémantique est exprimée à l'aide de prédicats de la logique du premier ordre. Afin de garder un système décidable, seul un sous-ensemble de la logique du premier ordre est utilisé. Le sous-ensemble est choisi selon les besoins du domaine. C'est pourquoi l'on parle de « logiques de description » au pluriel, car il existe un grand nombre

de logiques différentes, en fonction des sous-ensembles retenus : avec ou sans la négation, avec ou sans les unions,...

Les différentes logiques de descriptions sont nommées à l'aide de lettres représentant les éléments de la logique du premier ordre qu'elles autorisent. Par exemple, la logique $\mathcal{AL}$ autorise le concept universel $\top$, le concept vide $\bot$, la négation d'un concept atomique $\neg A$, l'intersection $A \sqcap B$, la restriction $\forall R.C$ et la quantificateur existentiel $\exists R.\top$ (mais limitée au concept universel $\top$). La logique $\mathcal{ALU}$ est une extension de la logique $\mathcal{AL}$ qui autorise en plus l'union $A \sqcup B$, *etc.*

Une ontologie est une description formelle de l'ensemble des concepts d'un domaine et des relations existant entre ces concepts. Les ontologies s'appuient pour cela sur les logiques de description. Le langage OWL (*Ontology Web Language*, standardisé par le W3C, *World Wide Web Consortium*) est l'un des plus utilisés pour formaliser les ontologies ; OWL pouvant prendre en charge un grand nombre de logiques de description différentes. Il existe de nombreux outils adaptés aux ontologies, comme l'éditeur Protégé.

Les ontologies ont deux finalités principales :

— **Le raisonnement automatique** : puisque l'ensemble des concepts, des relations et de leurs propriétés est décrit de manière formelle, il devient possible d'effectuer automatiquement des inférences logiques. Des raisonneurs indépendants du domaine de connaissance sont généralement utilisés à cette fin.

— **La réutilisation des connaissances** : puisqu'une ontologie peut importer d'autres ontologies, il est possible de réutiliser des connaissances déjà formalisées et publiées par d'autres personnes. Toutes les ontologies partagent de fait le même espace de nommage. Les connaissances peuvent ainsi être reliées entre elles pour aboutir au *web sémantique*.

Les ontologies reposent généralement sur **l'hypothèse du monde ouvert** (*open-world assumption*). Selon cette hypothèse, tout fait qui n'est ni asserté dans l'ontologie ni interdit par celle-ci est considéré comme possible. L'hypothèse contraire, dite du monde fermé, considère ces faits comme impossibles.

Ces dernières années, les ontologies sont utilisées largement dans le domaine biomédical [162, 196]. Elles influencent aussi de plus en plus les terminologies comme *Gene Ontology* (utilisé pour l'annotation des gènes) ou la SNOMED CT (*Systematized Nomenclature of Medicine - Clinical Terms*, utilisée pour le codage des dossiers patients), au point que l'on parle désormais de « ressources termino-ontologiques ».

Dans la suite de cette section, nous donnons un rappel des logiques de description et une définition formelle d'une ontologie ; pour plus de détails le lecteur pourra se référer à la littérature [13].

Les logiques de description correspondent à des fragments de la logique du premier ordre (*first-order predicate logic*) avec une seule variable libre. Elles sont équipées d'une sémantique formelle, fondée sur la logique. Une ontologie $\mathcal{O}$ peut être définie comme un ensemble d'axiomes logiques Φ. Ces axiomes sont construits à partir d'un ensemble d'individus $\mathbb{I} = \{i, j, ...\}$, d'un ensemble de concepts $\mathbb{C} = \{C, D, ...\}$, un ensemble de rôles $\mathbb{R} = \{R, S, ...\}$, et un ensemble de constructeurs $\mathbb{S}$. Ces constructeurs permettent de combiner des concepts et/ou des rôles (selon le constructeur) et définissent un nouveau concept ou rôle. Les types d'axiomes et les constructeurs dépendent de la logique de description considérée. Les principaux types d'axiomes sont les suivants : $C \sqsubseteq D$ (subsomption), $C \equiv D$ (équivalence), $C(i)$ (instantiation) et $R(i, j)$ (relation) et, pour la logique de description $\mathcal{ALCI}$, $\mathbb{S} = \{\neg, \sqcap, \sqcup, \forall, \exists, R^-\}$. Le tableau 2.1 liste les principaux types d'axiomes et les constructeurs.

L'ensemble des axiomes de l'ontologie $\mathcal{O}$ est classiquement divisé en une TBox $\mathcal{T}$ (*Terminological Box*, comprenant les axiomes portant sur les concepts et les rôles) et une ABox $\mathcal{A}$ (*Assertional Box*, comprenant les axiomes portant sur les individus et les relations).

Les logiques de description ont une sémantique théorique, qui est définie en termes d'interprétations. Pour une ontologie $\mathcal{O}$, une interprétation $\mathcal{I} = (\Delta, f)$ est un couple comprenant un domaine d'interprétation Δ (qui est un ensemble non vide) et une fonction d'interprétation $f()$ qui, pour chaque individu, concept, rôle, expression composée (définie à l'aide de constructeurs) et axiome, retourne son interprétation dans le domaine Δ, comme suit :

$$f(i \in \mathbb{I}) \in \Delta$$
$$f(A \in \mathbb{C}) \subseteq \Delta$$
$$f(R \in \mathbb{R}) \subseteq \Delta \times \Delta$$

La dernière colonne du tableau 2.1 montre l'interprétation associée à chaque axiome et constructeur. En utilisant la fonction d'interprétation, les axiomes logiques d'une ontologie peuvent être transformés en

		Syntaxe	Description	Sémantique
Constantes		$\top$	Thing / Top	Δ
		$\bot$	Nothing / Bottom	$\emptyset$ (ensemble vide)
Axiomes	TBox	$A \sqsubseteq B$	A est subsumé par B	$f(A) \subseteq f(B)$
		$A \equiv B$	A est équivalent à B	$f(A) = f(B)$
	ABox	$A(i)$	i est une instance de A	$f(i) \in f(A)$
		$R(i,j)$	i et j sont reliés par R	$(f(i), f(j)) \in f(R)$
Constructeurs		$\neg A$	Complément de A	$\Delta \backslash f(A)$
		$A \sqcap B$	Intersection de A et B	$f(A) \cap f(B)$
		$A \sqcup B$	Union de A et B	$f(A) \cup f(B)$
		$\exists R.B$	Restriction existentielle	$\{a \in \Delta \mid \exists b, (a,b) \in f(R) \wedge b \in f(B)\}$
		$\forall R.B$	Restriction universelle	$\{a \in \Delta \mid \forall b, (a,b) \in f(R) \rightarrow b \in f(B)\}$
		R^-	Rôle inverse	$\{(a,b) \in \Delta \times \Delta \mid (b,a) \in f(R)\}$
Disjonction		$A \sqcap B \sqsubseteq \bot$	A et B sont disjoints	$f(A) \cap f(B) = \emptyset$

TABLEAU 2.1 – Syntaxe et sémantique d'une ontologie en logique de description. A et B sont des concepts, R est un rôle, i et j des individus. La disjonction a été ajoutée pour faciliter sa compréhension, mais elle peut se déduire de l'intersection et de la subsomption.

formules ensemblistes, qui expriment la sémantique de l'ontologie. Par exemple, l'axiome $A \sqsubseteq B \sqcap C$ est traduit en $f(A \sqsubseteq B \sqcap C) = f(A) \subseteq f(B) \cap f(C)$.

Une interprétation $\mathcal{I}$ satisfait une ontologie $\mathcal{O}$ si elle satisfait tous les axiomes de $\mathcal{O}$ (*i.e.* $\forall \Phi \in \mathcal{O}, f(\Phi)$ est vrai). Une ontologie $\mathcal{O}$ est *consistante* s'il existe au moins une interprétation $\mathcal{I}$ qui satisfasse $\mathcal{O}$ ($\mathcal{O}$ est *inconsistante* dans le cas contraire). Un concept C est *satisfiable* dans $\mathcal{O}$ si (et seulement si) il existe au moins une interprétation $\mathcal{I}$ qui satisfasse $\mathcal{O}$ telle que $f(C) \neq \emptyset$ (*i.e.* il existe un individu i qui appartienne à C).

Pour une ontologie $\mathcal{O}$, $\Phi \in \mathcal{O}$ signifie que l'axiome Φ appartient à l'ensemble $\mathcal{O}$ (*i.e.* l'axiome a été *asserté* dans l'ontologie), et $\mathcal{O} \models \Phi$ signifie que l'axiome Φ peut être *inféré* à partir des axiomes dans $\mathcal{O}$. La simple transitivité entre les relations de subsomption n'est généralement pas considérée comme une inférence de ce point de vue : par exemple si $\mathcal{O} = \{A \sqsubseteq B, B \sqsubseteq C\}$ nous considérons que $A \sqsubseteq C \in \mathcal{O}$. De fait, les relations de subsomption indirectes peuvent être calculées facilement et un raisonneur n'est pas nécessaire pour cette tâche.

2.1.4 Spécificités des ontologies dans le domaine biomédical

Le domaine biomédical est très complexe et cette complexité conduit à deux difficultés propres aux ontologies de ce domaine :

1. Nous avons vu que la connaissance médicale était fondée sur une preuve statistique (*médecine fondée sur les preuves*). Par conséquent, **l'hypothèse du monde ouvert qui est sous-jacente aux ontologies n'est pas adaptée aux connaissances médicales**, car le raisonnement médical ne doit pas faire intervenir de nouvelles hypothèses qui remettraient en cause ou iraient au-delà les connaissances établies : seules les connaissances existantes sont fondées sur des preuves et donc elles seules doivent être prises en compte.

   Le monde ouvert est une hypothèse appropriée lorsque l'on raisonne sur les informations propres à un patient, par exemple cela permet au raisonneur de faire des hypothèses concernant des pathologies ou des états du patient encore inconnus dans un système d'aide au diagnostic. Au contraire, le monde ouvert n'est pas approprié lorsque l'on raisonne sur des connaissances médicales telles que les connaissances sur le médicament ou sur les recommandations pour la prise en charge des maladies (voir section 2.1.2). En effet, les connaissances médicales doivent être validées par des experts et des agences nationales : lorsqu'un médecin utilise un système d'aide à la décision, il ne s'attend donc pas

à ce que celui-ci fasse des hypothèses concernant des connaissances médicales encore inconnues à ce jour.

Par exemple, lorsqu'un médecin considère les effets indésirables des médicaments, il considère que tous les effets possibles sont connus (et listés dans les RCP) et que seuls les effets connus peuvent se produire, même si cela n'est pas entièrement vrai. Par conséquent, un système d'aide à la décision est supposé raisonner uniquement sur les effets indésirables connus et ne pas faire d'hypothèse concernant l'existence d'autres effets potentiels. De telles hypothèses conduiraient à des messages d'alerte stupides tels que : « Le patient présente une hyperkaliémie ; le médicament que vous lui avez prescrit n'a pas pour effet indésirable l'hyperkaliémie mais ce médicament pourrait avoir un effet indésirable d'hyperkaliémie encore inconnu. Veuillez vérifier le traitement ! ».

2. Nous avons vu que les principaux concepts médicaux font intervenir différents niveaux de granularité, notamment pour les maladies et les médicaments. Par conséquent, **de nombreuses entités biomédicales ne peuvent pas être représentées par des individus dans les ontologies, et nécessitent au contraire d'utiliser des classes pour les représenter**.

   Des relations d'héritage (*est-un*) sont nécessaires pour représenter les relations entre les différents niveaux de granularité (par exemple maladie de Crohn *est une* maladie inflammatoire de l'intestin). Les relations *est-un* n'existant pas entre individus, il n'est pas possible de concevoir les maladies comme des individus de la classe Maladie. Il est donc nécessaire de représenter les maladies et les médicaments par des classes dans les ontologies biomédicales. Cependant la manipulation des classes est plus complexe que celle des individus. Par exemple, pour asserter que la maladie de Crohn est chronique, si celle-ci est représentée par un individu dans une ontologie, il suffit d'un triplet RDF :

   (maladie_de_crohn, est_chronique, vrai)

   Tandis que si elle est représentée par une classe OWL, il faut définir une restriction qui nécessite 4 triplets et un noeud anonyme (*blank node*) :

   (Maladie_de_crohn, rdfs :subclassOf, _blank_node_x)
   (_blank_node_x, rdf :type, owl :Restriction)
   (_blank_node_x, owl :onProperty, est_chronique)
   (_blank_node_x, owl :hasValue, vrai)

   En conséquence, cela complexifie l'utilisation des ontologies dans le domaine biomédical.

2.1.5 Méthodologies pour la modélisation du domaine

La modélisation du domaine a pour objectif la construction d'un modèle conceptuel (par exemple en UML) ou d'une ontologie portant sur un domaine de connaissance donné. Parmi les différentes méthodologies proposées, nous distinguerons deux approches : les *méthodologies collaboratives*, qui cherchent à faire collaborer plusieurs personnes avec des compétences diverses, et les *méthodes automatiques et semi-automatiques*, qui cherchent à automatiser la production du modèle en s'appuyant sur des ressources ou des corpus de textes. Nous verrons également les approches proposées spécifiquement dans le domaine biomédical.

2.1.5.1 Méthodologies collaboratives

Les méthodologies collaboratives cherchent à construire un modèle du domaine en faisant collaborer plusieurs personnes réunissant à la fois des profils de compétences variées et des intérêts divers. Ces approches ont notamment été développées pour la construction d'ontologies. E. Simperl *et al.* [168] ont fait récemment une revue détaillée couvrant 6 méthodologies pour la construction collaborative d'ontologies : la méthode Holsapple and Joshi, Dogma-Mess, DILIGENT (*DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies*), HCOME (*Human Centered Ontology Engineering Methodology*), RapidOWL et Ontology maturing.

La construction collaborative peut être *centralisée* (les participants se réunissent en présentiel) ou *décentralisée* (les participants utilisent des outils collaboratifs tels que des Wikis pour échanger).

Parmi les participants, jusqu'à 5 *rôles* sont distingués :

— **l'utilisateur final** (*final user* en anglais) de l'ontologie ou de l'application qui exploitera celle-ci,
— **l'expert du domaine** (*domain expert*), qui maîtrise le domaine sur lequel portera l'ontologie et les connaissances qu'elle doit représenter,
— **l'ingénieur de la connaissance** (*knowledge engineer*), qui est chargé d'éliciter les connaissances des experts pour arriver à un modèle conceptuel du domaine,
— **l'ontologue** (*ontology engineer*), qui est chargé de formaliser les connaissances du modèle conceptuel sous forme d'ontologie,
— **l'éditeur en chef** (*lead editor*, parfois appelé administrateur ou modérateur), qui est généralement une personne unique ou un très petit groupe, en charge de coordonner le projet et d'approuver les modifications proposées par les autres participants.

Un même participant peut bien entendu jouer plusieurs rôles, par exemple un utilisateur peut être expert du domaine, ou l'ingénieur de la connaissance et l'ontologue peuvent être la même personne.

Les différentes méthodologies proposent des techniques pour arriver à un consensus parmi les participants, par exemple en utilisant la méthode Delphi (dans Holsapple and Joshi) ou le modèle d'argumentation IBIS (*Issue Based Information System*, dans DILIGENT et HCOME). Des logiciels spécifiques peuvent aussi être proposés pour faciliter la collaboration.

Bien que séduisantes en théorie, aucune approche pleinement collaborative n'a été entièrement adoptée pour réaliser de vrais projets concrets à base d'ontologie [168]. Les méthodologies collaboratives restent donc du domaine de la recherche.

2.1.5.2 Méthodes automatiques et semi-automatiques

La construction automatique ou semi-automatique d'ontologie (*ontology learning* en anglais) est une alternative à la construction manuelle. Elle s'appuie sur une source de connaissance non structurée ou semi-structurée à la place (ou en complément) des experts humains [66]. La source de connaissance est généralement un corpus de texte, mais il peut aussi s'agir de fichiers XML (*eXtendable Markup Language*), d'une base de données ou d'une description de service web. La construction de l'ontologie fait alors appel à différents sous-domaines de l'informatique : le traitement automatique de la langue naturelle, la gestion des bases de données, l'analyse formelle de concept, la recherche d'information, la fouille de données, l'apprentissage automatique et l'intelligence artificielle,...

Plusieurs systèmes ont été proposés pour la construction semi-automatique d'ontologie à partir d'un corpus de texte [190]. Ces méthodes utilisent des techniques statistiques, linguistiques et/ou logiques pour extraire les différentes parties d'une ontologie : les termes, les concepts, les relations taxonomiques, les autres relations et les axiomes.

Un exemple est la méthode Terminae [11, 10], qui comprend quatre étapes :

1. la constitution du corpus de texte,
2. l'analyse linguistique par des outils de traitement automatique du langage naturel, qui aboutit à une représentation terminologique,
3. la conceptualisation, incluant une phase de normalisation, qui aboutit à un modèle conceptuel,
4. la formalisation, qui aboutit à une ontologie formelle.

2.1.5.3 Méthodologies dans le domaine biomédical

Dans le domaine biomédical, J. Charlet [32] a proposé un processus en quatre étapes : (1) le recueil de données brutes, (2) la construction du schéma du modèle conceptuel, (3) la définition du modèle complet et son instanciation, et (4) l'opérationnalisation de ce modèle et l'implémentation de logiciels utilisant le modèle. J. Charlet distingue également des méthodes *ascendantes* (où l'on part d'un jeu de données et l'on arrive au modèle par abstraction) et des méthodes *descendantes* (où l'on part d'un modèle générique issu d'une bibliothèque de modèles, que l'on affine et que l'on adapte au domaine).

C. Kuziemsky *et al.* [98] ont proposé une méthodologie similaire pour la conception d'ontologie et de systèmes d'information à base d'ontologie, avec quatre étapes : (1) la spécification et la conceptualisation, incluant la collecte d'un jeu de données et son analyse, (2) la formalisation, aboutissant à une ontologie

formelle, (3) l'implémentation, aboutissant à un système d'information, et (4) l'évaluation et la maintenance du système. Cette méthodologie a été employée pour la conception d'un système d'information pour la prise en charge de la douleur en soin palliatif.

Ces deux méthodologies sont proches et on retrouve des étapes identiques (recueil de données, implémentation). En revanche, l'étape d'évaluation ne figure pas chez J. Charlet, tandis qu'il décompose davantage les premières étapes.

Par ailleurs, des méthodes de traitement automatique de la langue ont été utilisées, par exemple par C. Duclos pour réaliser un modèle conceptuel très complet de la pharmacocinétique [41].

2.1.6 Opérationnalisation des connaissances

Une fois les connaissances formalisées à l'aide d'une ontologie formelle, celle-ci doit être *opérationnalisée*, c'est-à-dire intégrée au sein d'un logiciel, d'un site web ou d'un système d'information et/ou d'une base de données. Cependant, l'*opérationnalisation des connaissances* est une tâche complexe, le langage des ontologies (généralement OWL) n'étant pas le même que celui des programmes informatiques ou des bases de données. Cette difficulté a également été décrite par P. Arapov [8] lors de la conférence IC 2014 (Ingénierie des Connaissances), et elle correspond au cas d'utilisation n°5 identifié par A. Rector [155] : *ontology programming interface*.

Actuellement, trois approches existent pour opérationnaliser les ontologies : les langages de requêtes, les API classiques, et la programmation orientée ontologie (voir exemples Figure 2.1).

2.1.6.1 Les langages de requêtes : SPARQL

SPARQL (*SPARQL Protocol and RDF Query Language*) est un langage de requête qui permet d'effectuer des recherches ou des modifications dans des graphes RDF. Il est proposé par le W3C. Ce langage s'inspire du langage SQL (*Structured Query Language*) qui est très utilisé pour les bases de données relationnelles. Il permet d'accéder facilement au contenu d'une ontologie *via* un *triple store*.

SPARQL présente cependant deux limites importantes. Tout d'abord, ce n'est pas une approche orientée objet. De la même manière que lorsqu'on utilise une base de données relationnelle dans un langage objet, le programmeur sera amené à encapsuler les requêtes dans les méthodes des différentes classes, ce qui peut être laborieux lorsqu'elles sont nombreuses.

Deuxièmement, SPARQL est basé sur RDF et non sur OWL. Il est donc possible d'interroger des individus de l'ontologie, mais plus difficile de manipuler des classes. En particulier, il n'est pas possible de requêter de manière transitive sur les restrictions existentielles d'une classe (c'est-à-dire rechercher $\forall R.C$, $\forall R.(\forall R.C)$, $\forall R.(\forall R.(\forall R.C))$...). Un exemple est la recherche de toutes les super-parties d'un organe donné, dans ce cas R est la relation *partie-de* et C la classe de l'organe. Ce second point est particulièrement gênant dans le domaine médical, car nous avons vu que les principaux concepts médicaux (maladies et médicaments notamment) doivent être représentés par des classes dans les ontologies médicales.

2.1.6.2 API classiques

Les API (*Application Programming Interface*) « classiques » fournissent des objets et des fonctions pour manipuler les éléments (classes, objets, relations, annotations) d'une ontologie, par exemple une classe de l'ontologie est, du point de vue du programmeur, une instance de la classe OWLClass. Plusieurs API de ce type existent, notamment OWLAPI [67] et Jena (https://jena.apache.org/) en Java.

Ces API sont largement utilisées. Cependant, elles sont compliquées à manipuler et conduisent souvent à un code source volumineux et peu lisible. De plus, les classes OWL étant traitées comme des instances par le langage de programmation, ces API ne permettent pas de tirer pleinement parti de la programmation orientée objet.

Dans l'exemple de la Figure 2.1, le calcul de la somme est implémenté dans une fonction statique getOrderCost(), alors que, d'un point de vue « objet », il aurait été préférable de créer une méthode Order.getCost(), ce que les API classiques ne permettent pas.

Langage de requête (SPARQL) :

```
PREFIX  <file://path/to/file.owl>
SELECT (SUM(?drug_price) AS ?total_price)
WHERE {
  ?order :name "My order" .
  ?order :has_drug ?drug .
  ?drug :has_price ?drug_price .
}
```

API classique (Java + OWLAPI) :

```
OWLDataFactory df = OWLManager.getOWLDataFactory();
public static float getOrderCost(OWLIndividual order) {
    float cost = 0.0;
    OWLObjectProperty drugProperty = df.getOWLObjectProperty(IRI.create("file.owl#has_drug"));
    OWLDataProperty priceProperty = df.getOWLDataProperty(IRI.create("file.owl#has_price"));
    Iterator drugs = order.listPropertyValues(drugProperty);
    while(purchases.hasNext()) {
        OWLIndividual drug = (OWLIndividual) drugs.next();
        Float price = (Float) drug.getPropertyValue(priceProperty);
        cost = cost + price.floatValue();
    }
    return cost;
}
```

API classique (Java + Jena) :

```
JenaOWLModel owlModel = ProtegeOWL.createJenaOWLModelFromURI("file.owl");
OntModel m = owlModel.getOntModel();
public static float getOrderCost(OWLIndividual order) {
    float cost = 0.0;
    ObjectProperty drugProperty = m.getObjectProperty("file.owl#has_drug");
    DatatypeProperty priceProperty = m.getDatatypeProperty("file.owl#has_price");
    Iterator drugs = order.listPropertyValues(drugProperty);
    while(purchases.hasNext()) {
        OWLIndividual drug = (OWLIndividual) drugs.next();
        Float price = (Float) drug.getPropertyValue(priceProperty);
        cost = cost + price.floatValue();
    }
    return cost;
}
```

Programmation orientée ontologie (Python + Owlready) :

```
class Order(Thing):
    def get_cost(order):
        cost = 0.0
        for drug in order.drugs: cost = cost + drug.price
        return cost
```

FIGURE 2.1 – Exemples d'utilisation d'une ontologie dans un programme informatique, en utilisant SPARQL (en haut), en utilisant Java avec une API classique (OWLAPI et Jena, au milieu) et en utilisant Python avec Owlready (en bas), le module de programmation orientée ontologie qui sera présenté plus loin. Chaque exemple calcule le coût total d'une ordonnance, en considérant une seule boîte pour chaque médicament (il s'agit donc simplement de faire la somme du prix de chaque médicament).

2.1.6.3 Programmation orientée ontologie

La programmation orientée objet est un paradigme largement connu et qui a fait ses preuves. Il serait donc intéressant de s'appuyer sur ce paradigme pour proposer des outils faciles d'emploi pour l'opérationnalisation des ontologies, d'autant plus qu'il existe des similitudes importantes entre ontologies et modèles objet [92] : les classes, propriétés et individus des ontologies correspondent aux classes, attributs et instances des modèles objets [91]. Les API classiques présentées précédemment ne tirent pas parti de ces similitudes.

La *programmation orientée ontologie* [56] cherche à rapprocher, voire unifier, l'ontologie et le modèle objet du langage de programmation utilisé ; par exemple une classe de l'ontologie devient une classe dans le langage de programmation. Le W3C a montré comment cette approche permettrait de réduire de moitié le volume de code source sur un exemple [91]. Sur l'exemple de la Figure 2.1 (en bas), la programmation orientée ontologie permet non seulement de réduire le volume de code, mais aussi de respecter le paradigme objet en définissant une méthode get_cost() au sein de la classe Order issue de l'ontologie.

Nous distinguons deux approches pour la programmation orientée ontologie :

— **L'approche statique** consiste à générer le code source des classes présentes dans l'ontologie. Cette approche utilise donc un « traducteur » lisant le fichier décrivant l'ontologie (par exemple en OWL) et générant un ou plusieurs fichiers de code source décrivant les classes. Cette approche permet d'accéder facilement à l'ontologie et de vérifier le typage à la compilation. En revanche, de par sa nature statique, elle ne permet pas d'effectuer des inférences ou de la classification automatique. Plusieurs outils de ce type ont été proposés en langage Java [80, 200] et C# [56].

— **L'approche dynamique** consiste à effectuer une « traduction » dynamique entre l'ontologie et le langage de programmation objet, lors de l'exécution du programme. Cette approche permet d'accéder facilement à l'ontologie mais aussi d'appeler un raisonneur pour effectuer des inférences (par exemple la classification automatique permet de modifier dynamiquement les classes des instances et les super-classes des classes) et même d'intégrer des méthodes au sein des classes de l'ontologie. En revanche, elle ne permet pas de vérifier le typage.
Un outil a été proposé en Common Lisp [92, 93], en utilisant un algorithme de subsomption spécifique pour l'inférence et non un raisonneur externe. Un prototype en Python a aussi été réalisé [15] en s'appuyant sur des métaclasses, cependant les possibilités de raisonnement et d'expression de restriction restent très limitées. D'autres auteurs ont cherché à concevoir de nouveaux langages spécifiquement adaptés à la programmation orientée ontologie, tel que Go ! [33].
Plus récemment, une approche semi-dynamique a été proposée en Java [172], elle permet la classification des instances par un raisonneur avec certaines limites (puisqu'il n'est pas possible de changer la classe d'un objet dans un langage statique comme Java, de nouveaux objets sont donc créés pour les instances et le programmeur ne doit plus utiliser les références aux anciens objets).

Encadrement associé à cette contribution :

1. Mobin Yasini, stagiaire de M2 puis doctorant en informatique médicale
(6 mois + 3 ans, 2010-2013, taux d'encadrement 70%)

Publications associées à cette contribution :

1. Yasini M, Duclos C, **Lamy JB**, Venot A. Facilitating access to laboratory guidelines by modeling their contents and designing a computerized user interface. Stud Health Technol Inform 2011 ;169 :487-91. Conférence Medical Informatics Europe MIE 2011, Oslo, Norvège, **best paper**
2. Yasini M, Duclos C, Venot A, Lepage E, **Lamy JB**. A guideline-derived model to facilitate the implementation of test-ordering rules within a hospital information system. Stud Health Technol Inform 2013 ;192 :719-23. Conférence MEDINFO 2013, Copenhague, Danemark
3. **Lamy JB**, Ugon A, Favre M, Duclos C, Venot A, Berthelot H. Comparaison et visualisation des contre-indications des médicaments. Actes du 3ème Symposium Ingénierie de l'Information Médicale SIIM 2015, Rennes
4. **Lamy JB**, Ugon A, Duclos C, Venot A, Berthelot H, Favre M. Une plate-forme visuelle pour une information comparative sur les nouveaux médicaments. Actes du congrès d'Ingénierie des Connaissances IC 2017, Caen

2.2 Contribution : Modélisation du domaine en médecine

Dans le milieu médical, un grand nombre d'acteurs interviennent et les connaissances sont souvent disponibles sous forme de textes. Malgré cela, la modélisation du domaine reste délicate, notamment sur ce qui touche aux aspects cliniques. Les approches collaboratives sont difficiles à mettre en oeuvre auprès des médecins pour de multiples raisons :

— les médecins sont peu nombreux,
— ils ont très peu de temps disponibles,
— ils ont peu d'affinité avec les nouvelles technologies (ce qui est en partie lié à la démographie médicale, les médecins étant dans l'ensemble âgés),
— l'indemnisation ou le défraiement des médecins (sous forme d'honoraire) a un coût élevé,
— en France, les médecins libéraux ont une grande tradition d'indépendance.

Une méthodologie collaborative impliquant un grand nombre de médecins travaillant régulièrement de manière décentralisée *via* des outils informatiques n'est donc pas vraiment envisageable à l'heure actuelle.

Les méthodes de modélisation automatiques pourraient être envisagées, notamment à partir des textes médicaux. Cependant, ces méthodes sont difficiles à appliquer à cause de la complexité intrinsèque aux connaissances médicales d'une part, et de la complexité propres aux textes d'autre part. En effet, les textes résultent souvent d'un consensus entre plusieurs experts aux points de vue et aux intérêts divergents, ce qui aboutit à des formulations compliquées voire alambiquées visant à « ménager la chèvre et le chou ». De plus, lorsqu'il s'agit des connaissances, le volume des textes est généralement suffisamment limité pour permettre un traitement manuel, par exemple lorsqu'il s'agit de guides de bonnes pratiques ou de Résumé des Caractéristiques Produit de médicaments (il en irait différemment s'il s'agissait de *données* médicales telles que des comptes-rendus hospitaliers).

2.2.1 Méthodologie

Nous proposons ici une méthodologie que nous appellerons PIGEOMS (Processus Intégré pour Générer et Enrichir des Ontologies et des Modèles en Santé). Elle n'a pas l'ambition d'être particulièrement originale ou innovante, la mise au point de méthodologies de modélisation étant un champ de recherche à part entière qui nous entraînerait très loin de la visualisation des connaissances. En revanche, notre méthodologie est spécifiquement adaptée au domaine médical et clinique. Elle est utilisée de manière informelle depuis des années comme une sorte de « savoir-faire » au LIM&BIO puis au LIMICS, et nous avons souhaité la rédiger et la décrire ici de manière explicite.

Cette méthodologie est fortement centralisée, elle s'appuie sur une personne centrale, le « modélisateur », qui doit nécessairement disposer de la double-compétence en santé (médecine, pharmacie) et en informatique

(ingénierie des connaissances). Seulement 3 rôles seront considérés :

— **le modélisateur**, qui est une personne unique qui cumule les rôles d'éditeur en chef, d'expert du domaine, d'ingénieur de la connaissance et d'ontologue (plus éventuellement celui d'utilisateur),
— **les évaluateurs**, qui sont des experts du domaine avec un niveau minimal de compétence en ingénierie des connaissances : les évaluateurs doivent savoir lire et utiliser un modèle conceptuel mais il n'est pas nécessaire qu'ils sachent en construire un.
— **les utilisateurs** finaux.

La Figure 2.2 présente l'ensemble de la méthodologie. Elle comprend 4 grandes phases :

1. **La phase de spécification** : elle vise à déterminer précisément l'objectif du modèle par des cas d'utilisation, à effectuer des choix préliminaires concernant le formalisme à utiliser (par exemple UML ou OWL) et les terminologies médicales à employer (par exemple la CIM10 ou la SNOMED CT pour le codage des maladies), et à rassembler les ressources nécessaires : évaluateurs, utilisateurs, textes et modèles préexistants dans la littérature.
2. **La phase de conceptualisation** : elle doit aboutir à un modèle conceptuel du domaine. Cette phase suit un processus itératif, chaque itération permettant d'enrichir le modèle en faisant appel à l'une des techniques suivantes :
   (a) l'utilisation des connaissances expertes personnelles du modélisateur,
   (b) la réutilisation d'éléments issus des modèles déjà existants dans la littérature,
   (c) l'analyse manuelle de textes du domaine,
   (d) la présentation du modèle en cours d'élaboration aux évaluateurs, qui donnent alors leur avis et peuvent faire des propositions,
   (e) l'élicitation de connaissances auprès des utilisateurs.
3. **La phase d'évaluation** : elle a pour objectif de vérifier que le modèle conceptuel permet bien de modéliser les connaissances attendues, et de le corriger si nécessaire. Durant cette phase, chaque évaluateur est chargé d'instancier le modèle dans un certain nombre de situations, typiquement à partir d'un jeu de texte. Lors de son travail, l'évaluateur note les difficultés rencontrées et les éléments de connaissance qu'il juge importants mais qui n'ont pas trouvé leur place dans le modèle. Les notes des évaluateurs sont ensuite rassemblées et recoupées. Enfin, le modélisateur corrige le modèle pour tenir compte des problèmes soulevés.
4. **La phase de formalisation** : elle a pour objectif de formaliser le modèle dans le formalisme choisi. Dans le cas où le modèle réalisé est une ontologie formelle, l'axiomatisation est réalisée lors de cette phase. Cette phase suit généralement un processus itératif consistant à établir un cas de test (correspondant par exemple à une inférence que l'on souhaite pouvoir réaliser avec l'ontologie), à réaliser la formalisation de la partie correspondante du modèle (par exemple en ajoutant des axiomes dans l'ontologie pour permettre les inférences) puis à tester le cas et à corriger la formalisation si nécessaire.

L'élicitation des connaissances auprès des utilisateurs peut se faire avec les méthodes habituelles ; nous suggérons deux d'entre elles que nous avons eu l'occasion d'employer à plusieurs reprises :

— **Les focus groupes** : il s'agit d'organiser une discussion ou une interview collective avec un petit groupe d'utilisateurs (6-15 personnes, au-delà il est préférable de réaliser plusieurs séances). La discussion peut s'appuyer sur des textes médicaux sur lesquels les utilisateurs auront travaillé préalablement ; les textes servant de support peuvent aussi être annotés par les utilisateurs et récupérés par la suite. Il est aussi recommandé d'avoir au moins deux organisateurs : un pour mener la discussion et l'autre pour prendre des notes. Les discussions peuvent aussi faire l'objet d'un enregistrement audio ou vidéo. Notre expérience montre que les focus groupes sont bien adaptés aux médecins généralistes ou de ville. Il est possible de passer par une société faisant de la formation auprès des médecins pour organiser le recrutement et fournir les locaux. De plus, si le focus groupe inclut un travail sur des textes médicaux, cela peut valider des heures de formation continue (obligatoire pour les médecins), et donc faciliter leur recrutement.

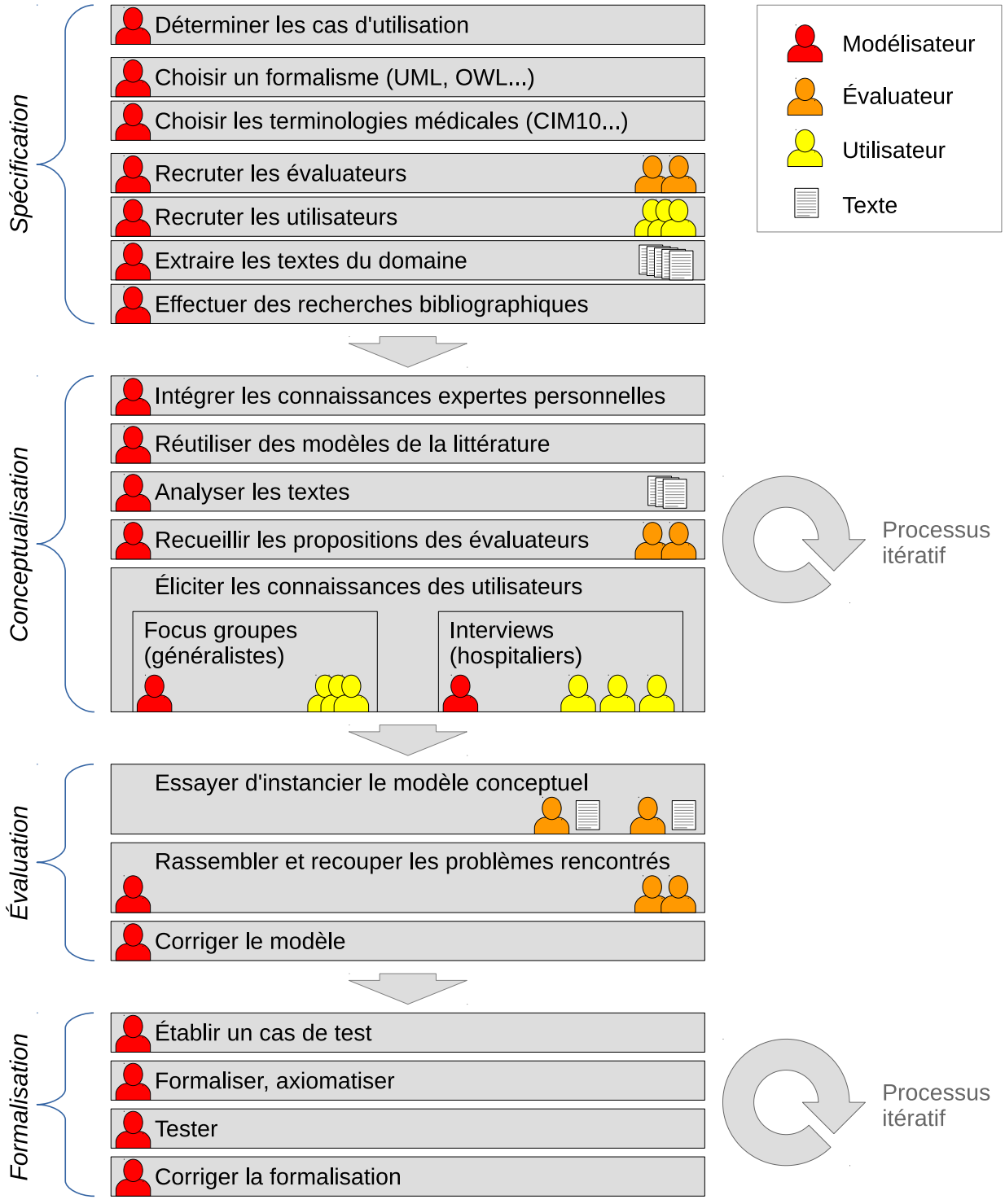


FIGURE 2.2 – Vue d'ensemble de la méthodologie PIGEOMS (Processus Intégré pour Générer et Enrichir des Ontologies et des Modèles en Santé) que nous proposons pour la modélisation du domaine en santé. Les icônes indiquent quels participants interviennent à chaque étape.

— **Les interviews individuelles** : il s'agit d'interviewer les utilisateurs un par un, en se déplaçant sur leur lieu de travail. Les interviews individuelles sont préférables pour les médecins hospitaliers. En effet, les focus groupes sont plus difficiles à organiser en milieu hospitalier : d'une part les médecins ont un emploi du temps plus complexe, d'autre part le défraiement des hospitaliers est plus compliqué d'un point de vue administratif, ce qui oblige à réduire au maximum le temps passé avec chaque médecin, dans la mesure où leur participation est généralement bénévole.

Les différentes phases se déroulent globalement dans l'ordre ci-dessus. La phase d'évaluation a généralement lieu avant la formalisation et l'axiomatisation car l'évaluation porte seulement sur la capacité du modèle à représenter les connaissances, mais pas sur sa capacité à permettre des inférences.

Cependant, il est possible de revenir sur les phases précédentes au cours des phases suivantes. Un exemple typique concerne les focus groupes ou les interviews d'utilisateurs qui interviennent dans la phase de conceptualisation : il est fréquent d'affiner les cas d'utilisation à ce moment-là avec les utilisateurs et donc de revenir sur la phase de spécification. Pour des raisons pratiques et matérielles, il est préférable de procéder ainsi et de ne réunir les utilisateurs qu'une seule fois plutôt que deux (lors des spécifications et lors de la conceptualisation).

La méthodologie n'impose pas un formalisme particulier ; elle a été utilisée avec des modèles UML comme avec des ontologies formelles en OWL. En revanche, l'utilisation de langages moins formels et plus souples tels que les cartes heuristiques (*mind maps*) n'est pas recommandée pour la modélisation du domaine, à cause de leur aspect idiosyncrasique (c'est-à-dire spécifique à la personne qui a réalisé la carte [48]).

Les textes du domaine interviennent à deux reprises, lors de la phase de conceptualisation et d'évaluation. Un même texte ne doit bien entendu pas être utilisé deux fois. Il faut répartir les textes en deux jeux : un pour la conceptualisation et un pour l'évaluation. En revanche, lors de l'évaluation, un même texte pourra être donné à deux évaluateurs différents, notamment lorsque le nombre de textes disponibles est faible.

Dans les sections suivantes, nous illustrerons la méthodologie PIGEOMS sur deux exemples, qui sont résumés dans le tableau 2.2.

2.2.2 Modélisation conceptuelle du domaine pour l'aide à la décision en biologie

Le premier exemple d'application de la méthodologie PIGEOMS concerne la construction d'un modèle conceptuel du domaine pour les recommandations en biologie médicale, dans le cadre d'un projet d'aide à la décision en biologie médicale. Le modèle a été construit par Mobin Yasini, en utilisant le formalisme UML et le langage Arden Syntax.

Arden Syntax [69] est un langage de programmation pour la connaissance médicale ; ce langage se veut facile à comprendre par un professionnel de santé. Il permet de générer des alertes, des interprétations, de filtrer et de gérer les messages... et aussi d'organiser les connaissances sous forme de Modules Logiques Médicaux (*Medical Logic Module*, MLM) qui peuvent être partagés et échangés entre différentes institutions. Un MLM comprend 3 blocs de code source pour son exécution :

— ***data*** : ce bloc indique comment récupérer les valeurs des paramètres et des données, par exemple en interrogeant une base de données *via* des requêtes SQL (*Structured Query Language*),
— ***logic*** : ce bloc comprend la logique du raisonnement médical, par exemple avec des conditions (*if*),
— ***action*** : ce bloc indique l'action à entreprendre, par exemple retourner une valeur.

Le MLM comprend aussi un nombre important de métadonnées pour le versionnage, ou pour renseigner sa fonction. Arden Syntax a par la suite été intégré dans le standard HL7 (*Health Level* 7) et il a notamment été utilisé dans des systèmes d'aide à la décision clinique [160].

Lors de la phase de spécification, le choix des terminologies médicales a fait l'objet d'une étude comparative poussée entre la CIM10 et la SNOMED CT ; cette dernière a finalement été retenue car elle permettait de coder un nombre plus important des conditions patient rencontrées dans les recommandations (80,1% contre 43,1%) [194]. LOINC (*Logical Observation Identifiers Names and Codes*) a été utilisé pour les coder examens.

La phase de conception du modèle s'est fondée sur les connaissances expertes de Mobin, l'analyse de différents guides issus des autorités de santé françaises (voir exemple Figure 2.3), et des interviews auprès de médecins biologistes hospitaliers et d'ingénieurs travaillant sur les logiciels utilisés pour la prescription de la biologie. De plus, le modèle a fait l'objet d'une mise en correspondance avec HL7 RIM (*Reference Information*

		Aide à la décision pour la biologie médicale	**VIIIP** (ontologie comparative des médicaments)
	Modéliseur	Mobin Yasini (médecin, doctorant en informatique médicale)	Jean-Baptiste Lamy (maître de conférence en informatique, pharmacien)
Spécif.	Objectif	Aide à la décision pour la prescription d'examen de biologie en milieu hospitalier	Formation continue des médecins en matière de nouveaux médicaments
	Formalisme	UML	OWL
	Terminologies	SNOMED CT (conditions patient) LOINC (examens)	MedDRA (effet indésirable) CIM10 (contre-indication) Thésaurus ANSM (interaction)
	Utilisateurs	4 médecins et biologistes hospitaliers 2 médecins-informaticiens travaillant pour des éditeurs de logiciels hospitaliers	17 médecins généralistes
	Textes	22 fiches en français produites par l'AP-HP 3 guides français de la HAS et l'ANSM 6 guides internationaux (GB, USA)	71 notices + 71 RCP de nouveaux médicaments commercialisés entre janvier 2011 et avril 2014
	Littérature	Modèle HL7 RIM	Schéma de la base de données Thériaque Modèle de Maia Iordatii
Concept.	Textes	22 fiches AP-HP + 3 guides français	19 nouveaux médicaments (notices + RCP)
	Élicitation des connaissances	Interviews individuelles	Focus groupes (2 séances)
Éval.	Évaluateurs	4 experts en informatique médicale	3 experts en informatique médicale 1 expert en médecine générale 1 expert en pharmacie
	Textes	6 guides internationaux	15 nouveaux médicaments (notices + RCP + ACT + EPAR)
Form.		Modélisation en UML puis implémentation en Arden Syntax.	Axiomatisation de l'ontologie et test des inférences avec le raisonneur HermiT.

TABLEAU 2.2 – Application de la méthodologie PIGEOMS dans deux exemples : la modélisation conceptuelle du domaine en biologie médicale et la construction d'une ontologie comparative des médicaments.

Model), un modèle standard pour l'échange d'informations médicales et d'éléments du dossier patient.

La phase d'évaluation a fait intervenir 4 évaluateurs qui ont essayé d'appliquer le modèle aux recommandations de 6 guides internationaux. Un premier modèle a été réalisé pour structurer les guides, et un second modèle pour les règles de recommandations qui est présenté en Figure 2.5. Un jeu de 201 règles de prescription a été extrait manuellement à partir des guides puis structuré selon ce modèle.

En pratique clinique, l'implémentabilité des règles est souvent délicate : pour pouvoir déclencher une règle, il faut que les données intervenant dans les conditions de la règle soient disponibles sous une forme structurée et codée. Ce point est problématique pour la prescription des examens de biologie car de nombreuses règles font appel aux caractéristiques du patient et à son état de santé, alors que le logiciel de prescription des examens n'est pas toujours connecté avec le dossier patient à l'hôpital. Pour évaluer l'importance de ce problème, nous avons mesuré l'implémentabilité des 201 règles en pratique clinique en s'appuyant sur la méthode GLIA (GuideLine Implementability Appraisal [164]). Au final, 37% des conditions de règles étaient implémentables sans connexion du logiciel de prescription au dossier patient, et 92% avec. Ce résultat montre l'importance de la connexion au dossier patient.

La phase de formalisation en Arden Syntax a porté sur une sélection de 32 règles de prescription définies selon le modèle UML. Ces règles ont été traduites en Arden Syntax (voir exemple Figure 2.4) et ensuite exécutées dans le moteur Experter intégré au logiciel ORBIS d'AGFA Healthcare. Elles ont pu être essayées dans un environnement de test mais pas dans l'hôpital, le logiciel ORBIS étant toujours en cours de déploiement à l'AP-HP (Assistance Publique des Hôpitaux de Paris) à l'époque.

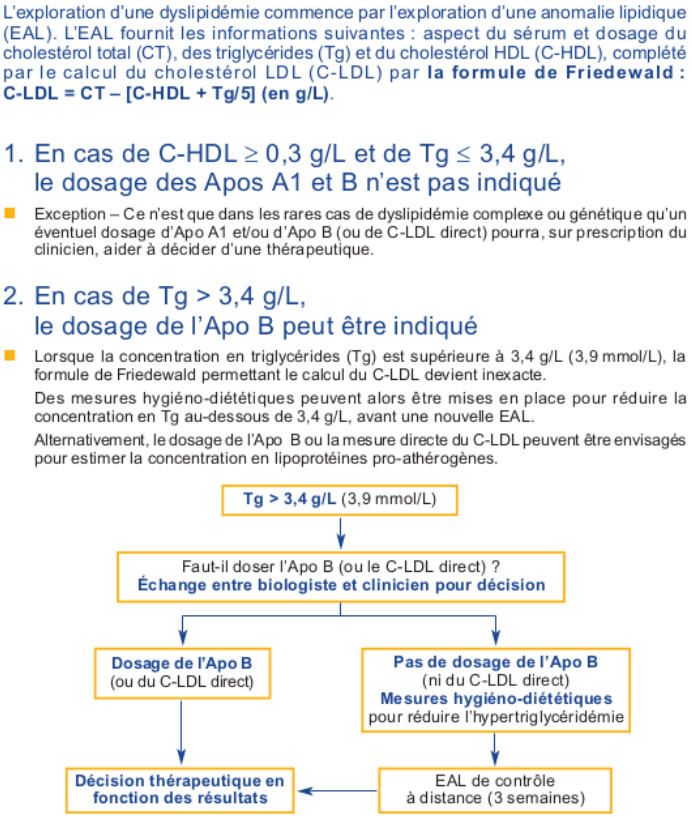


FIGURE 2.3 – Extrait d'un guide de bonnes pratiques sur la prescription des dosages des apolipoprotéines A1 et B pour l'exploration des dyslipidémies (source HAS 2009, http://www.has-sante.fr/portail/upload/docs/application/pdf/2009-06/fiche_butm_apolipo.pdf).

Maintenance | Library | Knowledge | Validation | Resources

Data

```
lab_code:=argument;
case_ID:=argument;

feedback:="";

(plannedsampletime, effectivetime, testname, prescriber) := READ LAST {
  (plannedsampletime,effectivetime, testname, prescriber)
  FROM REPORT "LabPrescription_case_code"
  WITH PARAMETERS (
    Case_ID SET TO case_ID;
    LabCode SET TO lab_code
  )
  USING COLUMNS (primary time=plannedsampletime)
}
;

HemoPrescripCode := ("5002000002");
```

Logic

```
IF lab_code is NOT IN HemoPrescripCode THEN
  conclude true;
ENDIF;

if (now - plannedsampletime) < 24 Hour then
  tmp := plannedsampletime formatted with "%.4t ";
  tmp := SUBSTRING 17 CHARACTERS FROM tmp;
  feedback:="Cette analyse a déjà été plannifiée le "|| tmp || " par " || prescriber ;
endif;
conclude true;
```

Action

```
return feedback;
```

FIGURE 2.4 – Exemple de MLM en Arden Syntax permettant de détecter une prescription redondante d'hémoculture.

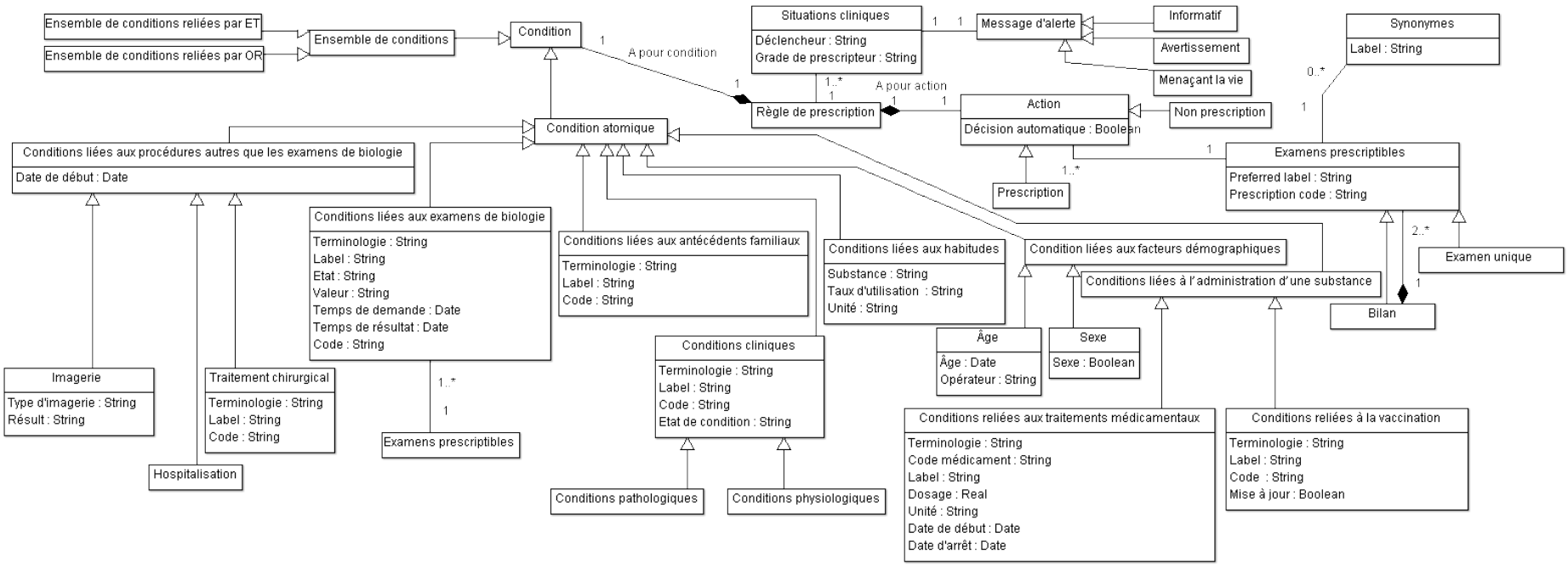


FIGURE 2.5 – Modèle conceptuel UML des règles de juste prescription en biologie médicale.

2.2.3 Construction d'une ontologie comparative des médicaments (projet VIIIP)

Le second exemple d'application de la méthodologie PIGEOMS concerne la construction d'une ontologie sur les propriétés des nouveaux médicaments, dans le cadre du projet VIIIP (voir Section 1.2.2). L'ontologie a été construite par moi-même (Jean-Baptiste Lamy) et formalisée en OWL. Cette ontologie représente à la fois des connaissances sur le nouveau médicament, mais aussi sur les médicaments existants de même indication, afin de permettre une comparaison entre le nouveau médicament et les anciens.

Lors de la phase de spécification, le formalisme OWL a été choisi pour le nombre important d'outils permettant de travailler sur les ontologies et pour sa capacité à produire des inférences. La terminologie MedDRA (*Medical Dictionary for Regulatory Activities*) a été choisie pour les effets indésirables, la CIM10 pour les contre-indications et le thésaurus ANSM pour les interactions.

La phase de conception du modèle s'est fondée sur les connaissances expertes du modélisateur mais aussi d'Hélène Berthelot, pharmacienne experte en base médicament, sur l'analyse de différents RCP et notices patient de médicaments, sur le schéma de la base de données sur le médicament Thériaque (http://theriaque.org), sur des modèles conceptuels simples (*mind maps*) réalisés précédemment au laboratoire lors de la thèse de Maia Iordatii [77], et sur des focus groupes réunissant des médecins généralistes (voir Tableau 2.2).

Les focus groupes ont réuni 17 généralistes en 2 séances. Le recrutement et l'organisation des séances se sont faits en partenariat avec la SFTG (Société de Formation Thérapeutique du Généraliste, `http://www.sftg.net/index.htm`), une association à but non lucratif avec laquelle nous travaillons depuis des années. La SFTG inclut une branche recherche qui s'intéresse à la recherche en médecine générale et en soins de premier recours, notamment dans les domaines de la qualité des soins et de l'organisation du système de santé.

La Figure 2.6 montre le programme détaillé des focus groupes. Ceux-ci comprenaient à la fois des discussions générales et un travail personnel des médecins sur 3 nouveaux médicaments. Au cours de ce travail, des documents sur les médicaments ont été distribués aux médecins ; ces documents, annotés par les médecins, ont été récupérés à la fin et analysés. Les focus groupes ont permis d'établir une liste des éléments de connaissance qui intéressaient les médecins, et aussi de préciser les cas d'utilisation et le périmètre du projet.

La phase d'évaluation a fait intervenir 5 évaluateurs qui ont essayé d'appliquer le modèle conceptuel sur un total de 15 nouveaux médicaments (chaque évaluateur traitant 3 médicaments avec des nouveautés différentes : une nouvelle molécule, une nouvelle forme ou voie d'administration, un nouveau dosage). Pour chaque nouveau médicament, plusieurs documents étaient disponibles : le RCP et la notice patient (voir section 2.1.2), l'ACT (Avis de la Commission de Transparence, si disponible) et le rapport d'évaluation (*European Public Assessment Reports*, EPAR, si disponible).

La conceptualisation et la formalisation de l'ontologie ont été réalisées à l'aide de l'éditeur Protégé. L'ontologie appartient à la famille $\mathcal{SHOIQ}(D)$ des logiques de description. La partie générale de l'ontologie (non spécifique à un médicament donné) comprend 240 classes, 167 propriétés, 154 individus et 2071 axiomes.

La Figure 2.7 montre les principales classes de l'ontologie et les différentes propriétés qui permettront la comparaison des médicaments. L'ontologie distingue les nouveaux médicaments (lesquels ne peuvent pas être déterminés automatiquement par rapport à leur date d'Autorisation de Mise sur le Marché (AMM), car la mise sur le marché effective est parfois très éloignée de la date d'AMM). Lorsqu'un médicament possède plusieurs indications, certaines propriétés sont valables pour le médicament (par exemple la composition ou les contre-indications), tandis que d'autres ne sont valables pour ce médicament que dans une indication donnée (par exemple la posologie, le coût ou la liste des comparateurs). De plus, certaines propriétés ne sont pertinentes que pour le nouveau médicament (par exemple l'ASMR, Amélioration du Service Médical Rendu, qui est donné à un temps t et pas toujours mis à jour). Par conséquent, nous avons créé quatre classes (Médicament, Médicament dans l'indication, Nouveau médicament et Nouveau médicament dans l'indication) avec des relations d'héritage, et nous avons rattaché chaque propriété à la classe appropriée. Les propriétés complexes (contre-indications, interactions et effets indésirables) sont représentées par des classes car elles-mêmes ont des attributs. De plus, elles peuvent être restreintes à une indication.

Pour les contre-indications, les interactions et les effets indésirables, des attributs supplémentaires sont présents : niveau de contre-indication ou d'interaction, fréquence et gravité des effets indésirables. La liste des interactions et des effets indésirables pouvant être longue, nous l'avons limitée aux interactions de niveaux

Présentation rapide du projet VIIIP (5 minutes)

Réflexion sur l'image de l'innovation pharmaceutique (10 minutes) : chaque participant écrit un mot ou une expression sur ce que lui évoque l'innovation pharmaceutique, positivement et négativement.

Tour de table sur les nouveaux médicaments (40 minutes), autour des questions suivantes :

- Vous arrive-t-il de prescrire des nouveaux médicaments ? lesquels ?
- Pourquoi prescrivez-vous (ou non) un nouveau médicament ?
- Quels sont les nouveaux médicaments que vous auriez envie de prescrire ?

Les réponses ont été écrites au tableau et réorganisées au fil de la discussion.

Discussion sur les connaissances sur les médicaments (30 minutes), autour des questions suivantes :

- Recherchez-vous de l'information sur les nouveaux médicaments avant de les prescrire ?
- À partir de quelles sources ? sites ? revues ? quand ? comment ?
- Quel est le type d'information dont vous avez besoin ? lisez-vous les RCP ? les avis de la commission de transparence ?

(pause 10 minutes)

Présentation du jeu de nouveaux médicaments (10 minutes). Les médicaments sont : Alvesco ou Cialis, Pylera, Jext. Les questions suivantes sont posées au préalable (avant l'étude des documents) :

- Est-ce que vous les connaissez ?
- Est-ce que vous les avez prescrits ?
- Est-ce que vous auriez envie de les prescrire ?

Travail individuel sur les documents portant sur le jeu de nouveaux médicaments (40 minutes). Les documents sont : promotion du labo, notice patient, RCP, ACT, tableaux récapitulatifs.

(pause 10 minutes)

Discussion autour des nouveaux médicaments étudiés et des documents associés (50 minutes) :

- Seriez-vous prêts à les prescrire après lecture des documents ? Pourquoi ?
- Quels sont les éléments qui vous manqueraient pour prendre cette décision ?
- Cette démarche longue est impossible au cabinet, comment souhaitez-vous que les informations vous soient présentées de façon synthétique ?
- En quoi le projet VIIIP pourrait-il vous aider sur la connaissance des médicaments ?
- Élaboration commune d'un document de synthèse sur un médicament à partir de questions aux participants.

FIGURE 2.6 – Programme des séances de focus groupes organisées lors du projet VIIIP.

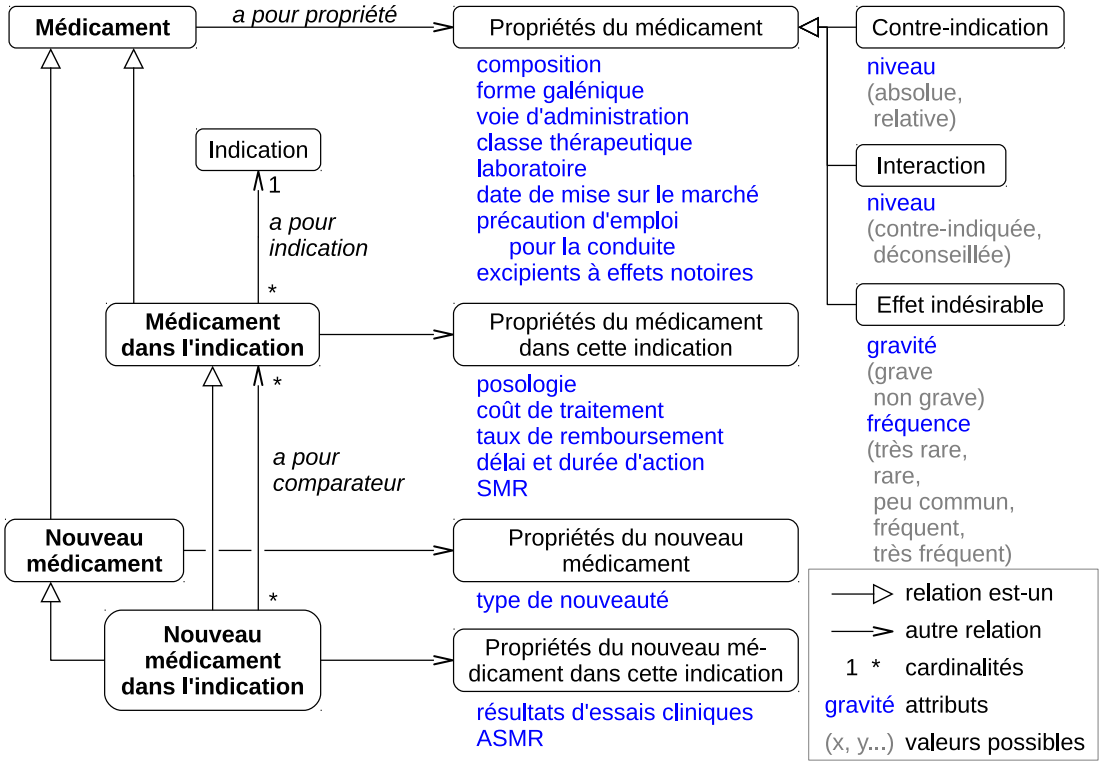


FIGURE 2.7 – Les principales classes de l'ontologie comparative des médicaments, les principales propriétés et le niveau où elles sont rattachées.

contre-indiqué et déconseillé, et aux effets indésirables graves et/ou fréquents (y compris très fréquents). Ce choix correspond aux demandes exprimées par les médecins lors des focus groupes.

Cette ontologie permet la comparaison des propriétés des médicaments. La principale difficulté rencontrée pour la comparaison est l'expression de propriétés proches à des niveaux de granularité différents, par exemple un nouveau médicament est contre-indiqué avec les maladies hémorragiques tandis qu'un de ses comparateurs est contre-indiqué avec les maladies hémorragiques *constitutionnelles* ou *acquises*, ce qui revient au même. L'ontologie permet des méthodes de raisonnements sémantiques pour résoudre ces problèmes [115]. Dans l'exemple précédent, « constitutionnelle » et « acquise » réalisent une partition, que l'on peut formaliser ainsi :

$Acquise \sqsubseteq Origine$
$Constitutionnelle \sqsubseteq Origine$
$Acquise \sqcap Constitutionnelle \sqsubseteq \bot$
$Origine \sqsubseteq (Acquise \sqcup Constitutionnelle)$

Il est ensuite possible de définir les trois maladies :

$Maladie \sqsubseteq (\exists aPourOrigine.Origine) \sqcap (\forall aPourOrigine.Origine)$
$MaladieHémorragique \sqsubseteq Maladie$
$MHAcquise \equiv MaladieHémorragique \sqcap \exists aPourOrigine.Acquise$
$MHConsti \equiv MaladieHémorragique \sqcap \exists aPourOrigine.Constitutionnelle$

Puis nous définissons un médicament m ayant deux contre-indications ciA et ciC :

$(ContreIndication \sqcap (\exists aPourMaladie.MHAcquise) \sqcap (\forall aPourMaladie.MHAcquise))(ciA)$
$(ContreIndication \sqcap (\exists aPourMaladie.MHConsti) \sqcap (\forall aPourMaladie.MHConsti))(ciC)$
$MHAcquise \sqsubseteq aPourMaladie^{-}.\{ciA\}$
$MHConsti \sqsubseteq aPourMaladie^{-}.\{ciC\}$
$(Médicament \sqcap (\forall aPourContreIndication.\{ciA, ciC\}))(m)$
$aPourContreIndication(m, ciA)$
$aPourContreIndication(m, ciC)$

Nous pouvons enfin définir la classe de l'ensemble des maladies contre-indiquées avec m :

$Maladie_CI_avec_m \equiv Maladie \sqcap (\exists aPourMaladie^{-}.(\exists aPourContreIndication^{-}.\{m\}))$

Un raisonneur permet alors d'inférer $MaladieHémorragique \sqsubseteq Maladie_CI_avec_m$ (et pas seulement $MHAcquise \sqsubseteq Maladie_CI_avec_m$ et $MHConsti \sqsubseteq Maladie_CI_avec_m$).

L'axiomatisation s'est ainsi poursuivie lors du projet VIIIP, au fur et à mesure que le besoin en inférences se faisait sentir, et en utilisant l'approche itérative que nous avons proposée.

2.2.4 Discussion

Dans cette section nous avons proposé la méthodologie PIGEOMS pour la construction de modèles et d'ontologies du domaine en santé. Cette méthodologie n'est pas particulièrement innovante et reste même en retrait : l'analyse des textes reste manuelle et la dimension collaborative est très limitée. Cependant, l'intérêt de cette méthodologie est son adaptation spécifique au domaine médical et en particulier clinique, c'est-à-dire lorsque les utilisateurs et experts du domaine sont des médecins. Par exemple la phase de choix des terminologies médicales est un élément spécifique à ce domaine.

Une particularité de cette méthodologie est de faire appel à une personne « clef de voûte » qui possède la double-compétence en ingénierie des connaissances et en santé. Cette personne peut être difficile à trouver, cependant, le laboratoire LIMICS étant coresponsable d'un Master d'informatique biomédicale, cela n'était pas un véritable problème en pratique. La méthodologie PIGEOMS s'avère même bien adaptée à une utilisation dans un cadre de recherche universitaire, où le rôle clef du modélisateur peut être joué par un doctorant en informatique médicale qui dispose de la double-compétence (tel que Mobin Yasini par exemple). Les encadrants pourront alors jouer le rôle d'évaluateurs.

La dimension collaborative reste limitée dans la méthodologie PIGEOMS, puisque l'essentiel de la conceptualisation est réalisé par le modélisateur. Celui-ci possède la double-compétence, ce qui est une alternative à la collaboration pour faire intervenir plusieurs compétences lors de la modélisation. De plus, reposer sur un petit nombre de personnes (voire une seule) fait qu'il est plus facile de s'assurer de l'indépendance des personnes en question. Ce point est notamment important lorsqu'on s'intéresse au médicament, puisqu'il existe des enjeux économiques très importants et que l'indépendance de certains experts est parfois discutable [139].

La méthodologie PIGEOMS fait appel à un corpus de texte, mais elle n'utilise pas de méthodes de traitement automatique pour ces textes. Néanmoins, elle pourrait facilement être adaptée pour un traitement automatique des textes.

PIGEOMS inclut une phase d'évaluation faisant intervenir un petit groupe d'évaluateurs qui testent le modèle sur un certain nombre de cas ou de textes. Lors de notre état de l'art, nous avons trouvé peu de méthodologies qui définissaient explicitement une phase d'évaluation : seules les méthodologies HCOME et celle proposée par C. Kuziemski [98] comprennent une phase d'évaluation ; celle-ci est différente de la nôtre puisqu'elle intervient après l'utilisation de l'ontologie dans des applications réelles. Il semblerait que, dans la plupart des méthodologies, l'aspect collaboratif et l'obtention d'un consensus soient jugés suffisants pour vérifier l'absence d'erreur et tiennent lieu « d'évaluation ». Une phase d'évaluation similaire à celle que nous proposons pourrait malgré tout enrichir les méthodologies existantes.

Publications associées à cette contribution :

1. **Lamy JB**, Berthelot H. Ontopy : programmation orientée ontologie en Python. Actes du congrès d'Ingénierie des Connaissances IC 2015, Rennes
2. **Lamy JB**. Ontology-Oriented Programming for Biomedical Informatics. Stud Health Technol Inform 2016 ;221 :64-68. Special Topic Conference STC 2016, Paris, France
3. **Lamy JB**. Owlready : Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. **Artificial Intelligence in Medicine** 2017 ;80 :11-28

2.3 Contribution : Module pour la programmation orientée ontologie

Dans l'état de l'art précédent, nous avons vu l'intérêt de la programmation orientée ontologie pour faciliter l'opérationnalisation des ontologies. Lors du projet VIIIP, il a fallu opérationnaliser l'ontologie comparative des médicaments (voir section 2.2.3) pour (1) peupler l'ontologie à partir de bases de données sur le médicament (base Thériaque) ou de saisies manuelles, (2) effectuer des inférences, puis (3) générer un site web à partir des connaissances de l'ontologie et des inférences obtenues. Nous souhaitions par ailleurs réutiliser des outils de gestion de terminologies médicales écrits en Python. Cependant, nous n'avons trouvé aucun outil satisfaisant pour manipuler les ontologies : les API classiques sont laborieuses à utiliser et il n'en existe pas en Python.

Nous avons donc développé un module pour la programmation orientée ontologie dynamique en Python, appelé Owlready (anciennement Ontopy). Le premier objectif d'Owlready est de rendre l'intégration entre Python et OWL la plus transparente possible. Le second objectif est de proposer des solutions aux deux difficultés que nous avons identifiées lors de l'utilisation des ontologies dans le domaine biomédical (section 2.1.4) : la nécessité de combiner raisonnement en monde ouvert et en monde fermé, c'est-à-dire de raisonner en *monde local fermé* (*local closed world reasoning*), et la nécessité de représenter la plupart des concepts médicaux par des classes.

La version d'Owlready décrite ici correspond à la version 2.

2.3.1 Architecture générale

La première version d'Owlready utilisait une architecture simple dans laquelle la totalité de l'ontologie était chargée en mémoire. Cette première version a permis de mettre au point l'outil et de le tester sur de petites ontologies, cependant elle montrait assez rapidement ses limites sur de plus grosses ontologies. La Figure 2.8 montre l'architecture générale de la seconde version d'Owlready. Elle s'appuie sur un *quad store* RDF et elle charge l'ontologie au fur et à mesure des besoins, en utilisant un parseur dynamique paresseux. Les classes et individus sont donc chargés à la demande. Par la suite, toute modification apportée sur les objets de l'ontologie est immédiatement reportée sur le graphe RDF.

Le *quad store* est implémenté avec la base de données relationnelle SQLite3, et permet de charger des fichiers au format RDF/XML, OWL/XML et NTriples, qui décrivent les classes, les propriétés et les individus de l'ontologie. En parallèle, un fichier de code Python (.py) peut être associé à l'ontologie (*via* une annotation dans celle-ci). Ce fichier contient les méthodes que l'on souhaite ajouter dans les classes de l'ontologie. Le module Owlready se connecte au *quad store*, charge le fichier de code Python et présente au programmeur une interface unifiée.

2.3.2 Comparaison du modèle objet et des ontologies formelles

Nous avons vu que le modèle objet utilisé par les langages de programmation et les ontologies formelles présentent de nombreuses notions communes : classes, propriétés, individus... Le Tableau 2.3 montre les principales différences entre modèles objet et ontologies, en distinguant le modèle objet statique du modèle objet dynamique. Nous pouvons constater que le modèle objet dynamique est plus proche des ontologies que ne l'est le modèle objet statique.

Le langage Python 3 a été choisi non seulement car nous souhaitions réutiliser des librairies programmées dans ce langage, mais aussi car il s'agit d'un langage objet dynamique avec héritage multiple, ce qui le

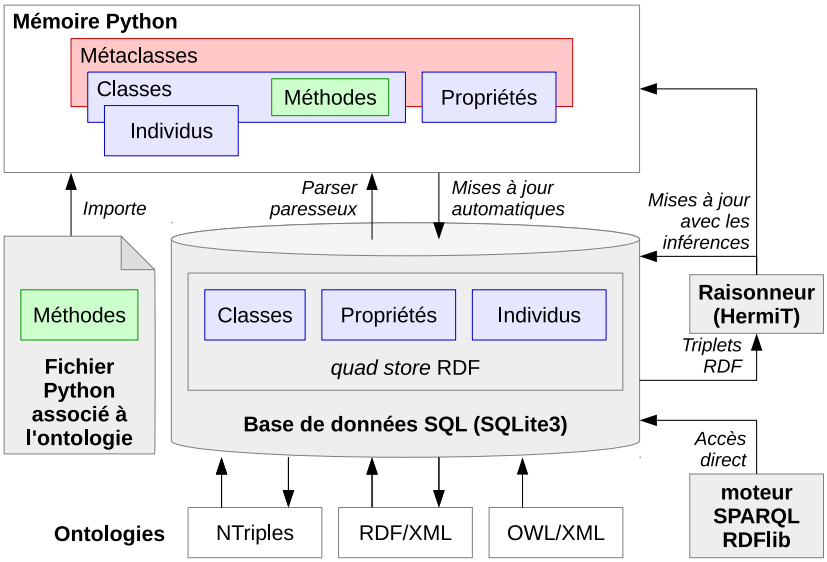


FIGURE 2.8 – Architecture du module Owlready (version 2).

Modèle objet statique (ex Java)	**Modèle objet dynamique (ex Python)**	**Ontologie (ex OWL)**
Hypothèse d'un monde fermé	Hypothèse d'un monde fermé	Hypothèse d'un monde ouvert
Disjonctions implicites entre classes (deux classes sont disjointes si l'une ne descend pas de l'autre)	Disjonctions implicites entre classes (deux classes sont disjointes si elles n'ont pas de descendante commune)	Pas de disjonctions implicites (les disjonctions doivent être assertées explicitement, deux classes peuvent être équivalentes)
Distinctions implicites entre instances (toutes les instances sont distinctes)	Distinctions implicites entre instances (toutes les instances sont distinctes)	Pas de distinctions implicites (les distinctions doivent être assertées explicitement, deux instances peuvent être identiques)
Héritage simple	Héritage multiple	Héritage multiple
Instanciation simple	Instanciation simple	Instanciation multiple
Une classe ne peut pas changer de superclasse	Une classe peut changer de superclasse durant l'exécution	Une classe peut changer de superclasse suite au raisonnement automatique
Une instance ne peut pas changer de classe	Une instance peut changer de classe durant l'exécution	Une instance peut changer de classe suite au raisonnement automatique
Une propriété est définie pour une classe donnée	Une propriété est définie pour une classe donnée, mais elles sont parfois considérées indépendamment des classes (*duck-typing*)	Les propriétés sont indépendantes des classes, il s'agit d'objets de premier ordre qui supporte l'héritage
Pas d'annotations	Pas d'annotations	Support des annotations

TABLEAU 2.3 – Comparaison des modèles objet (statique et dynamique) et du modèle des ontologies OWL.

rapproche des ontologies. En particulier, il permet de changer la classe d'un objet ou les superclasses d'une classe en cours d'exécution, par exemple suite à la classification automatique, ce que ne permet pas un langage statique comme Java. Il reste malgré tout plusieurs différences importantes entre le modèle objet dynamique de Python et celui des ontologies :

1. **Les ontologies reposent sur le principe du monde ouvert** : tout ce qui n'est pas expressément interdit est considéré comme possible. Au contraire, la programmation orientée objet part du principe inverse : tout objet ou classe qui n'est pas expressément créé n'existe pas.
2. **Les ontologies considèrent que deux classes nommées différemment peuvent être équivalentes**, de même deux individus nommés différemment peuvent être identiques. Au contraire, dans les modèles objets, deux instances sont implicitement considérées comme distinctes, et deux classes comme disjointes (à moins qu'il n'existe une classe fille héritant des deux).
3. **Les ontologies autorisent l'instanciation multiple** (une instance peut appartenir à plusieurs classes) en plus de l'héritage multiple. Le modèle objet ne supporte que l'instanciation simple.
4. **Les ontologies considèrent les propriétés comme des objets de premier ordre**, et permettent l'héritage entre propriétés. Récemment, le développement du « *duck-typing* » dans les langages de script tend à les rapprocher des ontologies. Dans le *duck-typing*, la sémantique d'une méthode est associée uniquement à son nom, alors que normalement les langages de programmation objet associent la sémantique d'une méthode au couple (Classe, nom de la méthode).
5. **Les ontologies autorisent l'ajout d'annotation en sus des propriétés**. La notion d'annotation n'existe pas dans la programmation orientée objet, en particulier, les annotations ne sont pas héritées.

2.3.3 Implémentation

Python permet de modifier en profondeur son modèle objet *via* un système de *métaclasses* (classe de classe). Le Tableau 2.4 montre les méthodes spéciales que nous avons redéfinies pour adapter le modèle objet de Python à OWL. Voici comment nous avons résolu les cinq principales différences :

1. **Pour ce qui est du monde ouvert**, nous avons fait le choix de garder les deux paradigmes en parallèle, ce qui permet de tirer parti des avantages de chacun selon le système employé : programmation orientée objet en Python avec monde fermé, et restriction OWL avec monde ouvert. De plus, nous proposerons plus loin une solution pour réaliser du raisonnement en monde fermé local.
2. **Pour ce qui est des équivalences et des disjonctions**, nous avons ajouté un attribut « equivalent_to » aux entités. Cet attribut permet de renseigner les entités équivalentes. De plus, nous avons modifié la manière dont Python détermine si une instance appartient à une classe, et si une classe est fille d'une autre classe, afin de prendre en compte les classes équivalentes (voir les méthodes de classe __instancecheck__ et __subclasscheck__ dans le Tableau 2.4).
3. **Pour gérer l'instanciation multiple**, lorsqu'une instance appartient à plusieurs classes, Owlready crée automatiquement une classe « intersection » héritant de ces différentes classes et l'instance est associée à cette classe.
4. **Pour permettre l'héritage entre propriétés**, nous avons considéré les propriétés comme des classes (il s'agit littéralement de « classes de relation »). Cela permet d'utiliser en Python l'héritage sur les propriétés.
5. **Pour gérer les annotations**, afin d'éviter que les annotations d'une classe ne soit héritée par ses classes-filles (ce qui est le comportement normal dans les ontologies), nous avons redéfini la lecture des attributs au niveau de la métaclasse (méthode __getattr__).

Pour permettre la mise à jour automatique du graphe RDF lorsque les propriétés d'une instance sont modifiées, nous avons créé en Python une sous-classe de liste spéciale nommée « CallbackList ». Cette liste appelle automatiquement une fonction (*callback* en anglais) lorsque la liste est modifiée. Cette fonction met à jour le graphe RDF.

	Méthode	Effet	Raison de la redéfinition
Métaclasse	__new__	Crée une nouvelle classe (c'est-à-dire une nouvelle instance de la métaclasse)	Combiner la nouvelle classe à la classe OWL de même nom, si elle existe
	__instancecheck__	Teste si un objet est une instance de la classe	Prendre en compte les classes équivalentes OWL
	__subclasscheck__	Teste si une classe est une sous-classe de la classe	Prendre en compte les classes équivalentes OWL
	mro	Calcule l'ordre de résolution des méthodes (*method resolution order*, MRO) notamment en cas d'héritage multiple	Ne pas déclencher d'erreur en cas de MRO temporairement incorrect lors du chargement de l'ontologie (les classes parentes étant ajoutées une à une)
	__getattr__	Obtient un attribut de la classe (appelé uniquement pour les attributs inexistants)	Retourner les annotations de la classe sans prendre en compte celle de ses superclasses
Classe	__setattr__	Modifie un attribut de l'objet	Mettre à jour le graphe RDF et, si elle existe, la propriété inverse
	__getattr__	Obtient un attribut de l'objet (appelé uniquement pour les attributs inexistants)	Interroger le graphe RDF pour obtenir la ou les valeurs de la propriété pour l'objet

TABLEAU 2.4 – Méthodes spéciales qui ont été redéfinies pour rendre le modèle objet de Python compatible avec OWL. Les méthodes de la métaclasse sont appliquées sur les classes, et celles des classes sur les instances.

Owlready cherche les fichiers OWL dans les répertoires de la liste appelée « *onto_path* », qui fonctionne de manière similaire au « *classpath* » de Java ou au « PYTHONPATH » de Python. À défaut, les ontologies sont téléchargées à partir de leur URI (*Uniform Resource Identifier*, correspondant à une adresse Internet).

Les éléments suivants d'OWL version 2.0 sont supportés :

— les classes,
— les propriétés (*ObjectProperty*, *DataProperty*, *FunctionalProperty*, *etc*),
— les individus,
— les disjonctions entre classes et les distinctions entre individus,
— les restrictions,
— les unions et intersections, la négation et les propriétés inverses,
— les annotations,
— les types de données suivants : booléen, entier, flottant, chaîne de caractères, date.

2.3.4 Syntaxe

La syntaxe d'Owlready s'inspire à la fois de la notation pointée utilisée dans la plupart des langages objet, et des notations utilisées dans Protégé pour décrire les expressions de classes. Le tableau 2.5 montre les principales syntaxes d'Owlready et les correspondances en notation formelle et dans Protégé.

Owlready utilise la notation pointée à cinq niveaux :

1. **Pour accéder à une entité dans une ontologie**. Chaque ontologie est considérée comme un espace de nommage, et la notation pointée permet d'accéder aux entités dans cet espace. Par exemple, si l'ontologie des pizzas (http ://www.co-ode.org/ontologies/pizza/pizza.owl) est chargée dans la va-

Formelle	**Protégé**	**Owlready**
$A \sqsubseteq B$	A subclass of B	class A(B) : ... (or) A.is_a.append(B)
$A \sqcap B$	A and B	A & B
$A \sqcup B$	A or B	A | B
$\neg A$	not A	Not(A)
$A \sqcap B = \emptyset$	A disjoint with B	AllDisjoint([A, B,...])
$A \equiv B$	A equivalent to B	A.equivalent_to.append(B)
{i, j,...}	{i, j,...}	one_of([i, j,...])
$\exists R.B$	R some B	R.some(B)
$\forall R.B$	R only B	R.only(B)
$=2R.B$	R exactly 2 B	R.exactly(2, B)
$\exists R.\{i\}$	R value i	R.value(i)
$\exists\ R.\top \sqsubseteq A$	R domain A	R.domain.append(A)
$\top \sqsubseteq \forall R.B$	R range B	R.range.append(B)
$S \equiv R^-$	S inverse of R	S.inverse = R
A(i)	i type A	i = A() (or) i.is_instance_of.append(A)
R(i, j)	i object property assertion j	i.R = j (R est fonctionnel) (or) i.R.append(j) (sinon)
R(i, n)	i data property assertion j	i.R = n (R est fonctionnel) (or) i.R.append(n) (sinon)
$\exists R.\{i\} \wedge (\exists R^-.A)(i)$	-	A.R = i (R est fonctionnel) (or) A.R.append(i) (sinon)

TABLEAU 2.5 – Correspondance entre les notations formelles, les notations de Protégé, et la syntaxe d'Owlready. A et B sont des classes, R et S des propriétés, i et j des individus, n un litéral (nombre, chaîne de caractères,...). La dernière ligne du tableau montre la syntaxe d'Owlready pour définir les restrictions de type « value » comme des propriétés de classes.

riable « pizza_onto », la classe Pizza (URI http ://www.co-ode.org/ontologies/pizza/pizza.owl#Pizza) pourra être récupérée par « pizza_onto.Pizza ».

De plus, Owlready permet également la création et la manipulation d'espace de nommage additionnel, sans qu'une ontologie leur soit associée.

2. **Pour accéder ou modifier la valeur d'une propriété pour une instance**. Par exemple, si la variable « test_pizza » contient une instance de la classe Pizza, nous pourrons obtenir les valeurs de la propriété « aPourGarniture » de la manière suivante : « test_pizza.aPourGarniture ». Les propriétés fonctionnelles retournent une valeur unique (ou None, la valeur nulle en Python, si aucune valeur n'est définie), les autres retournent une liste (qui s'écrivent entre crochets en Python). La méthode .append() de la liste permet d'ajouter des éléments, et la méthode .remove() d'en enlever.

   Owlready permet également, *via* l'annotation « python_name », de renommer les propriétés pour leur donner un nom qui sonne davantage « propriété » que « relation », par exemple « garnitures » au lieu de « aPourGarniture ».

3. **Pour accéder ou modifier les restrictions de type « value » associées à une classe** (*role-filler class constructs*, notées ∃R.{i}). D'un point de vue orienté objet, ces restrictions peuvent être vues comme des *attributs de classes*. C'est ainsi que Owlready les expose. La dernière ligne du tableau 2.5 montre cette syntaxe. Par exemple, pour asserter que toutes les pizzas appartenant à la classe PizzaMexicaine ont pour goût épicé (un individu de la classe Goût), c'est-à-dire formellement $PizzaMexicaine \sqsubseteq \exists aPourGoût.\{épicé\}$, nous écrirons « PizzaMexicaine.a_pour_goût = épicé » (si a_pour_goût est fonctionnelle). De plus, une contrainte existentielle est automatiquement créée sur l'individu, dans l'exemple précédent : $(\exists aPourGoût^{-}.PizzaMexicaine)(épicé)$. Cette syntaxe rend la manipulation des classes aussi facile que celle des instances.

4. **Pour accéder ou modifier les superclasses et les classes équivalentes** d'une classe. Les superclasses correspondent à l'attribut « is_a » de la classe et les classes équivalentes à l'attribut « equivalent_to ».

5. **Pour accéder ou modifier le domaine, le range et la propriété inverse** d'une propriété. Par exemple, le range de la propriété aPourGarniture peut être défini ainsi : « pizza_onto.aPourGarniture.domain = [Pizza] ».

OWL permet de « diviser » la description d'une classe donnée dans plusieurs ontologies : par exemple, si la classe C est décrite comme héritant de A et B, cela correspond aux deux assertions suivantes : $C \sqsubseteq A$ et $C \sqsubseteq B$, chacune pouvant être placée dans une ontologie différente. Au contraire, un langage de programmation objet comme Python ou Java oblige à décrire la totalité d'une classe dans un seul fichier.

Owlready utilise la syntaxe « with » de Python pour rendre possible cette fonctionnalité. Par défaut, chaque entité est associée à une ontologie (celle dans laquelle la première relation « rdf_type » a été créée) et les nouvelles assertions portant sur cette entité sont placées dans cette ontologie par défaut. La syntaxe « with *mon_ontologie* : » permet de délimiter un bloc de code ; toutes les assertions effectuées dans ce bloc seront placées dans *mon_ontologie* (et non dans les ontologies par défaut de chaque entité). L'exemple suivant permet de créer une classe C fille de A dans l'ontologie « onto_A », puis d'ajouter à C une classe mère B dans « onto_B » (notons que la redéfinition de la classe C ne crée pas une nouvelle classe mais étend la précédente) :

```
>>> with onto_A:
...     class C(A): pass
>>> with onto_B:
...     class C(B): pass
```

Les propriétés sont définies comme des classes filles de ObjectProperty ou DataProperty. D'autres classes mères comme FunctionalProperty, InverseFunctionalProperty,... peuvent être ajoutées. Par exemple :

```
>>> class ma_propriété(ObjectProperty, FunctionalProperty):
...     domain = [Domaine]
...     range  = [Range]
...     inverse = propriété_inverse
```

De plus, la syntaxe « >> » permet de définir facilement une propriété avec un domaine et un range :

```
>>> class ma_propriété(Domaine >> Range, FunctionalProperty):
...     inverse = propriété_inverse
```

2.3.5 Exemple

Nous donnons ici un exemple d'utilisation d'Owlready, sur l'ontologie des galettes et des crêpes présentées à la conférence IC2015 à Rennes. Owlready permet en quelques lignes de charger une ontologie, d'accéder à son contenu avec la notation pointée usuelle (par exemple *ontologie.Classe*), de lancer le raisonneur et d'accéder au résultat. Pour des exemples plus complexes sur Gene Ontology, se référer à la publication dans AIM [119].

```
# Importation du module Owlready
>>> from owlready2 import *

# Chargement de l'ontologie des galettes
>>> onto_path.append("/chemin/local/vers/les/ontos")
>>> onto=get_ontology("http://www.lesfleursdunormal.fr/static/_downloads/crepes_et_galettes.owl")
>>> onto.load()

# Création d'une instance de la classe Galette
>>> ma_galette = onto.Galette()

# Définition des propriétés de ma galette, et création de deux nouvelles instances
>>> ma_galette.a_pour_garniture = [ onto.Tomate(), onto.Viande() ]

# Exemple de définition d'une classe mixte incluant des expressions OWL et une méthode Python
>>> class GaletteNonVégétarienne(onto.Galette):
...     equivalent_to = [
...     onto.Galette
...     & ( onto.a_pour_garniture.some(onto.Viande)
...       | onto.a_pour_garniture.some(onto.Poisson)
...       ) ]
...     def manger(self): print("Beurk, je suis végétarien !")

# Appel du raisonneur HermiT
>>> sync_reasoner()

# La classe de ma_galette a été changée lors de la classification
>>> print(ma_galette.__class__)
onto.GaletteNonVégétarienne

# Appel de la méthode manger()
>>> ma_galette.manger()
Beurk, je suis végétarien !
```

2.3.6 Raisonnement automatique et monde fermé local

Owlready n'effectue aucune inférence tant que le raisonneur n'est pas appelé explicitement. Ce comportement est similaire à celui de Protégé. Pour le raisonnement, nous avons utilisé le raisonneur HermiT 1.3.8 [137]. Celui-ci a été légèrement modifié par l'ajout d'une option en ligne de commande pour afficher sur la sortie standard la classification des instances, en plus de celles des classes. La classification automatique se fait en 3 étapes : (1) exporter l'ontologie dans un fichier temporaire, (2) exécuter HermiT sur ce fichier, (3) récupérer la sortie d'HermiT et appliquer les résultats en mettant à jour les classes des individus et les superclasses des classes.

Le raisonnement en monde fermé local est rendu possible grâce à la fonction close_world() qui clôt le monde pour un individu ou une classe donnée en paramètre. Celle-ci crée les contraintes nécessaires pour

Algorithme 2.1 Algorithme pour la fonction close_world(). $\bigsqcup R$ indique l'union de tous les concepts dans l'ensemble R. L'algorithme est programmé en Python, donc « en monde fermé » : il ne prend en compte que les fait assertés.

fonction close_world(entité e) :
  si e est un individu :
    pour chaque propriété p dont le domaine est compatible avec e :
      $I = \{\ i \mid p(e, i)\ \}$
      $C = \{\ c \mid (\exists p.c)(e)\ \}$
      $R = C \cup \{\ I\ \}$
      si $R = \emptyset$: **asserte** $(\forall p.\bot)(e)$
      sinon si p n'est pas fonctionnelle : **asserte** $(\forall p.\bigsqcup R)(e)$
  sinon si e est une classe :
    $I = \{\ i \mid e(i)\ \}$
    si $I \neq \emptyset$: **asserte** e $\sqsubseteq I$
    pour chaque propriété p dont le domaine est compatible avec e :
      $I = \{i \mid \exists e' \sqsubseteq e \sqcap \exists p.\{i\}\ \}$
        $\cup\ \{i \mid \exists i',\ e(i') \wedge p(i', i)\ \}$
      $C = \{c \mid \exists e' \sqsubseteq e \sqcap \exists p.c\ \}$
        $\cup\ \{c \mid \exists i',\ e(i') \wedge (\exists p.c)(i')\ \}$
      $R = C \cup \{\ I\ \}$
      si $R = \emptyset$: **asserte** $e \sqsubseteq \forall p.\bot$
      sinon : **asserte** $e \sqsubseteq \forall p.(\bigsqcup R)$
    pour chaque classe $c \sqsubseteq e$: close_world(c)
    pour chaque individu i tel que $e(i)$: close_world(i)

clore partiellement le monde. Lorsqu'elle est appelée sur une classe, la fonction est récursive est clôt automatiquement les sous-classes et les instances de la classe.

L'algorithme 2.1 décrit la fonction close_world(). Lorsqu'elle est appelée sur un individu, pour chaque propriété, elle considère à la fois les valeurs assertées (ensemble I dans l'algorithme) et les contraintes existentielles (C). Puis la fonction crée une contrainte universelle (portant sur l'union de $\{I\}$ et C). Pour une classe, la fonction effectue trois étapes. Premièrement, si la classe a des instances, elle crée une contrainte de type « one-of » limitant les instances à celles créées (dans le cas contraire, nous supposons que les instances sont inconnues ; l'alternative serait de considérer la classe comme équivalent à Nothing, ce qui est rarement le résultat désiré). Deuxièmement, pour chaque propriété, la fonction crée une contrainte universelle, en fonction des contraintes définies sur la classe, ses ancêtres, ses descendantes et leurs individus. Troisièmement, la fonction s'appelle elle-même de manière récursive pour traiter les sous-classes et les individus.

2.3.7 Cas d'utilisation : raisonnement automatique sur les contre-indications des médicaments

Le cas d'utilisation que nous présentons ici est issu du projet VIIIP. Il s'agit d'automatiser le raisonnement et la comparaison des contre-indications. Le cas illustre les deux problèmes rencontrés avec les ontologies dans le domaine médical (le raisonnement en monde fermé local et l'utilisation massive de classes) et montre comment Owlready permet de les résoudre facilement.

Lors du projet VIIIP, nous souhaitions comparer les contre-indications de plusieurs médicaments de mêmes indications. Cependant, la comparaison est rendue difficile à cause de l'hétérogénéité dans l'expression des contre-indications : que ce soit dans les documents officiels (Résumés des Caractéristiques Produits) ou dans les bases de données sur le médicament, une même contre-indication est souvent décrite de manière différente d'un médicament à l'autre. Le Tableau 2.6 montre un exemple sur 3 médicaments et 3 maladies [2]. Le tableau

2. En réalité, les contre-indications peuvent porter sur des maladies mais également sur des symptômes, des états physiolo-

	ticagrélor	héparine	aspirine
maladie hémorragique	**CI 1**	-	-
maladie hémorragique acquise	-	-	**CI 3**
maladie hémorragique constitutionnelle	-	**CI 2**	**CI 4**

	ticagrélor	héparine	aspirine
maladie hémorragique	**CI 1**	?	**CI**
maladie hémorragique acquise	**CI**	Ok	**CI 3**
maladie hémorragique constitutionnelle	**CI**	**CI 2**	**CI 4**

TABLEAU 2.6 – Exemples de contre-indications pour 3 médicaments, telles qu'extraits d'une base de données sur le médicament (en haut) et telles qu'interprétée par un expert (en bas). « CI » signifie contre-indication, et « Ok » indique l'absence prouvée de contre-indication.

du haut correspond à ce qui est trouvé dans les textes ou les bases de données, mais il ne permet pas une comparaison facile. En effet, il existe entre les 3 maladies des relations de subsomption et de partition : par exemple, le ticagrélor est contre-indiqué avec les maladies hémorragiques, donc il l'est *a fortiori* avec les maladies hémorragiques acquises et les maladies hémorragiques constitutionnelles. De plus, l'aspirine est contre-indiquée avec les maladies hémorragiques acquises et constitutionnelles, ce qui implique qu'elle est contre-indiquée avec l'ensemble des maladies hémorragiques (par partition).

Le tableau du bas montre la manière dont un expert interpréterait les données : 3 contre-indications sont déduites (« CI » dans le tableau), ainsi qu'une absence de contre-indication (« Ok ») et une situation indécise (contre-indication dans certaines situations seulement, notée « ? »). Ce raisonnement peut devenir long et complexe lorsque l'on considère un nombre important de médicaments et de maladies, avec de multiples relations de subsomption et de partition. L'objectif du cas d'utilisation est d'automatiser ce raisonnement.

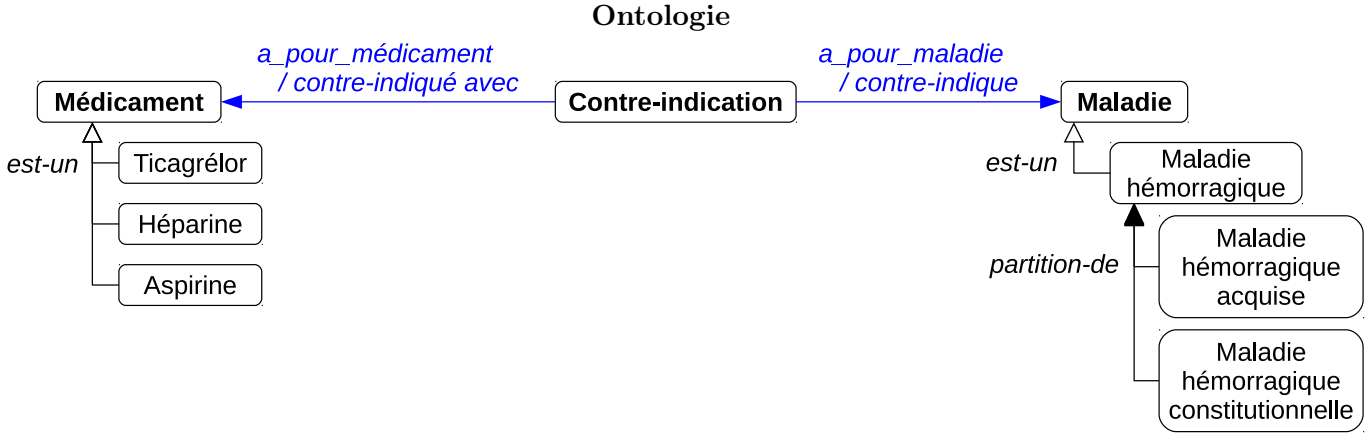
Nous avons utilisé l'ontologie comparative des médicaments, décrites à la section 2.2.3. Les médicaments ont été extraits à partir de la base de données médicamenteuse Thériaque. Les contre-indications peuvent être extraites de Thériaque ou bien extraites à la main par un expert à partir des RCP (nous avons rencontré un nombre d'erreurs assez important dans les bases de données, d'où l'intérêt d'une extraction manuelle). Les relations de subsomption ont été extraites de la CIM10 (Classification Internationale des Maladies, version 10) et enrichies à l'aide de méthodes de TAL (Traitement Automatique de la Langue naturelle) puis revues par un expert, de même pour les partitions. Comme vu précédemment (section 2.1.4), les médicaments et les maladies ne peuvent pas être représentés par des individus dans l'ontologie, mais nécessitent au contraire des classes pour pouvoir gérer la subsomption. L'ontologie comprend également une classe Contre-indication, reliée à la fois aux médicaments et aux maladies. Un schéma simplifié de l'ontologie est présenté en haut de la Figure 2.9.

Pour chaque médicament, les contre-indications sont extraites et ajoutées dans l'ontologie, sous forme d'individus de la classe Contre-indication, chaque individu étant relié à une classe de médicament et une classe de maladie. Cela peut être réalisé en 5 étapes avec Owlready (voir Figure 2.9) :

1. Créer un individu pour chaque contre-indication.
2. Relier chaque classe de médicament à ses contre-indications.
3. Relier chaque classe de maladie à ses contre-indications.
4. Asserter que tout est connu sur les contre-indications.
5. Asserter que tout est connu sur les médicaments.

Les étapes 2 et 3 nécessitent de créer des contraintes de type « value » (*role-filler*) sur les classes médicaments et maladies, et des contraintes existentielles sur les individus de la classe Contre-indication. Cela peut

giques tels que l'âge ou la grossesse, des antécédents... En médecine, le terme *conditions cliniques* est généralement utilisé pour les regrouper. Cependant, afin de garder un vocabulaire simple, nous parlerons de maladie ici, bien que condition clinique serait un terme plus approprié d'un point de vue médical.



	Python + Owlready	**Assertions en DL**
(1)	`ci1 = Contre_indication()`	$Contre_indication(ci1)$
	`ci2 = Contre_indication()`	$Contre_indication(ci2)$
	`ci3 = Contre_indication()`	$Contre_indication(ci3)$
	`ci4 = Contre_indication()`	$Contre_indication(ci4)$
(2)	`Ticagrélor.contre_indiqué_avec = [ci1]`	$Ticagrélor \sqsubseteq \exists contre_indiqué_avec.\{ci1\}$
		$(\exists a_pour_médicament.Ticagrélor)(ci1)$
	`Héparine.contre_indiqué_avec = [ci2]`	$Héparine \sqsubseteq \exists contre_indiqué_avec.\{ci2\}$
		$(\exists a_pour_médicament.Héparine)(ci2)$
	`Aspirine.contre_indiqué_avec = [ci3, ci4]`	$Aspirine \sqsubseteq \exists contre_indiqué_avec.\{ci3\}$
		$(\exists a_pour_médicament.Aspirine)(ci3)$
		$Aspirine \sqsubseteq \exists contre_indiqué_avec.\{ci4\}$
		$(\exists a_pour_médicament.Aspirine)(ci4)$
(3)	`MaladieHémorragique`	$MaladieHémorragique \sqsubseteq \exists contre_indique.\{ci1\}$
	`.contre_indique = [ci1]`	$(\exists a_pour_maladie.MaladieHémorragique)(ci1)$
	`MaladieHémorragiqueAcquise`	$MaladieHémorragiqueAcq. \sqsubseteq \exists contre_indique.\{ci3\}$
	`.contre_indique = [ci3]`	$(\exists a_pour_maladie.MaladieHémorragiqueAcq.)(ci3)$
	`MaladieHémorragiqueConstitutionnelle`	$MaladieHémorragiqueConst. \sqsubseteq \exists contre_indique.\{ci2\}$
	`.contre_indique = [ci2, ci4]`	$(\exists a_pour_maladie.MaladieHémorragiqueConst.)(ci2)$
		$MaladieHémorragiqueConst. \sqsubseteq \exists contre_indique.\{ci4\}$
		$(\exists a_pour_maladie.MaladieHémorragiqueConst.)(ci4)$
(4)	`close_world(Contre_indication)`	$(\forall a_pour_médicament.Ticagrélor)(ci1)$
		$(\forall a_pour_maladie.MaladieHémorragique)(ci1)$
		$(\forall a_pour_médicament.Héparine)(ci2)$
		$(\forall a_pour_maladie.MaladieHémorragiqueConst.)(ci2)$
		$(\forall a_pour_médicament.Aspirine)(ci3)$
		$(\forall a_pour_maladie.MaladieHémorragiqueAcq.)(ci3)$
		$(\forall a_pour_médicament.Aspirine)(ci4)$
		$(\forall a_pour_maladie.MaladieHémorragiqueConst.)(ci4)$
		$Contre_indication \sqsubseteq \{ci1, ci2, ci3, ci4\}$
(5)	`close_world(Médicament)`	$Ticagrélor \sqsubseteq \forall contre_indiqué_avec.\{ci1\}$
		$Héparine \sqsubseteq \forall contre_indiqué_avec.\{ci2\}$
		$Aspirine \sqsubseteq \forall contre_indiqué_avec.\{ci3, ci4\}$

FIGURE 2.9 – L'ontologie utilisée pour le cas d'utilisation (en haut) et le programme Python et Owlready pour créer les contre-indications du tableau 2.6 (en bas à gauche), avec les assertions correspondantes dans l'ontologie (en bas à droite). Les numéros à gauche correspondent aux étapes décrites dans le texte ; « Acq. » et « Const. » signifient « Acquise » et « Constitutionnelle » dans les formules.

Jeu de données	Exemple	Vitaros®	Pyléra®	Ciloxan®	Antarène codéine®
Nombre de médicaments	3	8	4	9	5
Nombre de maladies	3	29	17	14	26
Nombre de contre-indications assertées	4	70	30	42	50
Nombre de partitions	1	9	4	1	8
Nombre de disjonctions (sur les maladies)	1	640	166	82	499
Nombre d'inférences :					
- contre-indications	3	23	9	13	7
- absence de contre-indication	1	23	4	0	12

TABLEAU 2.7 – Métriques et nombre d'inférences obtenues dans l'exemple du Tableau 2.6 et sur les 4 jeux de données du projet VIIIP.

être réalisé très simplement avec la notation pointée proposée par Owlready : « Classe.propriété = valeurs ».

Les étapes 4 et 5 consistent à asserter que tout est connu sur les contre-indications et les médicaments (et donc que le raisonneur ne doit pas faire d'hypothèse sur l'existence éventuelle de contre-indications inconnues). Au contraire, les maladies doivent restées « en monde ouvert » si l'on souhaite pouvoir déduire de nouvelles maladies contre-indiquées pour les médicaments. Cela correspond donc à un raisonnement en monde fermé local.

Ensuite, il est possible de classer les maladies en maladies contre-indiquées avec l'aspirine et non contre-indiquées :

$$MaladieContreIndiquantAspirine \equiv Maladie \sqcap \exists contre_indique.(\exists a_pour_médicament.Aspirine)$$
$$MaladieContreOkAvecAspirine \equiv Maladie \sqcap \neg\exists contre_indique.(\exists a_pour_médicament.Aspirine)$$

Des classes similaires peuvent être définies pour les autres médicaments. L'exécution du raisonneur permet alors de classer automatiquement les maladies dans ces classes définies. Nous retrouvons alors les résultats du Tableau 2.6 (en bas). Les maladies contre-indiquées avec l'aspirine seront classées dans MaladieContreIndiquantAspirine, celles pour lesquelles l'absence de contre-indication a pu être prouvée seront classées dans MaladieOkAvecAspirine, et celles pour lesquelles une contre-indication existe uniquement dans certaines situations plus précises (« ? » dans le tableau) ne seront classées dans aucune de ces deux classes.

L'exemple décrit précédemment est très simple, avec seulement 3 médicaments, 3 maladies et 4 contre-indications. Le Tableau 2.7 montrent les métriques pour cet exemple et pour 4 jeux de données issus du projet VIIIP. Chacun de ces jeux de données comprend un nouveau médicament (Vitaros®, Pyléra®, Ciloxan® et Antarène codéine®) et plusieurs médicaments similaires. Les maladies et conditions cliniques ont été codées avec la CIM10 autant que possible. Le tableau donne également le nombre d'inférences obtenues pour chaque jeu de données.

Pour conclure, ce cas d'utilisation montre l'intérêt de la syntaxe de « haut niveau » proposée par Owlready pour le raisonnement en monde fermé local et la manipulation des classes : sur la Figure 2.9 le programme Python à gauche est indubitablement plus concis, plus clair et plus compréhensible que son équivalent en notation formelle à droite. Par ailleurs, un programme plus sophistiqué pourrait tirer parti des possibilités offertes par Python, par exemple pour interroger des bases de données, pour générer des interfaces graphiques pour que les experts saisissent ou vérifient les données, et pour produire une sortie sous la forme d'un site web en HTML.

2.3.8 Discussion

Dans cette section nous avons présenté Owlready, un outil pour la programmation orientée ontologie en Python. Cet outil permet d'unifier le système objet du langage Python et le modèle des ontologies. Il permet un accès et une manipulation très faciles des ontologies, qui se traduisent par un code source beaucoup plus

	Goldman 2003 [56]	Kalyampur 2004 [80]	Koide 2005 [92]	Babik 2006 [15]	Stevenson 2011 [172]	Lamy
Approche	statique	statique	dynamique	dynamique	semi-dynamique	dynamique
Langage	C#	Java	Common Lisp	Python	Java	Python
Classification des classes	-	-	-	-	-	X
Classification des individus	-	-	X	-	X	X
Syntaxe spécifique pour les expressions de classe OWL	-	-	X	-	-	X
Ajout de méthodes	-	-	-	-	-	X
Raisonnement en monde fermé local	-	-	-	-	-	X
Manipulation des classes (contraintes de type « value »)	-	-	-	-	-	X

TABLEAU 2.8 – Comparaison des différents outils proposés pour la programmation orientée ontologie.

concis par rapport aux API classiques. Ce point est important non seulement pour mesurer la facilité d'emploi, mais aussi car la longueur et la complexité d'un programme informatique sont généralement corrélées avec le nombre de bogues. Enfin, Owlready propose également deux fonctionnalités avancées qui répondent aux difficultés que nous avons rencontrées dans le domaine médical : le raisonnement en monde fermé local et la manipulation des classes.

2.3.8.1 Comparaison à la littérature

Le Tableau 2.8 compare les différents outils proposés dans la littérature pour la programmation orientée ontologie. Peu d'approches sont allées aussi loin dans l'unification entre modèle objet et ontologie que l'outil que nous avons proposé ici. En particulier la classification automatique des classes avec un raisonneur n'avait encore jamais été rapportée.

2.3.8.2 Raisonnement en monde fermé local

Dans la littérature, plusieurs auteurs ont également éprouvé le besoin d'un raisonnement en monde fermé local, notamment pour définir des interfaces en langage naturel [73] ou pour gérer des services web [62]. Cependant, la plupart des travaux existants ne se sont pas focalisés sur le raisonnement en monde fermé local, mais sur le raisonnement non monotone, qui est un problème plus large incluant aussi d'autres fonctionnalités comme l'héritage révisable (*defeasible inheritance*) ou l'utilisation de règles « par défaut » (par exemple considérer que le coeur est placé à gauche du corps du patient, *à moins que* le contraire ne soit précisé pour un patient particulier).

Les approches proposées dans la littérature s'appuient sur des contraintes d'intégrité inspirées des bases de données relationnelles [135], l'extension d'OWL avec des opérateurs spécifiques pour distinguer les faits connus des faits assumés [61], la construction de bases de connaissances hybrides combinant logiques de

description et règles de production (MKNF, *Minimal Knowledge and Negation as Failure*) [136] qui peuvent être rendus compatibles avec OWL [90], les logiques de descriptions circonscrites (*circumscriptive description logics*) [95] ou l'extension de la logique de description $\mathcal{SROIQ}$ avec une *NBox* spécifiant les propriétés à « fermer » et supportant la négation par l'échec (*negation as failure*) [156].

Ici, nous avons présenté une approche différente pour le raisonnement en monde fermé local. Cette approche est plus simple et plus limitée. Elle consiste simplement à ajouter des contraintes universelles à l'ontologie, de manière automatique. Ces contraintes sont générées par la fonction close_world() que nous avons décrite, cette fonction agit comme une sorte de préprocesseur pour OWL. Notre approche est donc 100% compatible avec OWL et les outils existant comme HermiT. La fonction close_world() permet de « fermer » un individus ou une classe ; en revanche elle ne permet pas de « fermer » une propriété, ce qui serait susceptible de rendre le système indécidable [22].

Notre approche présente cependant des limites. Premièrement, le statut de « fermé » n'est pas renseigné dans l'ontologie : si des propriétés sont ajoutées ou retirées à un individu, les contraintes générées ne sont plus à jour. Il est donc nécessaire d'appeler close_world() juste avant d'effectuer le raisonnement. Deuxièmement, l'algorithme proposé pour close_world() peut s'avérer trop permissif lorsque l'héritage est en jeu. Par exemple, considérons l'ontologie suivante : $B \sqsubseteq A$, $A \sqsubseteq \exists R.C$, $B \sqsubseteq \exists R.D$ (où A, B, C et D sont des classes et R une propriété). close_world(B) produira la contrainte suivante : $B \sqsubseteq \forall R.(C \sqcup D)$. Cependant, si $D \sqsubseteq C$, il serait préférable de produire la contrainte $B \sqsubseteq \forall R.D$. La fonction close_world() pourrait être améliorée en prenant en compte les héritages connus avant raisonnement.

Malgré ces limites, la fonction close_world() s'est révélée très utile lors du projet VIIIP.

2.3.8.3 Conclusion

Nous avons présenté la conception d'un module pour la programmation orientée ontologie en Python, intégrant des fonctionnalités de haut niveau pour les ontologies biomédicales. L'article publié dans *Artificial Intelligence in Medicine* [119] présente des exemples additionnels, notamment avec *Gene Ontology*, ainsi des mesures de performances et les procédures de tests qui ont été mises en oeuvre. Owlready a été testé jusqu'à cent millions de triplets RDF, mais peut potentiellement en supporter davantage.

Le module Owlready a été utilisé lors du projet VIIIP, puis réutilisé par la suite dans le projet de recette iconique en médecine traditionnelle africaine. Il est diffusé en tant que logiciel libre sous licence GNU LGPL v3 et disponible en ligne à l'adresse suivante : `https://bitbucket.org/jibalamy/owlready2`. La documentation complète d'Owlready est aussi disponible à l'adresse suivante : `http://owlready2.readthedocs.io/en/stable/`. De nombreux programmeurs et chercheurs l'ont utilisé, par exemple dans des applications portant sur du raisonnement sémantique sur l'intensité de la douleur [58], le traitement automatique de la langue [86, 85], et la conception de pièces mécaniques dans les hélicoptères [199]. Plus généralement, le langage Python et les ontologies sont tous deux très utilisés en biologie, et il n'existait pas jusque là d'outil pour accéder aux ontologies OWL en Python. Owlready répond donc à un réel besoin.

Les perspectives pour Owlready incluent la génération automatique de boîtes de dialogue pour éditer les individus ou les classes, ainsi que l'utilisation d'Owlready pour la visualisation des connaissances décrites dans une ontologie. La programmation orientée ontologie permet d'obtenir rapidement un prototype fonctionnel, ce qui est très important lorsque l'on met au point des outils de visualisation en s'appuyant sur des méthodes de développement agile et des cycles courts.

Chapitre 3

Visualisation iconique des connaissances

« Cette multiplication de caractères témoignait-elle [...] d'un sentiment d'angoisse devant le fait que le sens, toujours intimement lié à l'ombre d'un son unique, se dérobe éternellement ? »
(L'autre ville, Michal Ajvaz, éditions Mirobole, 2015)

3.1 Introduction et contexte

Une première approche pour visualiser les connaissances consiste à utiliser des icônes. Il s'agit de traduire sous forme d'éléments graphiques (généralement couleurs et pictogrammes) la *nature* des connaissances, par exemple les organes touchés par les effets indésirables d'un médicament. Les icônes peuvent ensuite être ajoutées dans un texte pour le baliser, ou bien être intégrées dans des interfaces graphiques plus sophistiquées.

Dans ce chapitre, nous définirons d'abord les icônes, les langages iconiques et les langages diagrammatiques, puis nous étudierons les approches existantes pour formaliser les langages graphiques. Ensuite, nous présenterons une première contribution concernant la création d'un langage graphique à base d'icônes pour visualiser les recettes et les remèdes dans la médecine traditionnelle d'Afrique de l'Ouest, et une seconde contribution proposant une méthode à base d'ontologie pour formaliser la sémantique des langages iconiques.

3.1.1 Les icônes simples

Une *icône* est un symbole visuel de petite taille, constituée d'un assemblage de pictogrammes et de couleurs. Lorsqu'elles représentent des objets réels, les icônes sont généralement simplifiées ou stylisées, et ne recherchent pas une ressemblance exacte. Les icônes peuvent aussi comprendre des lettres ou des mots, mais ne se limitent pas à du texte. Une photographie ou un mot seul ne constitue donc pas une icône. Selon la théorie du *double codage* [146], les mécanismes cognitifs mis en jeu lors de la perception sont distincts selon le type de signe, textuel ou visuel. Il y aurait deux modes de représentation mentale distincts : un premier système pour les représentations verbales ou propositionnelles correspondant au traitement du texte, et un second système pour les représentations visuelles telles que les icônes. Des travaux récents en neuro-imagerie ont montré que les icônes stimulent le système sémantique dans le cerveau, mais qu'elles sont traitées comme des images d'un point de vue cognitif, plutôt que comme des mots [72]. Les deux modes de représentations sont complémentaires et peuvent se potentialiser l'un l'autre, notamment en ce qui concerne la mémorisation.

Nous appelons *icônes simples* des jeux d'icônes contenant une liste fixe d'icônes prédéfinies, sans règles particulières pour la création de nouvelles icônes. De nombreux exemples existent dans la vie quotidienne tels que l'étiquetage des produits chimiques dangereux, les étiquettes pour l'entretien des textiles, les icônes indiquant les possibilités de recyclage sur les emballages ou les panneaux de signalisation dans les lieux publics (voir exemples Figure 3.1).

Plusieurs travaux se sont intéressés à catégoriser les icônes dans des taxonomies selon différentes approches. La plupart des taxonomies classent les icônes en fonction des stratégies de représentation iconique [142], avec trois grandes catégories :



FIGURE 3.1 – Exemple d'icônes simples. De haut en bas : étiquetage des produits chimiques dangereux (produit inflammable, comburant, explosif, corrosif, dangereux pour l'environnement, irritant / nocif, toxique), étiquettes pour l'entretien des textiles (laver à 40°, laver à la main, ne pas repasser), emballage recyclable.

— **L'analogie ou similarité visuelle**, par exemple en reprenant la forme des organes à l'aide de pictogrammes,
— **L'indice** ou association sémantique, en représentant un objet fréquemment associé à ce que l'on veut représenter, par exemple les toilettes pour représenter une diarrhée,
— **La convention arbitraire**, par exemple les trois couleurs du feu rouge.

D'autres les classent selon l'objectif dans lequel les icônes sont employées [126] : les auteurs distinguent des icônes d'*action* qui représentent une activité (par exemple enregistrer un fichier) et des icônes de *connaissance* qui représentent l'état d'un objet (par exemple le type d'un fichier dans un gestionnaire de fichiers).

3.1.2 Les langages iconiques

Un *langage iconique* est un langage comprenant un jeu de primitives graphiques (des pictogrammes ou formes et des couleurs), ainsi que des règles grammaticales permettant d'assembler ces primitives pour former des icônes. Les icônes peuvent aussi être créées par assemblage de « morceaux d'icône » que l'on appelle des *iconèmes*. Les langages iconiques sont particulièrement intéressants lorsque l'on a besoin d'un grand nombre d'icônes : ils permettent de créer beaucoup d'icônes à partir d'un nombre limité de primitives, ce qui facilite l'apprentissage tout en garantissant une certaine cohérence entre les icônes. Cette cohérence peut faciliter l'utilisation des icônes : par exemple il est plus facile de rechercher visuellement des objets partageant un attribut en commun si cet attribut est représenté de la même manière sur les icônes de chaque objet.

Des langages iconiques ont été développés dans plusieurs domaines où le nombre d'icônes nécessaires était élevé, tels que la signalisation routière. Les icônes utilisées dans les logiciels informatiques suivent parfois des règles grammaticales, par exemple l'ajout d'un iconème « étoile » à une icône représentant un format de fichier pour signifier « nouveau fichier de ce format » ; cependant il ne s'agit pas d'un langage unique standardisé. D'autres langages iconiques ont été réalisés afin d'offrir une alternative visuelle et analogique à l'écriture, tels que la Sémantographie de C.K. Bliss. Ces langages visent généralement les touristes, les personnes illettrées et/ou les personnes dont les aires cérébrales dédiées au langage symbolique ont été endommagées suite à un accident. Enfin, l'écriture de certaines langues, comme le chinois ou les hiéroglyphes égyptiens, était à l'origine des langages iconiques. Ces langues se sont cependant éloignées de leur nature iconique originelle, avec la stylisation des pictogrammes et l'utilisation de symboles phonétiques.

La conception des langages iconiques peut s'appuyer sur les travaux du cartographe J. Bertin [18] sur

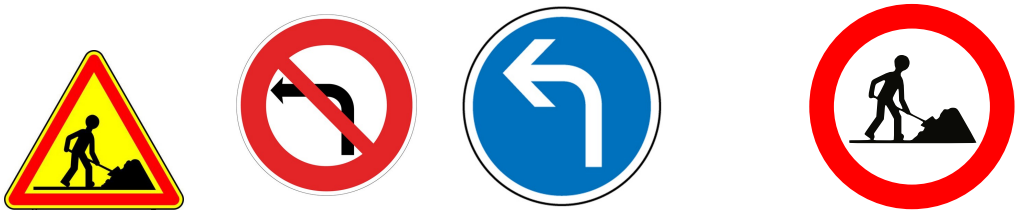


FIGURE 3.2 – Exemples de panneaux qu'il est possible de former en utilisant la signalisation routière comme un langage iconique. De gauche à droite : attention travaux, interdiction de tourner à gauche, virage à gauche obligatoire, interdiction de travaux. Ce dernier panneau est un exemple de panneau inconsistant (et inexistant), car les automobilistes ne font pas de travaux.

la sémiologie graphique et les variables rétiniennes, sur les règles de perception issues de la psychologie de la forme (*gestalt theory*) [147] et sur les analogies et conventions existantes. Pour plus d'information sur la conception d'un langage iconique, se référer à mes travaux de thèse [118] et à la communication effectuée à l'atelier EGC (Extraction et Gestion des Connaissances) [105].

Nous présentons ici deux exemples de langages iconiques : la signalisation routière et le langage VCM.

3.1.2.1 La signalisation routière

Les panneaux de signalisation routière reposent sur un langage iconique simple et bien connu qui permet de donner des consignes visuelles aux automobilistes. Les icônes de ce langage sont les panneaux routiers. Quatre principaux types de panneaux peuvent être distingués : les panneaux d'interdiction (avec un rond rouge, parfois barré), les panneaux d'obligation (rond bleu), les panneaux informant d'un danger à venir (triangle rouge) et les panneaux apportant d'autres types d'informations (carré). Les panneaux temporaires ont un fond jaune et les permanents un fond blanc. Enfin, un pictogramme placé au centre vient préciser l'objet du panneau.

Une grammaire combinatoire permet de combiner ces trois éléments entre eux (forme, fond et pictogramme, voir exemple Figure 3.2). Cependant, certaines combinaisons n'ont pas de sens et des règles sémantiques sont nécessaires en plus des règles purement grammaticales pour empêcher la construction de panneaux inconsistants. Par exemple, le pictogramme « virage à gauche » peut figurer à la fois sur des panneaux d'ordre (virage à gauche obligatoire), d'interdiction (interdit de tourner à gauche) ou d'information de danger (attention virage à gauche dangereux). Au contraire, le pictogramme « travaux » peut figurer sur des panneaux d'information de danger (attention travaux), mais pas sur des panneaux d'ordre ou d'interdiction (voir quatrième panneau Figure 3.2).

3.1.2.2 Le langage VCM (Visualisation des Connaissances Médicales)

Le langage VCM a été conçu durant mes travaux de thèse [118, 104, 106]. Ce langage iconique propose des icônes pour représenter les principales conditions cliniques d'un patient, dont les symptômes, les maladies, les états physiologiques (tels que la grossesse ou les classes d'âge), les risques et les antécédents de maladie, les traitements médicamenteux ou non, les examens de biologie et les procédures de suivi. Le langage VCM comprend un ensemble de primitives graphiques (pictogrammes, formes et couleurs) pouvant être combinées selon une grammaire pour générer un grand nombre d'icônes. Ces icônes n'ont pas pour objectif d'être aussi précises que les textes médicaux, mais visent au contraire à les compléter de manière synthétique. Un didacticiel est disponible sur le site Internet dédié à VCM : `http://vcm.univ-paris13.fr/`.

La Figure 3.3 montre l'assemblage des icônes VCM et la Figure 3.4 des exemples d'icônes. Une icône VCM se compose d'une couleur, d'une forme de base avec un ensemble de modificateurs de forme, d'un pictogramme central, ainsi qu'éventuellement d'un exposant associant un pictogramme et une couleur, et d'un second exposant. Une icône simple combine (1) une couleur indiquant l'aspect temporel de l'icône : rouge pour un état actuel du patient, marron pour un antécédent, orange pour un risque ; (2) une forme de base : un cercle pour un état physiologique ou un carré pour un état pathologique ; et (3) un pictogramme central indiquant la localisation anatomo-fonctionnelle (par exemple cardiaque, pulmonaire...) ou la caractéristique

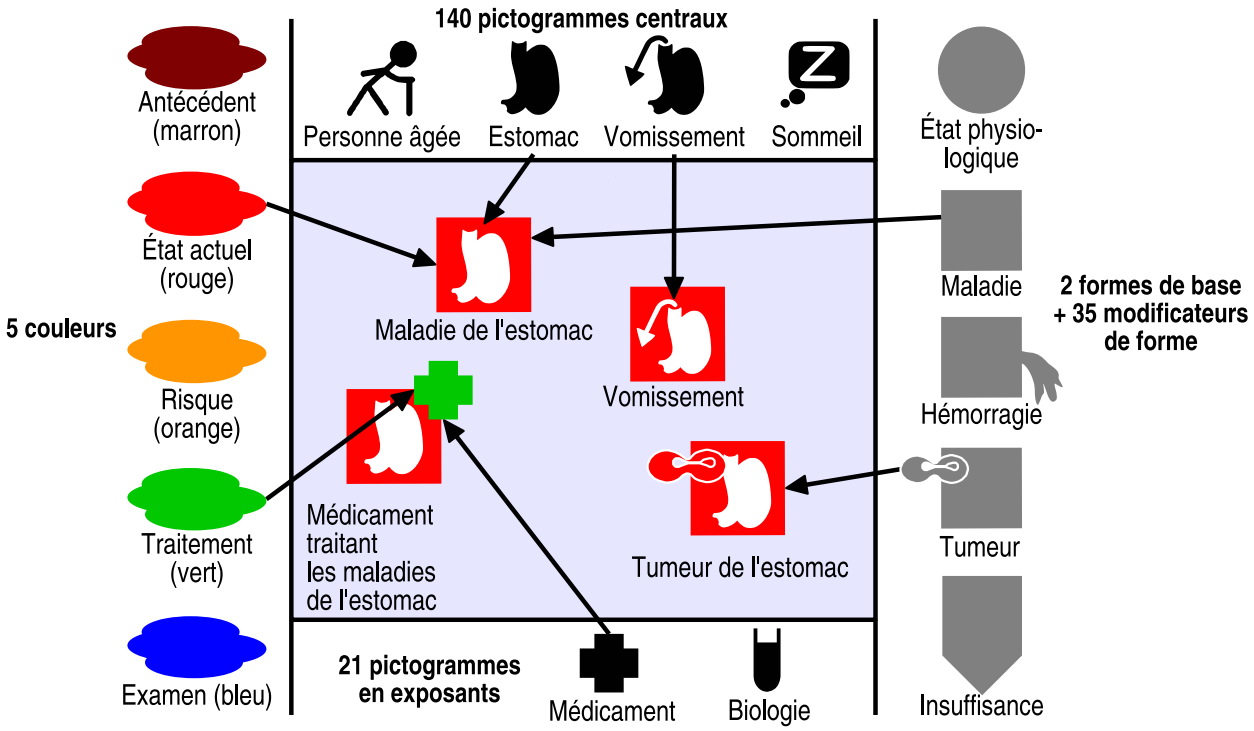


FIGURE 3.3 – Création d'icônes VCM par la combinaison de plusieurs éléments.

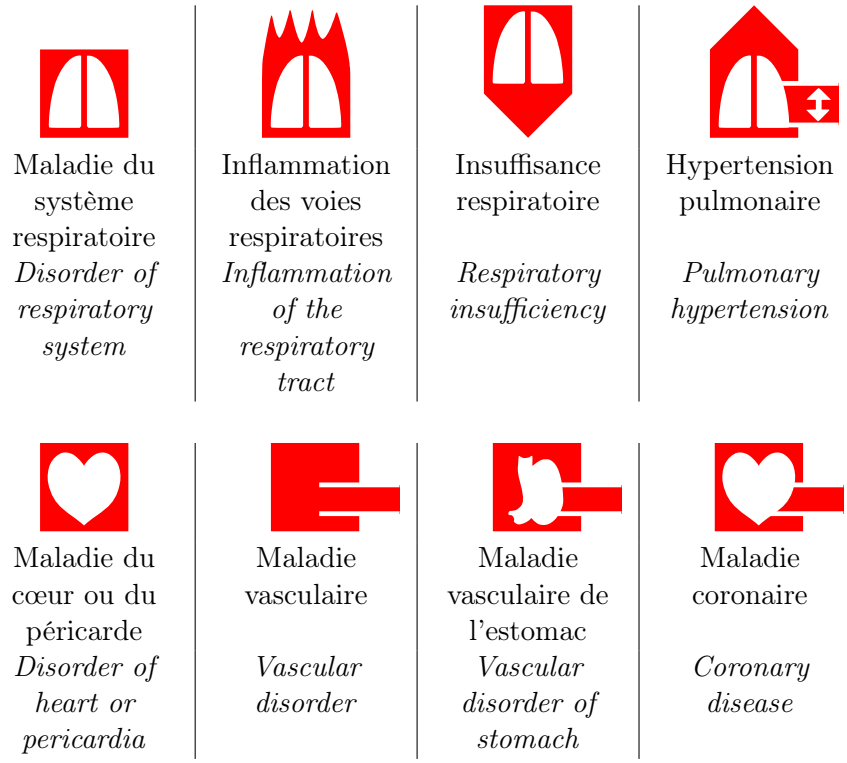


FIGURE 3.4 – Exemples d'icônes VCM avec leurs libellés en français et en anglais.

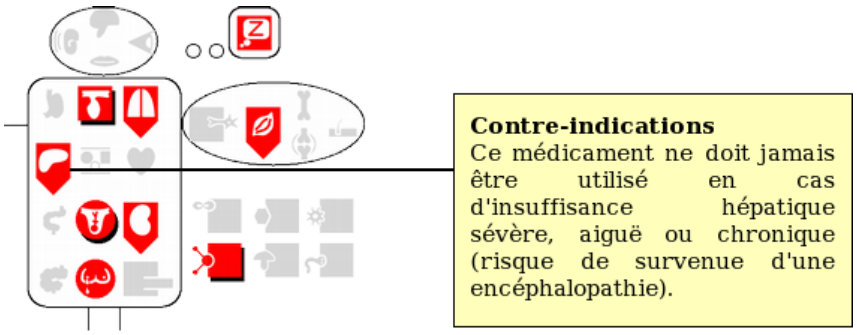


FIGURE 3.5 – Exemple d'interface « Monsieur VCM » présentant ici les contre-indications du Stilnox®. L'utilisateur a cliqué sur l'icône « insuffisance hépatique », ce qui a déclenché l'affichage du texte correspondant.

patient (par exemple grossesse) impliquée. Les structures anatomiques et les fonctions qu'elles effectuent sont représentées par le même pictogramme (par exemple « poumon » et « respiration »).

Les icônes de maladie ou de symptôme peuvent ensuite être précisées, selon deux approches différentes : (1) pour les maladies ou symptômes spécifiques à un système anatomo-fonctionnel (par exemple les vomissements, qui sont spécifiques à l'estomac), le pictogramme central est modifié ; (2) pour les maladies ou symptômes génériques pouvant être décrits comme une morphologie associée à un système anatomo-fonctionnel (par exemple les tumeurs, les infections ou les insuffisances d'une fonction), un modificateur de forme est ajouté à la forme de base (par exemple, les tumeurs sont représentées par deux cellules en division). Ces deux approches peuvent être combinées, et plusieurs modificateurs de forme peuvent être présents du moment qu'ils ne se recouvrent pas spatialement.

Les icônes de traitements et d'examens sont construites à partir de l'icône de la maladie traitée ou du risque surveillé, en ajoutant un pictogramme en exposant, de couleur verte pour les traitements ou bleue pour les examens. Ce pictogramme en exposant indique le type de traitement (par exemple chirurgical) ou d'examen (par exemple imagerie). Un second exposant peut être ajouté pour représenter un professionnel de santé ou un document en relation avec une maladie, par exemple l'icône « cardiologue » sera construite en ajoutant l'exposant « professionnel de santé » à l'icône « maladie cardiaque ».

Les icônes VCM peuvent être insérées en marge d'un texte ou dans une interface utilisateur. Elles peuvent aussi être présentées sur l'interface « Monsieur VCM » (Figure 3.5) qui rassemblent toutes les icônes d'un type donné dans un document (par exemple les contre-indications d'un médicament) sur un schéma anatomique stylisé où chaque pictogramme occupe une place fixe (les organes dans le « bonhomme », les étiologies en dehors en bas à droite). Les icônes sont alors placées selon leurs pictogrammes. Lorsque plusieurs icônes occupent le même espace, une ombre noire est ajoutée. Les emplacements qui ne reçoivent aucune icône affichent un pictogramme grisé, ce qui permet à l'utilisateur de voir d'un seul coup d'œil qu'il n'y a pas d'information concernant cet organe ou cette étiologie. Enfin, les icônes sont interactives : l'utilisateur peut cliquer sur une icône pour afficher l'extrait de texte correspondant.

Les icônes VCM ont initialement été développées pour présenter les connaissances sur les médicaments (contre-indications, interactions médicamenteuses, précautions d'emploi, effets indésirables, *etc.*). Dans cette application, nous avons comparé une interface interactive utilisant VCM par rapport à une interface « classique » textuelle. Les résultats montrent que l'interface iconique permet un accès deux fois plus rapide à l'information mais aussi une diminution significative des erreurs [116]. Par la suite, le langage iconique a été étendu et d'autres applications ont été développées : (a) l'association d'icônes dans les dossiers patients, pour en faciliter la lecture [167], (b) l'aide à la consultation des guides de bonnes pratiques cliniques, qui contiennent souvent de nombreux cas particuliers que les icônes permettent d'identifier plus facilement [148], et (c) l'utilisation dans les moteurs de recherche médicaux, pour faciliter la lecture des listes de résultats parfois longues et fastidieuses [60]. Ces applications seront présentées en détail à la section 5.1.

À l'heure actuelle, VCM est un des langages iconiques les plus poussés, tous domaines confondus.

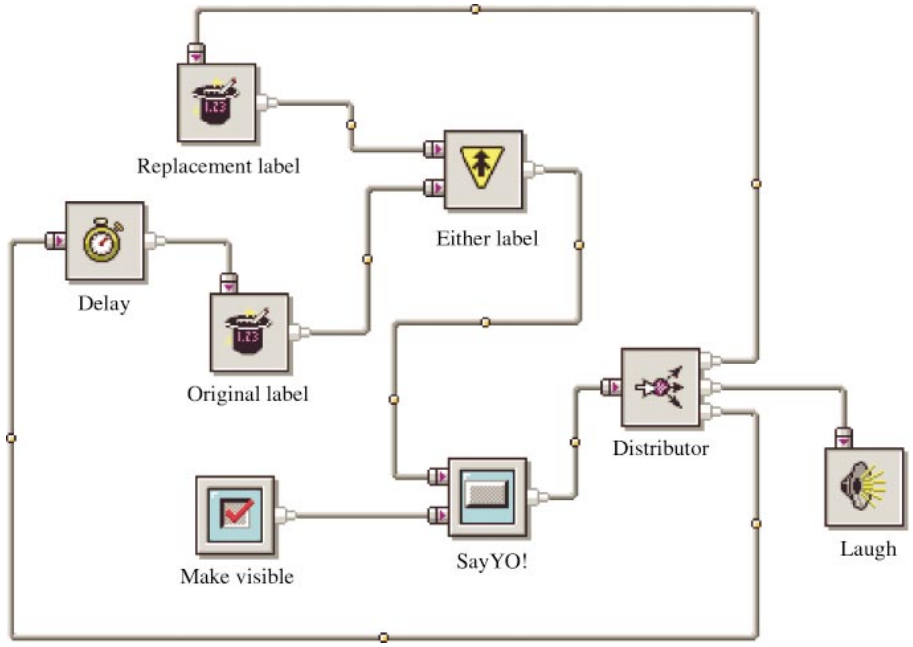


FIGURE 3.6 – Exemple de langage diagrammatique (d'après [21]).

3.1.3 Les langages diagrammatiques

Les *langages diagrammatiques* représentent des informations ou des connaissances à l'aide d'un diagramme composé de plusieurs éléments reliés par des traits ou des flèches (voir exemple Figure 3.6). Ces langages ont notamment été utilisés pour la représentation de processus en général, et plus particulièrement la programmation visuelle [189] ou le requêtage visuel de bases de données. Un exemple bien connu de langage diagrammatique est le langage UML (*Unified Modeling Language*) pour la modélisation objet.

L'utilisation d'icônes ou de pictogrammes dans les langages diagrammatiques offre des possibilités intéressantes, notamment en termes de mémorisation [21]. Il y a une complémentarité naturelle entre langage iconique et langage diagrammatique.

3.1.4 La formalisation des langages graphiques

L'essentiel de la littérature sur la formalisation des langages graphiques remonte à 20-30 ans, ce qui correspond à l'émergence des interfaces graphiques et des langages visuels de modélisation. Aujourd'hui, les icônes sont omniprésentes mais paradoxalement, l'intérêt apporté à leur formalisation et leur sémantique semble avoir quasiment disparu. Les travaux récents sur les icônes se focalisent sur le design graphique [71] et l'évaluation de l'utilisabilité [54]. Dans le domaine médical, deux exemples de ces travaux concernent le design d'icônes pour des interfaces utilisateurs pour le suivi de patient à distance [81], et pour un système d'information aux urgences, en utilisant un design participatif [159]. Dans la suite de cette section, nous nous intéresserons uniquement aux travaux sur la sémantique des icônes et plus généralement des langages graphiques ou visuels.

Historiquement, les langages graphiques (iconiques ou diagrammatiques) ont d'abord fait l'objet d'études sémiologiques [131, 132] qui consistaient généralement à décomposer les signes graphiques tels que les panneaux routiers (voir Figure 3.7), afin d'associer à chaque élément (couleur, pictogramme, ...) une partie du sens du signe. Bien que faciles à comprendre pour un humain, ces travaux ne permettaient pas une véritable formalisation de la sémantique.

Ensuite, les travaux se sont orientés vers la conception de grammaires pour les langages visuels. Plusieurs approches ont été proposées. On peut citer les grammaires de multi-ensembles à contraintes (*constraint multiset grammars*) [129] qui s'inspirent des grammaires de Chomsky pour définir la grammaire des langages visuels. Les grammaires positionnelles (*positional grammars*) [35] représentent une autre approche qui s'appuie sur une grammaire textuelle classique complétée par un évaluateur de position (PE) chargé de traduire les

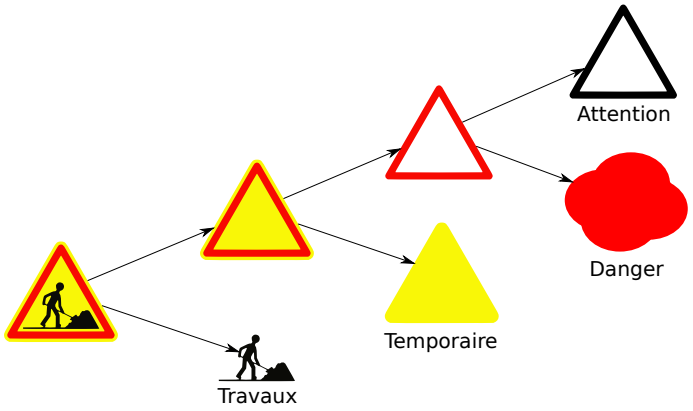


FIGURE 3.7 – Structure sémiotique du panneau routier « attention travaux », d'après JG Meunier.

phrases de manière graphique. Ces approches grammaticales ont notamment été appliquées aux langages diagrammatiques. Ces grammaires ont permis la réalisation d'analyseurs (*parsers*) et de compilateurs pour ces langages.

Une autre approche consiste à transformer les « textes » d'un langage graphique en graphes orientés étiquetés, sans nécessairement passer par une grammaire, grâce à l'utilisation d'une syntaxe abstraite visuelle (*abstract visual syntax*) [49, 44]. Le graphe constitué permet ensuite d'exprimer de la sémantique et d'effectuer des démonstrations. Des syntaxes abstraites visuelles ont notamment été définies pour des langages graphiques dérivant du lambda calcul.

Deux approches ont cherché à décrire les langages graphiques de manière lexicale et syntaxique. L'utilisation de logiques de description (DL) est une première approche de ce type, et s'appuie sur des définitions logiques telles que « une relation d'héritage est une ligne qui touche deux rectangles de classe et qui comporte un triangle à l'une de ses extrémités » [64]. Une autre approche a utilisé des schémas UML pour définir la syntaxe concrète d'un langage visuel [14]. La définition de la syntaxe peut être vérifiée à l'aide de la logique du premier ordre, ce qui permet de vérifier qu'un diagramme correspond bien à un seul modèle (absence d'ambiguïté). Plus récemment, une approche basée sur le contexte local a été proposée [36] pour les langages graphiques schématiques. Elle permet une définition lexicale et syntaxique, en définissant les éléments du langage (boîtes, flèches, ...) et la manière de les relier entre eux (cardinalités, *etc*). Ces deux approches permettent de décrire les langages graphiques mais restent limitées dans l'expression de la sémantique.

NC Kuicheu *et al.* ont proposé une ontologie des icônes appelées IcOnto [96], avec pour objectif la formalisation des icônes à l'aide de logiques de description. Les auteurs définissent une icône comme l'intersection d'une partie physique, l'icône décrite par ses attributs graphiques (couleurs, forme, *etc.*), et une partie logique, un ensemble d'attributs ou de caractéristiques communément associés avec l'icône (par exemple des émotions, des activités, des sons). L'ontologie permet l'inférence d'attribut *via* la subsomption mais ne permet pas le raisonnement sémantique au-delà de cette situation simple.

En conclusion, diverses approches ont donc été proposées pour formaliser les langages graphiques. Cependant deux problèmes sont couramment rencontrés :

— Plusieurs approches sont de nature grammaticale, ce qui pose un problème car la grammaire et la sémantique ne sont pas la même chose : par exemple, la phrase « la souris mange le chat » est grammaticalement correcte mais sémantiquement absurde.

— La majorité des approches se sont intéressées aux langages diagrammatiques tels qu'UML (*Unified Modeling Language*) plutôt qu'aux langages iconiques.

Encadrement associé à cette contribution :

1. Appoh Kouame, doctorant en informatique
   (taux d'encadrement 50%)

Publications associées à cette contribution :

1. Kouame A, **Lamy JB**, Lo M, Brou KM. Vers un système iconique d'aide à la décision pour les praticiens de la médecine traditionnelle. Accepté au Colloque Africain sur la Recherche en Informatique et Mathématiques CARI 2016
2. Kouamé A, **Lamy JB**, Brou KM, Lo M. Modelization of recipe in African traditional medicine with visual ontology approach, iconic sketch. Africomm conference 2016, Ouagadougou, Burkina Faso

3.2 Contribution : Création d'un langage graphique pour représenter les plantes médicinales, les recettes et les remèdes

Lors de ses travaux de thèse, Appoh Kouame a construit une ontologie de la médecine traditionnelle d'Afrique de l'Ouest, puis il s'est intéressé à la présentation graphique des individus de cette ontologie (sous ma direction pour la partie graphique). La représentation graphique a porté plus spécifiquement sur les recettes de remèdes traditionnels et sur les plantes utilisées comme ingrédient dans ces recettes.

La Figure 3.8 montre l'extrait de l'ontologie portant sur les remèdes et leurs recettes. Nous avons fait la distinction entre les remèdes, qui sont des objets physiques, et les recettes, qui sont des processus permettant de fabriquer les remèdes. Chaque recette comprend un ou plusieurs ingrédients, et un unique mode de préparation (par exemple décoction). En effet, lors de ses recherches, Appoh n'a pas rencontré de recette avec des processus complexes incluant plusieurs étapes. Nous avons donc décidé de nous limiter à des recettes simples avec une seule étape dans laquelle tous les ingrédients sont mélangés selon un mode de préparation unique. Un ingrédient se compose d'une plante d'une espèce donnée et d'une partie de plante (par exemple la feuille ou la racine).

3.2.1 Représentation iconique des plantes

L'utilisation de photos pour identifier les plantes est problématique car au sein d'une même espèce, l'aspect d'une plante varie beaucoup, en fonction de son âge, de l'environnement et du pays : par exemple, la morphologie du Baobab est très différente en Côte d'Ivoire et au Sénégal. Appoh s'est donc orienté vers l'utilisation d'icônes pour représenter les plantes. Le nombre important d'espèces de plantes fait qu'il n'est

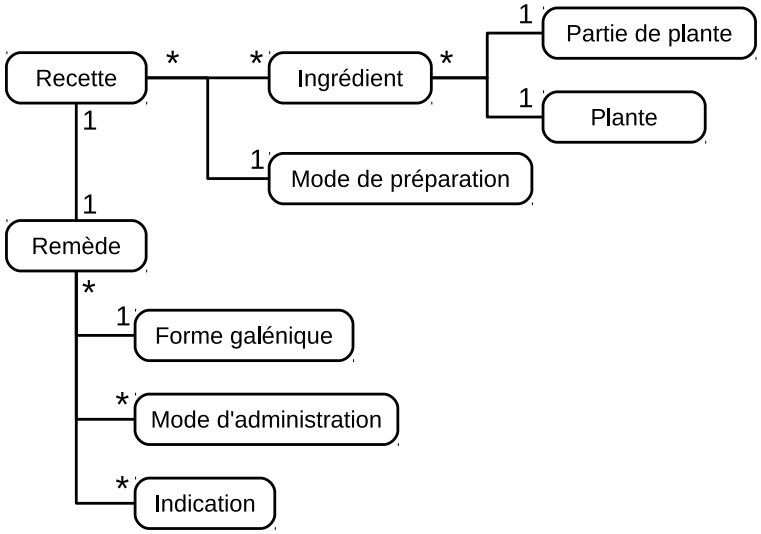


FIGURE 3.8 – Extrait de l'ontologie de la médecine traditionnelle d'Afrique de l'Ouest, concernant la description des recettes et des remèdes.

Recette	Plante	Partie	Préparation	Administration
1	Senna occidentalis L. Azadirachta indica A. Juss.	Feuilles Feuilles	Décoction du mélange	Boisson et bain
2	Senna occidentalis L.	Feuilles	Décoction	Boisson
3	Adenia lobata (Jacq.) Engl.	Feuilles	Décoction	Boisson
4	Senna alata (L.) Roxb.	Feuilles	Infusion	Boisson
5	Carica papaya L.	Fruits	Pétrissage avec de l'eau	Absorption

TABLEAU 3.1 – Exemples de recettes en médecine traditionnelle de l'Afrique de l'Ouest pour le traitement du paludisme. La recette 1 combine deux plantes.

pas possible de dessiner ou de mémoriser une icône spécifique pour chaque espèce. Nous avons donc décidé de créer un langage iconique pour représenter les espèces de plantes, avec une icône pour chaque espèce, en nous appuyant sur des critères botaniques. La méthode que nous avons suivie pour la construction du langage iconique s'est inspirée de celle mise au point pour la construction de VCM.

3.2.1.1 Extraction d'un jeu de données

Dans un premier temps, Appoh a extrait un jeu de données portant sur le traitement du paludisme, à partir de la littérature [4, 94, 87, 145, 144, 9, 198] et de ses rencontres directes avec des praticiens de la médecine traditionnelle en Côte d'Ivoire et au Sénégal. Le jeu de données comprend 30 recettes utilisant 22 plantes (certaines recettes utilisent les mêmes plantes, avec des modes de préparation ou d'administration différents). Un extrait figure dans le Tableau 3.1.

3.2.1.2 Détermination des critères

Ensuite, nous avons recherché les critères permettant de caractériser les plantes. Ces critères sont principalement de nature botanique (silhouette générale et dimensions de la plante, forme ou couleur des fleurs et fruits, position des feuilles sur les tiges,...) mais d'autres relèvent de la vie pratique (comestibilité de la plante). 40 critères ont été définis et renseignés pour chacune des 22 plantes, à partir de sources botaniques. Pour certains critères comme la forme du fruit, il existe des valeurs *génériques* telles que « fruit rond » ou « fruit allongé » mais il y a aussi des formes de fruit très connues et caractéristiques, telles que la banane ou l'ananas. Dans ce cas, nous avons défini à la fois des valeurs génériques et des valeurs spécifiques, qui pourront faire l'objet d'un pictogramme spécifique.

3.2.1.3 Sélection des critères

Une fois les critères établis, nous avons cherché à réduire la liste des critères pour sélectionner ceux que nous allons retenir pour construire les icônes. En effet, le nombre de critères (40) est trop élevé pour les faire tous apparaître sur une seule icône. Le choix des critères sélectionnés a pris en compte plusieurs paramètres :

— **La facilité avec laquelle les valeurs du critère pouvaient être représentées graphiquement**. Par exemple, il est facile de représenter graphiquement la couleur de la fleur ou du fruit, en revanche il est plus difficile de représenter la présence d'épines, car les épines ne sont pas toujours situées aux mêmes endroits sur les plantes (elles peuvent être placées sur les tiges, le tronc, les feuilles ou les fruits).

— **La capacité du critère à distinguer les différentes plantes**, par rapport aux autres critères. Par exemple, si toutes les plantes ont des feuilles vertes, la couleur des feuilles est un mauvais critère pour

Poids	Critère	Valeurs possibles	Sélectionné
2,845	Forme du fruit	(générique) allongée, elliptique, graine ailée, en haricot vert, poire, rond, triangulaire (spécifique) banane, ananas, tubercule,...	X
2,799	Couleur du fruit	vert, orange, rouge,...	X
2,215	Couleur de la fleur	blanc, jaune, rouge,...	X
2,197	Forme des feuilles	(générique) lancéolée, oblongue, falciforme, aiguillon, ovale, palmatilobée, palmatiséquée, palmatipartite (spécifique) feuille de noix de coco, feuille de fougère,...	X
1,182	Taille du fruit	petit, moyen, grand	-
1,770	Taille du tronc	petit, moyen, grand	-
1,717	Silhouette de la plante	arbre, arbuste, palme, liane, herbe	X
1,624	Pétiole	absent, court, long	-
1,529	Pennation	paripennée, imparipennée, bipennée, tripennée, absence de pennation régulière	X

TABLEAU 3.2 – Extrait de la liste des critères triés par ordre de poids calculés par l'algorithme *Ranker*, avec les 6 critères sélectionnés.

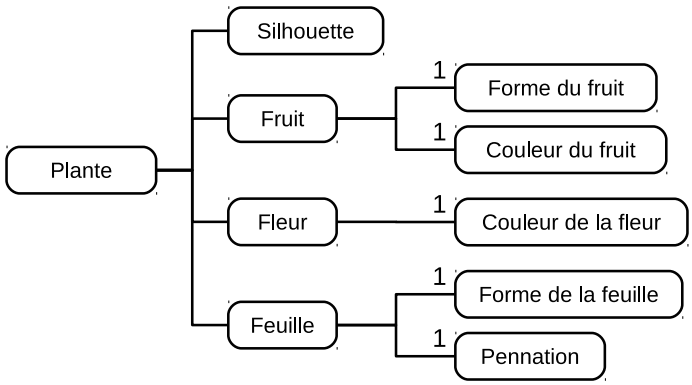


FIGURE 3.9 – Extrait de l'ontologie de la médecine traditionnelle d'Afrique de l'Ouest, concernant la description des plantes.

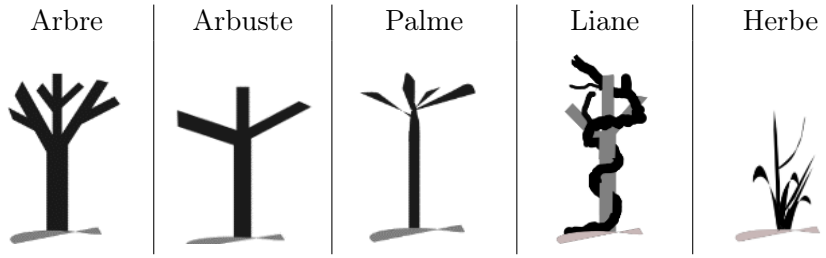


FIGURE 3.10 – Icônes pour les 5 silhouettes de plantes.

identifier les plantes. Pour cela, nous avons utilisé l'algorithme *Ranker* de la plate-forme Weka [51] d'apprentissage automatique et d'analyse de données. Cet algorithme permet, à partir d'un ensemble d'éléments définis par une classe et une valeur pour chaque critère, de donner un poids à chacun des critères. Plus le poids est élevé, plus l'attribut permet la distinction entre les différentes classes. Dans notre cas, nous avons utilisé l'espèce des plantes comme classe (chaque plante appartenait donc à une classe différente).

La sélection des critères s'est faite de manière itérative, en réduisant progressivement le nombre de critères. À chaque étape, nous avons utilisé l'algorithme *JRip* de Weka. Il s'agit d'un algorithme de classification automatique qui produit une liste de règles de classification. Nous l'avons utilisé pour vérifier que les critères sélectionnés permettaient bien de distinguer nos 22 plantes. Pour cela, nous avons construit les règles à partir de la totalité du jeu de données, et nous l'avons validé sur le jeu de donnée lui-même, en vérifiant que 100% des plantes étaient bien classées. Nous avons aussi modifié deux paramètres de l'algorithme : nous avons choisi un nombre minimum d'éléments par règle de 1 (au lieu de 2 par défaut), et nous avons désactivé l'élagage (*pruning*) afin de garder la totalité des règles.

Le Tableau 3.2 montre les critères classés en premier par *Ranker*, et ceux que nous avons sélectionnés. La Figure 3.9 montre la représentation des plantes dans l'ontologie.

3.2.1.4 Représentation graphique des valeurs des critères

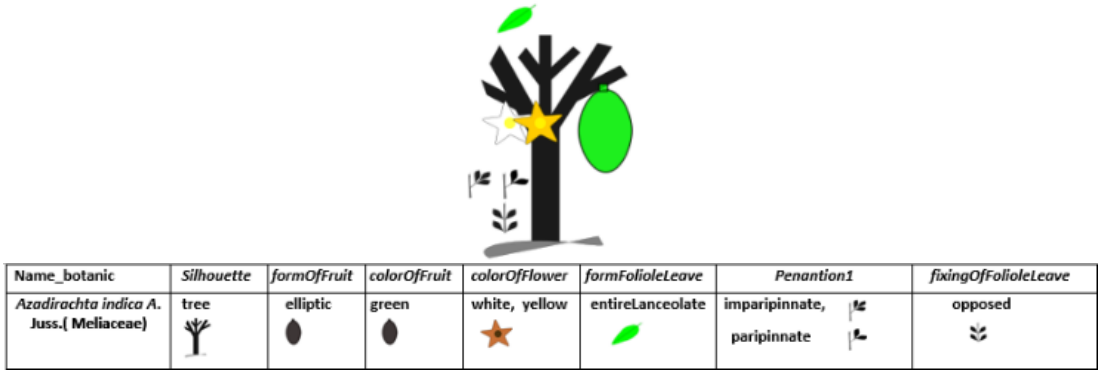
Nous avons représenté graphiquement chaque valeur possible pour chaque critère. Deux variables rétiniennes ont été utilisées : la couleur, pour la couleur des fruits et des fleurs, et la forme, pour les autres critères. Nous avons fait le choix de représenter la silhouette globale de la plante par une icône et les autres critères par des iconèmes placés à côté de la silhouette, à la manière des planches « éclatées » que l'on utilise en botanique. La Figure 3.10 montre les icônes de silhouettes. L'ensemble des icônes et des iconèmes ont été dessinés par Appoh dans le logiciel Inkscape au format vectoriel SVG.

3.2.1.5 Assemblage des icônes

Enfin, nous avons assemblé les icônes de silhouette, les iconèmes de forme de fruits, de forme de feuilles et de pennation, et les couleurs de fruits et de fleurs pour constituer des icônes. Les premiers assemblages ont été réalisés dans le logiciel de dessin Inkscape à la main. Par la suite, nous avons développé un programme Python pour générer automatiquement les icônes des 22 plantes à partir de leurs descriptions dans l'ontologie de la médecine traditionnelle. Pour cela, nous avons réutilisé le module Owlready pour la programmation orientée ontologie (voir section 2.3). Owlready permet de charger l'ontologie et d'accéder très facilement aux individus Plantes et à leurs attributs. Les fichiers SVG étant des fichiers XML, il est facile de les manipuler comme des fichiers texte. Le programme Python ouvre ensuite les fichiers SVG des icônes et des iconèmes requis, modifie les couleurs des iconèmes fruits et fleurs, assemble le tout en une seule image SVG et l'enregistre. La Figure 3.11 montre un exemple d'icône de plante produite.

3.2.2 Représentation graphique des recettes et des remèdes

Nous avons ensuite cherché à représenter les recettes sous la forme de schéma à l'aide d'un langage diagrammatique (voir section 3.1.3) intégrant les icônes.



Name_botanic	Silhouette	formOfFruit	colorOfFruit	colorOfFlower	formFolioleLeave	Penantion1	fixingOfFolioleLeave
Azadirachta indica A. Juss.(Meliaceae)	tree	elliptic	green	white, yellow	entireLanceolate	imparipinnate, paripinnate	opposed

FIGURE 3.11 – Exemple d'icône de plante générée automatiquement à partir de l'ontologie, pour la plante Azadirachta indica, ainsi que la liste des iconèmes utilisés pour construire cette icône.



FIGURE 3.12 – Icônes pour représenter les parties des plantes utilisées comme ingrédient dans la recette. De gauche à droite : feuille, fleur, fruit, tige ou tronc, écorce de tige ou tronc, écorce de racine, racine, sève, plante entière.

3.2.2.1 Icônes de parties de plantes, de modes de préparation et de modes d'administration

Dans notre langage diagrammatique, les plantes seront représentées avec les icônes de plantes créées à la section précédente. Il nous reste à décrire les parties de plantes utilisées (par exemple, feuille, fruit ou racine), les modes de préparation des recettes (par exemple décoction ou infusion) et les modes d'administration des remèdes (par exemple boisson ou bain). Le nombre de valeurs possibles pour chacun de ces trois attributs étant faible (moins de 10), nous avons simplement dessiné une icône pour chacun sans chercher à construire un langage iconique. La Figure 3.12 montre les icônes réalisées pour les parties de plantes ; les autres sont encore en cours d'élaboration.

3.2.2.2 Diagrammes

Notre langage diagrammatique distingue donc quatre catégories d'icônes qui correspondront à quatre niveaux :

1. les plantes,
2. les parties des plantes utilisées,
3. les modes de préparation,
4. les modes d'administration.

Les icônes de ces quatre niveaux sont ensuite reliées par deux types de flèches. Le premier type de flèche part d'une plante et conduit à un mode de préparation, au milieu de la flèche figure la partie de plante. Le second type de flèche part d'un mode de préparation et conduit à un mode d'administration. L'ensemble du diagramme correspond donc à un graphe acyclique. La Figure 3.13 montre un premier résultat de représentation de recette et remède.

3.2.3 Discussion

Dans cette section nous avons présenté d'une part la construction d'un langage iconique pour représenter des plantes médicinales à partir de critères botaniques, et d'autre part un langage diagrammatique pour représenter des recettes de remèdes à base de plantes, dans le contexte de la médecine traditionnelle de l'Afrique de l'Ouest. L'approche iconique et graphique est particulièrement intéressante dans ce contexte, car

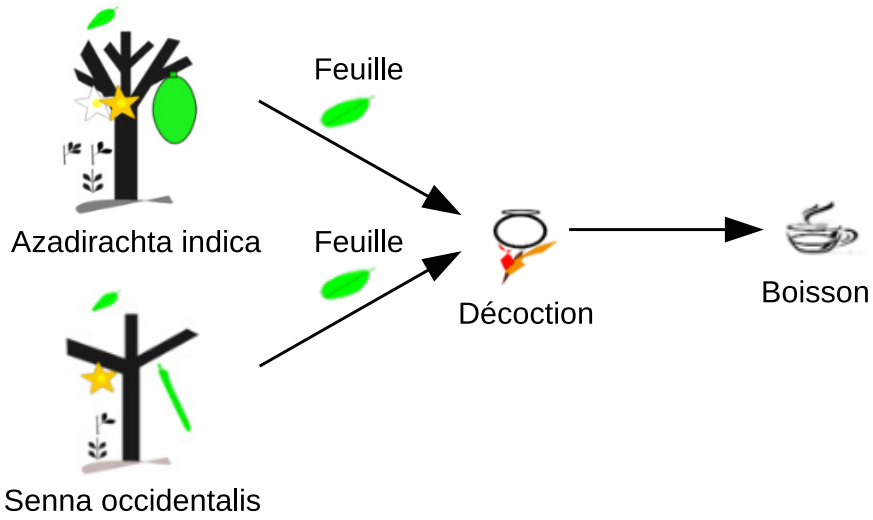


FIGURE 3.13 – Exemple de remède et de recette en langage diagrammatique. La recette utilise les feuilles de deux plantes que l'on mélange ensemble pour en faire une décoction. Le produit résultant sera bu. Les étiquettes textuelles ont été ajoutées pour faciliter la compréhension mais ne seront pas nécessairement conservées dans les diagrammes finaux.

la majorité des praticiens de la médecine traditionnelle africaine sont illettrés. Le travail est toujours en cours et se poursuit, notamment sur la partie langage diagrammatique pour les recettes et les remèdes.

Lors de la sélection des critères, nous avons vérifié que les critères sélectionnés permettaient bien de distinguer nos 22 plantes. En revanche, cela ne garantit pas l'absence de confusion avec d'autres plantes, c'est-à-dire l'existence d'une plante absente de notre ontologie qui aurait les mêmes valeurs pour nos critères qu'une des 22 plantes. Dans ce cas, une confusion est possible entre les deux plantes, qui pourraient conduire un praticien à utiliser une mauvaise plante à la place de la plante indiquée. Pour corriger ce problème, il faudrait idéalement inclure toutes les plantes existantes en Afrique de l'Ouest dans notre ontologie, et revoir la sélection des critères en fonction. Pour des raisons pratiques, le nombre de plantes étant très élevé, cela n'est pas envisageable. Nous pourrions cependant inclure quelques plantes parmi les plus communes pour éviter les confusions les plus probables.

Les icônes et les iconèmes ont été dessinés par Appoh. Les dessins pourraient être améliorés en faisant appel à un graphiste professionnel pour arriver à un rendu plus esthétique. Cependant, cela ne remet pas en cause la signification des pictogrammes ni les choix de représentation qui ont été faits.

La démarche suivie pour la conception du langage iconique s'est inspirée de celle qui avait été mise au point pour VCM [118, 105]. En particulier, nous avons utilisé pour les formes des fruits et des feuilles à la fois des formes génériques (rond, allongé,...) et des formes spécifiques (banane, ananas,...). Dans VCM, j'avais utilisé une technique similaire pour représenter les maladies, en distinguant des maladies génériques qui peuvent d'appliquer à plusieurs organes (infections, inflammations,...) et des maladies spécifiques à un organe (vomissement pour l'estomac par exemple).

Les icônes et les recettes iconiques n'ont pas encore été testées auprès des praticiens. Il serait cependant très intéressant d'évaluer cette approche, et de la comparer à d'autres (approche textuelle par exemple).

Les perspectives concernent l'utilisation des recettes iconiques au sein d'un wiki sémantique pour permettre aux praticiens de rédiger et d'échanger leurs recettes sans avoir besoin de savoir lire ni écrire. Plus simplement, nous pouvons aussi envisager un livre de recettes iconiques au format papier ou électronique.

Encadrement associé à cette contribution :

1. Soufiane Cafile, stage de M2 Informatique BioMédicale (6 mois en 2009)

Publications associées à cette contribution :

1. **Lamy JB**, Soualmia LF, Kerdelhué G, Venot A, Duclos C. Validating the semantics of a medical iconic language using ontological reasoning. J Biomed Inform 2013 ;46(1) :56-67
2. **Lamy JB**, Tsopra R, Venot A, Duclos C. A Semi-automatic Semantic Method for Mapping SNOMED CT Concepts to VCM Icons. Stud Health Technol Inform 2013 ;192 :42-6. Conférence MEDINFO 2013, Copenhague, Danemark
3. Vaillant P, **Lamy JB**. Using graph transformation algorithms to generate natural language equivalents of icons expressing medical concepts. Lecture Notes in Computer Sciences 2014 ;LNAI 8655 :353–362. Conférence Text, Speech and Dialogue, Brno, République Tchèque
4. Griffon N, Kerdelhué G, Soualmia LF, Merabti T, Grosjean J, **Lamy JB**, Venot A, Duclos C, Darmoni SJ. Evaluating alignment quality between iconic language and reference terminologies using similarity metrics. BMC medical informatics and decision making 2014 ;14 :17
5. **Lamy JB**, Soualmia LF, Venot A, Duclos C. Validation de la sémantique d'un langage iconique médical à l'aide d'une ontologie : méthodes et applications. Actes du congrès d'Ingénierie des Connaissances (IC) 2014, Clermont-Ferrand
6. **Lamy JB**, Soualmia LF, Duclos C, Venot A. Formalisation de la sémantique des langages iconiques : méthode à base d'ontologie et applications. Revue d'intelligence artificielle (RIA) 2016 ;30(5) :579-606
7. **Lamy JB**, Soualmia LF. Formalization of the semantics of iconic languages : an ontology-based method and four semantic-powered applications. **Knowledge-Based System** 2017 ;135 :159-179

3.3 Contribution : Formalisation des langages iconiques

Lors de ces travaux, initiés en tant que postdoc puis poursuivis en tant que maître de conférences au cours des projets L3IM et SiFaDo, j'ai cherché à mettre au point une méthode pour formaliser la sémantique d'un langage iconique tel que le langage VCM développé lors de ma thèse. En effet, VCM a été conçu il y a plus de 10 ans et a été intégré dans une demi-douzaine d'applications, certaines étant utilisées en production. La sémantique du langage était relativement informelle au début. Cependant, au fil du temps, le besoin d'une sémantique claire s'est fait de plus en plus sentir, alors que le langage se complexifiait et que les applications se multipliaient : des utilisateurs créaient des icônes incohérentes, des experts utilisaient un même pictogramme avec plusieurs sens légèrement différents, des développeurs réclamaient des correspondances entre VCM et les ressources terminologiques qu'ils utilisaient.

Ce travail a été réalisé avec la collaboration de Lina Soualmia, maître de conférence au LIM&BIO puis au LITIS à Rouen, pour les aspects de formalisation.

3.3.1 Méthode proposée pour la formalisation d'un langage iconique

Notre méthode pour formaliser la sémantique d'un langage iconique cherche à séparer les aspects syntaxiques et les aspects sémantiques, afin d'éviter toute confusion entre les objets graphiques (les icônes et leurs composants : pictogrammes, couleurs, *etc.*, par exemple le pictogramme « ♡ ») et les objets du domaine qu'ils représentent (par exemple l'organe cœur). Pour cela, nous avons fait le choix de représenter dans deux ontologies séparées la syntaxe du langage (*ontologie des icônes*) et les objets du domaine représentés (*ontologie du domaine*), puis de les relier entre elles dans une troisième ontologie (*ontologie de correspondance*). La Figure 3.14 montre la structure générale des trois ontologies.

La méthode comporte donc les étapes suivantes : (1) construire *l'ontologie des icônes* décrivant la syntaxe des icônes du langage, (2) construire *l'ontologie du domaine* (ou réutiliser une ontologie existante) décrivant les objets représentés par le langage iconique, (3) établir des liens entre les concepts des deux ontologies à l'aide des relations « icône - représente - objet du domaine » et « objet du domaine - est représenté par - icône » ; ces relations constituent *l'ontologie de correspondance*.

Il est possible d'utiliser des relations filles de « représente / est représenté par » pour distinguer les différents types de représentation possibles. La plupart des auteurs en distinguent trois types : (a) les « icônes

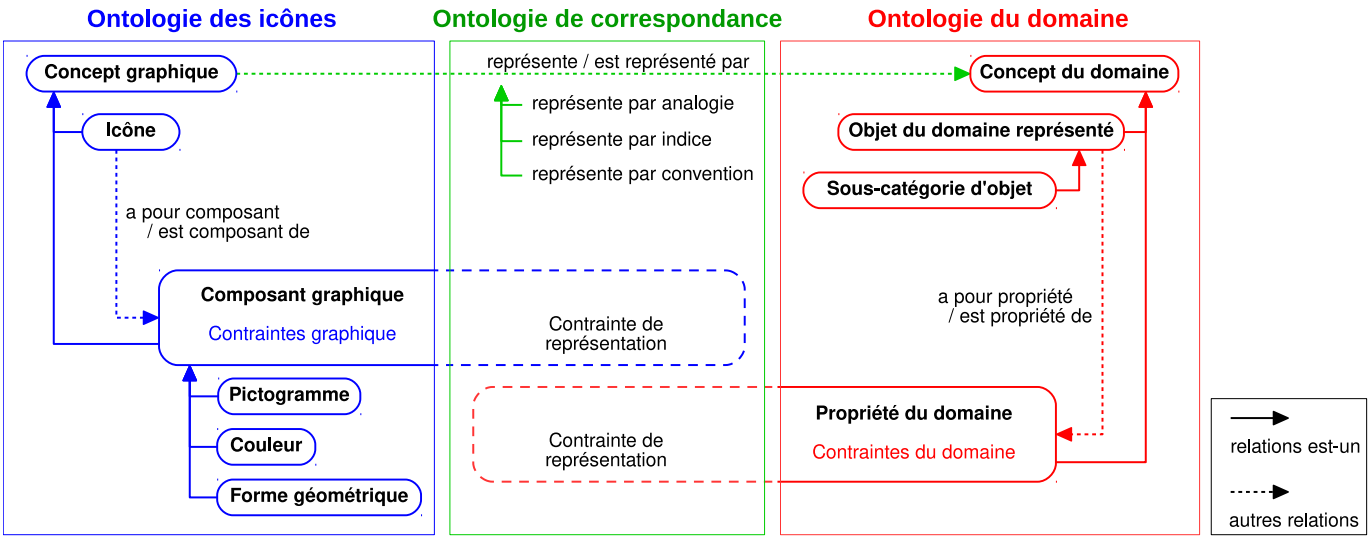


FIGURE 3.14 – Structure générale de l'ontologie des icônes (en bleu), de l'ontologie du domaine (en rouge) et de l'ontologie de correspondance (en vert).

vraies » qui s'appuie sur l'analogie et ressemblent à l'objet représenté (par exemple le pictogramme «  » ressemble à la forme en haricot caractéristique d'un rein), (b) les indices qui sont des choses associées à l'objet (par exemple une disquette pour l'action d'enregistrer) et (c) les symboles qui sont des associations arbitraires (convention, par exemple « H » pour « hôpital »).

Des contraintes peuvent être formulées à trois niveaux :

1. ***des contraintes graphiques*** dans l'ontologie des icônes (par exemple « un panneau routier contient au plus un pictogramme »),
2. ***des contraintes du domaine*** dans l'ontologie du domaine (par exemple « les travaux peuvent être associés à une consigne de signalisation routière annonçant un danger, mais pas à des consignes d'ordre ou d'interdiction », c'est-à-dire donnant l'ordre de faire des travaux ou interdisant d'en faire, car les automobilistes ne font pas de travaux),
3. ***des contraintes de représentations*** dans l'ontologie de correspondance (par exemple « le pictogramme 'travaux' est présent uniquement sur des panneaux représentant des consignes associées aux travaux » et « les travaux sont uniquement associés à des consignes représentées par des panneaux incluant le pictogramme 'travaux' »).

Les relations et contraintes de représentation permettent ainsi de « propager » les contraintes de chacune des deux premières ontologies vers l'autre : en effet, les contraintes graphiques peuvent avoir un impact sur ce que peut représenter le langage et limiter sa sémantique, de même que les contraintes du domaine peuvent avoir des implications graphiques et rendre certaines icônes inconsistantes.

Il est naturel d'exprimer les correspondances entre Composants graphiques et Concepts du domaine sous forme de triplets, par exemple (Composant graphique A, représente, Concept du domaine A). Cependant, les contraintes logiques de l'ontologie de correspondance sont plus complexes que cela. D'une part, une contrainte est requise pour chaque Composant graphique et chaque Concept du domaine (donc l'exemple de triplet précédent correspond en fait à deux contraintes). D'autre part, si un Composant graphique ou un Concept du domaine intervient dans plusieurs triplets, il faut les combiner ensemble. De plus, les relations d'héritage doivent être prises en compte. Écrire manuellement les contraintes de représentation serait donc complexe et fastidieux.

Par conséquent, nous proposons de gérer ces correspondances en utilisant trois formats de fichier :

1. Un fichier texte contenant des triplets, pour pouvoir écrire facilement les correspondances à la main,
2. Un fichier OWL (l'ontologie de correspondance), pour effectuer le raisonnement automatique,
3. Une base de données SQL, pour effectuer des recherches rapidement.

	Triplets	Contraintes
1	(g_A, représente, d_A)	$g_A \sqsubseteq (\forall est_composant_de.(\forall représente.(\exists a_pour_propriété.d_A)))$ $d_A \sqsubseteq (\forall est_propriété_de.(\forall est_représenté_par.(\exists a_pour_composant.g_A)))$
2	(g_A, représente, d_A) (g_A, représente, d_B)	$g_A \sqsubseteq (\forall est_composant_de.(\forall représente.(\exists a_pour_propriété.(d_A \sqcup d_B))))$ $d_A \sqsubseteq (\forall est_propriété_de.(\forall est_représenté_par.(\exists a_pour_composant.g_A)))$ $d_B \sqsubseteq (\forall est_propriété_de.(\forall est_représenté_par.(\exists a_pour_composant.g_A)))$
3	(g_A, représente, d_A+d_B)	$g_A \sqsubseteq (\forall est_composant_de.(\forall représente.(\exists a_pour_propriété.(d_A \sqcap d_B))))$ $d_A \sqsubseteq (\forall est_propriété_de.(\forall est_représenté_par.(\exists a_pour_composant.g_A)))$ $d_B \sqsubseteq (\forall est_propriété_de.(\forall est_représenté_par.(\exists a_pour_composant.g_A)))$

TABLEAU 3.3 – Exemples de triplets de correspondance et les contraintes qu'ils génèrent. g signifie "Composant graphique" et d "Concept du domaine" et, dans les triplets, le + est utilisé pour indiquer l'intersection.

Les deux derniers fichiers peuvent être produits à partir du premier. Les triplets peuvent être stockés dans une table SQL, et nous avons utilisé un programme Python pour générer l'ontologie de correspondance à partir des triplets. Le tableau 3.3 montre des exemples de triplets et les contraintes auxquelles ils conduisent. L'exemple 1 est une correspondance simple (1-1) entre un Composant graphique et un Concept du domaine. L'exemple 2 est un cas de polysémie : un Composant graphique associé à plusieurs Concepts du domaine. L'exemple 3 est un cas d'intersection : un Composant graphique représentant l'intersection de plusieurs Concepts du domaine. Ce cas se rencontre lorsqu'un pictogramme est dessiné spécifiquement pour une combinaison de Concepts du domaine. Mentionnons pour exemple les maladies obstructives dans VCM : elles sont représentées en ajoutant un « bouchon » sur le pictogramme de l'organe obstrué, cependant la position du bouchon dépend de la forme de l'organe. Il n'est donc pas possible de suivre une approche combinatoire (pictogramme de l'organe + pictogramme du bouchon), au contraire un pictogramme spécifique doit être dessiné pour chaque organe. Chacun de ces pictogrammes (par exemple « vaisseau sanguin obstrué ») représente l'intersection des concepts Obstruction et de l'organe concerné.

Notons que toutes les contraintes générées se focalisent sur la relation représente/est-représenté-par entre l'icône et l'objet du domaine représenté. Aucune contrainte n'est générée directement entre Composant graphique et Concept du domaine, par exemple la première contrainte dans le tableau 3.3 est $g_A \sqsubseteq (\forall est_composant_de.(\forall représente.(\exists a_pour_propriété.d_A)))$ et non $g_A \sqsubseteq (\forall représente.d_A)$. Des contraintes directes seraient plus simples, cependant elles ne permettent pas le raisonnement automatique avec les raisonneurs OWL. En effet, il faudrait asserter le fait que, pour toute icône I, I représente $O \Rightarrow \forall g \in I, \exists d \in O \mid g$ représente d. Or cette assertion nécessite plusieurs variables libres alors que les logiques de description n'ont pas de variables et correspondent à des formules à une seule variable libre [13].

Enfin, l'héritage est aussi pris en compte : si g_A et $g_{A'}$ sont deux Composants graphiques avec $g_{A'} \sqsubseteq g_A$, nous considérons que le triplet ($g_{A'}$, représente, d_A) implique automatiquement le triplet (g_A, représente, d_A) (en effet, sans cela, les contraintes produites seraient inconsistantes).

D'un point de vue formel, nous pouvons décrire une icône I par un ensemble de Composants graphiques, tels que des pictogrammes, des formes géométriques, ou des couleurs, et le sens M de l'icône par un ensemble de Concepts du domaine (Algorithme 3.1). Au contraire, n'importe quel ensemble de Composants graphique ou de Concepts du domaine ne constitue pas une icône ou un sens (respectivement) valide, à moins de respecter les contraintes définies précédemment. Lorsqu'un pictogramme peut être présent à différentes positions sur l'icône, chaque paire (pictogramme, position) valide doit être considérée comme un Composant graphique distinct.

Nous pouvons aussi définir la fonction get_meaning(I), qui retourne le sens d'une icône donnée. La fonction utilise tout d'abord les relations *représente* / *est-représenté-par* pour calculer, pour chaque Composant graphique, l'ensemble des Concepts du domaine qu'il peut représenter. Chaque Composant graphique représente un Concept du domaine, ou plusieurs en cas de polysémie. D est le tuple contenant tous les ensembles obtenus. Ensuite nous calculons P, le produit cartésien de tous les ensembles dans D. Les éléments de P sont les *interprétations candidates* possibles pour l'icône I, c'est-à-dire un choix possible parmi toutes les

Algorithme 3.1 Définition d'une icône et de son sens.

Notons :

- $\mathcal{O}_{icone}$ l'ontologie des icônes
- $\mathcal{O}_{domaine}$ l'ontologie du domaine
- $\mathcal{O}_{corresp}$ l'ontologie de correspondance
- $T_{corresp}$ l'ensemble des triplets de correspondance,
$$T_{corresp} = \{(g, \text{représente}, d)\} \text{ où } g \sqsubseteq Composant_graphique \in \mathcal{O}_{icone}$$
$$\text{et } d \sqsubseteq Concept_du_domaine \in \mathcal{O}_{domaine}$$
- I une icône, $I \subseteq \{g \sqsubseteq Composant_graphique \in \mathcal{O}_{icone}\}$
- M le sens d'une icône, $M \subseteq \{d \sqsubseteq Concept_du_domaine \in \mathcal{O}_{domaine}\}$

fonction **get_meaning**(I) :

$D = \{\{d \mid d \sqsubseteq Concept_du_domaine \in \mathcal{O}_{domaine} \text{ et } (g, \text{représente}, d) \in T_{corresp}\} \mid g \in I\}$

$P = \{\{d_1, ..., d_{|I|}\}, \forall (d_1, ..., d_{|I|}) \in \prod D\}$ (P est le produit Cartésien non-ordonné des ensembles de D)

$P' = \{\{d_1, ..., d_{|I|}\} \in P \mid d_1 \sqcap ... \sqcap d_{|I|} \text{ est satisfiable dans } \mathcal{O}_{domaine} \cup \mathcal{O}_{corresp} \cup \mathcal{O}_{icone}\}$

$M = \bigcup P'$

retourne M

polysémies. Ensuite, P est filtré pour ne conserver que les ensembles de Concepts du domaine qui sont sont consistants (en prenant en compte les trois types de contraintes définies précédemment), ce qui produit P'. Enfin, l'union des ensembles de P' conduit à M, le sens de l'icône, c'est-à-dire l'ensemble des Concepts du domaine signifiés par l'icône.

La formalisation que nous proposons ici permet non seulement de décrire la syntaxe et la sémantique des langages iconiques, mais rend également possibles plusieurs applications : déterminer automatiquement les icônes inconsistantes, générer des correspondances avec des concepts issus d'autres ressources termino-ontologiques, automatiser la production d'un lexique du langage iconique et des libellés des icônes. Dans la section suivante, nous appliquons cette formalisation à un exemple simple : les panneaux de la signalisation routière. Dans la section d'après, nous l'appliquons au langage VCM et nous donnerons un exemple d'application possible.

3.3.2 Applications à la signalisation routière

Les panneaux routiers peuvent facilement être décrits à l'aide d'une ontologie (ontologie des icônes) ainsi que les consignes associées (ontologie du domaine), ce qui conduit à l'ontologie de la Figure 3.15. Dans l'ontologie du domaine, nous distinguons deux grandes catégories de consignes : les ordres et les informations. Les ordres se subdivisent en obligations et interdictions, et les informations en informations de danger et autres informations. Chaque consigne est associée à une temporalité (permanente ou temporaire) et un objet, qui peut être (a) une *action de l'automobiliste* (par exemple tourner à gauche, rouler à 50 km/h, dépasser un autre véhicule, conduire un tracteur, *etc.*) ou (b) une *rencontre* portant sur un élément extérieur à l'automobiliste (par exemple un virage à gauche, un hôpital, des animaux sauvages, *etc.*). De manière évidente, les ordres ne peuvent porter que sur des actions de l'automobiliste (on ne peut pas ordonner à l'automobiliste de rencontrer quelque chose) et les informations ne peuvent porter que sur des rencontres (la signalisation routière n'est pas là pour informer l'automobiliste de ses propres actions). Nous traduisons cela par deux contraintes sémantiques dans l'ontologie du domaine.

La fonction get_meaning() définie précédemment (Algorithme 3.1) permet d'obtenir le sens d'un panneau routier. Par exemple, le panneau « interdiction de tourner à gauche » peut être formalisé ainsi :

I = { Cercle, Couleur de bord rouge, Couleur de fond blanc, Pictogramme flèche à gauche }

Ensuite, la fonction crée D, le tuple des ensembles de Concepts du domaine représentés par chacun des Composants graphiques dans I. Les relations *est-un* doivent être prises en compte, par exemple le composant Cercle peut représenté le Concept du domaine Ordre, mais aussi ces deux descendants, Interdiction et

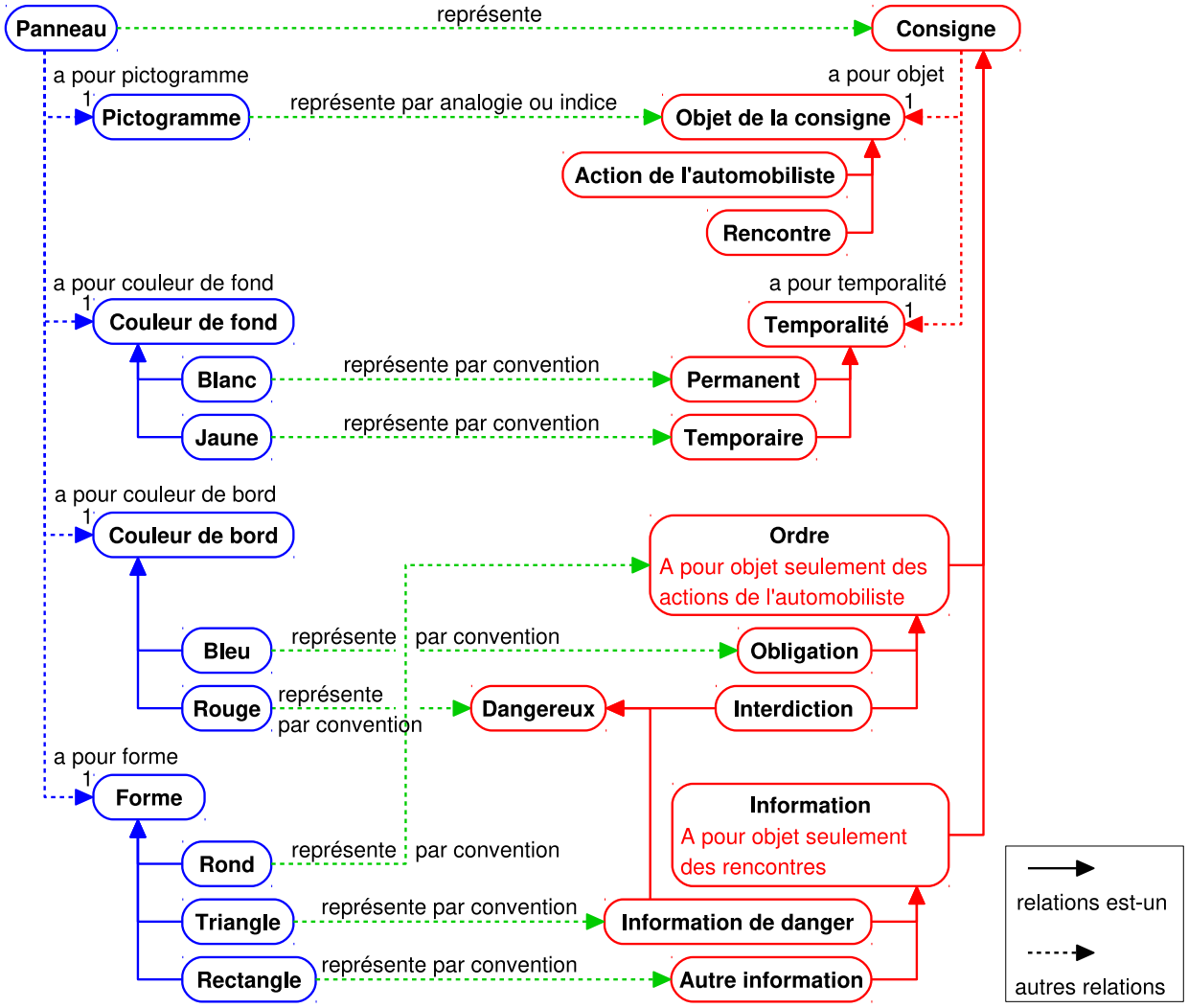


FIGURE 3.15 – Ontologie des panneaux routiers (en bleu) et des consignes associées (en rouge).

Obligation.

D = ({ Ordre, Interdiction, Obligation },
{ Danger, Interdiction, Information de danger },
{ Permanent },
{ Rencontre virage à gauche, Action tourner à gauche })

P est l'ensemble des interprétations candidates de I, et correspond au produit cartésien des ensembles de D.

P = { { Interdiction, Action tourner à gauche, Permanent },
{ Ordre, Danger, Action tourner à gauche, Permanent },
{ Ordre, Danger, Rencontre virage à gauche, Permanent },
{ Obligation, Danger, Action tourner à gauche, Permanent },
... (18 éléments en tout) }

P' est le sous-ensemble de P comprenant les interprétations consistantes. Dans notre exemple, le troisième ensemble est inconsistant car une rencontre ne peut pas être associé à un ordre dans l'ontologie du domaine, et le dernier ensemble est inconsistant car une obligation ne peut pas être associé à un danger.

P' = { { Interdiction, Action tourner à gauche, Permanent },
{ Ordre, Danger, Action tourner à gauche, Permanent },
... }

Enfin, le sens M est l'union des ensembles dans P'.

M = { Interdiction, Ordre, Danger, Action tourner à gauche, Permanent }

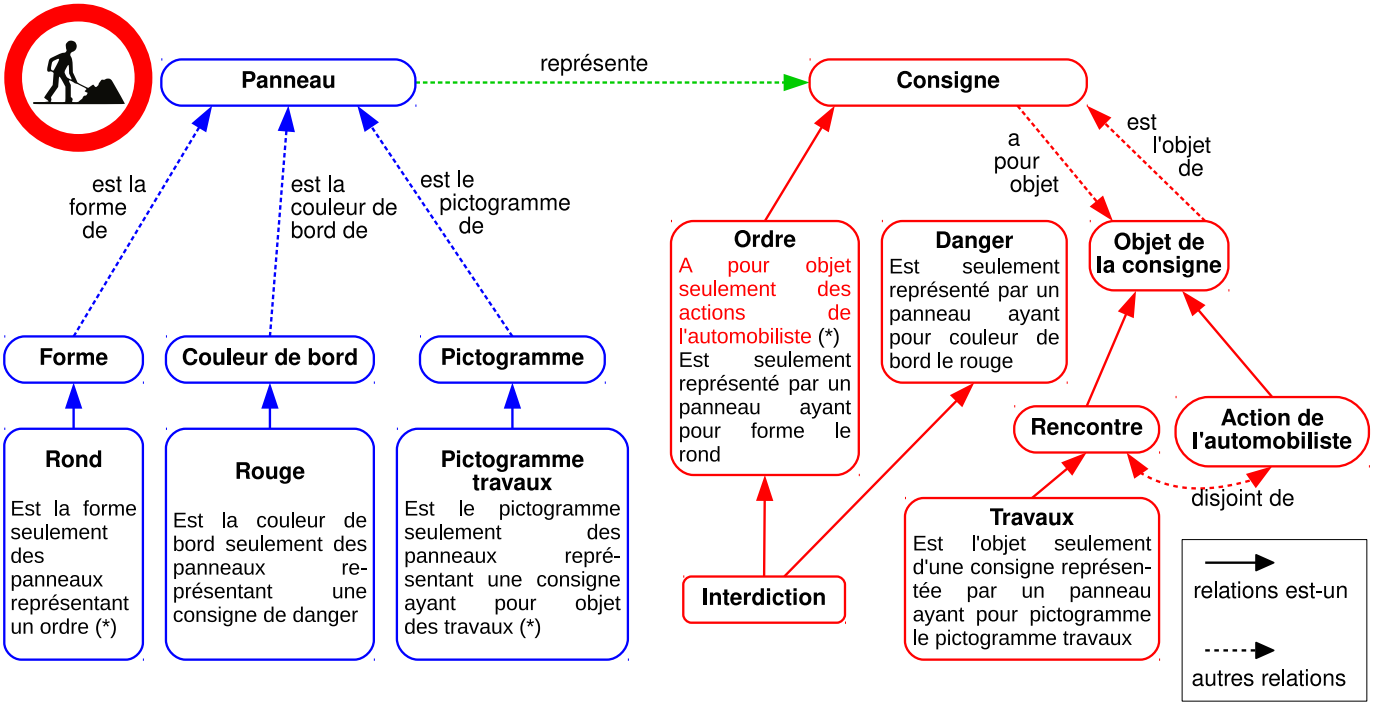


FIGURE 3.16 – Représentation du panneau inconsistant « interdiction de travaux » dans l'ontologie et la sémantique associée. Les trois contraintes marquées d'une étoile (*) sont celles qui permettent d'inférer l'inconsistance sémantique de ce panneau. Deux contraintes de représentation permettent de déduire que ce panneau représente un ordre ayant pour objet les travaux, or une contrainte du domaine indique que les ordres ne peuvent avoir pour objet que des actions de l'automobiliste et non des rencontres, alors que les travaux sont une rencontre.

$$
\begin{aligned}
\text{Interdiction_de_travaux} \equiv\ & \text{Icône} \\
\sqcap\ & \forall\, a_pour_forme.\text{Cercle} \\
\sqcap\ & \forall\, a_pour_couleur_de_bord.\text{Rouge} \\
\sqcap\ & \forall\, a_pour_pictogramme.\text{Pictogramme_travaux} \\
\text{Cercle} \sqsubseteq\ & \text{Forme} \\
\sqcap\ & \forall\, est_la_forme_de.(\text{Icône} \sqcap \forall\, représente.\text{Ordre}) \\
\text{Rouge} \sqsubseteq\ & \text{Couleur} \\
\sqcap\ & \forall\, est_la_couleur_de_bord_de.(\text{Icône} \sqcap \forall\, représente.\text{Danger}) \\
\text{Pictogramme_travaux} \sqsubseteq\ & \text{Pictogramme} \\
\sqcap\ & \forall\, est_le_pictogramme_de.(\text{Icône} \sqcap \forall\, représente.(\text{Consigne} \sqcap \exists\, a_pour_objet.\text{Travaux})) \\
\\
\text{Ordre} \sqsubseteq\ & \text{Consigne} \\
\sqcap\ & \forall\, a_pour_objet.\text{Action_de_l'automobiliste} \\
\sqcap\ & \forall\, est_représenté_par.(\text{Icône} \sqcap \exists\, a_pour_forme.\text{Cercle}) \\
\text{Danger} \sqsubseteq\ & \text{Consigne} \\
\sqcap\ & \forall\, est_représenté_par.(\text{Icône} \sqcap \exists\, a_pour_couleur_de_bord.\text{Rouge})) \\
\text{Interdiction} \sqsubseteq\ & \text{Ordre} \\
\sqcap\ & \text{Danger} \\
\text{Travaux} \sqsubseteq\ & \text{Rencontre} \\
\sqcap\ & \forall\, est_l'objet_de.(\text{Consigne} \sqcap \forall\, est_représenté_par. \\
& \qquad (\text{Icône} \sqcap \exists\, a_pour_pictogramme.\text{Travaux})) \\
\text{Rencontre} \sqcap & \\
\text{Action_de_l'automobiliste} \sqsubseteq\ & \bot
\end{aligned}
$$

FIGURE 3.17 – Extrait de la TBox correspondant au panneau « interdiction de travaux » de la Figure 3.16.

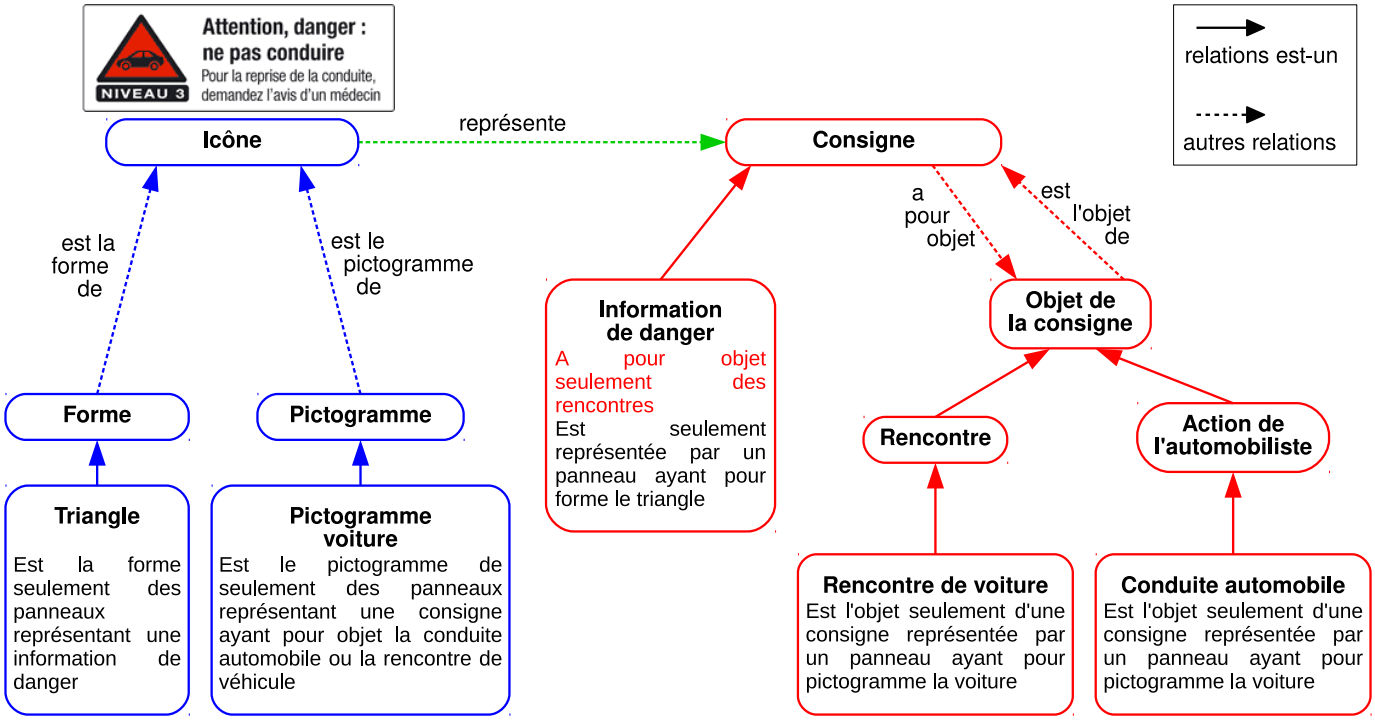


FIGURE 3.18 – Icône « attention danger : ne pas conduire » qui figure sur certains médicaments, et sa représentation dans l'ontologie des panneaux routiers. Cette représentation conduit à un sens différent pour cette icône : « attention : risque de rencontrer des voitures », car la forme triangle est normalement associée à une information de danger et à une rencontre, mais pas à une interdiction.

L'ontologie présentée permet de formaliser la sémantique des panneaux routiers. En associant les différents éléments constitutifs des panneaux routiers, il est possible de construire les panneaux existants ou bien d'en inventer de nouveaux. Une première application consiste à déterminer la consistance et la sémantique de ces panneaux. Lorsqu'on essaie de décrire un panneau inconsistant comme le panneau « interdiction de travaux » (Figure 3.16), les contraintes présentes dans l'ontologie permettent de déduire l'inconsistance (Figure 3.17). La déduction peut être automatisée à l'aide d'un raisonneur, comme nous le détaillons avec VCM (section 3.3.3).

Les panneaux routiers ont souvent été utilisés comme source d'inspiration pour la construction d'icônes hors du domaine de la signalisation routière. Un exemple est l'icône « attention danger : ne pas conduire » qui figure sur certains médicaments. Une seconde application de notre ontologie consiste à vérifier que ces langages iconiques dérivés respectent bien la sémantique originelle des panneaux routiers. Lorsque l'on décrit l'icône « attention danger : ne pas conduire » dans l'ontologie des panneaux routiers (Figure 3.18), le sens obtenu n'est pas celui attendu. En effet, dans la signalisation routière, la forme « triangle » doit être associée à une information de danger et à une rencontre. Le sens devient donc « attention : risque de rencontrer des voitures » !

Une troisième application consiste à améliorer la cohérence des panneaux routiers. Par exemple certains panneaux d'interdiction sont barrés et d'autres ne le sont pas. De même, certains panneaux partagent un même élément sémantique qui est représenté visuellement de manière similaire mais pas identique ; par exemple la flèche à gauche est différente sur les panneaux « interdiction de tourner à gauche » et « virage à gauche obligatoire » (voir deuxième et troisième panneaux en Figure 3.2).

3.3.3 Applications à VCM

3.3.3.1 L'ontologie VCM

L'ontologie VCM [110] a pour objectif d'aider à la validation de la sémantique du langage VCM. Nous présentons tout d'abord les principes généraux suivis lors de la construction de cette ontologie, puis nous la décrivons plus en détail. L'ontologie, ainsi que les alignements et les programmes que nous présentons ensuite, sont diffusés en logiciel libre et intégrés dans PyMedTermino [1] [117], un ensemble de modules permettant d'accéder aux terminologies médicales en langage Python.

Le premier principe général que nous avons appliqué lors de la construction de l'ontologie a été de distinguer d'un côté les icônes et les primitives graphiques de VCM (*ontologie des icônes*) et de l'autre les concepts médicaux correspondants (*ontologie du domaine*). En effet, le pictogramme « poumon » est distinct de l'organe « poumon ».

Le second principe a été d'employer au maximum les relations de subsomption « est-un » plutôt que d'autres types de relations comme les relations méréologiques « partie-de », puisque la plupart des outils d'édition et des moteurs d'inférences s'appuient sur la subsomption. En particulier, les objets anatomiques ont été désignés comme « structure + adjectif » plutôt que par leur nom d'organe ; nous dirons par exemple « une *structure stomacale* est une *structure digestive* » plutôt que « l'*estomac* est une partie du *tube digestif* ». Les organes peuvent ensuite être ajoutés comme fils (relation *est-un*) des structures anatomiques : le poumon *est une* structure pulmonaire. Bien que contraire au principe du « biais minimal d'encodage », on retrouve cette approche dans plusieurs terminologies médicales, dont la SNOMED CT (*Systematized Nomenclature of Medicine - Clinical Terms* [34]) et nous l'avons adoptée.

Le troisième principe a été que toute icône décrit un état du patient, y compris les icônes de traitement ou d'examen. Par exemple l'icône « antiasthmatique » a été modélisée comme « patient traité par un antiasthmatique ». Cela correspond à la manière dont VCM représente les traitements et les examens, en reprenant la pathologie qui est traitée ou le risque qui est surveillé.

Le quatrième principe a été d'utiliser uniquement des classes et non des individus. Dans l'ontologie du domaine, seules des connaissances générales sont représentées, et non des informations spécifiques à un patient : par exemple l'organe « poumon » est une classe que l'on pourrait instancier pour obtenir le poumon d'un patient donné (cependant cette instanciation n'est pas nécessaire pour l'expression de la sémantique du langage). La difficulté à employer des instances dans le domaine médical a été abordée en détail à la section 2.1.4.

Nous retrouvons les trois types de contraintes dans l'ontologie :

1. ***des contraintes graphiques*** dans l'ontologie des icônes, qui portent sur les primitives de VCM : par exemple les modificateurs de forme « tumeur » et « virus » occupent le même emplacement sur les icônes, et ne peuvent donc pas être conjointement associés,
2. ***des contraintes du domaine médical*** dans l'ontologie du domaine, qui portent sur les structures anatomiques, les morphologies... représentées par les primitives : par exemple une tumeur est une morphologie qui ne peut s'appliquer qu'à une structure anatomique, mais pas à une fonction biologique,
3. ***des contraintes de représentation*** dans l'ontologie de correspondance, qui relient les éléments de l'ontologie des icônes à ceux de l'ontologie du domaine (par exemple pour relier le pictogramme « poumon » à l'organe « poumon » et à la fonction « respiration »).

Lors de la construction de l'ontologie, un jeu d'une centaine d'icônes test a été mis au point de manière itérative, et un raisonneur a été utilisé pour tester la consistance de l'ontologie et vérifier les résultats obtenus sur les icônes test.

L'ontologie VCM globale (Figure 3.19) comprend trois modules. Le premier (ontologie des icônes, comprenant 240 concepts, 21 relations et 2 597 axiomes) décrit les primitives et les icônes VCM. Il a été généré automatiquement à partir de la liste des primitives du langage VCM et de scripts Python. Le fichier OWL obtenu a ensuite été importé puis édité manuellement dans Protégé pour y ajouter les règles de composition des icônes et les contraintes graphiques. Ces règles restreignent les composantes d'une icône (par exemple, au plus un pictogramme central) et prennent en compte les contraintes spatiales.

1. `https://pypi.python.org/pypi/PyMedTermino`

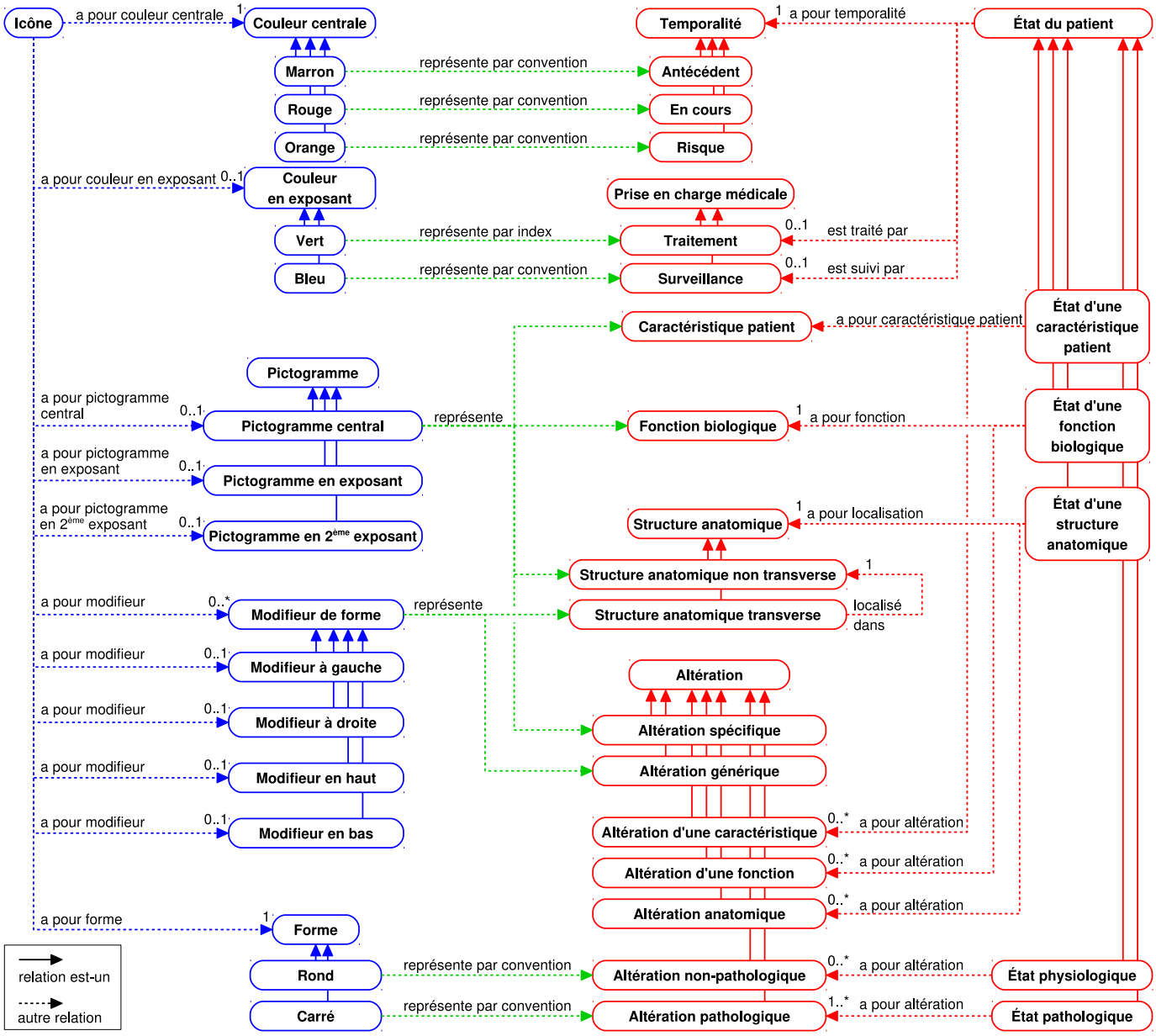


FIGURE 3.19 – Principaux concepts et relations dans l'ontologie VCM. Les couleurs correspondent aux trois ontologies.

Le second module (ontologie du domaine, 369 concepts, 18 relations et 828 axiomes) décrit les concepts médicaux représentés par les primitives du langage VCM : structures anatomiques, fonctions biologiques, morphologies, processus pathologiques, caractéristiques patient (telles que les classes d'âge) et types de traitements et de surveillances. L'ontologie contient les concepts médicaux de base et les règles de combinaison, mais ne comporte pas l'ensemble des maladies, traitements... qui peuvent être générés par combinaison (il s'agit donc de *post-coordination*). Ce second module a été modélisé avec Protégé, en s'appuyant sur les terminologies médicales (SNOMED CT en particulier) et le réseau sémantique de l'UMLS (*Unified Medical Language System*) [82], et en se limitant au niveau de granularité élevé qui est celui de VCM.

Le troisième module (ontologie de correspondance, 509 axiomes) relie par des contraintes sur les relations « représente » et « est représenté par » les concepts des deux ontologies précédentes. Par exemple, le pictogramme central « poumon » est seulement présent sur des icônes qui représentent des concepts médicaux liés à une « structure pulmonaire » ou à la « fonction respiratoire ». Ces relations ont été générées automatiquement à partir d'un fichier texte comprenant des triplets, comme décrit précédemment.

L'ontologie dans sa totalité comprend 609 concepts, 41 relations et 3 934 axiomes, définis à l'aide du lan-

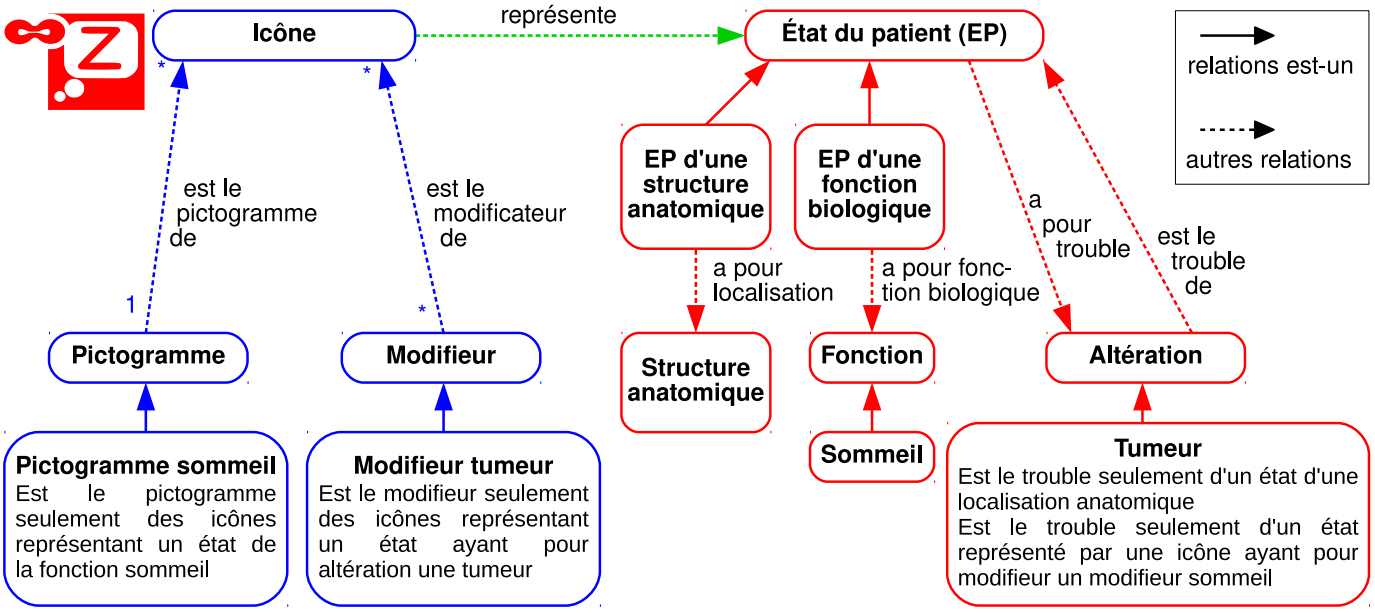


FIGURE 3.20 – Représentation de l'icône inconsistante « tumeur du sommeil » dans l'ontologie.

gage OWL-DL. Elle appartient à la famille *ALCRIQ* des logiques de description (*Attribute Language*, *Complex concept negation*, *Role*, *Inverse property*, *Qualified cardinality restriction*), pour laquelle il est démontré que le raisonnement est décidable [68].

3.3.3.2 Vérification de la consistance des icônes

La première application de l'ontologie VCM a été la vérification de la consistance des icônes [110]. En effet, certaines combinaisons du langage VCM sont inconsistantes : par exemple une icône associant le modificateur de forme « tumeur » et le pictogramme central « sommeil » signifie « tumeur du sommeil », ce qui est absurde du point de vue médical. Ces icônes inconsistantes peuvent poser problème, notamment lorsque des utilisateurs sont amenés à construire eux-mêmes une icône en sélectionnant plusieurs primitives. De la même manière que pour les panneaux routiers, l'ontologie VCM permet de vérifier la consistance des icônes.

La Figure 3.20 montre un exemple de représentation dans l'ontologie d'une icône inconsistante. L'inconsistance peut être déduite à partir des contraintes modélisées dans l'ontologie : (a) l'icône a le pictogramme « sommeil », et donc elle représente un état de la fonction « sommeil », (b) l'icône a le modificateur de forme « tumeur », et donc elle représente un état qui a pour altération la morphologie « tumeur », (c) la morphologie « tumeur » est une altération anatomique, qui ne peut s'appliquer qu'à des états d'une structure anatomique, (d) les structures anatomiques et les fonctions biologiques sont disjointes.

Ce raisonnement a pu être mis en œuvre avec le raisonneur HermiT [137], soit pour une seule icône (un individu), soit pour une classe d'icônes, en créant dans l'ontologie le concept des icônes partageant telles ou telles composantes. Par exemple, si l'on souhaite déterminer la consistance de l'icône « maladie du système respiratoire », nous pouvons la décrire par la classe I suivante :

$$
\begin{aligned}
I \sqsubseteq Icône \quad & \sqcap \quad \exists a_pour_forme.Carré \\
& \sqcap \quad \forall a_pour_forme.Carré \\
& \sqcap \quad \exists a_pour_couleur_centrale.Rouge \\
& \sqcap \quad \forall a_pour_couleur_centrale.Rouge \\
& \sqcap \quad \exists a_pour_pictogramme_central.Pictogramme_poumon \\
& \sqcap \quad \forall a_pour_pictogramme_central.Pictogramme_poumon \\
& \sqcap \quad \neg\exists a_pour_modificateur_de_forme.Thing \\
& \sqcap \quad \neg\exists a_pour_couleur_en_exposant.Thing \\
& \sqcap \quad \neg\exists a_pour_pictogramme_en_exposant.Thing \\
& \sqcap \quad \neg\exists a_pour_pictogramme_en_second_exposant.Thing
\end{aligned}
$$

Ensuite le raisonneur permet de tester la consistance de I.

De manière similaire, il est possible de déterminer les Concepts du domaine associé à l'icône (fonction get_meaning()). Par exemple, pour tester si l'icône I « signifie » l'organe poumon, nous pouvons définir la classe I' comme suit :

$$I' \sqsubseteq I \sqcap (\exists représente.(\exists a_pour_localisation.Poumon))$$

Le raisonneur permet de tester la consistance de I'. Si (et seulement si) I' est consistante, alors l'organe poumon fait parti du sens de I (soit $Poumon \in M$). Il est ainsi possible de tester les différents concepts du domaine.

En pratique, l'implémentation dans un serveur d'icône est cependant rendue difficile, à cause d'une part du temps de calcul du raisonneur (son lancement demande plus d'une seconde) qui ne permet pas son exécution dynamiquement lors de la génération des icônes, et d'autre part du nombre très important d'icônes possibles (environ 200 millions) qui fait qu'il n'est pas possible de stocker les résultats pour chacune dans un cache. Par conséquent, le cache contient des informations sur des groupes d'icônes (par exemple l'ensemble des icônes partageant un pictogramme donné et un modificateur de forme donné) conduisant aux mêmes résultats, et non sur des icônes individuelles. Pour accélérer le temps de calcul, les icônes sont groupées par paquet de 10 000 et envoyées au raisonneur, en faisant tourner un raisonneur par processeur. La génération complète du cache prend environ 3 heures et la base de données produite occupe environ 120 Mo.

La détermination de la consistance a ensuite été évaluée [110]. Pour illustrer les applications possibles, nous avons vérifié la consistance des 521 icônes présentes dans le didacticiel du langage VCM. 25 de ces icônes ont été détectées comme inconsistantes par l'ontologie ; une vérification manuelle a montré que 14 d'entre elles correspondaient à des erreurs dans le didacticiel qui ont été corrigées par la suite, 5 étaient des icônes incomplètes qui étaient utilisées comme résultat intermédiaire pour expliquer la construction des icônes, et 6 étaient en réalité consistantes.

3.3.3.3 Autres applications

L'ontologie des icônes VCM a ensuite été utilisée pour trois autres applications :

1. La génération semi-automatique d'une correspondance entre VCM et la SNOMED-CT,
2. La génération automatique d'un lexique des pictogrammes bilingues français-anglais,
3. La génération automatique de libellés bilingues pour chaque icône VCM.

Ces applications sont décrites en détail dans l'article publié en français dans la Revue d'Intelligence Artificielle [109] et en anglais dans Knowledge Based System [108].

3.3.4 Discussion

Dans cette partie, nous avons proposé une méthode générique pour formaliser la sémantique d'un langage iconique, à l'aide d'une ontologie formelle qui décrit les *icônes* du langage et leur syntaxe d'une part, les *objets du domaine* représentés par les icônes d'autre part, ainsi que les relations « représente » qui existent entre les deux. Nous avons montré comment cette méthode pouvait être appliquée à deux langages iconiques : la signalisation routière et le langage VCM. Nous avons aussi montré l'intérêt de cette formalisation à travers différentes applications. Ces applications auraient pu être accomplies avec des représentations des connaissances spécifiques à chacune d'elles (par exemple des règles de grammaire pour vérifier la syntaxe des icônes), cependant l'ontologie développée a permis de les réaliser toutes à partir d'une même source de connaissance. La méthode proposée pourrait faciliter la création de nouveaux langages iconiques, améliorer la cohérence et la qualité de ces langages, faciliter leur apprentissage par la production automatique de lexiques et de libellés, et enfin faciliter leur utilisation en générant des alignements avec des ressources existantes.

3.3.4.1 Comparaison à la littérature

La méthode que nous avons proposée se rapproche de plusieurs approches présentées dans l'état de l'art (section 3.1.4) : (a) les études sémiotiques de [131, 132] pour la décomposition des icônes en éléments

(pictogrammes, formes géométriques, couleurs), (b) les syntaxes abstraites visuelles (*abstract visual syntax*) [49, 44] pour l'emploi d'une syntaxe non graphique plus abstraite et formelle que les langages visuels et (c) les travaux de V. Haarslev [64] pour l'emploi d'une ontologie formelle en logiques de description (LD). Cependant, contrairement à V. Haarslev, nous avons cherché à représenter dans l'ontologie la syntaxe du langage graphique mais aussi sa sémantique. Par rapport aux approches existantes, la nôtre permet d'aller beaucoup plus loin dans le raisonnement sémantique.

3.3.4.2 Grammaire ou sémantique ?

Dans l'état de l'art (Section 3.1.4), nous avons vu que plusieurs approches proposées pour formaliser la sémantique des langages graphiques sont en fait de nature grammaticale voire syntaxique. Par exemple, nous aurions pu déterminer la consistance des panneaux routiers ou des icônes VCM à l'aide d'une grammaire formelle, telle que la suivante :

<panneau> : := <panneau d'interdiction> | <panneau d'obligation>
| <panneau d'information de danger>
| <autre panneau d'information>
<panneau d'information de danger> : := triangle rouge avec <rencontre> au milieu
<panneau d'interdiction> : := rond rouge avec <action> au milieu
<rencontre> : := virage à gauche, travaux
<action> : := virage à gauche, rouler à plus de 50 km/h

Cependant, cette formalisation intègre sans différenciation des éléments hétérogènes : des éléments graphiques (rond rouge) et des éléments non graphiques (travaux). Elle ne sépare pas l'objet représenté de sa représentation, ni la syntaxe de la sémantique. En réalité, le panneau « interdiction de travaux » ou l'icône VCM « cancer du sommeil » sont parfaitement corrects d'un point de vue syntaxique, même si tous deux sont absurdes d'un point de vue sémantique.

L'utilisation d'une grammaire pour exprimer la sémantique nous semble peu appropriée car la sémantique fait souvent appel à des relations d'héritage (est-un), pour lesquelles une ontologie est plus adaptée qu'une grammaire. Par ailleurs, toutes les applications que nous avons présentées n'auraient pas pu être réalisées si nous n'avions pas exprimé la sémantique indépendamment de la syntaxe. Par exemple, il aurait été difficile de générer un lexique des pictogrammes à partir d'une grammaire dans laquelle pictogrammes et concepts médicaux auraient été mélangés.

3.3.4.3 Perspectives

Au cours de travaux futurs, il serait intéressant d'essayer d'étendre notre approche dans des contextes où le langage d'icônes est moins formalisé, par exemple pour les icônes des interfaces utilisateurs dans les logiciels, et d'évaluer ce qu'une telle approche pourrait apporter. La méthode que nous avons proposée pour formaliser la sémantique pourrait également être envisagée au-delà des langages iconiques, pour formaliser la sémantique de manière plus générale.

L'alignement de l'ontologie VCM avec une ontologie de fondement (*top ontology*) est une autre perspective.

Chapitre 4

Visualisation structurelle des connaissances

« Si je décris les sortilèges des deux magiciens en termes de formes et de couleurs, c'est parce que je ne comprends pas leur nature. »
(La reine de Vendôme volume 1, Colin Marchika, éditions Mnémos, 2001)

4.1 Introduction et contexte

Une seconde approche pour rendre les connaissances visuelles consiste à utiliser des techniques de visualisation d'information. Ces techniques permettent de traduire graphiquement la structure de l'information, par exemple traduire sous forme de graphe les relations existantes entre plusieurs éléments.

Dans ce chapitre, nous définirons la visualisation d'information et nous présenterons les principales techniques de visualisation existantes, puis nous décrirons plus en détail un sous-domaine : la visualisation d'ensembles non disjoints. Nous verrons ensuite comment appliquer ces techniques à la visualisation des connaissances. Enfin, nous présenterons une contribution portant sur la conception et l'évaluation d'une nouvelle technique pour la visualisation d'ensembles non disjoints. Cette technique est particulièrement bien adaptée pour des connaissances et notamment les connaissances sur le médicament.

4.1.1 Visualisation d'information

La *visualisation d'information* est une discipline de l'informatique qui « utilise des représentations visuelles et interactives de données abstraites sur un ordinateur pour amplifier la cognition » [31], avec six objectifs possibles : afficher une grande quantité d'information, faciliter la recherche d'une information donnée, aider à la détection de motifs (fouille de données ou *data-mining* visuelle [12]), permettre des inférences visuelles, surveiller la survenue d'événements, explorer le jeu de données. La visualisation d'information s'intéresse notamment à des données abstraites, pour lesquelles il n'y a pas de représentation graphique évidente (par opposition à la visualisation de signaux purement numériques ou à la visualisation d'information géographique basée sur des cartes).

Il existe de nombreuses techniques pour la visualisation d'information. La plupart des taxonomies existantes classent les techniques en fonction de la nature des informations visualisées (nombre de variables et nature de celles-ci : qualitative ou quantitative) [180, 149], cependant les différentes catégories sont loin d'être exclusives, notamment lorsque l'on travaille sur des connaissances où les éléments sont souvent complexes et hétérogènes. C'est pourquoi nous proposons ici une classification différente, fondée sur la stratégie de visualisation, selon quatre grands groupes.

4.1.1.1 Graphes

Ce groupe rassemble les visualisations où l'on relie différents éléments entre eux, en fonction des relations existants entre les éléments.

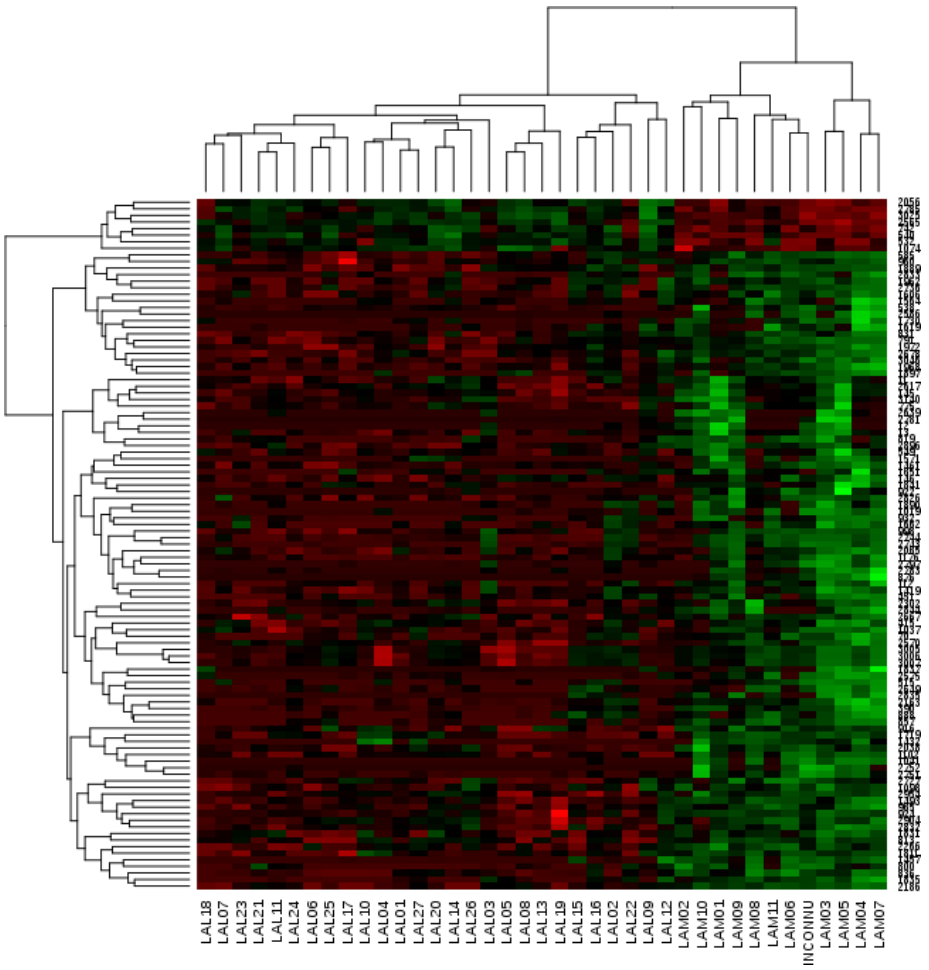


FIGURE 4.1 – Exemple de carte thermique (au centre) avec deux dendrogrammes (en haut et à gauche) montrant les résultats d'une étude par puce à ADN sur des patients souffrant de leucémies aiguës lymphoïdes (LAL) ou myéloïdes (LAM). L'analyse permet de classer un patient pour lequel le type de leucémie est inconnu (INCONNU). Les patients sont en colonne et les gènes différentiellement exprimés en ligne. Dans chaque case la couleur indique le niveau d'expression du gène (rouge : surexprimé, vert : sous-exprimé, noir : moyennement exprimé). Cet exemple est issu de mes enseignements de bioinformatique.

Il est possible de visualiser sous forme de graphes un ensemble d'objets et leurs relations, chaque objet est représenté par un nœud et chaque relation par un arc reliant deux nœuds. Il existe plusieurs algorithmes pour générer automatiquement des visualisations en graphes et pour placer les nœuds dans l'espace, avec des résultats souvent difficiles à lire lorsque le nombre de nœuds est important.

Les arbres sont un type de graphe particulier qui permettent de représenter des hiérarchies (correspondant à un héritage simple). **Le dendrogramme** (voir Figure 4.1) est un type d'arbre particulier dans lequel la longueur des branches est proportionnelle à l'éloignement entre un nœud fils et son père.

4.1.1.2 Tableaux et représentations tabulaires

Ce groupe rassemble les tableaux et les matrices, et plus généralement toute représentation dérivée, c'est-à-dire tirant parti des capacités de la vision humaine à distinguer plus facilement les lignes horizontales et verticales que les lignes obliques [27, 187]. Les éléments visualisés sont donc placés en lignes ou en colonnes.

Un exemple de visualisations dérivées des tableaux est la carte thermique (*heatmap*) qui est notamment très utilisée en bioinformatique (Figure 4.1). La carte thermique utilise des couleurs dans chaque case du tableau, et les lignes et les colonnes sont ordonnées de sorte à rapprocher les lignes et les colonnes similaires. Un autre exemple est le *DataTube* [24, 23], qui consiste à « plier » le tableau pour obtenir un tube en 3D à l'intérieur duquel l'utilisateur peut naviguer.



FIGURE 4.2 – Exemple de *treemap* pour l'analyse fonctionnelle de séquences biologiques (d'après [158]). Chaque région rectangulaire représente une séquence possible, et est découpée en deux parties, rouge et verte, qui indiquent la proportion d'individus malades et sains pour cette séquence.

4.1.1.3 Positions

Ce groupe rassemble les approches qui consistent à positionner les éléments à visualiser dans l'espace et à associer une signification à leurs positions.

Une première approche cherche à partitionner l'espace en différentes régions. **Les diagrammes de Venn, d'Euler et leurs nombreuses variantes**, consiste à placer les éléments à visualiser dans l'espace sous forme de point, puis à découper l'espace pour définir plusieurs régions, chaque région correspondant à un sous-ensemble des éléments partageant une propriété commune (voir exemple Figure 4.5 en bas à droite). Les différentes régions peuvent présenter des intersections si nécessaires. Les diagrammes de Venn et d'Euler sont faciles à comprendre, en revanche ils peuvent devenir difficiles à lire lorsque le nombre d'ensembles devient élevé. De plus, il n'existe pas d'algorithme permettant de générer automatiquement un diagramme de Venn ou d'Euler optimal dans le cas général.

Les *treemap* consistent à diviser l'espace disponible en différentes régions, chaque région représentant un élément à visualiser (voir exemple Figure 4.2). Contrairement aux diagrammes de Venn ou d'Euler, il n'y a pas d'intersection entre régions (hors inclusion totale) : les régions peuvent être imbriquées de manière hiérarchique, mais chaque région n'a qu'un seul parent. La couleur et la taille de chaque région peuvent varier en fonction des attributs des éléments et/ou du nombre d'éléments présents.

Une seconde approche cherche à calculer la position exacte de chaque élément en fonction de ses propriétés. Un exemple type est **l'analyse en composante principale** (ACP) suivie d'une visualisation en nuage de points sur deux dimensions (voir exemple Figure 4.3).

4.1.1.4 Facettes

Ce groupe inclut les approches qui décomposent les éléments à visualiser selon plusieurs facettes. Un élément ne sera donc pas représenté par un objet visuel unique et facilement identifiable, mais par un ensemble d'objets visuels distincts les uns des autres.

Une première approche est la visualisation en coordonnées parallèles [76] (voir exemple Figure 4.4). Cette visualisation considère un ensemble d'objets avec n attributs par objet. Au lieu de placer un point par objet dans un repère à n dimension, chaque attribut définit un repère à une 1 dimension et chaque objet est représenté par n points, un pour chaque attribut. Les points d'un même objet sont ensuite reliés par une ligne brisée.

Une autre approche a été proposée par M.J. Graham pour visualiser des taxonomies parallèles [57]. Chaque objet a une position différente dans chacune des taxonomies, et est représenté une fois pour

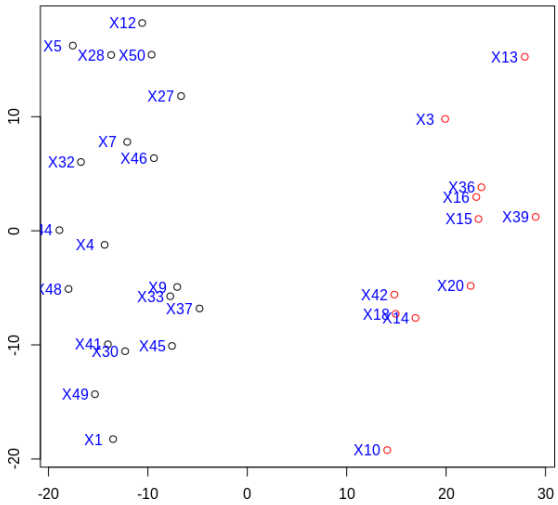


FIGURE 4.3 – Exemple de nuage de points issu d'une analyse en composantes principales (ACP). Chaque point correspond à un patient dont l'expression des gènes a été déterminée à l'aide de puce à ADN (22 283 gènes par patient). Cet espace à 22 283 dimensions a ensuite été ramené à un espace à deux dimensions par ACP et projection sur les deux premières dimensions. Cet exemple est issu de mes enseignements de bioinformatique.

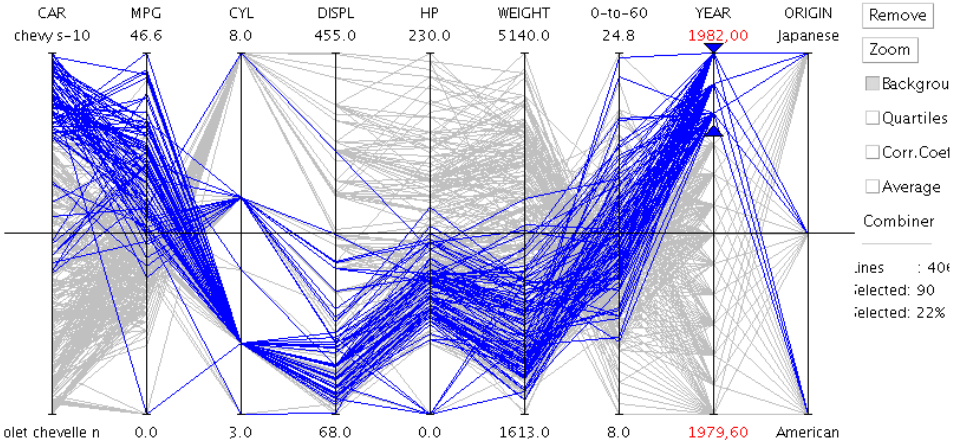


FIGURE 4.4 – Exemple de coordonnées parallèles, représentant les caractéristiques de plusieurs modèles de voitures. La sélection des modèles récents (en bleu, sélection sur l'avant-dernier axe) permet de voir une tendance à la baisse du poids du véhicule et du nombre de cylindres.

chaque taxonomie.

L'interactivité est très souvent utilisée en complément des différentes techniques de visualisation. Elle intervient dans le « mantra » de la recherche visuelle : parcourir une vue d'ensemble, zoomer, filtrer puis demander davantage de détails (*overview, zoom, filter, details-on-demand*) [165]. L'interactivité est nécessaire ici pour choisir la partie de la visualisation à zoomer, choisir les éléments à filtrer ou à exclure, et pour choisir l'élément sur lequel l'utilisateur souhaite obtenir des détails.

4.1.2 Visualisation d'ensembles non disjoints

La visualisation d'ensembles non disjoints (ou ensembles chevauchants, *overlaping sets* en anglais) est un sous-domaine de la visualisation d'information dans lequel l'objectif est de visualiser le plus clairement possible des éléments ainsi que plusieurs ensembles composés de ces éléments (un même élément pouvant appartenir à plusieurs ensembles). Il s'agit d'un problème « symétrique », c'est-à-dire que les rôles joués par les éléments et les ensembles peuvent être échangés.

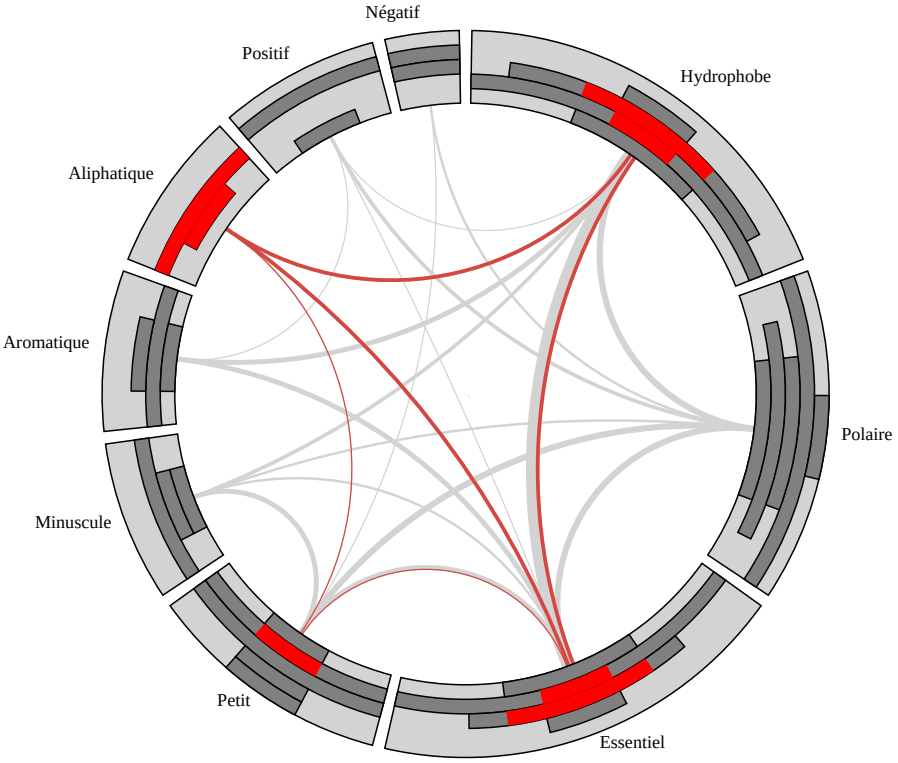
Un exemple bien connu d'ensembles non disjoints est celui des propriétés des acides aminés. Il existe 20 acides aminés (Alanine, Proline... que l'on peut abréger en codes de 3 lettres : Ala, Pro...). Plusieurs acides aminés ont des propriétés physiques ou chimiques communes, telles que leur *petite* taille ou la présence d'un cycle *aromatique*. Une dizaine de propriétés sont traditionnellement définies. Les acides aminés peuvent être considérés comme des éléments, et les propriétés comme des ensembles regroupant les acides aminés partageant cette propriété.

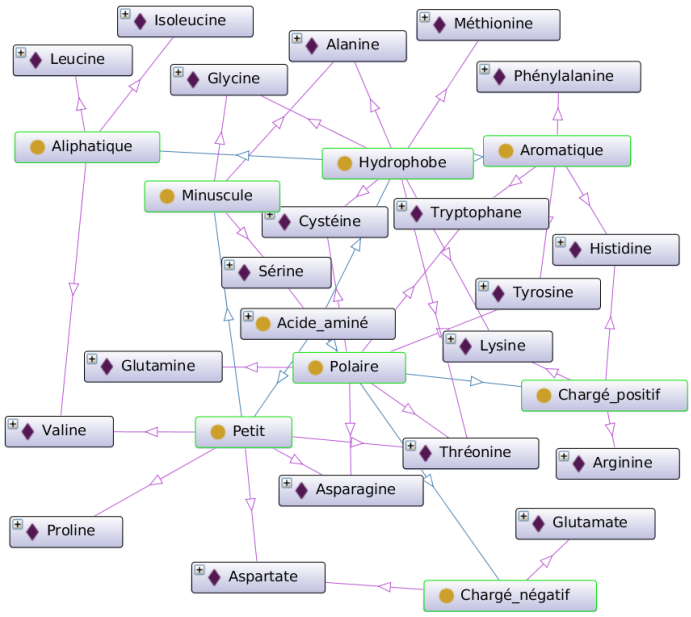
Dans la littérature, la visualisation d'ensembles non disjoints a donné lieu à de nombreux travaux dont B. Alsakallah *et al.* ont effectué une revue récente [7]. Nous retrouvons les 4 approches de visualisation que nous avons décrites précédemment :

- **Les graphes** : les éléments et les ensembles figurent sur la visualisation, et des arcs relient les éléments aux ensembles auxquels ils appartiennent.
- **Les tableaux** : les éléments figurent dans un tableau en ligne et les propriétés en colonne (ou l'inverse).
- **Les positions** :
  - les diagrammes d'Euler, de Venn et leurs dérivés,
  - les nuages de points : chaque élément est visualisé par un point, les coordonnées de chaque point étant calculé en fonction des ensembles auxquels appartient l'élément de sorte à rapprocher les éléments similaires, sans nécessairement faire apparaître les ensembles de manière explicite,
  - les approches cartographiques pour des informations géographiques.
- **Les facettes** : il s'agit d'approches par agrégation où les éléments ne figurent pas en tant que tels mais sont agrégés ; la visualisation montre alors des données agrégées et stratifiées par ensemble, telles que les fréquences (nombres d'éléments appartenant à l'ensemble X et à Y autres ensembles) ou les coocurrences (nombres d'éléments appartenant à l'ensemble X et Y). Un exemple est le *radial sets* [6].

La Figure 4.5 montre un exemple de chaque approche sur les acides aminés. Les approches graphes et facettes conduisent ici à des visualisations peu lisibles : le graphe à cause du nombre important de relations, et les *radial sets* ont été développés pour visualiser des jeux de données très volumineux (ce qui n'est pas le cas ici). En biologie, c'est le diagramme de Venn qui est traditionnellement utilisé pour représenter les propriétés des acides aminés.

	Petit	Minuscule	Hydrophobe	Aliphatique	Aromatique	Polaire	Positif	Négatif
Ala	X	X	X					
Arg						X	X	
Asn	X					X		
Asp	X					X		X
Cys	X	X	X			X		
Glu						X		X
Gln						X		
Gly	X	X	X					
His			X		X	X	X	
Ile			X	X				
Leu			X	X				
Lys						X	X	
Met			X					
Phe			X		X			
Pro	X							
Ser	X	X				X		
Thr	X		X			X		
Trp			X		X	X		
Tyr			X		X	X		
Val	X		X	X				





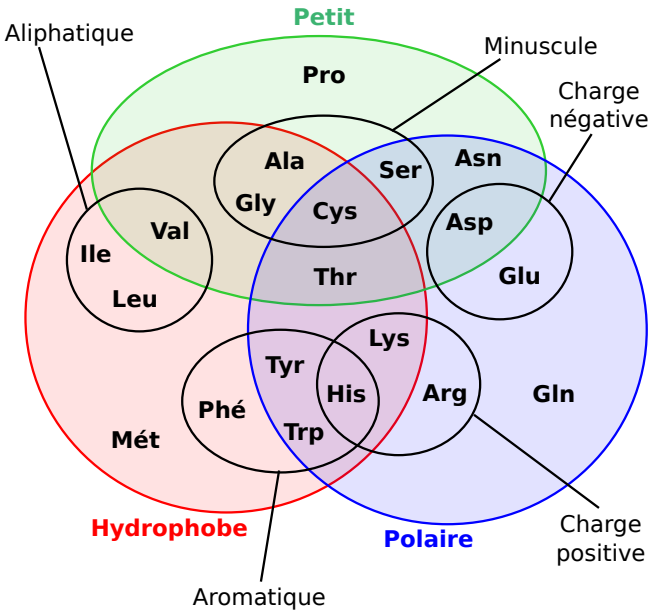


FIGURE 4.5 – Visualisation des propriétés des acides aminés selon les quatre approches : tableau (en haut à gauche), facettes (*radial sets*, en haut à droite), graphe (généré avec OntoGraph, en bas à gauche) et positions (diagramme de Venn, en bas à droite). Sur les *radial sets*, les ensembles sont placés en cercle et pour chacun d'entre eux figure un histogramme indiquant le nombre d'éléments de cet ensemble qui appartiennent à 0, 1, 2, ..., n autres ensembles. L'épaisseur des arcs placés au centre du cercle indique le nombre de cooccurrence, c'est-à-dire le nombre d'éléments appartenant aux deux ensembles reliés par l'arc. L'utilisateur peut sélectionner les éléments d'un ensemble et voir comment ceux-ci sont répartis parmi les autres ensembles (ici, l'ensemble des aliphatiques a été sélectionné).

Encadrement associé à cette contribution :

1. Coralie Capron, stage de DUT Génie biologique option bioinformatique (2 mois en 2016)

Publications associées à cette contribution :

1. **Lamy JB**, Berthelot H, Favre M. Rainbow boxes : a technique for visualizing overlapping sets and an application to the comparison of drugs properties. Information Visualization 2016, Lisbonne, Portugal
2. **Lamy JB**, Berthelot H, Favre M. Les boîtes arc-en-ciel : une nouvelle technique pour visualiser les relations hiérarchiques dans les ontologies. Congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA) 2016, Clermont-Ferrand
3. **Lamy JB**, Berthelot H, Capron C, Favre M. Rainbow boxes : a new technique for overlapping set visualization and two applications in the biomedical domain. **Journal of Visual Language and Computing** 2017 ;43 :71-82
4. Tsopra R, **Lamy JB**. Design of a visual interface for comparing antibiotics using rainbow boxes. Stud Health Technol Inform. 2017 ;235 :529-533. Conférence Medical Informatics Europe MIE 2017, Manchester, Angleterre (taux de sélection 28%), **best paper**
5. **Lamy JB**, Tsopra R. Translating visually the reasoning of a perceptron : the weighted rainbow boxes technique and an application in antibiotherapy. Information Visualization 2017, Londres, Angleterre, **best paper**

4.2 Contribution : Les boîtes arc-en-ciel

Lors du projet VIIIP, l'un des objectifs était de mettre au point des méthodes visuelles pour comparer les propriétés de plusieurs médicaments de même indication. La comparaison des contre-indications, interactions et effets indésirables de plusieurs médicaments est un problème de visualisation d'ensembles non disjoints : les médicaments sont les éléments et les propriétés sont les ensembles, par exemple *l'ensemble des médicaments contre-indiqués en cas d'hypertension* ou *l'ensemble des médicaments pouvant provoquer des nausées* (un effet indésirable). Un même médicament peut bien sûr appartenir à plusieurs de ces ensembles.

Dans cette section, nous présentons les *boîtes arc-en-ciel* (*rainbow boxes*) une nouvelle technique de visualisation d'ensembles non disjoints que nous avons mise au point pour visualiser entièrement des jeux de données de taille relativement faible (2-25 éléments et 5-100 ensembles environ). Nous présentons d'abord cette technique sur l'exemple simple des acides aminés (voir section 4.1.2), puis nous montrons son application à trois exemples complexes : (1) la comparaison des propriétés de médicaments similaires lors du projet VIIIP, (2) la visualisation d'annotations de gènes extraites de Gene Ontology, et (3) la traduction visuelle du raisonnement d'un perceptron, avec une application en antibiothérapie. Nous évaluons ensuite cette technique au cours de deux expérimentations utilisateur. Enfin, nous discuterons les résultats obtenus et les perspectives ouvertes.

4.2.1 Principe général des boîtes arc-en-ciel

Dans les boîtes arc-en-ciel, chaque élément est représenté par une colonne et chaque ensemble par une boîte rectangulaire qui occupe les colonnes correspondant aux éléments appartenant à cet ensemble. Le nom de l'ensemble est affiché à l'intérieur de la boîte. Lorsque cela est nécessaire, des « trous » sont présents dans les boîtes ; les éléments sont ordonnés de sorte à limiter au maximum les trous, à l'aide d'un algorithme décrit en section 4.2.3. Les boîtes les plus larges sont placées en bas (la largeur étant calculée globalement, en incluant les trous), ensuite les boîtes sont empilées comme les blocs d'un jeu de « Tétris » ; plusieurs boîtes peuvent donc être côte à côte du moment qu'elles n'occupent pas les mêmes colonnes. La Figure 4.6 présente les propriétés des acides aminés sous forme de boîtes arc-en-ciel.

Par défaut, les boîtes arc-en-ciel n'utilisent pas la couleur pour apporter une information supplémentaire, mais plutôt pour faciliter la lecture et rendre la visualisation plus attractive. Chaque élément reçoit une couleur différente, du rouge (à gauche) au violet (à droite) en traversant tout le spectre selon l'ordre des colonnes. La couleur d'une boîte est la moyenne des couleurs (en coordonnées RVB, Rouge-Vert-Bleu) des éléments de l'ensemble correspondant. Par conséquent, les boîtes regroupant peu d'éléments ont tendance à prendre la couleur de ces éléments tandis que les boîtes avec beaucoup d'éléments tirent sur le gris.

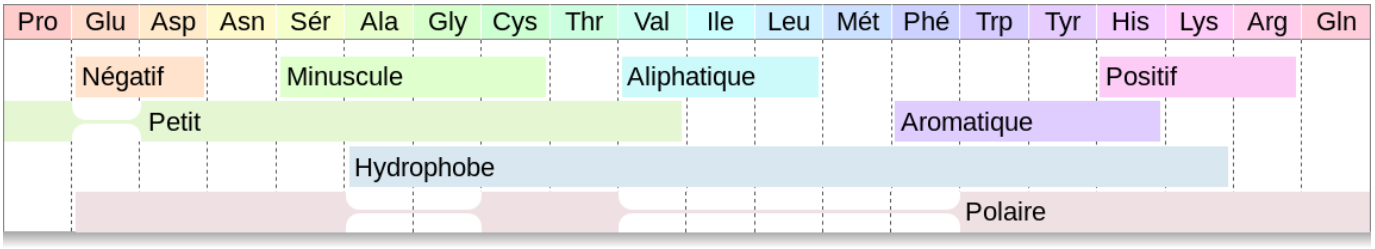


FIGURE 4.6 – Boîtes arc-en-ciel montrant 8 propriétés des 20 acides aminés.

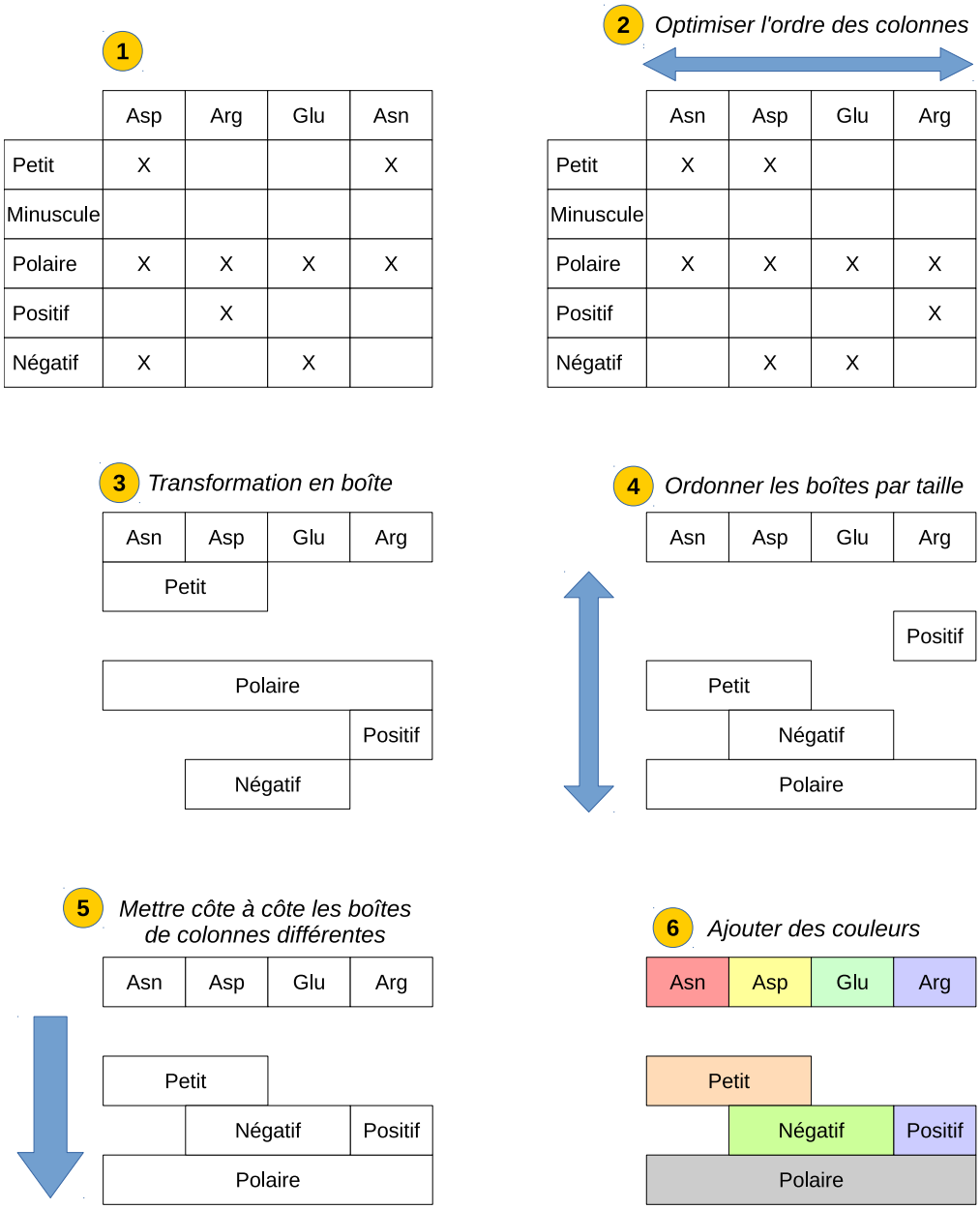


FIGURE 4.7 – Les 6 étapes de la mise au point des boîtes arc-en-ciel.

Néanmoins les couleurs peuvent aussi être spécifiées manuellement pour chaque couple (ensemble, élément) afin d'apporter une information complémentaire (voir exemple sur les effets indésirables en section 4.2.5.2). Les boîtes peuvent aussi inclure des hachures pour indiquer une appartenance « partielle » d'un élément à un ensemble, par exemple pour les acides aminés semi-essentiels sur la Figure 4.6. Enfin, la hauteur des boîtes peut aussi être utilisée comme variable visuelle.

Les boîtes arc-en-ciel indiquent l'appartenance des éléments aux différents ensembles, mais permettent aussi de découvrir de nouvelles relations ou similarités. Sur la Figure 4.6, il est facile de remarquer que les acides aminés *minuscules* sont tous *petits* puisque la boîte *petit* couvre la totalité des colonnes couvertes par la boîte *minuscule*. De même, aucun acide aminé n'est à la fois *petit* et *aromatique*. Par ailleurs, il est frappant de constater que la classe des acides aminés *essentiels* correspond quasiment à celle des *hydrophobes*.

4.2.2 Méthodologie pour la mise au point des boîtes arc-en-ciel

La conception de la technique des boîtes arc-en-ciel s'est faite en partant d'une visualisation des ensembles non disjoints en tableau simple (voir étape 1 Figure 4.7). Nous avons le constat suivant : le tableau contient une part importante de « vide » (c'est-à-dire des cases sans croix). Nous avons donc cherché à optimiser l'espace pour condenser la visualisation. Ensuite, la conception s'est faite de manière exploratoire et itérative (Figure 4.7), chaque étape correspondant :

- À ordonner les lignes ou les colonnes du tableau (étapes 2 et 4).
- À appliquer des principes généraux pour améliorer la visualisation, issus de la littérature [188], tels que :
  - Regrouper en un seul objet visuel ce qui va ensemble (étape 3).
  - Compacter les espaces vides (étape 5).
  - Ajouter des couleurs pour rendre la visualisation plus attractive (étape 6).

Après chaque itération, nous nous sommes demandé « comment améliorer la visualisation ? » et « quel principe de visualisation ou de design visuel pouvons-nous appliquer ? ».

4.2.3 Optimisation de l'ordre des colonnes

Pour tracer des boîtes arc-en-ciel avec le plus petit nombre possible de trous, l'ordre des colonnes (c'est-à-dire des éléments) doit être optimisé. Le nombre d'ordres possibles est $n!$ (factoriel n) où n est le nombre d'éléments. La complexité du problème est donc $\mathcal{O}(n!)$. Pour des valeurs de n relativement élevées (au-delà de 10), il n'est pas possible de tester tous les ordres possibles. Par exemple, pour les acides aminés, il y a $20! \approx 2{,}43\text{x}10^{18}$ ordres possibles.

Pour résoudre ce problème dans un temps satisfaisant, nous proposons ici une heuristique pour obtenir un ordre (quasi-)optimal dans un temps raisonnable. Cette heuristique génère dans un premier temps une liste réduite d'*ordres candidats* (204 dans l'exemple des acides aminés) et ensuite teste chacun de ces ordres candidats pour garder le meilleur (c'est-à-dire celui conduisant au plus petit nombre de trous et aux trous les plus courts dans la visualisation).

L'Algorithme 4.1 génère la liste des ordres candidats. Les ordres candidats sont construits progressivement, en commençant par *e0*, l'élément qui appartient au plus grand nombre d'ensembles et qui est donc le plus susceptible de créer des trous puisqu'il fait intervenir le plus grand nombre de boîtes. Ensuite, les éléments restants sont ajoutés soit au début soit à la fin de l'ordre en cours de construction. À chaque étape, l'élément à ajouter et sa position (début ou fin) sont déterminés à l'aide d'un score d'insertion, l'insertion avec le meilleur score étant retenue. Ce score prend en compte (a) le nombre d'ensembles auxquels appartiennent à la fois l'élément à ajouter et son voisin (c'est-à-dire le premier ou le dernier élément actuel de l'ordre en cours de construction, selon que l'on ajoute au début ou à la fin, respectivement) et (b) le nombre d'ensembles auxquels n'appartiennent ni l'élément à ajouter ni aucun des éléments déjà présents dans l'ordre en cours de construction. La partie (a) favorise l'ajout d'un élément qui appartient aux mêmes ensembles que son futur voisin, tandis que la partie (b) favorise l'ajout d'un élément qui n'appartient pas à des ensembles qui sont pour l'instant absents de l'ordre en cours de construction, c'est-à-dire que cela tend à « retarder » l'ajout de nouvelles boîtes dans la visualisation. Enfin, lorsque plusieurs *ex aequo* sont présents pour choisir *e0* ou

Algorithme 4.1 Heuristique pour trouver les ordres candidats (quasi-)optimaux (*ordres_candidats*). *éléments* est l'ensemble de tous les éléments et *ensembles* l'ensemble de tous les ensembles.

premiers_éléments_candidats = { $e0 \in$ *éléments* | $e0$ appartient au plus grand nombre d'ensembles dans *ensembles* }
ordres_en_construction = { ($e0 \in$ *éléments*) | $e0 \in$ *premiers_éléments_candidats* }
ordres_candidats = { } (ensemble vide)

Tant que *ordres_en_construction* $\neq$ { } :
    ordre = Prendre et enlever un tuple de *ordres_en_construction*
    restant = *éléments* $\setminus$ *ordre*
    insertions_possibles = $\{(e, \text{"}début\text{"}) \mid e \in restant\} \cup \{(e, \text{"}fin\text{"}) \mid e \in restant\}$
    Pour chaque couple (e, *position*) dans *insertions_possibles* avec le plus grand **score_insertion**(e, *position*, *ordre*) :
        Si *position* est "début" : *nouvel_ordre* = (e, *ordre*)
        Sinon si *position* est "fin" : *nouvel_ordre* = (*ordre*, e)
        Si *éléments* $\setminus$ *nouvel_ordre* = { } : Ajouter *nouvel_ordre* dans *ordres_candidats*
        Sinon : Ajouter *nouvel_ordre* dans *ordres_en_construction*

Fonction **score_insertion**(*élément*, *position*, *ordre*) :
    Si *position* est "début" : *voisin* = Premier élément de *ordre*
    Sinon si *position* est "fin" : *voisin* = Dernier élément de *ordre*
    score = 0
    Pour chaque ensemble X dans *ensembles*, tel que (*élément* $\in X$) et (*voisin* $\in X$) : *score* = *score* + 2
    Pour chaque ensemble Y dans *ensembles*, tel que (*élément* $\notin Y$) et ($\nexists\, j \in$ *ordre*, $j \in Y$) : *score* = *score* + 1
    Retourne *score*

pour choisir l'élément à ajouter ou sa position, toutes les possibilités sont considérées, ce qui conduit à la génération de plusieurs ordres candidats.

4.2.4 Implémentation

Le logiciel permettant la génération des boîtes arc-en-ciel et l'optimisation de l'ordre des colonnes a été développé en langage Python 3 dans un module appelé « RainbowBox ». Ce logiciel génère une page HTML (*HyperText Markup Language*) avec feuille de style CSS (*Cascading Style Sheet*) et code JavaScript pour l'affichage des boîtes arc-en-ciel dans un navigateur web. Le code source est disponible à l'adresse suivante : http://bitbucket.org/jibalamy/rainbowbox, sous licence libre (GNU LGPL v3).

4.2.5 Application à la comparaison des propriétés des médicaments

Nous décrirons ici l'application des boîtes arc-en-ciel à la comparaison des contre-indications et des effets indésirables des médicaments, lors du projet VIIIP. L'application aux interactions médicamenteuses a aussi été réalisée, cependant elle ne diffère pas sensiblement de l'application aux contre-indications.

4.2.5.1 Comparaison des contre-indications

La Figure 4.8 présente sous forme de boîtes arc-en-ciel les contre-indications de 8 médicaments pour la dysfonction érectile. Les boîtes arc-en-ciel « basiques » décrites à la section 4.2.1 ont été enrichies avec : (a) des en-têtes indiquant les classes pharmacothérapeutiques, au-dessus des en-têtes de colonnes, tels que « Prostaglandine E1 » pour Vitaros®, (b) des hachures pour représenter les situations où le médicament est déconseillé, c'est-à-dire des contre-indications relatives et non absolues, telles que l'insuffisance rénale sévère pour Cialis®, (c) des *widgets*, pour ajouter des informations spécifiques sur un couple (médicament, contre-indication) donné, tels que l'étiquette « (non-controllée) » pour la contre-indication hypertension artérielle de Cialis® ou les barres colorées dans la boîte « Âge », (d) des icônes VCM à droite des boîtes.

Les boîtes arc-en-ciel de la Figure 4.8 montrent les contre-indications de 8 médicaments, mais elles montrent aussi les similarités entre médicaments et en particulier les ressemblances qui existent au sein de chacune des deux classes pharmacothérapeutiques. Par exemple, la contre-indication avec la prédisposition au priapisme est clairement une propriété commune à la classe des prostaglandines E1. De même, il est facile de repérer que Muse® est le médicament le plus proche de Vitaros® en termes de contre-indication.

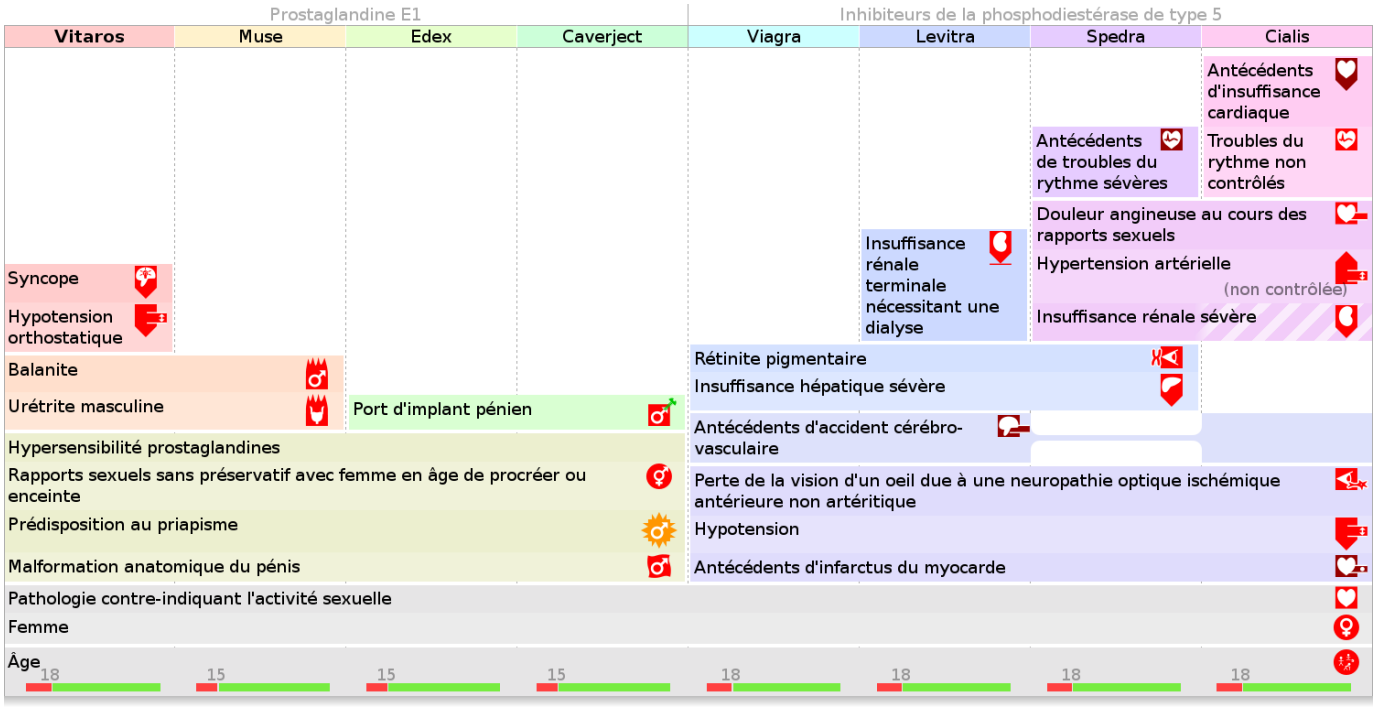


FIGURE 4.8 – Boîtes arc-en-ciel montrant les 79 contre-indications (26 distinctes) de 8 médicaments pour traiter la dysfonction érectile.

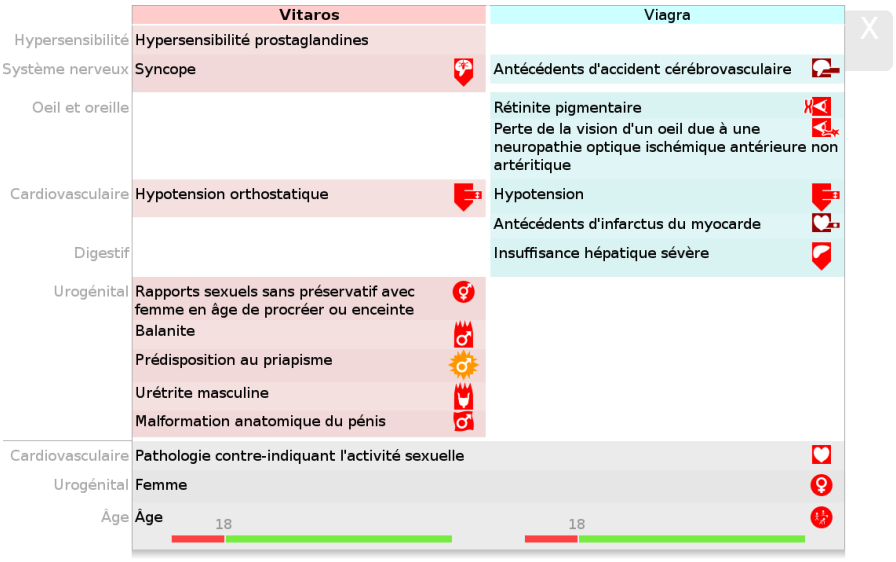


FIGURE 4.9 – Comparaison des contre-indications de deux médicaments (Vitaros® et Viagra®).

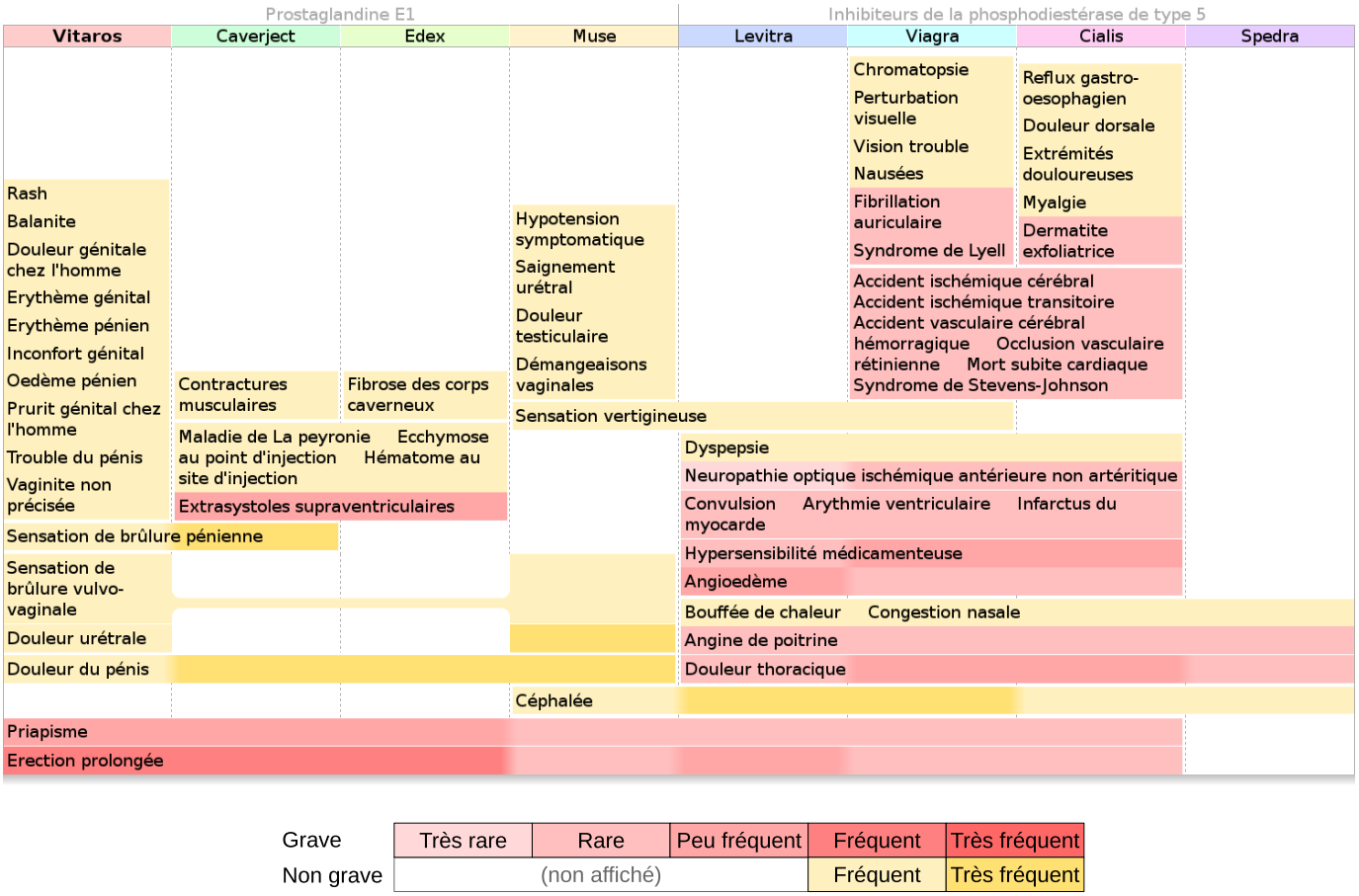


FIGURE 4.10 – Boîtes arc-en-ciel montrant les 116 effets indésirables (56 distincts) de 8 médicaments pour traiter la dysfonction érectile. Des couleurs personnalisées sont utilisées pour indiquer la gravité (teinte rouge ou orange) et la fréquence (saturation) des effets.

La visualisation a été rendue interactive : en cliquant sur les colonnes, l'utilisateur peut afficher une comparaison limitée aux deux médicaments choisis (Figure 4.9). Cette comparaison comprend trois parties : en haut à gauche, les contre-indications spécifiques du premier médicament, en haut à droite, celles du second médicament, et en bas, les contre-indications communes aux deux médicaments. Dans chaque partie, les contre-indications sont triées par organes.

4.2.5.2 Comparaison des effets indésirables

La Figure 4.10 montre un exemple de boîtes arc-en-ciel pour présenter les effets indésirables des huit médicaments. Les effets indésirables sont plus complexes à visualiser que les contre-indications, car ils sont caractérisés par deux attributs additionnels : la gravité (attribut booléen) et la fréquence de l'effet (mesurée sur l'échelle standard de cinq valeurs : très rare, rare, peu fréquent, fréquent, très fréquent). Seuls les effets graves et/ou fréquents (ou très fréquents) ont été retenus, en effet les médecins ne sont pas intéressés *a priori* par les effets non graves et peu fréquents, d'après les focus groupes réalisés au début du projet VIIIP.

Les deux attributs ont été représentés à l'aide de couleurs personnalisées dans chaque cellule. La teinte indique la gravité (rouge : grave, orange : non grave) tandis que la saturation indique la fréquence (les couleurs plus saturées correspondent à des effets plus fréquents). Pour certaines boîtes, la couleur peut changer d'une colonne à l'autre (par exemple la boîte « Douleur du pénis » est moins saturée pour Vitaros® car cet effet est moins fréquent avec ce médicament qu'avec les autres prostaglandines E1).

La Figure 4.10 donne une vue globale des effets indésirables et permet plusieurs constats tels que : Vitaros® a de nombreux effets indésirables non graves, Viagra® a de nombreux effets graves, la plupart étant communs avec Cialis®, Spedra® est le médicament qui présente le moins d'effets indésirables.

4.2.6 Application à la visualisation d'annotations de gènes extraites de Gene Ontology

L'annotation des gènes consiste à associer à chaque produit de gène (ARNm ou protéine) un ou plusieurs termes d'annotations qui indiquent l'activité du produit ou sa localisation dans la cellule (Figure 4.11). Gene Ontology est une ressource qui est utilisée pour cela : elle comprend des termes d'annotations groupés en trois catégories (composants de la cellule, processus biologiques et fonctions moléculaires) et structurés par des relations de subsomption (est-un), de méronymie (partie-de) ou de régulation (active, inhibe, régule). Ces termes d'annotations sont ensuite associés aux produits de gènes.

L'annotation des gènes conduit à un gros volume de données qu'il est souvent difficile de visualiser. Lorsqu'on s'intéresse à une famille de gènes ou de protéines, la visualisation des annotations qui leur sont associées peut se voir comme un problème de visualisation d'ensembles non disjoints. Nous avons donc voulu étudier la possibilité de visualiser des annotations de gènes avec les boîtes arc-en-ciel.

Ce travail sur Gene Ontology a été réalisé par Coralie Capron, étudiante en DUT génie biologique option bioinformatique, sous ma direction. Tout d'abord, Gene Ontology a été intégré au module PyMedTermino [117] qui permet la gestion des terminologies médicales en Python. L'intégration s'est faite à partir de la base de données MySQL accessible en ligne sur le serveur de Gene Ontology. Ensuite, PyMedTermino a été utilisé pour accéder à Gene Ontology en Python, pour récupérer les données nécessaires et pour générer les boîtes arc-en-ciel avec le module RainbowBox développé lors du projet VIIIP.

Si l'on considère les protéines d'une même famille, chaque protéine est un élément et chaque terme d'annotation est un ensemble comprenant les protéines annotées avec ce terme. Les annotations peuvent alors être visualisées à l'aide de boîtes arc-en-ciel, avec les protéines en colonnes et les termes d'annotation dans les boîtes. Des couleurs permettent de distinguer les catégories de termes d'annotation.

La Figure 4.12 montre un exemple de visualisation des annotations de gènes au sein de la famille des histones chez l'être humain. Les histones sont des protéines qui se lient à l'ADN ; elles sont responsables du repliement de l'ADN et jouent également un rôle dans la régulation de l'expression des gènes. Cinq grandes catégories d'histones existent : H1, H2A, H2B, H3 et H4. Chaque catégorie se subdivise en sous-catégories (H1.0, H1.1,...). Sur la Figure 4.12, les histones ont été ordonnées selon l'algorithme décrit précédemment. Il est intéressant de noter que l'on retrouve, de manière empirique, la classification des histones : par exemple les six histones H1 ont été placées côte à côte.

Plusieurs constats sont possibles à partir de la visualisation :

— DNA binding (fonction chimique), chromosome, nucleosome, nucleus (composants cellulaires) sont les termes d'annotation communs à toutes les histones.
— Chromatin DNA binding (fonction chimique) est spécifique et caractéristique des histones H1.
— L'histone H4 est celle qui a le plus d'annotations, notamment des processus biologiques (en bleu).
— Les boîtes vertes et rouges sont plus larges et positionnées plus bas que les boîtes bleues. Les composants cellulaires et les fonctions chimiques sont donc communs à plusieurs histones, tandis que les processus biologiques semblent être davantage spécifiques de sous-type d'histones précis.

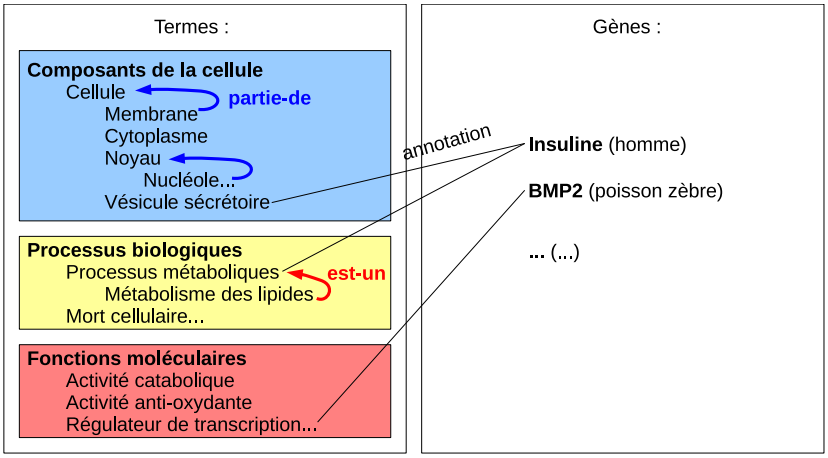


FIGURE 4.11 – Exemple d'annotations de gènes à l'aide de termes d'annotation issus de Gene Ontology.

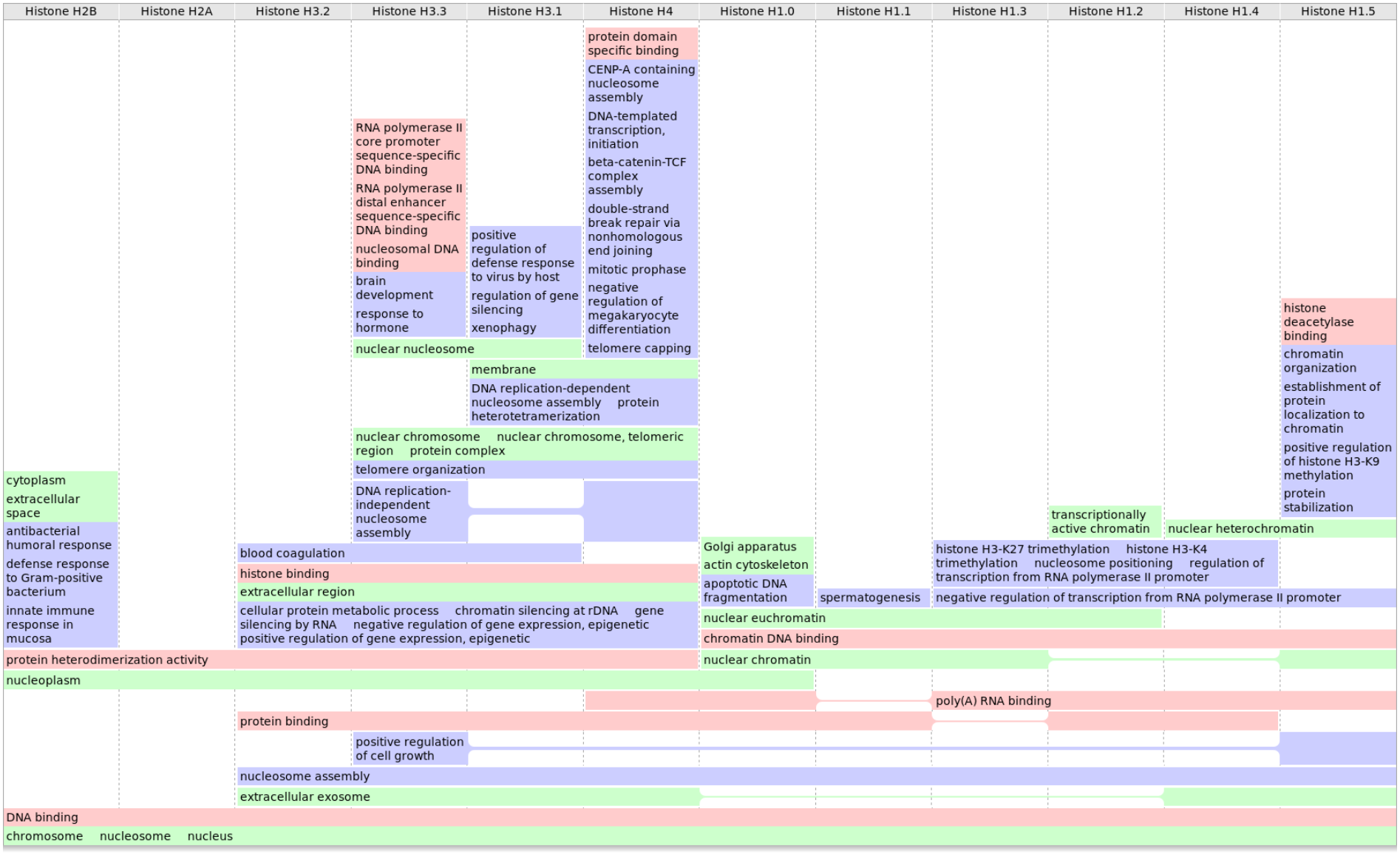


FIGURE 4.12 – Exemples de boîtes arc-en-ciel représentant les annotations de gènes au sein d'une famille de protéines : les histones. La couleur des boîtes indique la catégorie de terme d'annotation : composant cellulaire (vert clair), fonction moléculaire (rouge clair) et processus biologique (bleu clair).

4.2.7 Visualisation du raisonnement d'un perceptron et application en antibiothérapie

Les boîtes arc-en-ciel ont ensuite été étendues par l'ajout d'une nouvelle variable visuelle : la hauteur des boîtes. Cela a abouti aux boîtes arc-en-ciel pondérées (*weighted rainbow boxes*). Celles-ci sont particulièrement intéressantes car elles permettent de reproduire visuellement le raisonnement effectué par un neurone formel ou *perceptron* [2].

Un perceptron a n entrée I et calcule une sortie O. Un poids w_i est associé à chaque entrée I_i. Le perceptron calcule la somme pondérée de ses entrées, en multipliant la valeur de chaque entrée par le poids correspondant et en faisant la somme, et ensuite il applique la *fonction d'activation* f sur le résultat. Par conséquent, la sortie peut être calculé avec la formule suivante : $O = f(I_1 \times w_1 + I_2 \times w_2 + ... + I_n \times w_n)$.

Nous allons considérer un perceptron avec (a) des entrées booléennes (0 ou 1), (b) une sortie qui est un nombre réel positif, (c) des poids strictement positifs, et (d) la fonction d'activation $f_{noop}(x) = x$. Il est alors possible de représenté un ensemble de vecteurs d'entrée V à l'aide de boîtes arc-en-ciel, en considérant chaque vecteur d'entrée $(I_1, I_2, ..., I_n) \in V$ comme un élément, et chaque entrée comme un ensemble $\mathcal{I}_i$ auquel appartiennent tous les vecteurs d'entrée pour lesquels l'entrée est vraie : $\mathcal{I}_i = \{(I_1, I_2, ..., I_n) \in V \mid I_i = 1\}$. Les boîtes arc-en-ciel représentent chaque ensemble $\mathcal{I}_i$ par une boîte ; la hauteur de la boîte $\mathcal{I}_i$ peut être utilisée pour représenter le poids w_i associé à l'entrée i (voir Figure 4.13). La hauteur totale des boîtes empilées peut être « calculée » visuellement très rapidement, elle correspond à la valeur de sortie du perceptron. Les boîtes arc-en-ciel pondérées traduisent donc visuellement le raisonnement du perceptron.

J'ai appliqué ce système en antibiothérapie, en collaboration avec Rosy Tsopra (Assistante Hospitalo-Universitaire (AHU) au laboratoire LIMICS). Rosy a conçu une base de connaissances sur les antibiotiques [182, 183] à destination des médecins généralistes. Cette base regroupe l'ensemble des antibiotiques existants, des situations cliniques et des propriétés des antibiotiques. Une situation clinique correspond à une indication, c'est-à-dire une maladie infectieuse associée à un terrain (par exemple femme enceinte, enfant, adulte,...). Il y a 6 propriétés que le médecin doit prendre en compte pour choisir un antibiotique à prescrire, qui correspondent à des inconvénients. Il s'agit de : (1) une efficacité modérée, (2) un protocole de prise complexe, (3) un risque important d'émergence de résistance bactérienne, (4) un risque important d'effet indésirable, (5) un spectre bactérien trop large, et (6) l'appartenance à une classe précieuse (à préserver pour des pathologies graves). Pour chaque triplet *(situation, antibiotique, propriété)*, la base renseigne si la propriété est présente (vrai, 1) ou non (faux, 0).

Le raisonnement médical pour choisir un antibiotique se fait par l'exclusion progressive des plus mauvais antibiotiques, en considérant les propriétés selon l'ordre lexicographique suivant : 1 puis 2 puis 3 puis à égalité 4,5 et 6. Dans un premier temps, il s'agira donc d'éliminer les médicaments ayant une efficacité modérée, puis ceux ayant un protocole complexe, *etc*, jusqu'à arriver à un seul médicament (ou plusieurs *ex-aequo*).

Pour une situation clinique donnée, ce raisonnement peut être modélisé avec un perceptron de la manière suivante : les entrées correspondent aux 6 propriétés, les vecteurs d'entrées aux antibiotiques (chaque vecteur décrit les propriétés d'un antibiotique), les poids traduisent l'ordre lexicographique, et la sortie est un score. L'antibiotique avec le score le plus faible est celui qui a le moins d'inconvénients et est donc le meilleur. La Figure 4.14 montre le perceptron obtenu.

La Figure 4.15 montre un exemple de boîtes arc-en-ciel pondérées. La hauteur des boîtes est proportionnelle à l'importance chaque inconvénient, ce qui permet de calculer un score pour chaque antibiotique, en sommant visuellement les hauteurs des boîtes empilées dans chaque colonne. Ici, les scores les plus faibles correspondent aux meilleurs antibiotiques, et donc la nitrofurantoine est le meilleur traitement.

Cette approche visuelle a fait l'objet d'une première évaluation qualitative sur un petit échantillon de médecins généralistes. Pour davantage d'information, se référer aux publications [111, 181].

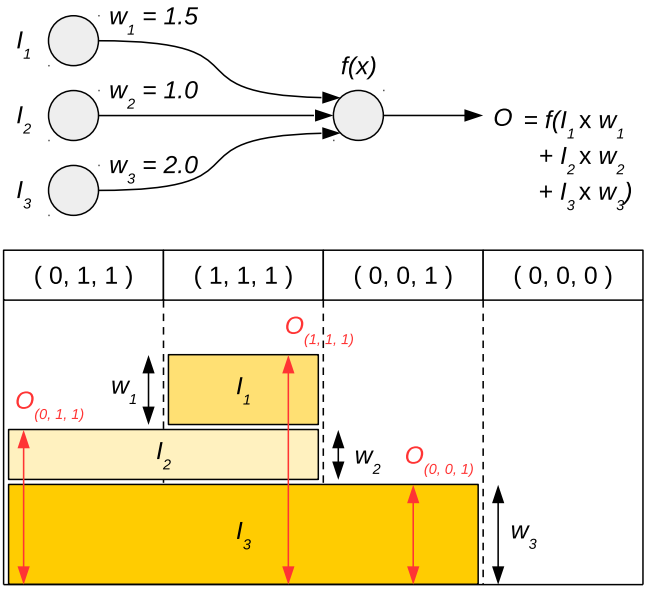


FIGURE 4.13 – Exemple de perceptron (en haut) avec 3 entrées, et sa représentation visuelle sous forme de boîtes arc-en-ciel pondérées, pour les 4 vecteurs d'entrée (I_1, I_2, I_3) suivants : (0, 1, 1), (1, 1, 1), (0, 0, 1) et (0, 0, 0) (en bas).

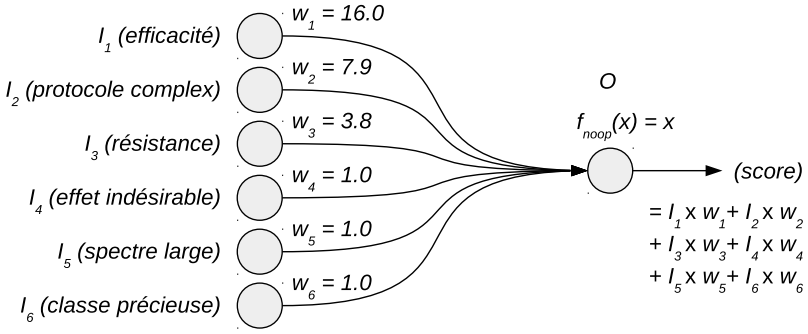


FIGURE 4.14 – Perceptron permettant d'associer à chaque antibiotique un score.

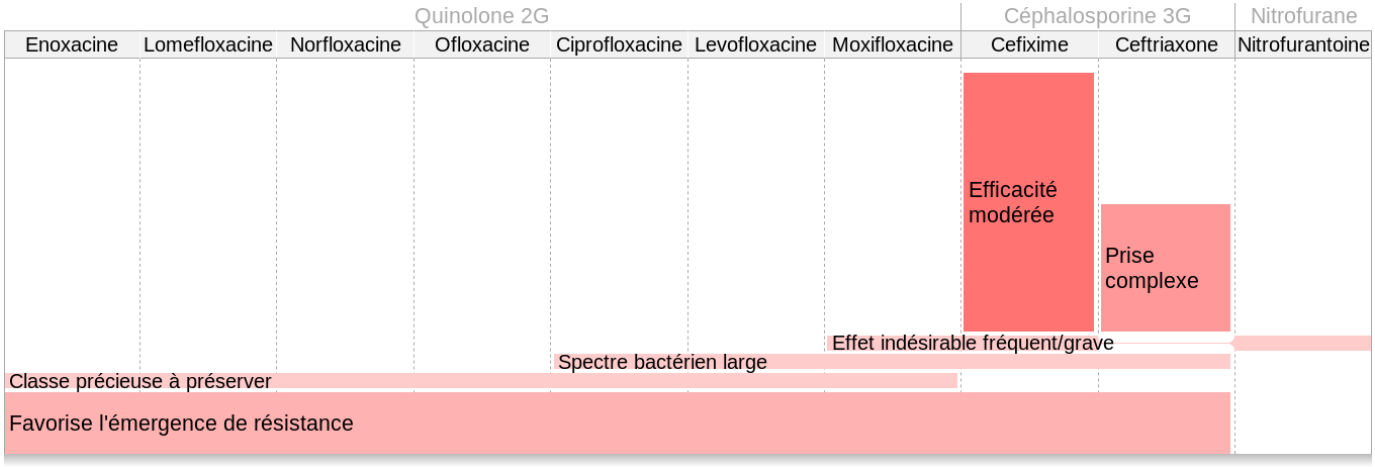


FIGURE 4.15 – Exemples de boîtes arc-en-ciel pondérées représentant les antibiotiques disponibles pour traiter les cystites de l'adulte à risques de complications, ainsi que les inconvénients de chaque médicament. Les 10 antibiotiques sont en colonnes, et les 6 inconvénients possibles sont représentés par des boîtes rectangulaires. Chaque boîte recouvre les colonnes correspondant aux antibiotiques qui partagent un même inconvénient.

4.2.8 Expérimentations utilisateur

Nous avons réalisé plusieurs expérimentations utilisateurs comparant les boîtes arc-en-ciel avec d'autres types de visualisation. Nous présenterons ici une comparaison du diagramme de Venn des acides aminés (Figure 4.5 en bas à droite) avec les boîtes arc-en-ciel équivalentes (Figure 4.6). L'étude a été conduite auprès des étudiants en licence et master de biologie de l'université Paris 13.

L'étude a été réalisée selon un plan d'étude croisé (*cross-over*). Deux séries de 5 questions portant sur les propriétés des acides aminés ont été préparées. Chaque étudiant devait répondre sur un site web à une première série de questions avec l'un des deux diagrammes (Venn ou boîtes arc-en-ciel), puis à la seconde série de questions avec l'autre diagramme. Le site web a permis de mesurer le taux d'erreur et le temps de réponse. De plus, une dernière question demandait à l'étudiant quel diagramme il a préféré. Les données ont été analysées avec le logiciel R version 3.3.2, en acceptant un risque α de 5%.

Nous avons recruté 78 étudiants (56 en L3 + 22 en M1), ce qui a conduit à 390 réponses (78×5) par diagrammes. Nous avons observé 38 erreurs avec le diagramme de Venn (taux d'erreur 9,7%) et 19 avec les boîtes arc-en-ciel (4,9%). **Les boîtes arc-en-ciel conduisent donc à deux fois moins d'erreurs que le diagramme de Venn** ; cette différence est significative ($p = 0,013$, test exact de Fisher). Le temps de réponse moyen était de 11,57 secondes avec le diagramme de Venn et de 9,63 s avec les boîtes arc-en-ciel. Cette différence est significative ($p < 10^{-6}$, test t de Student sur le log du temps de réponse). Pour la préférence, 19 étudiants ont préféré le diagramme de Venn et 51 les boîtes arc-en-ciel, les autres n'exprimant pas d'avis. La proportion est significativement différente de 50-50% ($p = 0,00013$, test du Chi-2).

Pour conclure, les boîtes arc-en-ciel sont meilleures que le diagramme de Venn en termes d'erreurs, de temps de réponse et de préférence utilisateur. Ces résultats sont en cours de publication. Nous avons également réalisé une évaluation entre boîtes arc-en-ciel et tableau sur l'application aux propriétés des médicaments [101, 99], qui a montré que les boîtes arc-en-ciel conduisent à une lecture plus rapide. Une évaluation plus globale du site web VIIIP intégrant les boîtes arc-en-ciel sera présentée en section 5.2.3.

4.2.9 Discussion

Dans cette section, nous avons décrit la conception des boîtes arc-en-ciel, une nouvelle technique de visualisation pour les ensembles non disjoints. Nous avons montré l'application de cette technique à l'exemple simple des acides aminés, à la comparaison visuelle des propriétés des médicaments, à la visualisation d'extraits de Gene Ontology et à la traduction visuelle du raisonnement d'un perceptron. Enfin, nous avons présenté une expérimentation utilisateur qui montre que cette technique conduit à de meilleurs résultats que le diagramme de Venn.

4.2.9.1 Boîtes arc-en-ciel

Les boîtes arc-en-ciel permettent de visualiser des ensembles non disjoints et de favoriser la découverte de similarités entre les éléments ou les ensembles.

Les boîtes arc-en-ciel partagent certaines caractéristiques avec le diagramme linéaire [157], inventé par Leibniz en 1686. Elles ont cependant l'avantage de représenter chaque ensemble par un seul objet visuel, ce qui n'est pas le cas du diagramme linéaire.

La visualisation d'ensembles non disjoints a de nombreuses applications possibles. Si l'on considère le problème d'un point de vue « ontologie », comme c'est le cas dans le projet VIIIP, les boîtes arc-en-ciel permettent de visualiser les relations d'instanciation entre instances et classes (*est-instance-de*) dans une ontologie [100], tout en restant lisibles lorsque ces relations sont nombreuses et complexes (notamment en cas d'instanciation multiple). En revanche, les relations de subsomption entre classes (*est-un*) ne sont pas représentées de manière explicite. Par exemple, il est facile de constater sur la Figure 4.6 que tous les acides aminés *aliphatiques* sont aussi *essentiels*, cependant les boîtes arc-en-ciel n'indiquent pas si ce fait est purement fortuit ou bien s'il existe une relation de subsomption entre la classe des acides aminés *aliphatiques* et celle des *essentiels*.

En l'état, les boîtes arc-en-ciel peuvent être utiles pour découvrir des similarités entre instances ou entre classes, et faire des hypothèses sur des relations de subsomption. Elles pourraient cependant être améliorées

en prenant en compte les relations de subsomption entre classes. L'héritage simple entre classes pourrait par exemple être représenté en incluant les boîtes des classes plus spécifiques dans les boîtes des classes plus générales (par exemple la boîte des acides aminés *minuscules* pourrait être incluse dans celle des *petits*). De même, la visualisation des disjonctions entre classes pourrait être facilitée si l'on plaçait les classes disjointes sur la même ligne. Les connaissances ontologiques pourraient ainsi intervenir davantage dans la visualisation.

Le nombre de colonnes présentes sur les boîtes arc-en-ciel est limité par la largeur de l'écran. Il est donc difficile d'envisager la visualisation de plus de 25 éléments actuellement. En revanche, les boîtes arc-en-ciel pourraient être couplées à un algorithme de regroupement (*clustering*), ce qui permettrait de rassembler les éléments proches au sein d'une même colonne et donc de visualiser un nombre d'éléments beaucoup plus grand. La largeur des colonnes pourrait alors être proportionnelle au nombre d'éléments qu'elles regroupent. De la sorte, il serait possible d'envisager les boîtes arc-en-ciel pour visualiser de gros volumes de données ouvertes (*open datasets* [124]).

4.2.9.2 Optimisation de l'ordre des colonnes

Pour l'optimisation de l'ordre des colonnes, nous avons testé d'autres approches qui n'ont pas conduit à des résultats aussi bons que l'heuristique présentée ici. Nous avons essayé l'*analyse en composante principale* (ACP) après avoir transformé le jeu de données en un espace à n dimensions (où n est le nombre d'ensembles) dans lequel chaque élément est un point dont les coordonnées sont 1 (si l'élément appartient à l'ensemble associé à la dimension) ou 0 (sinon). L'ACP permet ensuite de calculer l'axe permettant le mieux de séparer les points ; les éléments sont alors placés dans l'ordre où ils figurent sur cet axe. Nous avons aussi essayé des méthodes de construction d'arbre (*unweighted pair group method with arithmetic mean*, UPGMA, et *neighbor-joining*, NJ). Ces méthodes permettent de regrouper les éléments similaires de manière hiérarchique, cependant elles ne définissent pas un ordre à proprement parler, en particulier les fils de chaque nœud de l'arbre ne sont pas ordonnés. Il faut alors tester tous les ordres possibles satisfaisant l'arbre, et cela conduit à tester un nombre d'ordres bien plus grand que l'heuristique que nous avons proposée (pour les acides aminés, 524 288 pour UPGMA et 786 432 pour NJ). Les algorithmes de classification double (*biclustering*) pourraient aussi être envisagés, avec néanmoins le même problème : l'objectif est ici de trouver un ordre optimal et non pas un ensemble de classes (*clusters*).

L'utilisation des boîtes arc-en-ciel et de l'algorithme qui ordonne les colonnes à des fins de classification est une perspective envisageable. En effet, nous avons vu que nous retrouvons des classifications existantes sur plusieurs exemples (tels que les histones). Cette capacité à établir des classifications est un point qu'il serait intéressant d'évaluer plus rigoureusement.

4.2.9.3 Application à la comparaison des propriétés des médicaments

L'expérimentation utilisateur a montré que les médecins avaient une légère préférence pour les tableaux, mais qu'ils trouvaient l'information significativement plus rapidement avec les boîtes arc-en-ciel. Ces résultats ne sont pas contradictoires : en effet le score SUS mesure l'*utilisabilité perçue* mais celle-ci n'est pas nécessairement corrélée à l'utilisabilité réelle. Temps de réponse, justesse des réponses et satisfaction de l'utilisateur sont trois paramètres à mesurer lors des expérimentations utilisateurs et ceux-ci doivent être considérés *a priori* comme non corrélés [52].

Les difficultés rencontrées par les médecins avec les boîtes arc-en-ciel peuvent s'expliquer par le fait que cette visualisation est nouvelle pour eux, alors qu'ils sont habitués aux tableaux, et par l'âge élevé d'un certain nombre des évaluateurs (certains mentionnant eux-mêmes ce point dans les questionnaires). Les boîtes arc-en-ciel pourraient être utiles pour la formation initiale et continue, par exemple pour les enseignements de pharmacologie ou pour diffuser les connaissances sur les nouveaux médicaments.

Dans la littérature, les seules approches de visualisation utilisées pour la comparaison des propriétés de plusieurs médicaments sont des tableaux ou des matrices. Wroe *et al.* [191] ont proposé DOPAMINE, un outil à base de matrices, principalement à des fins de rédactions et de vérifications des propriétés dans les bases de données. Iordatii *et al.* [78] ont également proposé une approche à base de tableaux pour comparer les contre-indications et les effets indésirables d'un nouveau médicament à un médicament de référence. Les *Drug*

Fact Boxes [163] offrent une information comparative sur le médicament, mais limitée à un faible nombre d'entre eux et destinée aux patients plutôt qu'aux médecins.

4.2.9.4 Application à la visualisation d'annotations de gène

Nous avons également montré que les boîtes arc-en-ciel pouvaient être appliquées à d'autres domaines que la comparaison des médicaments. Pour cela, nous avons appliqué cette technique à la visualisation d'annotations de gène extraites de Gene Ontology. Les visualisations n'ont cependant pas encore fait l'objet d'une analyse détaillée par des biologistes. Par ailleurs, la visualisation pourrait être améliorée avec une meilleure prise en compte des relations entre termes d'annotation (subsomption et méronymie notamment).

Une autre perspective est la visualisation des résultats des analyses fonctionnelles issues des données de puces à ADN ou de NGS (*Next Generation Sequencing*, ou séquençage à très haut débit). Lors de ces études, l'objectif est de déterminer les gènes différentiellement exprimés entre deux conditions (par exemple patients sains et patients atteints d'une maladie donnée). L'analyse fonctionnelle consiste à regarder à quelles fonctions correspondent ces gènes différentiellement exprimés.

Dans la littérature, d'autres approches ont été proposées pour visualiser l'annotation des gènes. La plupart des approches existantes s'appuient sur des techniques à base de graphes [43, 70]. L'utilisation de *treemaps* a également été suggérée pour visualiser les annotations des gènes issues d'une analyse par puce à ADN [16].

4.2.9.5 Application à la visualisation du raisonnement d'un perceptron

Nous avons vu que les boîtes arc-en-ciel pondérées permettaient de reproduire visuellement le raisonnement d'un perceptron. Cela est intéressant car les réseaux de neurones ont pour principal inconvénient leur aspect « boîte noire », c'est-à-dire que le système fait de bonnes prédictions mais est incapable de les expliquer. Ce point est souvent bloquant pour l'utilisateur, en particulier lorsqu'il s'agit d'un médecin.

Les perspectives de ce travail incluent une évaluation plus poussée de l'application en antibiothérapie et l'extension à des réseaux de plusieurs neurones formels.

Chapitre 5

Applications médicales de la visualisation des connaissances

Cuisiniers, cuisiniers,
Faites-nous viandes cuites,
Nous ferons grand souper,
Mais où est Marguerite ?
Elle est dedans sa chambre,
Sur sa taille bien coupée
Elle met une robe dorée
À point pour le souper
(La blanche biche, d'après une chanson traditionnelle)

Dans ce chapitre, nous présenterons deux exemples de visualisation des connaissances appliquées au domaine médical. Nous décrirons les applications, mais aussi leurs évaluations, les protocoles utilisés et les résultats obtenus.

Le premier exemple concerne le projet L3IM et porte sur l'intégration du langage iconique VCM au sein de 4 nouvelles applications : le dossier patient en cabinet, le dossier patient hospitalier, les moteurs de recherche et les guides de bonnes pratiques cliniques. Cet exemple combine un langage iconique avec des techniques simples de visualisation.

Le second exemple concerne le projet VIIIP et la conception d'un site web pour comparer les propriétés des nouveaux médicaments à celles des médicaments existants. Cet exemple utilise une technique avancée de visualisation structurelle, les boîtes arc-en-ciel, et, dans une moindre mesure, le langage iconique VCM.

Encadrements associés à cette contribution :

1. Christian Simon, stagiaire de M2 Informatique BioMédicale (6 mois en 2010)

Publications associées à cette contribution :

1. **Lamy JB**, Duclos C, Hamek S, Beuscart-Zéphir MC, Kerdelhué G, Darmoni S, Favre M, Falcoff H, Simon C, Pereira S, Serrot E, Mitouard T, Hardouin E, Kergosien Y, Venot A. Towards iconic language for patient records, drug monographs, guidelines and medical search engines. Stud Health Technol Inform 2010 ;160 :156-60. Conférence MEDINFO 2010, Le Cap, Afrique du Sud
2. **Lamy JB**, Beuscart-Zéphir MC, Boog C, Darmoni S, Favre M, Guigue L, Hamek S, Hassler S, Kerdelhué G, Leroy N, Mitouard T, Pereira S, Simon C, Venot A, Duclos C. Langage iconique et interfaces interactives en médecine : application aux dossiers patients, guides de bonnes pratiques et moteurs de recherche médicaux. IRBM 2012 ;33(2) :129-136
3. Simon C, Hassler S, Beuscart-Zephir MC, Favre M, Venot A, Duclos C, **Lamy JB**. Using an iconic language to improve access to electronic medical records in general medicine. Stud Health Technol Inform 2014 ;205 :333-7. Conférence Medical Informatics Europe MIE 2014, Istanbul, Turquie
4. Pereira S, Hassler S, Hamek S, Boog C, Leroy N, Beuscart-Zéphir MC, Favre M, Venot A, Duclos C, **Lamy JB**. Improving access to clinical practice guidelines with an interactive graphical interface using an iconic language. **BMC medical informatics and decision making** 2014 ;14(1) :77
5. Griffon N, Kerdelhué G, Hamek S, Hassler S, Boog C, **Lamy JB**, Duclos C, Venot A, Darmoni SJ. Design and usability study of an iconic user interface to ease information retrieval of medical guidelines. **J Am Med Inform Assoc** 2014 ;21(e2) :e270-7

5.1 Contribution : Nouvelles applications intégrant VCM

Dans cette section, nous montrerons comment les icônes VCM, initialement développées pour les connaissances sur le médicament et les bases médicaments, ont été intégrées à 4 autres types de logiciels médicaux : le dossier patient en cabinet, le dossier patient hospitalier, les moteurs de recherche et les guides de bonnes pratiques cliniques. Ces travaux ont été réalisés lors du projet ANR L3IM.

Nous décrirons tout d'abord la méthode générale suivie pour intégrer VCM et pour évaluer l'apport des icônes dans les applications. Ensuite, nous verrons chacune des 4 applications en détail.

5.1.1 Méthode générale pour intégrer VCM dans les applications médicales

L'intégration de VCM dans les différentes applications s'est faite en plusieurs étapes, selon une méthode de conception centrée utilisateur [79] prenant en compte les facteurs humains [20] :

1. **Établir des correspondances** entre VCM et les terminologies médicales utilisées par l'application.
2. **Réaliser des focus groupes** avec des médecins utilisateurs du logiciel concerné, pour recueillir leur avis et leurs besoins.
3. **Concevoir des maquettes** d'interfaces incluant VCM dans l'application. Les maquettes sont généralement faites dans un logiciel de dessin, en retouchant des copies d'écran du logiciel.
4. **Réaliser un prototype** fonctionnel à partir des maquettes les plus prometteuses.
5. **Réaliser une évaluation ergonomique** du prototype par des ergonomes spécialistes.
6. **Corriger le prototype** en fonction des problèmes identifiés lors de l'évaluation ergonomique.
7. **Réaliser une évaluation de satisfaction et de performance** auprès des médecins utilisateurs. Un groupe de médecins a été recruté. Les évaluations comprenaient 3 volets :
   (a) **Une évaluation objective**, consistant à mesurer les performances des médecins avec les logiciels incluant VCM et à les comparer aux performances obtenues sans VCM. Cette évaluation s'appuyait sur plusieurs scénarios ou cas cliniques, chaque médecin ayant la moitié des scénarios à traiter avec un logiciel intégrant VCM et l'autre moitié sans VCM, selon un plan d'étude croisé (*cross-over*). Les performances mesurées incluaient à la fois un temps de réponse et une mesure de la qualité du travail effectué par le médecin avec le logiciel (variable selon les applications, voir détails ci-dessous).

(b) **Une évaluation subjective**, consistant à mesurer la satisfaction des utilisateurs et l'utilisabilité perçue à l'aide du questionnaire SUS (*System Usability Scale*) standard [26].

(c) **Une discussion générale avec les évaluateurs**.

Les fonctionnalités sémantiques développées autour du langage VCM ont largement été utilisées pour faciliter son intégration. Notamment, le lexique des pictogrammes généré à partir de l'ontologie (voir section 3.3.3) a été mis à disposition des experts chargés d'établir les correspondances entre VCM et les terminologies médicales. L'ontologie des icônes VCM a ensuite permis de valider les icônes créées par les experts dans la correspondance (voir section 3.3.3.2).

5.1.2 Intégration dans un dossier patient de cabinet : éO Médecin

éO Médecin est un logiciel de gestion de cabinet de médecine (générale ou de spécialité) édité par Silk Informatique. Le logiciel gère les aspects administratifs et financiers, mais aussi les dossiers informatisés des patients. Le dossier patient d'éO suit un modèle « orienté problème », c'est-à-dire que le médecin peut classer les éléments d'un dossier par pathologie. Ce principe facilite la gestion des dossiers complexes de patients ayant plusieurs pathologies chroniques (par exemple hypertension artérielle + diabète de type 2).

L'intégration de VCM à éO Médecin a été réalisée par Christian Simon, dans le cadre de son stage de M2 Informatique BioMédicale. Christian est par ailleurs le principal développeur d'éO.

Dans éO Médecin, les pathologies peuvent être codées selon deux terminologies de référence : la CIM10 et la CISP2 (Classification Internationale des Soins Primaires, version 2). L'intégration de VCM dans éO Médecin a donc nécessité d'établir des correspondances entre VCM et les terminologies CIM10 et CISP2.

Les icônes VCM ont été intégrées à de nombreux endroits dans le logiciel éO Médecin :

— **Dans la liste des consultations** du patient (voir Figure 5.1). Chaque consultation reçoit une icône VCM en fonction de la pathologie qui lui a été associée. De plus, l'ensemble des différentes icônes figure en haut à droite de la fenêtre et permet de filtrer la liste des consultations pour n'afficher que les consultations associées à l'icône choisie.

— **Dans la liste des antécédents** du patient (voir Figure 5.2). Les icônes correspondant aux antécédents du patient sont rassemblées et visualisées sur un « Monsieur VCM ». De plus, un axe chronologique permet de « remonter le temps » pour visualiser les pathologies du patient à une date passée.

— **Dans la liste des comptes-rendus** du patient. Les rédacteurs des comptes-rendus sont identifiés par des icônes VCM représentant les grandes spécialités médicales (cardiologie, néphrologie,...).

— **Dans la liste des médicaments prescrits** pour un patient. L'ordonnance est visualisée sous la forme d'un « Monsieur VCM », les icônes représentant les médicaments prescrits par leur indication.

— **Dans le tableau de bord des pathologies suivies**. Le tableau de bord se présente sous la forme d'une boîte à onglet, avec un onglet par pathologie suivie. Une icône VCM a été ajoutée à chaque onglet, pour identifier la pathologie.

Pour l'évaluation, nous avons recruté 20 médecins généralistes utilisateurs du logiciel éO Médecin au quotidien. Deux dossiers patients ont été extraits, anonymisés et codés proprement. Les évaluateurs devaient consulter chacun de ces dossiers, soit avec VCM soit sans, et identifier les éléments d'information qu'ils jugeaient importants pour la prise en charge du patient. Le temps de consultation du dossier a été mesuré, et les informations relevées ont été comparées à un *gold standard* réalisé par deux médecins travaillant sur le projet. Chaque évaluateur avait un dossier avec VCM et un sans.

Les résultats de l'évaluation montrent que le nombre d'éléments d'information pertinents extraits avec VCM est légèrement plus faible que sans (6,65 contre 8,0 en moyenne) mais la différence n'est pas significative (test de Wilcoxon). En ce qui concerne le temps de consultation du dossier, les résultats sont différents sur les deux dossiers patients : sur le premier dossier, la consultation est plus rapide avec VCM (613 secondes en moyenne sans VCM contre 457 avec, p=0,25, différence non significative), sur le second dossier c'est le contraire (440 secondes en moyenne sans VCM contre 676 avec, p=0,0312, différence significative).

Le test SUS a donné un score moyen de 65, ce qui est un résultat plutôt bon. Lors de la discussion finale, les médecins nous ont dit qu'ils avaient particulièrement apprécié la vue d'ensemble du patient offerte par VCM, ainsi que la possibilité de mettre en évidence les informations-clefs en leur associant une icône.

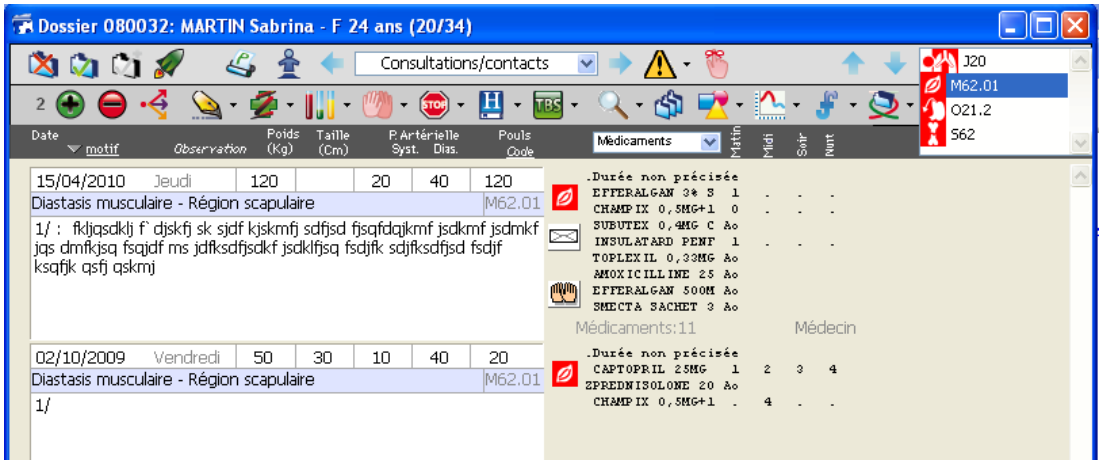


FIGURE 5.1 – Copie d'écran de la fenêtre listant les consultations d'éO Médecin, avec intégration d'une icône VCM pour chaque consultation.

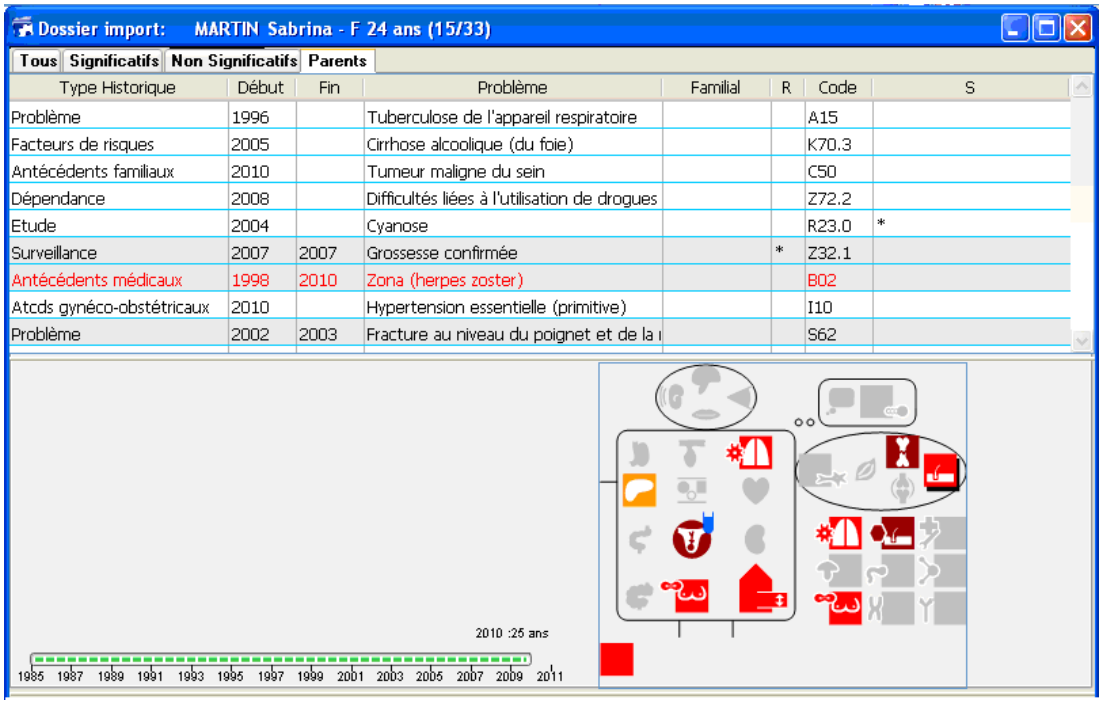


FIGURE 5.2 – Copie d'écran de la fenêtre des antécédents d'éO Médecin, avec intégration d'un « Monsieur VCM » et d'un axe chronologique.

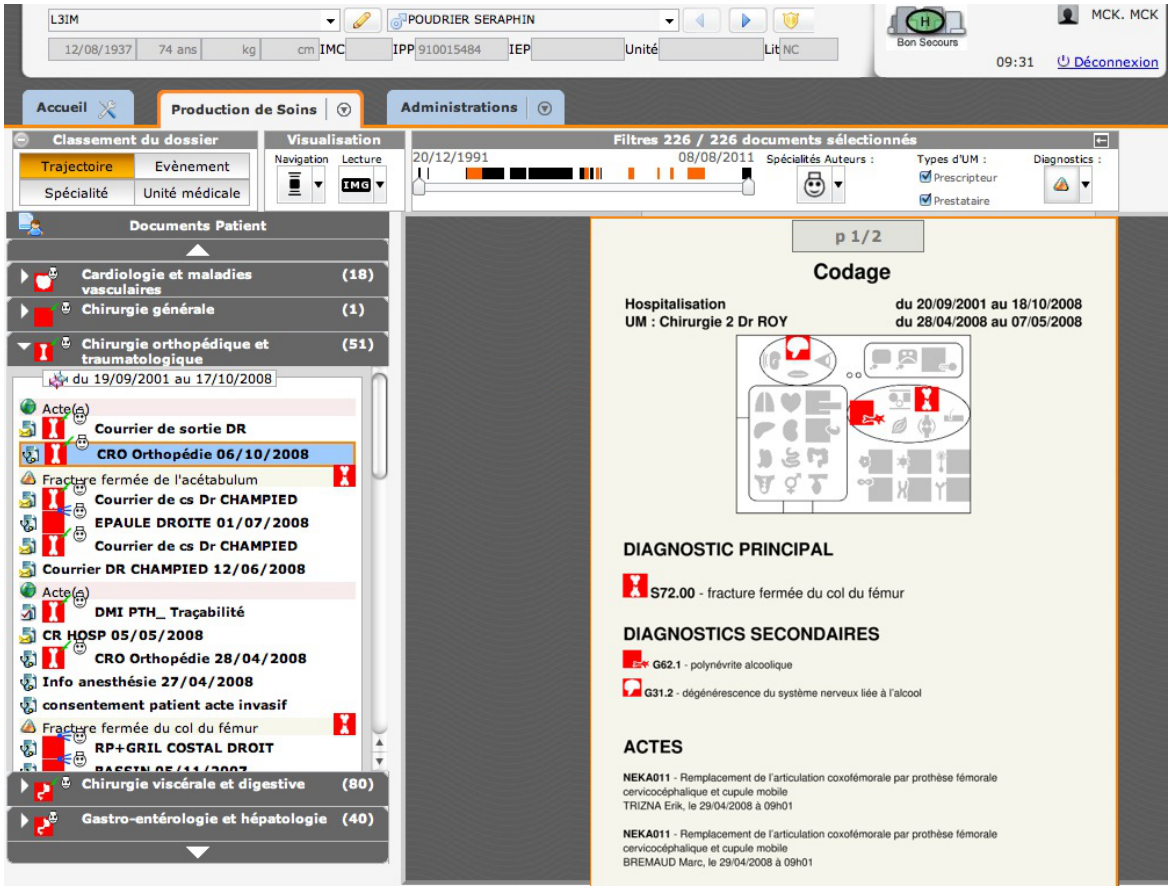


FIGURE 5.3 – Copie d'écran du logiciel Crossway Hôpital, avec l'intégration des icônes VCM et d'un « Monsieur VCM ».

5.1.3 Intégration dans un dossier patient hospitalier : Crossway Hôpital

Crossway Hôpital est un logiciel qui permet la gestion des dossiers patients à l'hôpital. Il était édité par McKesson France à l'époque du projet L3IM (devenu par la suite MainCare). Le dossier patient hospitalier se présente principalement comme une liste de documents : lettres d'admission, comptes-rendus d'opération ou de consultation, courriers de sortie... Il contient aussi le codage PMSI (Programme de Modernisation du Système d'Information) qui s'appuie sur la terminologie CIM10 pour coder le diagnostic principal et les éventuels diagnostics secondaires. Ce codage sert notamment à la Tarification à l'Activité (T2A) : en fonction des codes de diagnostics et des actes réalisés, l'hôpital recevra plus ou moins d'argent pour la prise en charge du patient.

L'intégration de VCM dans éO Médecin a nécessité d'établir une correspondance entre VCM et la terminologie CIM10.

Les icônes VCM ont été intégrées à deux endroits dans le logiciel Crossway Hôpital :

— **Dans la liste des documents** (voir Figure 5.3 à gauche). Les documents sont classés par service (cardiologie, chirurgie orthopédique,...). Chaque service est associé à une icône VCM. Chaque document peut aussi recevoir une icône, lorsque son contenu porte sur une pathologie ou un acte clairement identifié.

— **Dans le codage PMSI** (voir Figure 5.3 à droite). Chaque code CIM10 est associé à une icône dans la liste des diagnostics. De plus, un « Monsieur VCM » rassemble l'ensemble de ces icônes sur un bonhomme anatomique synthétique.

Pour l'évaluation, nous avons recruté 20 médecins de l'hôpital de Rouen, utilisateurs du logiciel Crossway Hôpital au quotidien. Deux dossiers patients ont été extraits, anonymisés et codés proprement. Les évaluateurs devaient consulter chacun de ces dossiers, soit avec VCM soit sans, et identifier les éléments d'information qu'ils jugeaient importants pour la prise en charge du patient. Le temps de consultation du dossier a été

mesuré, et les informations relevées ont été comparées à un *gold standard* réalisé par deux médecins travaillant sur le projet. Chaque évaluateur avait un dossier avec VCM et un sans.

Les résultats de l'évaluation montrent que le nombre d'éléments d'information pertinents extraits avec VCM est légèrement plus important que sans (6,8 contre 5,6 en moyenne) ; la différence est significative sur le second dossier (p = 0,0274, test de Wilcoxon). Le temps de navigation dans l'interface augmente légèrement avec les icônes VCM (191.85 secondes en moyenne contre 164.6, différence non significative). Cette tendance à l'augmentation du temps de navigation pourrait cependant être liée soit au fait que le médecin relève un nombre plus élevé d'informations avec VCM, soit au fait que le médecin ait voulu « jouer » avec l'interface et les icônes.

Le test SUS a donné un score moyen de 57, ce qui est un résultat moyen. Lors de la discussion finale, les médecins nous ont dit qu'ils avaient besoin de plus de temps pour se familiariser avec les nouvelles interfaces. En effet, l'intégration de VCM s'est faite dans la nouvelle version de Crossway Hôpital alors que les médecins étaient habitués à l'ancienne interface. Les évaluateurs découvraient donc à la fois la nouvelle interface et les icônes. De plus, les icônes VCM sont aussi mélangées dans l'interface à des icônes « maison » de Crossway Hôpital, se qui diminue la lisibilité (par exemple les items de la liste de documents de la Figure 5.3 ont deux icônes). Enfin, le « Monsieur VCM » n'était pas interactif, alors que les médecins auraient aimé pouvoir cliquer sur les icônes pour extraire les documents correspondants.

5.1.4 Intégration dans un moteur de recherche médical : CISMeF

CISMeF (Catalogue et Index des Sites Médicaux Francophones, http://www.cismef.org/) est un moteur de recherche dédié aux sites médicaux. Le moteur de recherche fonctionne à partir d'une saisie de mots-clefs textuels, et affiche une liste de résultats. Le moteur intègre de nombreuses fonctionnalités lexicales et sémantiques, telles que la correction orthographique ou l'expansion de requêtes, et l'indexation des ressources s'appuie sur une traduction française de la terminologie MeSH (*Medical Subject Heading*) utilisée par PubMed. L'intégration de VCM dans éO Médecin a donc nécessité d'établir une correspondance entre VCM et la terminologie MeSH.

Nous avons travaillé sur la partie « bonnes pratiques » (CISMeF BP) du site, qui recherche parmi les guides de bonnes pratiques cliniques ; ce trvail a été réalisé en collaboration avec l'équipe CISMeF de Rouen. Les icônes VCM ont été intégrées :

- **Dans la liste des résultats** (voir Figure 5.4). À chaque résultat de recherche, une ou plusieurs icônes VCM ont été associées en fonction des termes MeSH qui indexent le document.
- **Dans la zone de filtre** (voir Figure 5.5). Une zone de filtre a été ajoutée, elle permet de filtrer les résultats de la recherche selon divers termes MeSH organisés en 3 colonnes : les termes correspondant aux mots-clefs saisis par l'utilisateur, les termes correspondant à des maladies autres mais souvent associées dans les résultats, et les termes correspondant à des terrains (âge, grossesse,...). Chaque terme MeSH du filtre est accompagné d'une icône VCM.

Pour l'évaluation, nous avons recruté 20 médecins (10 généralistes membres de la Société de Formation Thérapeutique du Généraliste, SFTG + 10 hospitaliers de l'hôpital de Rouen). Quatre scénarios ont été réalisés, chaque scénario correspondant à une recherche dans CISMeF BP : dyslipidémie, asthme, infections urinaires et antibiotique pour traiter une pneumonie. Pour chaque scénario, le médecin devait trouver une ressource cible identifiée par deux médecins travaillant sur le projet. Le temps de navigation a été mesuré, ainsi que l'obtention ou non de la ressource cible.

Les résultats de l'évaluation montrent que la ressource cible est trouvée dans 97,5% des cas (1 échec sur 40 recherches) avec VCM et dans 77,5% des cas sans VCM (9 échecs sur 40 recherches). La différence est significative (p = 0,01, test de Fisher exact). En revanche, le temps de navigation est légèrement plus long avec VCM (99 secondes en moyenne contre 66, différence significative : p = 0,04, ANOVA).

Le test SUS a donné un score moyen de 74, ce qui est un très bon résultat. Lors de la discussion finale, les médecins nous ont dit que le filtre iconique était trop complexe. Les 3 colonnes ne sont pas clairement identifiées par un titre, et elles comprennent de nombreuses icônes proches qu'il est difficile de distinguer. Ce problème sur le filtre pourrait expliquer l'augmentation du temps de navigation avec VCM.

Pour davantage d'information, se référer aux publications [59, 60].



FIGURE 5.4 – Copie d'écran du moteur de recherche CISMeF, avec intégration d'icônes VCM dans la liste des résultats.

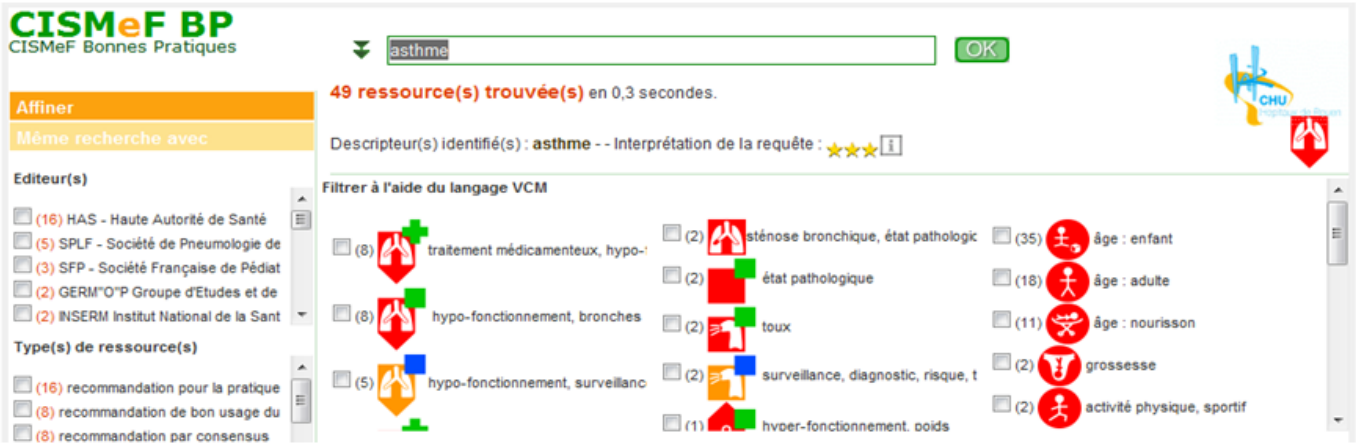


FIGURE 5.5 – Copie d'écran du moteur de recherche CISMeF, montrant le filtre iconique.

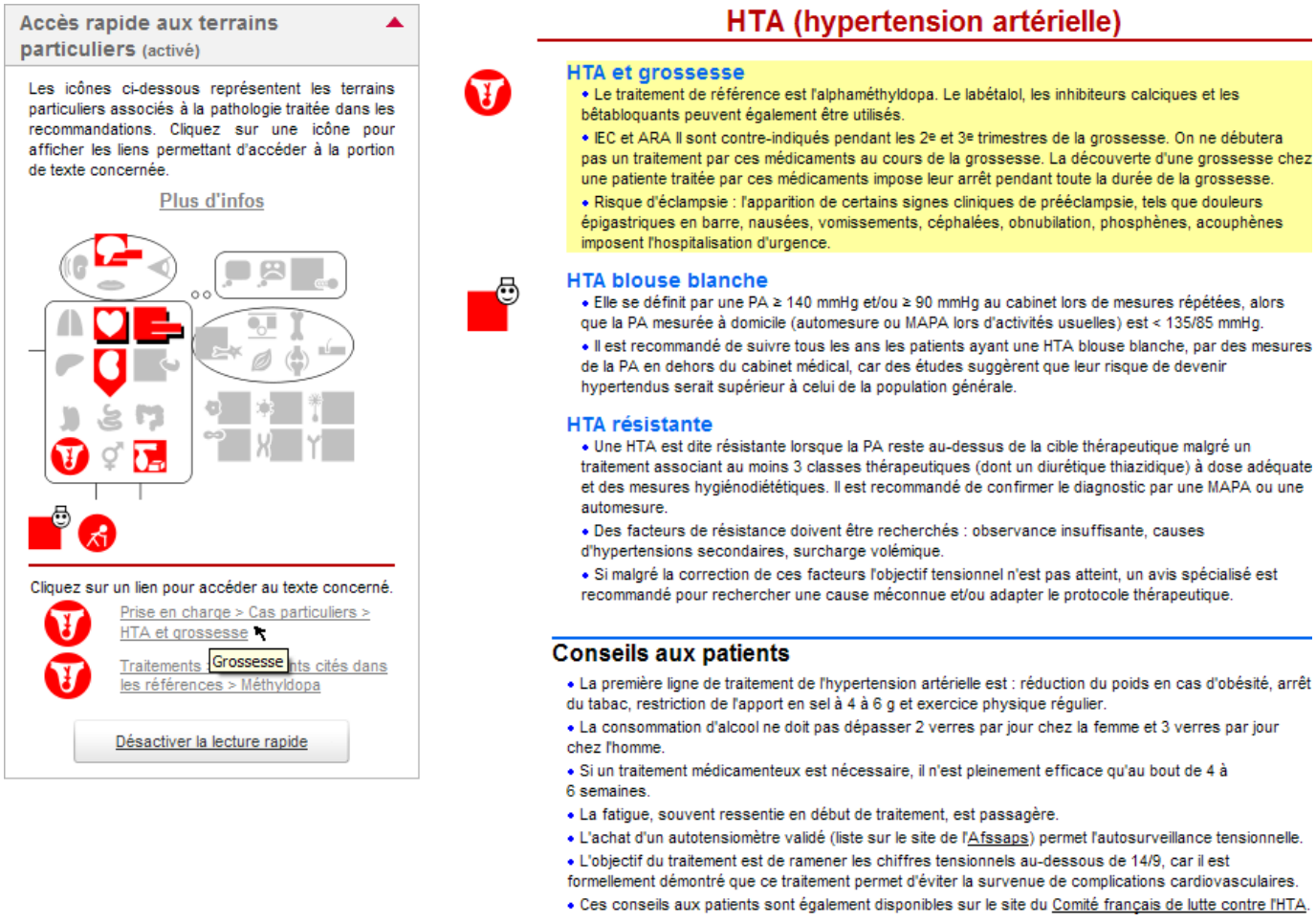


FIGURE 5.6 – Copie d'écran de Vidal Reco avec intégration des icônes VCM.

5.1.5 Intégration dans les guides de bonnes pratiques : Vidal reco

Vidal Reco est un produit de la société Vidal (partenaire du projet L3IM). Vidal Reco rassemble 165 guides de bonnes pratiques cliniques pour différentes maladies. Les guides sont tous rédigés selon un plan identique et structuré permettant de trouver rapidement l'information. Ils incluent aussi des arbres de décision sous forme de schéma. Cette compilation de guide est disponible au format papier sous forme de livre, ou au format électronique dans un site web. C'est sur ce site web que nous avons travaillé. Les icônes ont été intégrées à deux niveaux (Figure 5.6) :

— **Des icônes ont été placées en marge du texte** du guide, sur la gauche. Ces icônes représentent les différents cas particuliers traités dans le guide. Ces cas particuliers sont généralement des comorbidités (c'est-à-dire des maladies que présente le patient en plus de celle à laquelle on s'intéresse, par exemple traitement de l'hypertension chez un patient diabétique) ou des caractéristiques patient telles que l'âge ou la grossesse.

— **L'ensemble des icônes présentes dans le texte ont été regroupées et rassemblées sur un « Monsieur VCM »**, placé à gauche du texte. Ce « Monsieur VCM » est interactif et permet, en cliquant sur une icône, d'afficher en dessous la liste des paragraphes évoquant le cas particulier. Un second clic dans la liste des paragraphes permet d'aller directement au texte.

Pour l'évaluation, nous avons recruté 20 médecins (10 généralistes membres de la Société de Formation Thérapeutique du Généraliste, SFTG + 10 hospitaliers de l'hôpital de Rouen). Deux scénarios médicaux ont été rédigés, portant sur le guide pour l'hypertension et celui de la colite néphrétique. Chaque médecin a recherché le traitement recommandé pour chaque scénario dans Vidal Reco, soit avec VCM soit sans. Nous avons mesuré le temps de réponse et la justesse de la réponse, par comparaison à un *gold standard* issu du guide.

Les résultats sont les suivants : 2 erreurs ont été constatées sans VCM contre zéro avec (différence non significative). Le temps de réponse moyen est de 45,6 secondes sans VCM et de 25,2 secondes avec ; cette différence est significative (p = 0,04, ANOVA après transformation logarithmique). Le temps de réponse est 1,8 fois inférieur avec VCM.

Le test SUS a donné un score moyen de 75, ce qui est un très bon résultat. Lors de la discussion finale, les médecins nous ont dit qu'ils avaient apprécié VCM et en particulier le « Monsieur VCM », cependant ils auraient aimé que celui-ci soit davantage visible.

5.1.6 Discussion

Dans cette section, nous avons décrit l'intégration des icônes VCM à 4 nouvelles applications médicales : le dossier patient en cabinet, le dossier patient hospitalier, les moteurs de recherche et les guides de bonnes pratiques cliniques.

Chaque application a fait l'objet d'une évaluation auprès des utilisateurs, avec un volet objectif où nous avons mesuré les performances obtenues par les utilisateurs avec et sans VCM (temps de réponse et qualité des réponses), et un volet subjectif où nous avons demandé l'avis des utilisateurs. Ce protocole d'évaluation s'est révélé bien adapté et a produit de nombreux résultats. Cependant, il aurait été préférable d'avoir une plus grande puissance statistique, en augmentant soit le nombre d'évaluateurs soit le nombre de scénarios.

L'intégration de VCM dans les applications relatives aux connaissances médicales (moteur de recherche et guides de bonnes pratiques) a globalement donné de bons résultats, avec des scores SUS élevés (74 et 75) et des gains significatifs de performances (moins d'erreurs dans le moteur de recherche, gain de temps dans les guides de bonnes pratiques). Ces résultats sont similaires à ceux obtenus sur les connaissances sur le médicament lors de ma thèse [118].

En revanche, l'intégration de VCM dans les dossiers patients (cabinet et hôpital) a donné des résultats mitigés, que ce soit en termes de scores SUS (65 et 57) ou de performances. Plusieurs raisons peuvent expliquer ces résultats :

- Tout d'abord, les dossiers patients sont plus éloignés de l'application originelle de VCM que les deux autres applications. L'adaptation de VCM au dossier patient était donc plus difficile, et aussi plus risquée. En particulier, les concepts médicaux mentionnés dans les connaissances médicales (guides de bonnes pratiques par exemple) sont souvent exprimés à un niveau de granularité élevé (par exemple diabète) alors que, dans les dossiers patients, le niveau de granularité est beaucoup plus fin (par exemple diabète de type 2 de découverte tardive). Or VCM représente seulement des niveaux de granularité élevés.
- Les icônes VCM affichées dépendent des correspondances réalisées entre VCM et les terminologies médicales. L'établissement des correspondances manuelles entre VCM et les terminologies des dossiers patients (CIM10 et CISP2) s'est avéré délicat, à cause du nombre important de termes (plus de 10 000 pour la CIM10). Certaines des difficultés rencontrées par les médecins sont liées à des problèmes dans les correspondances.
- De plus, dans le dossier hospitalier, le codage CIM10 effectué pour le PMSI est réalisé à des fins médico-économiques, et non à des fins d'utilisation en clinique. Certains choix de codage se justifient donc par des raisons économiques : par exemple un diagnostic secondaire pourra être omis si ce code n'a pas d'impact sur la T2A ; ce qui ne signifie pas qu'il n'a pas d'intérêt en clinique mais dans ce cas, l'icône sera manquante dans l'interface, et le médecin ne comprendra pas pourquoi.
- Les interfaces des deux logiciels de dossier patient étaient déjà très complexes sans les icônes VCM : l'interface d'éO est réputée riche mais compliquée, et l'interface de Crossway Hôpital avait été entièrement modifiée, les utilisateurs devaient donc s'approprier le changement d'interface en plus de l'ajout des icônes. Au contraire, les interfaces de Vidal Reco et de CISMeF BP sont beaucoup plus simples.

Dans la littérature, plusieurs travaux ont porté sur l'utilisation d'icônes pour des applications médicales. La plupart des travaux utilisent des icônes pour transmettre au patient des recommandations sur la prise d'un traitement médicamenteux (posologie, moment de prise, méthode d'administration pour les dispositifs complexes,...) [1, 197, 74, 75, 40, 170, 39].

Stabilis 4 [185] est base de données sur la stabilité et la compatibilité des médicaments injectables, consultable en ligne (http://www.stabilis.org). Stabilis utilise un jeu d'icônes pour indiquer les classes thérapeutiques des médicaments, les conditions de conservations, les stabilités dans différentes solutions, les incompatibilités,... L'utilisation de pictogrammes permet notamment de s'affranchir de la langue, alors que les sources de connaissance (littérature médicale et livres de référence spécialisés) sont quasi exclusivement en anglais. La base est ainsi disponible dans 29 langues différentes.

UVAL-MED [150, 152, 151] est un langage graphique permettant de décrire à l'aide de schémas les principales caractéristiques des pathologies, et notamment les symptômes, afin de faciliter le diagnostic. Il comprend un ensemble de pictogrammes de base ainsi qu'une grammaire graphique simple permettant de combiner ces pictogrammes. L'objectif est de faciliter l'apprentissage de la physiopathologie par des étudiants en médecine.

Des icônes représentant des éléments du dossier patient ont aussi été intégrées dans un système d'aide à la décision clinique [141]. Les icônes sont construites selon un langage iconique et indiquent pour chaque événement dans le dossier le type d'événement (par exemple acte, prescription de médicament, rechute,...), la spécialité médicale concernée (cardiologie, pneumologie,...). Les icônes incluent des étiquettes textuelles pour préciser la date et des informations chiffrées. D'autres chercheurs ont aussi mis en lumière l'importance de la visualisation des informations du dossier patient, sans aller jusqu'à envisager des icônes [179].

Enfin, l'utilisation de pictogrammes a été proposée pour présenter l'Indice de Masse Corporelle (IMC) des patients [88].

Encadrements associés à cette contribution :

1. Lilia Kraiem, stage de M1 et de M2 informatique biomédicale (2 mois en 2013 + 6 mois en 2014)
2. Mouhssin Gaouar, stage de M1 informatique biomédicale (2 mois en 2015)
3. Camelia Marnia, stage de M1 informatique biomédicale (2 mois en 2015)

Publications associées à cette contribution :

1. Ugon A, Berthelot H, Venot A, Favre M, Duclos C, **Lamy JB**. Abstractive Summarization of Drug Dosage Regimens for Supporting Drug Comparison. Stud Health Technol Inform 2015 ;213 :79-82. Conférence Informatics, Management, and Technology in Healthcare IMTH 2015, Athène, Grèce
2. **Lamy JB**, Ugon A, Berthelot H, Favre M. A visual decision support system for helping physicians to make a decision on new drugs, Decision Support Systems VII. Data, information and knowledge visualization in decision support systems, Lecture Notes in Business Information Processing, International Conference on Decision Support System Technology ICDSST 2017, Namur, Belgique
3. **Lamy JB**, Berthelot H, Favre M, Ugon A, Duclos C, Venot A. Using visual analytics for presenting comparative information on new drugs. **Journal of Biomedical Informatics** 2017 ;71 :58-69

5.2 Contribution : Le site VIIIP pour la comparaison des médicaments

Dans cette section, nous décrirons la plate-forme et le site web issus du projet VIIIP, ainsi que l'évaluation du site web réalisée auprès de médecins généralistes.

L'objectif principal du site web VIIIP est d'aider le médecin à prendre une décision concernant la place d'un nouveau médicament dans l'arsenal thérapeutique : le médecin peut-il envisager de le prescrire ? en première intention ou en seconde ? pour n'importe quel patient ou seulement pour certains patients selon leur âge, leur sexe ou leurs pathologies ? Les objectifs secondaires sont d'aider à la mémorisation des propriétés des médicaments et à la découverte de similarités au sein d'un groupe de médicaments.

Pour atteindre ces objectifs, le site ne s'appuie pas sur des avis d'experts comme les revues et les sites existants, mais sur une information comparative et visuelle. Le site propose la présentation visuelle des propriétés du nouveau médicament, et la comparaison de ces propriétés à celles des autres médicaments déjà existants dans la même indication.

5.2.1 Prototype

Un prototype de la plate-forme VIIIP a été implémenté en langage Python. Il s'appuie sur :

— **L'ontologie comparative des médicaments** (voir section 2.2.3).
— **Le module Owlready pour la programmation orientée ontologie** (voir section 2.3), pour accéder à l'ontologie et pour générer automatiquement des boîtes de dialogue pour la saisie des données.
— **La technique de visualisation des boîtes arc-en-ciel** (voir section 4.2) pour permettre la comparaison visuelle des propriétés cliniques des médicaments.
— **Les icônes VCM** (voir section 3.1.2.2) pour visualiser les contre-indications.

Le prototype génère à partir de l'ontologie des pages HTML incluant des feuilles de styles CSS et des scripts JavaScript. Chaque page présente un nouveau médicament dans une indication donnée, et le compare à un ou plusieurs comparateurs, c'est-à-dire des médicaments déjà existants qui ont la même indication. Lorsqu'un nouveau médicament possède plusieurs indications distinctes (cas du Ciloxan), une page différente est proposée pour chaque indication, avec éventuellement des comparateurs différents selon l'indication. Des liens hypertextes permettent de passer d'une indication à une autre.

Les données correspondant à 4 nouveaux médicaments et leurs comparateurs ont été extraites manuellement par Hélène Berthelot, pharmacienne experte dans les bases médicaments. Les quatre nouveaux médicaments sont les suivants :

— **Vitaros® 300 µg** (alprostadil, crème) :

— Indication : dysfonction érectile, 7 comparateurs (Edex®, Caverject®, Muse®, Viagra®, Cialis®, Levitra®, Spedra®).
— **Pyléra® 140 mg/125 mg/125 mg** (sous-nitrate de bismuth potassique + métronidazole + chlorhydrate de tétracycline, gélule) :
— Indication : éradication de Helicobacter pylori et prévention des récidives d'ulcères gastro-duodénaux, 3 comparateurs (clarithromycine + amoxicilline, clarithromycine + métronidazole, amoxicilline + métronidazole).
— **Antarène Codéine® 400 mg/60 mg** (ibuprofène + codéine, comprimé pelliculé) :
— Indication : douleur, 4 comparateurs (Dafalgan Codéine®, Izalgi®, Lamaline®, Ixprim®).
— **Ciloxan® 3 mg/ml** (ciprofloxacine, solution pour instillation auriculaire) :
— Indication n°1 : otites externes aiguës, 6 comparateurs (Ciloxadex®, Antibio Synalar®, Framyxone®, Polydexa®, Panotile®, Auricularum®).
— Indication n°2 : otorrhées purulentes sur cavité d'évidement et sur otites chroniques non ostéitiques à tympan ouvert, 2 comparateurs (Oflocet®, Otofa®).

5.2.2 Présentation du site web

Chaque page du site web présente un nouveau médicament dans une indication donnée. Les pages sont découpées en 8 parties (voir exemple Figure 5.7) :

1. **La zone de titre**, avec le nom du médicament, son ou ses principes actifs et son type d'innovation.
2. **La synthèse** des propriétés du nouveau médicament (information non-comparative) et la liste des comparateurs.
3. Un tableau comparatif (incluant le nouveau médicament et les comparateurs) comportant :
   — **Les informations cliniques ponctuelles** : le terrain (adulte, enfant,...), la posologie, le SMR (Service Médical Rendu) et l'ASMR (Amélioration du Service Médical Rendu), le délai et la durée d'action, le niveau de précaution d'emploi par rapport à la conduite.
   — **Les informations économiques** : le coût et le taux de remboursement.
4. **Les principaux résultats des essais cliniques** portant sur le nouveau médicament, sous la forme de diagrammes en bâtons.
5. **La comparaison des contre-indications** du nouveau médicament avec les comparateurs.
6. **La comparaison des interactions** du nouveau médicament avec les comparateurs.
7. **La comparaison des effets indésirables** du nouveau médicament avec les comparateurs. Seuls les effets graves et/ou fréquents sont affichés ; les effets non graves et peu fréquents, rares ou très rares ne sont pas affichés (ceci correspond à une demande de la part des médecins, afin de réduire le volume d'effets indésirables).
8. **La comparaison des excipients à effet notoire** du nouveau médicament avec les comparateurs.
9. Un tableau comparatif comportant :
   — **Les informations administratives** : dénomination commune, dosage, forme, présentation, nom du laboratoire, date d'AMM (Autorisation de Mise sur le Marché).
   — **Les liens vers les documents officiels** : RCP, notice, ACT.

Deux outils différents sont proposés pour comparer les propriétés cliniques du médicament (contre-indications, interactions et effets indésirables, parties 5, 6 et 7) : les *tableaux dynamiques* et les *boîtes arc-en-ciel*.

Les tableaux dynamiques affichent les médicaments en colonne et les propriétés en ligne. Pour les contre-indications et les interactions, 3 symboles sont utilisés : une croix rouge pour indiquer une contre-indication absolue, un triangle orange pour indiquer une contre-indication relative (prescription déconseillée) et une coche verte pour les autres situations (absence de contre-indication mais aussi interaction de niveau « à prendre en compte » et « précaution d'emploi » qui n'empêche pas la prescription). Pour les effets indésirables, la fréquence de l'effet est indiquée avec 1 à 5 petits carrés orange correspondant aux 5 niveaux usuels de fréquence (très rare, rare, peu fréquent, fréquent, très fréquent) et les effets graves sont mentionnés en rouge.

Le tableau dynamique permet de masquer certaines lignes ; dans ce cas le nombre de propriétés cachées est indiqué en dessous du tableau pour chaque médicament. Le tableau peut afficher (voir exemple Figure 5.8) :

- **Les propriétés du nouveau médicament** (affichage par défaut). Seules les lignes du tableau qui correspondent aux propriétés du nouveau médicament sont affichées. Les propriétés des comparateurs n'apparaissent que si le nouveau médicament partage la même propriété.
- **La comparaison des propriétés du nouveau médicament avec un comparateur** choisi par l'utilisateur. Cet affichage est obtenu en cliquant sur l'un des comparateurs pour le sélectionner. Seules les lignes du tableau qui correspondent aux propriétés du nouveau médicament ou du comparateur choisi sont affichées. Les lignes où les deux médicaments diffèrent sont affichées en surbrillance.
  Cet affichage permet de comparer le nouveau médicament à un seul comparateur, par exemple le médicament que le médecin a l'habitude de prescrire et qui lui sert de référence.
- **Les « absences notables »**. Seules les lignes correspondant à des propriétés que possèdent plus de la moitié des comparateurs et que ne possède pas le nouveau médicament sont affichées. Cet affichage permet par exemple d'avoir les situations où la majorité des comparateurs sont contre-indiqués et où le nouveau médicament ne l'est pas, c'est-à-dire les situations où le nouveau médicament peut être intéressant.
- **L'ensemble les propriétés**. Cet affichage est proposé, même s'il conduit généralement à des tableaux très longs et peu exploitables en pratique.

Les boîtes arc-en-ciel sont une nouvelle technique de visualisation mise au point lors du projet VIIIP. Elles ont été décrites en section 4.2. Dans les tableaux comme dans les boîtes arc-en-ciel, des icônes VCM ont été ajoutées pour visualiser les contre-indications.

Le prototype de site web incluant les quatre nouveaux médicaments peut être consulté sur Internet à l'adresse suivante : http://www.lesfleursdunormal.fr/static/viiip_proto/html.

5.2.3 Évaluation du site web

L'objectif de l'évaluation était de répondre à la question suivante : « les médecins peuvent-ils se faire une opinion sur un nouveau médicament à partir de la comparaison des propriétés de ce nouveau médicament avec celles des médicaments existants dans la même indication, sans faire intervenir les avis d'experts habituels ? ». Pour cela, nous avons évalué l'ensemble du site web selon deux aspects : (a) une évaluation subjective de la satisfaction et de l'opinion des médecins sur le site web, et (b) une mesure objective de l'impact de la consultation du site web sur les connaissances des médecins et sur leur décision d'accepter de prescrire ou non des nouveaux médicaments donnés.

L'évaluation s'est effectuée à partir du prototype de site web présenté précédemment avec 4 nouveaux médicaments.

5.2.3.1 Recrutement des médecins

Le recrutement a été réalisé par le Dr Madeleine Favre membre de la SFTG (Société de Formation Thérapeutique du Généraliste). 22 médecins ont été recrutés. Le Tableau 5.1 montre les caractéristiques démographiques des 22 médecins participants. L'âge moyen des médecins (54,6 ans) est élevé, mais cela correspond à la démographie médicale en France.

Deux séances d'évaluation ont été organisées, le mardi 26 janvier 2016 de 20 à 23h (12 médecins) et le samedi 6 février 2016 de 9h30 à 12h30 (10 médecins). Un identifiant arbitraire a été attribué à chaque médecin, à partir du numéro de poste informatique qu'il occupait, auquel nous avons ajouté 100 lors de la deuxième séance. Les deux séances d'évaluation ont suivi le même programme.

5.2.3.2 Protocole

Nous avons choisi un protocole comparatif de type « avant - après ». Avant de consulter le site web, les médecins ont répondu à un premier questionnaire dans lequel ils ont indiqué pour chacun des quatre nouveaux médicaments s'ils le connaissaient, s'ils étaient prêts à le prescrire (ou l'avaient déjà prescrit) ou

VITAROS 300 microgrammes, crème

alprostadil (Prostaglandine E1)
Nouvelle voie d'administration pour un principe actif existant

Synthèse

Aide ?

Indications
Chez les hommes à partir de 18 ans, traitement de la dysfonction érectile, définie par l'incapacité à atteindre ou à maintenir une érection suffisante du pénis pour permettre une performance sexuelle satisfaisante.

Efficacité
SMR : faible
ASMR : absence de progrès thérapeutique (V)
2 études cliniques

Comparateurs
Prostaglandine E1
Edex (alprostadil)
Caverject (alprostadil)
Muse (alprostadil)
Inhibiteurs de la phosphodiestér…
Viagra (sildénafil)
Cialis (tadalafil)
Levitra (vardénafil)
Spedra (avanafil)

Contre-indications
Hypersensibilité prostaglandi…
Syncope
Hypotension orthostatique
Pathologie contre-indiquant l…
Rapports sexuels sans préser…
Femme
Balanite
Prédisposition au priapisme
Urétrite masculine
Malformation anatomique d…
Enfant de moins de 18 ans

Interactions
(aucune)

Effets indésirables
(grave)
Priapisme
Erection prolongée
(fréquent)
Rash
Douleur urétrale
Douleur du pénis
Erythème pénien
Oedème pénien
Sensation de brûlure pénienne
Trouble du pénis
Prurit génital chez l'homme
Vaginite non précisée
Balanite
Douleur génitale chez l'homme
Erythème génital
Inconfort génital
Sensation de brûlure vulvo-vaginale

Excipients à effet notoire
(aucun)

Terrain, posologie, coût, efficacité et conduite

		Prostaglandine E1			Inhibiteurs de la phosphodiestérase de type 5			
	Vitaros	Edex	Caverject	Muse	Viagra	Cialis	Levitra	Spedra
Terrain	Adulte Homme + 18 ans	Adulte Homme + 15 ans	Adulte Homme + 15 ans	Adulte Homme + 15 ans	Adulte Homme + 18 ans	Adulte Homme + 18 ans	Adulte Homme + 18 ans	Adulte Homme + 18 ans
Posologie	1 unidose 1 fois par jour 2 à 3 fois par semaine.	5 à 20 µg 1 fois par jour 1 à 2 fois par semaine.	5 à 20 µg 1 fois par jour 1 à 2 fois par semaine.	1 bâton 1 à 2 fois par jour. Maximum 7 bâtons par semaine.	1 comprimé 1 fois par jour.	2,5 mg et 5 mg : 1 comprimé 1 fois par jour. 10 et 20 mg : 1 comprimé 1 fois par jour 1 à 2 fois par semaine.	1 comprimé 1 fois par jour.	1 comprimé 1 fois par jour.
Coût de dose	10,05 €	11,47 €	10,79 €	16,50 - 18,90 - 23,60 €	6,35 (Gé) à 12,62 € (VIAGRA)	8,95 €	10,54 €	6,33 €
Taux de remboursement	0%	0%	0%	0%	0%	0%	0%	0%
Délais d'action	5 à 30 mn	5 à 10 mn	5 à 10 mn	5 à 10 mn	60 mn	30 mn	25-60 mn	30 mn
Durée d'action	1 à 2 h	30 à 60 mn	30 à 60 mn	30 à 60 mn	4 h	17 heures	4 h	6 à 17 heures
Service Médical Rendu	faible	modéré	modéré	-	important	important	important	-
Amélioration du Service Médical Rendu	absence de progrès thérapeutique (V)							
Conduite de véhicule	niveau 1	niveau 1	niveau 1	niveau 1	niveau 1	niveau 1	niveau 1	niveau 1

(tout est affiché)

Résultats d'essai clinique

Aide ?

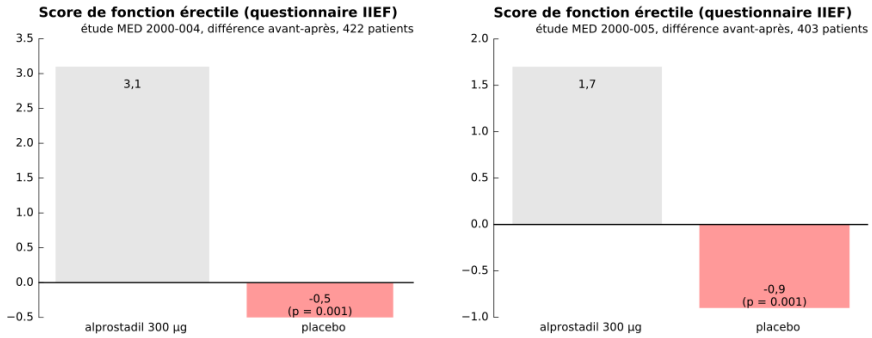
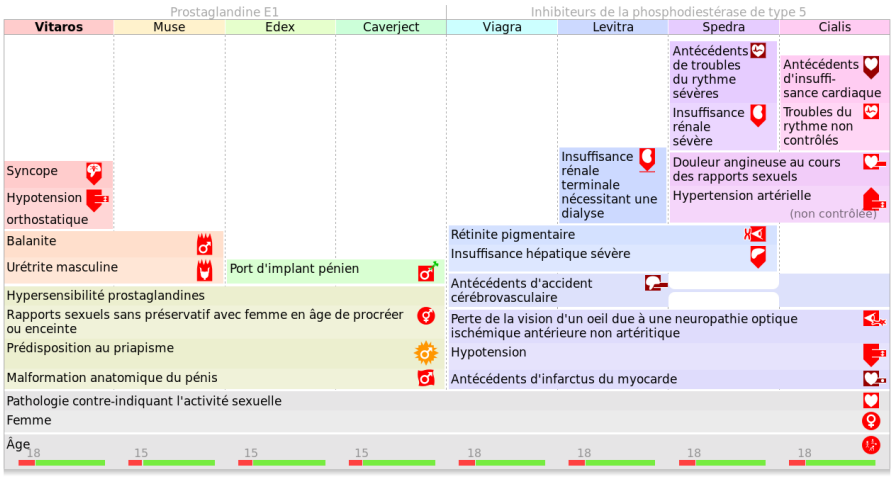


FIGURE 5.7 – Page VIIIP pour Vitaros®, créée par extraction et saisie manuelle des données (les contre-indications sont en boîtes arc-en-ciel, les interactions et les effets indésirables en tableaux dynamiques ; la page permet à l'utilisateur de choisir l'un ou l'autre)[108]

Contre-indications
Tableau **Boîtes arc-en-ciel**

Aide ?



Interactions (contre-indiquées et déconseillées)
Tableau Boîtes arc-en-ciel

Aide ?

	Prostaglandine E1				Inhibiteurs de la phosphodiestérase de type 5			
	Vitaros	Edex	Caverject	Muse	Viagra	Cialis	Levitra	Spedra
Clarithromycine	✓							✗
Dapoxetine	✓				△	△	△	△
Doxazosine	✓				△	△	△	△
Dérives nitrés et apparentés	✓				✗	✗	✗	✗
Inhibiteurs de protéases boostes par ritonavir	✓				△	△	✗	✗
Itraconazole	✓						△	✗
Jus de pamplemousse	✓						△	△
Kétoconazole	✓						△	✗
Ombitasvir + paritaprevir	✓				✗			
Riociguat	✓				✗	✗	✗	✗
Telaprevir	✓				△		△	✗
Telithromycine	✓							✗

Afficher tout
Afficher les absences notables

Effets indésirables (fréquents, très fréquents ou graves)
Tableau Boîtes arc-en-ciel

Aide ?

	Prostaglandine E1				Inhibiteurs de la phosphodiestérase de type 5			
	Vitaros	Edex	Caverject	Muse	Viagra	Cialis	Levitra	Spedra
Affections de la peau et du tissu sous-cutané								
Rash	■■■■							
Affections du rein et des voies urinaires								
Douleur urétrale	■■■■			■■■■■				
Affections des organes de reproduction et du sein								
Douleur du pénis	■■■■	■■■■■	■■■■■	■■■■■				
Erythème pénien	■■■■							
Oedème pénien	■■■■							
Sensation de brûlure pénienne	■■■■		■■■■■					
Trouble du pénis	■■■■							
Priapisme	■■■	■■■	■■■	■■	■■	■■	■■	
Prurit génital chez l'homme	■■■■							
Vaginite non précisée	■■■■							
Balanite	■■■■							
Douleur génitale chez l'homme	■■■■							
Erythème génital	■■■■							
Inconfort génital	■■■■							
Erection prolongée	■■■■	■■■■	■■■■	■■	■■	■■	■■■	
Sensation de brûlure vulvo-vaginale	■■■■			■■■■				
		5 autres	5 autres	6 autres	25 autres	23 autres	13 autres	5 autres

Afficher tout
Afficher les absences notables

Excipients à effet notoire

Aide ?

	Prostaglandine E1				Inhibiteurs de la phosphodiestérase de type 5			
	Vitaros	Edex	Caverject	Muse	Viagra	Cialis	Levitra	Spedra
Lactose					✗	✗		

(tout est affiché)

Identification et documents

	Prostaglandine E1				Inhibiteurs de la phosphodiestérase de type 5			
	Vitaros	Edex	Caverject	Muse	Viagra	Cialis	Levitra	Spedra
Dénomination Commune	alprostadil	alprostadil	alprostadil	alprostadil	sildénafil	tadalafil	vardénafil	avanafil
Dosage	300 µg	10 et 20 µg	10 et 20 µg	250 µg, 500 µg et 1000 µg	25 mg, 50 mg et 100 mg	2,5 mg, 5 mg, 10 mg et 20 mg	5 mg, 10 mg et 20 mg	50 mg, 100 mg et 200 mg
Forme	Crème	Poudre et solvant pour solution injectable	Poudre et solvant pour solution injectable	Bâton pour usage urétral	Comprimé et comprimé orodispersible (Gé)	Comprimé	Comprimé et comprimé orodispersible	Comprimé
voie	Topique	Intracaverneuse	Intracaverneuse	Intra-urétrale	Orale	Orale	Orale	Orale
Présentation	Récipient unidose	Cartouche	Seringue	Bâton				
Laboratoires	MAJORELLE	UCB PHARMA	PFIZER	MEDA AB	PFIZER	LILLY	BAYER	MENARINI
Date d'AMM	2013	1997	1994	1999	1998	2002	2003	2013
Résumé des Caractéristiques Produit								
Notice patient								
Avis de la Commission de Transparence								

(tout est affiché)

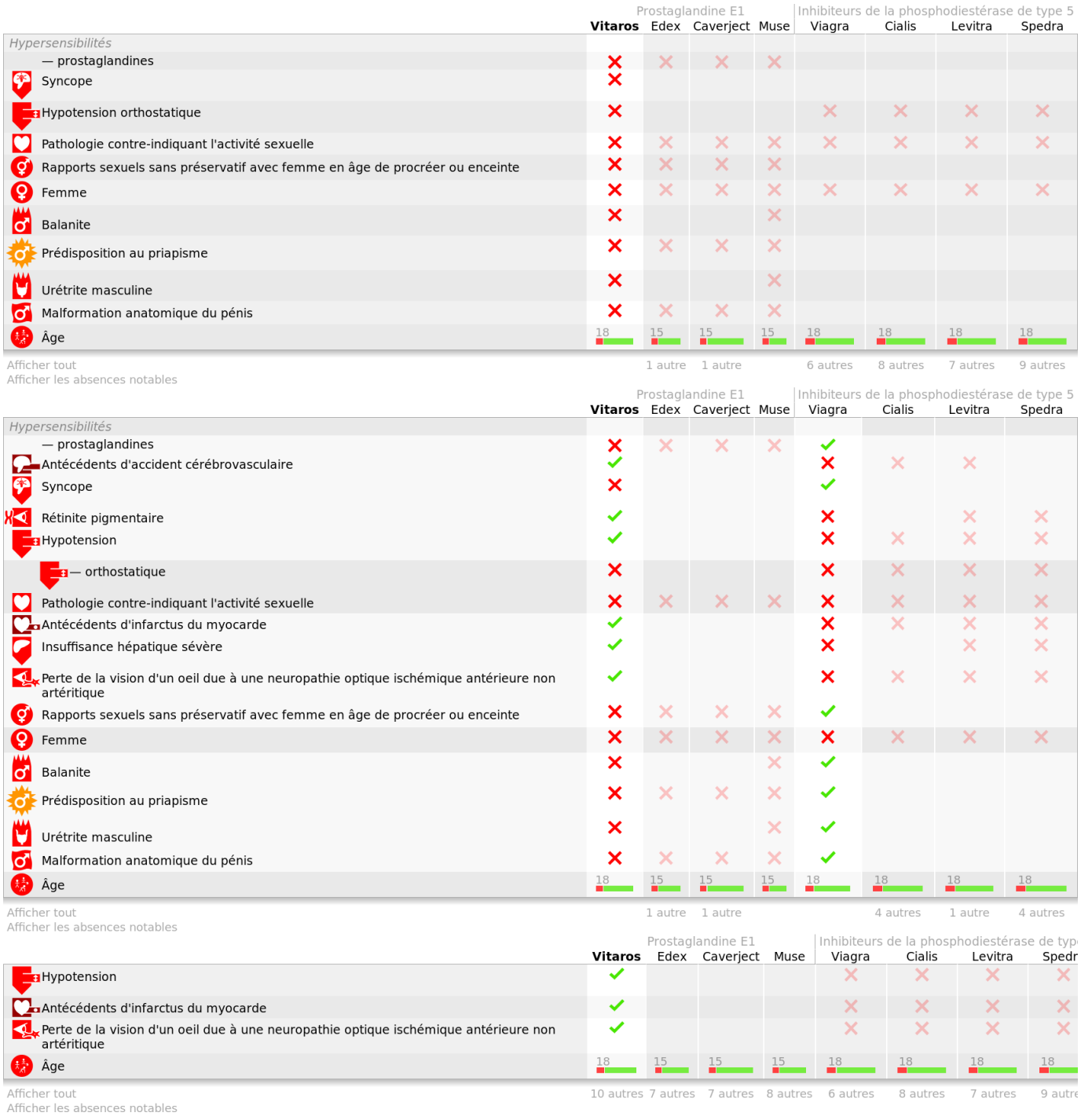


FIGURE 5.8 – Exemple de tableau dynamique présentant les contre-indications de Vitaros® et de 7 comparateurs. En haut, le tableau affiche les propriétés de Vitaros®. Au milieu, il affiche une comparaison Vitaros® - Viagra®. En bas, il affiche les absences notables.

Utilisateur n°	sexe	Âge	Années exercice
2	F	55	20
3	M	52	26
4	M	52	24
5	M	55	26
6	M	39	10
7	M	38	10
9	F	52	5
11	M	28	1
12	F	65	5
20	F	61	33
21	M	69	39
22	M	61	29
101	M	53	25
102	F	42	15
104	F	62	38
105	F	53	3
106	F	56	26
107	F	60	35
109	F	60	33
111	M	62	37
112	M	58	25
120	M	69	40
Moyenne :	12 H, 10 F	54,6	23,0

TABLEAU 5.1 – Caractéristiques démographiques des 22 médecins recrutés pour l'évaluation (M : masculin, F : féminin).

Connaissez-vous Vitaros® ? ☐ Oui ☐ Non

Connaissez-vous ses propriétés ? ☐ Oui ☐ Non

Seriez-vous prêt à le prescrire ?
☐ Je l'ai déjà prescrit ☐ Je suis prêt à le prescrire ☐ Je ne suis pas prêt à le prescrire

Si vous l'avez prescrit ou êtes prêt à le faire, pour quelle(s) raison(s) ?

- ☐ Il est efficace
- ☐ Il a peu de contre-indications et/ou d'interactions
- ☐ Il a peu d'effets indésirables
- ☐ Il est bon marché
- ☐ Mes correspondants spécialistes le prescrivent
- ☐ Mes patients le réclament
- ☐ Autre(s) :

Si vous n'êtes pas prêt à le prescrire, pour quelle(s) raison(s) ?

- ☐ Il n'est pas assez efficace
- ☐ Il a trop de contre-indications et/ou d'interactions
- ☐ Il a trop d'effets indésirables
- ☐ Il coûte trop cher
- ☐ Il est trop récent
- ☐ Je manque d'information sur ce médicament
- ☐ Je préfère d'autres médicaments qui ont la même indication
- ☐ Autre(s) :

FIGURE 5.9 – Extrait du questionnaire n°1 portant sur le nouveau médicament Vitaros®.

non, et quels étaient leurs arguments pour justifier ce choix (voir extrait du questionnaire Figure 5.9). Les arguments proposés en faveur de la prescription étaient les suivantes : l'efficacité, le peu de contre-indications et d'interactions, le peu d'effets d'indésirables, le faible coût, la prescription par les correspondants spécialistes, la demande par les patients. Les arguments proposés contre la prescription étaient : l'efficacité faible, le peu de contre-indications et interactions, le peu effets d'indésirables, le coût élevé, le fait qu'il soit trop récent, le manque d'information, le choix de préférer d'autres médicaments. D'autres arguments pouvaient être ajoutés librement, dans les deux cas.

Ensuite, les médecins ont consulté le prototype de site à leur guise durant environ 45 minutes. Puis ils ont répondu à un second questionnaire comprenant deux parties : la première demandait leur opinion sur le site web, son utilisation et ses composantes, et la seconde reprenait les mêmes questions que le questionnaire précédent (décision de prescrire ou non chacun des quatre nouveaux médicaments, et arguments pour justifier ce choix), afin de voir si l'avis des médecins et leurs arguments avaient changé suite à la consultation du site web. Les médecins étaient autorisés à consulter le site pendant qu'ils répondaient au second questionnaire.

Une discussion générale a été organisée en fin d'évaluation. Celle-ci a été enregistrée et des notes ont également été prises.

L'analyse statistique a été réalisée avec le logiciel R version 3.2.3. Lors de l'analyse, les données manquantes dans les questionnaires (lorsqu'un médecin n'a coché ni la case « oui » ni la case « non » par exemple) ont été considérées comme ayant la valeur la plus défavorable (en général « non »).

5.2.3.3 Résultats : opinion générale des médecins sur le site web

Le Tableau 5.2 montre les réponses obtenues aux questions sur l'opinion générale. 100% (22/22) des médecins ont trouvé que le site web leur avait permis de se forger une opinion sur les 4 nouveaux médicaments. 77% ont facilement appris à se servir du site Web et 91% l'ont trouvé facile à utiliser une fois pris en main. 100% ont trouvé intéressant le principe d'une information comparative sur les médicaments. 95% recommanderaient ce site Web à leurs collègues.

91% des médecins seraient prêts à utiliser régulièrement ce site Web ; pour ce qui est de la fréquence, 3 médecins évoquent une utilisation quotidienne, 6 médecins une utilisation hebdomadaire, 1 médecin une utilisation mensuelle et 5 médecins une utilisation au coup par coup (lors de la prescription de nouveaux médicaments par exemple).

50% des médecins ont mentionné la présence d'information manquante. Les informations complémentaires demandées concernent : davantage de détails et d'explication sur les études cliniques (3 médecins), les précautions d'emploi (2 médecins), le mode d'administration, les médicaments similaires, l'accès au site web par pathologie ou par ancien médicament en plus de l'accès par nouveau médicament (1 médecin pour chaque item).

Les informations jugées les plus utiles sont la comparaison des effets indésirables (95%) et des coûts de traitement (86%), et la liste des comparateurs (86%).

Cette partie de l'évaluation montre que **les médecins ont apprécié le site web et les principes sur lesquels il repose** (l'information comparative en particulier). Le site web pourrait cependant être enrichi, notamment en ce qui concerne les essais cliniques et les précautions d'emploi.

5.2.3.4 Résultats : impact sur le choix de prescrire

La Figure 5.10 résume les résultats de cette partie de l'évaluation sous forme de diagrammes en bâton. On notera le nombre important de médecins qui manquaient d'information sur un ou plusieurs des nouveaux médicaments ; après la consultation du site web, le manque d'information n'a été signalé qu'1 seule fois (1%) par les médecins. L'évaluation impliquait 22 médecins et 4 médicaments, soit 88 (= 22 x 4) cas. Sur ces 88 cas, nous avons observé 29 (33,0%) changements d'avis sur la décision de prescrire le nouveau médicament. Ces 29 changements d'avis se décomposent en 18 basculements vers la prescription (9 pour Ciloxan, 7 pour Vitaros, 2 pour Pyléra) et 11 basculements vers la non-prescription (6 pour Pyléra, 2 pour Antarène codéine, 2 pour Vitaros, 1 pour Ciloxan). Sur les 59 cas où l'avis du médecin n'a pas changé, dans 40 cas (67,8%) les arguments donnés pour justifier son choix diffèrent avant et après avoir consulté le site web (seuls ont été pris en compte les 4 premiers arguments, efficacités, contre-indications et interactions, effets indésirables et

Questions	% (intervalle à 95%)	Oui	Non	Absent
Ce site Web m'a permis de me forger une opinion sur chacun des 4 nouveaux médicaments	100%	22	0	0
J'ai appris facilement à me servir du site Web	77% (60 - 95)	17	3	2
Une fois pris en main, j'ai trouvé le site Web facile à utiliser	91% (79 - 100)	20	2	0
J'utiliserais couramment ce site Web s'il était systématiquement mis à jour pour chaque nouveau médicament	91% (79 - 100)	20	1	1
J'ai trouvé que certaines informations manquaient	50% (29 - 71)	11	10	1
Je trouve intéressant d'avoir une information comparative (nouveau médicament - comparateurs) plutôt qu'une information sur le nouveau médicament seul	100%	22	0	0
Dans le site Web, j'ai trouvé utile :				
...la synthèse	59% (39 - 80)	13		
...la liste des comparateurs	82% (68 - 98)	18		
...les résultats cliniques du nouveau médicament	45% (25 - 66)	10		
...la comparaison des contre-indications	77% (60 - 95)	17		
...la comparaison des interactions médicamenteuses	64% (44 - 84)	14		
...la comparaison des effets indésirables	95% (87 - 100)	21		
...la comparaison des excipients à effet notoire	50% (29 - 71)	11		
...la comparaison des posologies	50% (29 - 71)	11		
...la comparaison du coût de traitement	82% (68 - 98)	18		
Pour les comparaisons, j'ai trouvé utile :				
...les comparaisons 2 à 2	55% (34 - 75)	12	6	4
...la vue d'ensemble	86% (72 - 100)	19	1	2
Je conseille la consultation d'un tel site Web à mes collègues	95% (87 - 100)	21	0	0

TABLEAU 5.2 – Réponses obtenues aux différentes questions posées aux médecins pour mesurer leur satisfaction et leur opinion vis-à-vis du prototype VIIIP.

coût, les autres n'étant pas liés aux propriétés intrinsèques du médicament). Sur l'ensemble des 88 cas, 48 arguments sont donnés en tout avant et 111 après. Le nombre d'arguments pour la prescription (23 avant et 55 après) et pour la non-prescription (25 avant et 56 après) est similaire. Le Tableau 5.3 montre l'évolution des avis et des arguments des médecins.

Cette partie de l'évaluation montre que **le site web a permis aux médecins de changer leur opinion sur le nouveau médicament dans 33,0% des cas** (29/88) ; ces changements d'opinion correspondent :

- dans 20,5% des cas (18/88), à des médecins qui ne prescrivaient pas le nouveau médicament car ils ne le connaissaient pas ou ils manquaient d'information à son sujet, et qui décident ensuite de le prescrire,
- dans 12,5% des cas (11/88), à des médecins qui étaient prêts à le prescrire (ou le prescrivaient déjà) et qui choisissent de ne plus le prescrire.

De plus, la consultation du site web a permis aux médecins de mieux argumenter leur choix dans 45% des cas, lorsqu'il ne les a pas faits changé d'avis (40/88, dont 17 cas où le médecin manquait d'information et 23 cas où ils ne manquaient pas d'information). Enfin, les taux importants de changements d'avis et d'arguments montrent que **le site web VIIIP à permis aux médecins de se faire une opinion mais aussi que les médecins ont fait confiance au site web et à son contenu**.

5.2.3.5 Résultats : discussion générale

Les médecins ont été très enthousiastes lors des deux séances d'évaluation. Voici quelques citations de médecins :

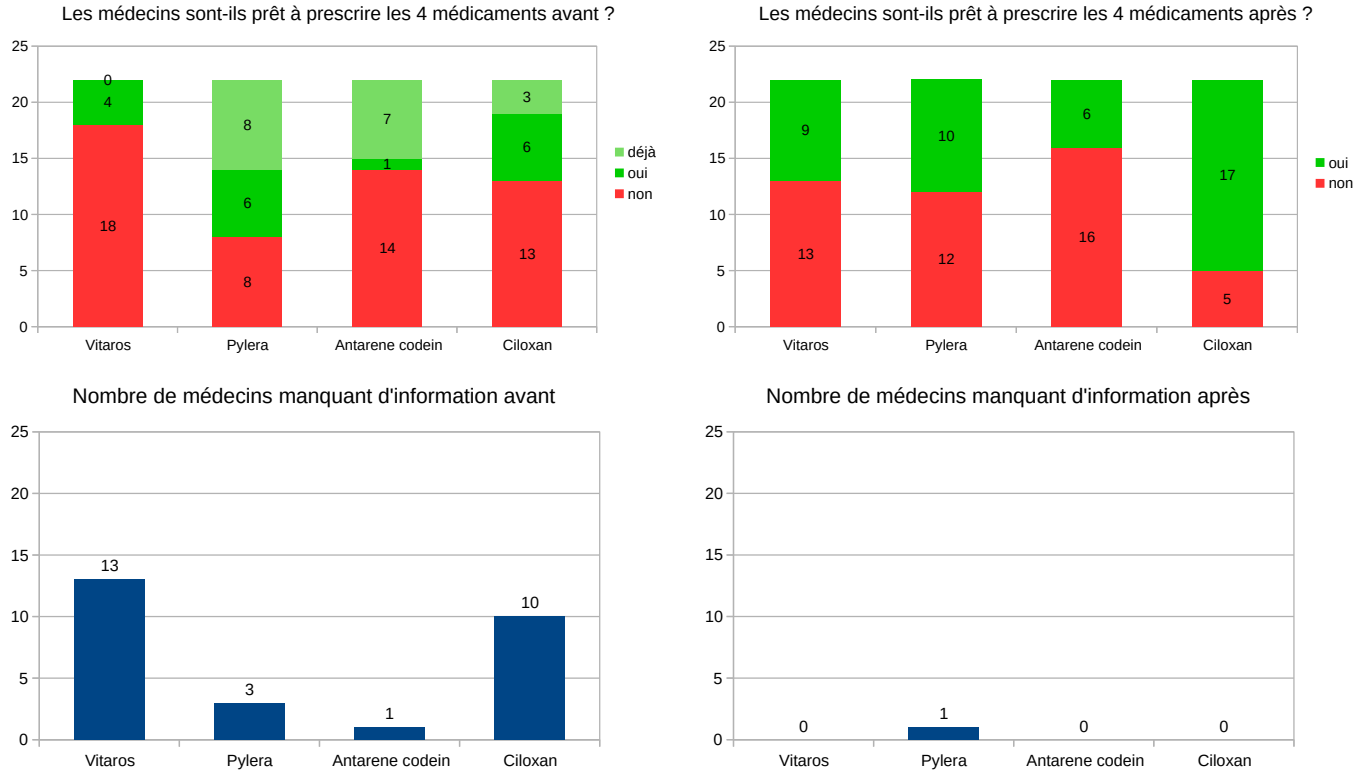


FIGURE 5.10 – Diagrammes en bâtons montrant pour chacun des quatre nouveaux médicaments le nombre de médecins prêts à le prescrire avant de consulter le site web (en haut à gauche) et après (en haut à droite), et le nombre de médecins manquant d'information avant (en bas à gauche) et après (en bas à droite).

Avant la consultation du site web		Après
Le médecin est prêt à prescrire le nouveau médicament (35 / 88, 39,7%)		Il change d'avis et n'est plus prêt à prescrire ce médicament (11 / 35, 31,4%)
		Il ne change pas d'avis mais justifie son choix avec des arguments différents (15 / 35, 42,9%)
		Il ne change ni son avis ni ses arguments (9 / 35, 25,7%)
Le médecin n'est pas prêt à prescrire le nouveau médicament (53 / 88, 60,2%)	Il ne connaît pas le nouveau médicament ou manque d'information à son sujet (39 / 88, 44,3%)	Il est prêt à prescrire le médicament (18 / 39, 46,2%)
		Il n'est pas prêt à prescrire le médicament (21 / 39, 53,8%)
	Il connaît le nouveau médicament (14 / 88, 15,9%)	Il change d'avis et est désormais prêt à prescrire ce médicament (0 / 14, 0%)
		Il ne change pas d'avis mais justifie son choix avec des arguments différents (8 / 14, 57,1%)
		Il ne change ni son avis ni ses arguments (6 / 14, 42,9%)

TABLEAU 5.3 – Évolution de l'avis et des arguments des médecins avant et après la consultation du site web (sur 4x22 = 88 cas).

« On l'avait rêvé, vous l'avez créé ! »
« Mieux que le Doroz »
« Vous avez créé un besoin »
« Indispensable »
« Prescription raisonnée »
« VIIIP garde une certaine neutralité »
« Il n'existe pas d'outils de ce type »

Les médecins ont trouvé le site web très riche et très complet. Quelques-uns l'ont même trouvé trop riche et trop dense en information (« Trop d'information tue l'information ») tandis que d'autres au contraire ont trouvé que l'aspect « visuel » du site web permettait une analyse rapide des informations, et particulièrement pour décider si le nouveau médicament « doit aller à la poubelle » ou non. Par rapport à la revue Prescrire, les médecins ont trouvé le site web VIIIP plus synthétique (notamment grâce aux tableaux et aux comparatifs) et, bien que le site ne donne pas directement un avis sur le nouveau médicament, les médecins ont trouvé qu'ils leur permettaient de s'en faire un.

L'ensemble des médecins était d'accord pour dire que le site web était adapté à une utilisation en formation. Mais une partie d'entre eux le considérait aussi comme adapté à une utilisation en consultation, notamment pour choisir un médicament approprié à un patient compliqué présentant de nombreuses contre-indications. L'utilisation en consultation est une situation que nous n'avions pas vraiment envisagée jusque là. Plusieurs médecins ont suggéré l'intégration de la plate-forme aux logiciels de prescription.

Des médecins ont trouvé que le site web était trop succinct sur les résultats d'essais cliniques ; ils auraient souhaité davantage d'informations pour arriver à quelque chose qui se rapproche de la librairie Cochrane, avec le nombre d'études publiées, le nombre d'études fiables, *etc.*

Les icônes VCM présentes au niveau des contre-indications ont été jugées difficiles à comprendre et surchargeant l'interface (au niveau des couleurs en particulier) par plusieurs médecins. Cependant, les médecins n'avaient préalablement reçu aucune formation à VCM.

Pour de plus amples détails, se référer aux publications [102, 113, 114].

5.2.4 Discussion

Dans cette section, nous avons décrit la plate-forme et le site web mis au point lors du projet VIIIP pour la diffusion d'une information indépendante sur les nouveaux médicaments, *via* la comparaison des propriétés des nouveaux médicaments aux propriétés des médicaments existants. Nous avons aussi présenté l'évaluation de ce site web auprès de médecins généralistes. Cette évaluation a montré le très grand enthousiasme des médecins vis-à-vis du site et valide notre hypothèse : les médecins ont pu se faire une opinion sur quatre nouveaux médicaments sans recevoir d'avis d'expert, par la simple comparaison visuelle des propriétés des nouveaux médicaments avec celles des médicaments existants dans la même indication.

Le protocole d'évaluation a été particulièrement délicat à mettre au point, à cause de l'absence d'outils similaires existants pour effectuer une comparaison. Nous nous sommes donc finalement orientés vers un protocole comprenant d'une part une évaluation subjective portant sur la satisfaction globale, et d'autre part une évaluation objective de type « avant - après ». Contrairement aux évaluations réalisées lors du projet L3IM, cette évaluation objective ne mesurait pas les performances mais plutôt l'effet du système et sa capacité à faire changer d'avis le médecin, sans pour autant considérer de « bons » et de « mauvais » avis.

Nous avons observé un nombre important de changements d'avis des médecins concernant la décision de prescrire ou non des nouveaux médicaments. Cependant, tous les médecins qui ont changé d'avis dans le sens de la prescription (le médecin n'était pas prêt à prescrire le nouveau médicament avant de consulter le site, puis il change d'avis après et devient prêt à prescrire) disaient manquer d'information sur ce médicament avant la consultation du site. Nous n'avons pas rencontré de médecins qui n'étaient pas prêts à prescrire avant la consultation du site pour des raisons cliniques (effets indésirables par exemple) ou économiques et qui ont changé d'avis en consultant les propriétés du nouveau médicament sur le site web.

Les médecins ont proposé plusieurs pistes pour enrichir le site web. En particulier, il semble intéressant d'améliorer la visualisation des résultats des essais cliniques.

Le prototype de site web évalué comprenait quatre nouveaux médicaments et leurs comparateurs. Les données concernant ces médicaments ont été extraites à partir des documents de référence (RCP, notices et ACP) par une pharmacienne experte en base médicament puis saisies manuellement. Cependant cette extraction manuellement demande un temps de travail important à l'expert (environ 40h pour chaque nouveau médicament). C'est pourquoi nous avons essayé deux alternatives :

- **L'extraction automatique à partir d'une base médicament**. Nous avons travaillé sur la banque Thériaque ; de nombreux problèmes ont été rencontrés dans les informations extraites, nous les avons classés sur quatre niveaux :
  - **Des problèmes de manque d'information**, lorsque l'information que nous recherchons n'est pas présente dans la base (par exemple la gravité des effets indésirables).
  - **Des problèmes de structure** de la base, lorsque l'information que nous recherchons est présente mais que sa structuration ne permet pas une extraction satisfaisante (par exemple les posologies sont en texte libre).
  - **Des problèmes de qualité** de l'information, lorsque la structure de la base est bonne mais que les données qui figurent dedans ne correspondent pas aux documents de référence telles que les RCP (par exemple, pour Izalgi, l'effet indésirable « insomnie » a pour fréquence « fréquent » dans Thériaque, alors que dans le RCP, la fréquence est « peu fréquent »).
  - **Des problèmes de codage hétérogène**, lorsque les informations sont bonnes mais que le codage est hétérogène, c'est-à-dire que des codes différents sont utilisés pour coder des informations identiques. Dans ce cas, il est possible de consulter les propriétés de chaque médicament sans problème, mais il n'est pas possible de comparer les médicaments entre eux : les bases médicaments n'ont pas été conçues pour permettre la comparaison entre médicaments. Par exemple, pour deux médicaments antiagrégants plaquettaires, l'un a pour contre-indication « maladie hémorragique » et l'autre a une dizaine de contre-indications (« déficit des facteurs de la coagulation », « thrombopénie »...) couvrant les principales maladies hémorragiques (sans être nécessairement exhaustif).

  Une étude a été menée par Hélène Berthelot sur les deux autres bases médicaments disponibles en France (Vidal et Banque Claude Bernard, BCB). Cette étude a montré des problèmes similaires dans les autres bases. Lors du projet VIIIP, nous avons donc conclu à l'impossibilité d'extraire automatiquement des données fiables et comparables à partir des bases médicaments.
- **L'extraction automatique à partir des textes**, en utilisant des méthodes de Traitement Automatique de la Langue naturelle (TAL). Dans le projet VIIIP, nous avons envisagé la mise au point d'outils d'extraction automatique à partir du texte pour les excipients à effet notoire et pour les effets indésirables [112]. Cependant, nous ne sommes pas parvenus à des résultats satisfaisants ; l'extraction de connaissances à partir des textes reste un sujet de recherche qui pourrait faire l'objet de travaux ultérieurs avec des partenaires spécialisés en TAL.

Pour davantage d'information sur l'extraction des connaissances sur les médicaments et les problèmes rencontrés, se référer au rapport scientifique du projet VIIIP [103].

Chapitre 6

Discussion et conclusion

« Nos savoirs actuels ne sont nullement des vérités absolues et définitives, mais seulement des étapes dans l'histoire mouvante des savoirs. »
(Bestiaire du Moyen-Âge, chapitre le Moyen-Âge et l'animal, Michel Pastoureau, Somogy éditions d'art, 2004)

« Nous avions servi des hommes. Nous aurions dû servir l'humanité. »
(Les seigneurs de l'Instrumentalité, Cordwainer Smith, Folio SF éditions Gallimard, 2004)

Dans ce chapitre, nous présenterons tout d'abord un état de l'art sur la visualisation des connaissances. Ensuite, nous reviendrons sur trois points importants : l'évaluation des outils visuels pour d'aide à la décision, l'aspect multidisciplinaire de notre travail, et les questions éthiques sous-jacentes. Enfin, nous présenterons notre conclusion.

6.1 De la visualisation des connaissances à la connaissance des visualisations

Malgré ses origines anciennes (voir chapitre 1), la visualisation des connaissances peine aujourd'hui à émerger comme un domaine de recherche à part entière. Nous voyons plusieurs raisons à cela : l'absence de définition claire du domaine, les recoupements importants qui existent avec d'autres domaines comme la visualisation d'information, la représentation des connaissances ou la communication, et le fait que certaines approches visuelles n'utilisent pas d'ordinateur, ce qui rend leur appartenance à la discipline informatique moins évidente.

6.1.1 État de l'art

Dans la littérature, plusieurs définitions de la visualisation des connaissances (*knowledge visualisation*) ont été proposées, parfois contradictoires :

— **R.A. Burkhard** définit en 2004 la visualisation des connaissances comme « la discipline qui examine l'utilisation de représentations visuelles pour améliorer le transfert de connaissance entre au moins deux personnes ou groupes de personnes » [29, 30, 28]. Les travaux de R.A. Burkhard ont porté sur la visualisation des connaissances organisationnelles à l'aide de métaphores visuelles, par exemple la présentation des étapes d'un projet à la manière d'un plan de métro (6.1). Il destine principalement la visualisation des connaissances aux gestionnaires (*managers*) et cible 3 problèmes fréquemment rencontrés : la surcharge d'information (*information overload*), la mauvaise interprétation des informations et leur mauvaise utilisation. L'auteur s'inspire des méthodes utilisées par les architectes, avec plusieurs modes de visualisation tels que les plans, les photos, les schémas, les maquettes, *etc.* Il insiste particulièrement sur la dimension « communication » et mentionne aussi l'emploi de méthodes de visualisation sans ordinateur.

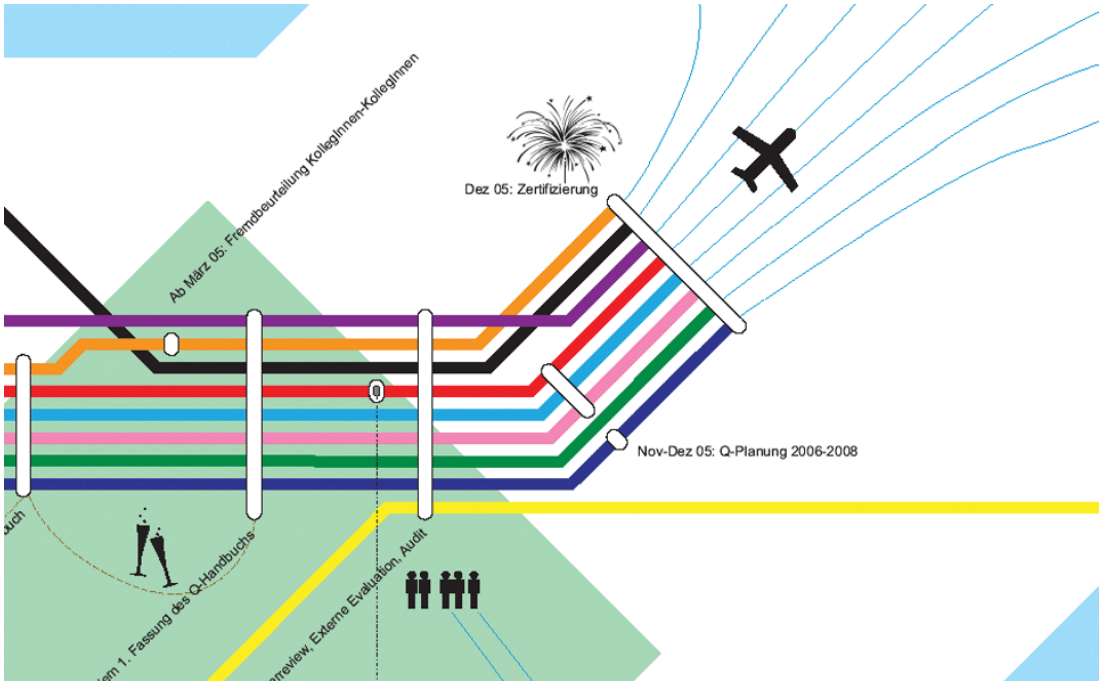


FIGURE 6.1 – Visualisation des étapes d'un projet sous la forme d'un plan de métro (*tube map*). Les lignes de métro représentent les différentes équipes, et les stations de métro les étapes (les lignes qui s'y arrêtent correspondent aux équipes participant à cette étape). L'aéroport avec l'avion qui décolle symbolise la fin du projet. Extrait d'après [30].

Selon lui, la visualisation d'information a pour objectif de présenter graphiquement les informations afin de susciter de nouvelles découvertes (*insights*) tandis que la visualisation des connaissances a pour objectif de transférer ces découvertes d'une personne à l'autre. Le même auteur propose un cadre conceptuel pour la visualisation des connaissances s'appuyant sur 4 dimensions : le type de connaissance (s'agit-il d'une réponse à la question qui, quoi, comment ou pourquoi ?), le type de cible (individu, groupe, équipe, *etc*), la fonction recherchée (coordonner les individus, capturer l'attention, faciliter la mémorisation, motiver, élaborer et découvrir de nouvelles connaissances) et le type de visualisation (plans, photos, maquettes, *etc*).
La définition originelle de R.A. Burkhard nous semble trop restrictive et limitée, puisqu'elle exclut du champ de la visualisation des connaissances tous les travaux qui ne portent pas expressément sur le transfert de connaissance. Cette définition exclue notamment l'aide à la décision. La thèse de R.A. Burkhard selon laquelle la visualisation des connaissances tirerait son origine de l'architecture [30] apparaît également discutable, surtout au regard des éléments historiques que nous avons reportés en introduction.

— **R. Meyer** [134], en 2008, s'appuie sur la définition originelle de R.A. Burkhard.
— **S.O. Tergan, T. Keller et R.A. Burkhard** [178] proposent en 2006 une définition plus large : la visualisation des connaissances « étudie la capacité des formats visuels pour représenter les connaissances. Il vise à aider les processus cognitifs impliqués dans la génération, la représentation, la structuration, le partage et l'utilisation de la connaissance ».
— **M.J. Eppler et R.A. Burkhard** [47] précisent cette définition en 2007 : la visualisation des connaissances « examine l'utilisation de représentations visuelles pour améliorer la gestion des connaissances à tous les niveaux (personnel, interpersonnel, équipe, organisationnel, interorganisationnel et sociétal) ». Plusieurs applications sont proposées : le transfert de connaissances mais aussi la création de nouvelles connaissances, l'identification de connaissances, l'évaluation des connaissances, l'application des connaissances et le « marketing » des connaissances. Le cadre conceptuel de R.A. Burkhard est repris et enrichi d'une cinquième dimension, la situation (durant un dialogue, durant une présentation, dans un rapport papier, *etc*).

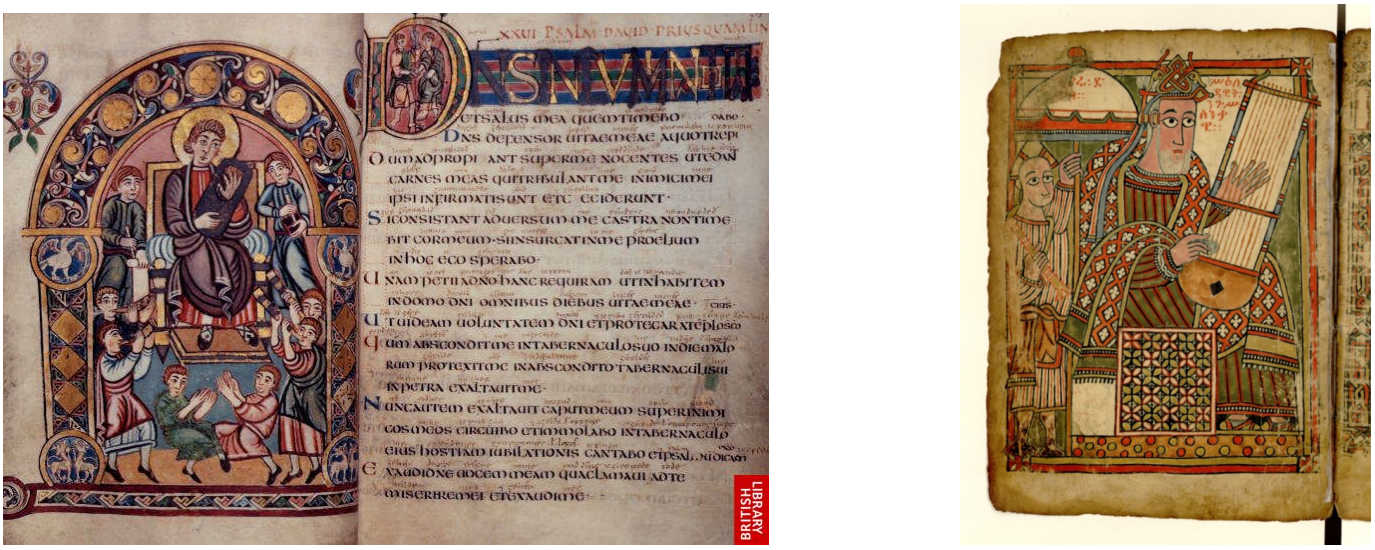


FIGURE 6.2 – Enluminures montrant des joueurs de lyre médiévale. À gauche : psautier anglais de l'école de Cantorbéry (VIII^{ème} siècle), à droite : manuscrit éthiopien d'Abbadie 105 folio 13v (XV^{ème} siècle). Ces enluminures contiennent des connaissances diffuses sur la manière de jouer de la lyre, par exemple il est possible d'étudier la tenue de l'instrument et la position des mains.

- **S. Bertschi** a réuni plusieurs points de vue sur la visualisation des connaissances [19] :
  - Pour **S. Bresciani**, « la visualisation des connaissances relie des concepts de manière graphique en structurant les textes et les éléments visuels de manière pertinente ».
  - Pour **A. Vande Moere**, « la visualisation des connaissances peut être considérée comme la visualisation de données dans un certain contexte », celui d'un récit (*storytelling*).
  - Pour **V. Sabol**, « la visualisation des connaissances est une discipline qui cherche à représenter la connaissance différemment, [...] en exprimant les faits de manière graphique ».
  - Pour **R. Goebel**, la visualisation des connaissances est « la transformation de données concrètes, pour les "expliquer" de manière visuelle avec l'intention d'évoquer les mêmes perspectives pour tous les utilisateurs ». L'auteur ne distingue la visualisation des connaissances de l'art visuel que par les données utilisées.
  - Pour **W. Kienriech**, « la visualisation des connaissances utilise des représentations visuelles pour améliorer la communication des connaissances entre deux personnes ou plus ». Cette définition se rapproche de la définition originelle de R.A. Burkhard.

  En conclusion, les auteurs situent principalement la visualisation des connaissances dans le champ de la communication.
- **La visualisation d'ontologie formelle** peut être considérée comme un sous-domaine de la visualisation des connaissances, même si celui-ci n'est pas toujours présenté comme tel. La visualisation des ontologies a fait l'objet de deux revues [83, 42]. L'objectif est de faciliter la conception et la gestion des ontologies, et la navigation au sein des ontologies [83], ou bien d'aider à l'édition, l'inspection, l'apprentissage et le partage des ontologies [42].

  La plupart des approches présentent les relations hiérarchiques sous forme de graphes, deux cas particuliers bien connus étant les arbres et les listes indentées. Ces méthodes ont par exemple été appliquées pour l'anatomie [37] et pour les terminologies médicales [5]. Une étude récente [161] dans le domaine médical a comparé 8 outils de visualisation à base de graphes sur une ontologie médicale de grande taille. Les diagrammes de Venn et les *treemaps* ont aussi été utilisés et ont donné lieu à des variantes telles que Jambalaya [123] et CropCircle [186]. Cependant, une étude [84] comparant quatre modes de visualisation a montré que la liste arborescente indentée (présente de base dans l'éditeur Protégé), bien que simpliste, donnait les meilleurs résultats en termes de temps de réponse.
- **Plusieurs auteurs travaillant sur la visualisation d'objets culturels** (oeuvres d'art ou pièces historiques) se rangent également dans la visualisation des connaissances :

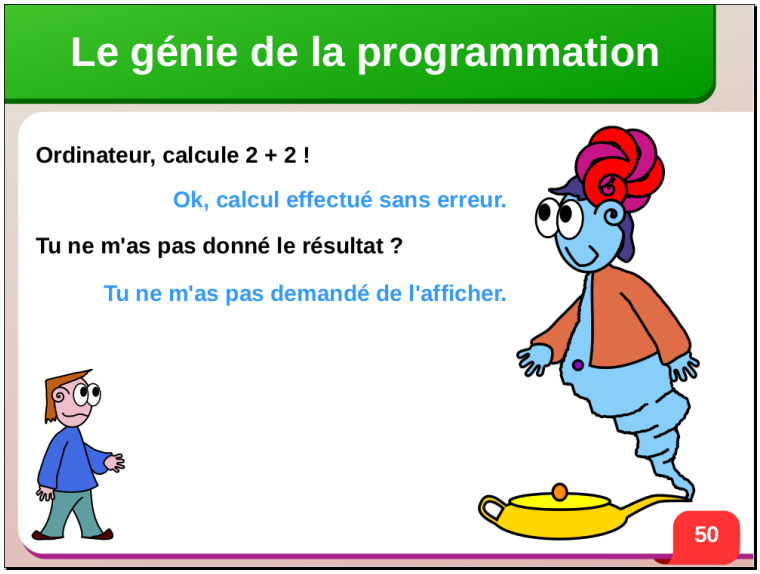


FIGURE 6.3 – Exemple de métaphore visuelle, issu de mes supports de cours de programmation. Le génie de la lampe d'Aladin, obéissant à la lettre mais dont l'obéissance même finit par se retourner contre son maître, est utilisé ici comme métaphore de l'ordinateur et des difficultés rencontrées lors de la programmation.

- **B. Yang** *et al.* [193] proposent une technique de visualisation inspirée de la cartographie pour visualiser des corpus littéraires, *via* une ontologie formelle.
- **L. Rajaonarivo** *et al.* [154] proposent un musée virtuel en 3D pour naviguer au sein d'un ensemble d'œuvres, les salles du musée se construisant au fur et à mesure du parcours de l'utilisateur en fonction de ses choix.
- **A. Hayashi** *et al.* [65] présente un outil pour visualiser les partitions de musique classique, en mettant en évidence les différents mouvements musicaux et les répétitions qu'elles comprennent.
- **T. Laska** *et al.* [120] cherchent à restaurer virtuellement (sur ordinateur) des peintures murales anciennes qui ont été fortement abîmées. La restauration est assistée par ordinateur en s'aidant de photos de peintures similaires mais mieux conservées, et l'objectif du projet n'est pas seulement de restaurer les peintures mais aussi d'acquérir de la connaissance sur les peintures de cette époque et les motifs et sujets associés – connaissance qui pourra ensuite servir pour de nouvelles restaurations.
- **T. Wyeld** [192] fait appel à la visualisation pour étudier des tableaux de la Renaissance. L'objectif est de comprendre quelles techniques les peintres de cette époque ont utilisées pour dessiner des perspectives complexes (par exemple des réflexions dans un miroir). La visualisation est utilisée afin de faire des expériences sur la perspective et de les comparer aux tableaux.

L'aspect « connaissance » est présent dans l'ajout de fonctionnalité à base de connaissances pour améliorer la visualisation, et/ou dans la volonté affichée de permettre à l'utilisateur d'acquérir des connaissances (par exemple dans le domaine historique, voir exemple Figure 6.2). Les auteurs ne donnent pas une définition précise de la visualisation des connaissances, cependant certains de ces travaux (notamment les derniers) s'éloignent des définitions précédentes.

- Enfin, l'utilisation de schémas, d'aides ou de métaphores visuels (Figure 6.3) est parfois considérés comme une forme très simple de visualisation des connaissances.

En conclusion de cet état de l'art, nous constatons tout d'abord que **la visualisation des connaissances n'a que rarement été proposée comme une approche d'aide à la décision** à ce jour, sinon de manière marginale (par exemple les travaux de R.A. Burkhard ciblent des décideurs mais ne s'inscrivent pas vraiment dans l'aide à la décision) ou limitée à des approches de fouille visuelle dans lesquelles la visualisation porte sur des données [45]. Dans la littérature, le terme « aide à la décision visuelle » (*visual decision support*) est parfois rencontré mais porte sur la visualisation de données ou d'information. Cependant, les choses évoluent rapidement puisque le congrès d'aide à la décision ICDSST 2017 (*International Conference on Decision Support System Technology*), qui s'est tenu à Namur, avait pour thème « *Data, Information and Knowledge*

Visualisation in Decision Making ».

Nous pouvons distinguer cinq grandes catégories de travaux relevant *a priori* de la visualisation des connaissances :

1. **La visualisation d'ontologie** visualise une ontologie formelle avec des techniques génériques, et a pour objectif l'édition, l'inspection ou la navigation dans l'ontologie.
2. **La visualisation des connaissances pour l'aide à la décision** visualise des connaissances structurées (voire formalisées) avec des méthodes prenant en compte les spécificités de ces connaissances (nature et/ou structure), et a pour objectif l'aide à la décision, la compréhension, l'apprentissage et la découverte de connaissances.

   Les travaux que nous avons présentés dans ce document relèvent de cette catégorie.
3. **La fouille visuelle de données** utilise des méthodes de visualisation de données et d'information, et a pour objectif la découverte de connaissances.
4. **Le transfert visuel de connaissances** présente visuellement les connaissances tacites d'un (ou plusieurs) individus à l'aide de métaphores, et a pour objectif le transfert de connaissances à d'autres individus.
5. **La visualisation d'objets culturels** visualise des pièces historiques ou des œuvres d'art sur ordinateur, et a pour objectif la diffusion de la culture, la restauration des oeuvres anciennes et la découverte de connaissances littéraires ou artistiques.

6.1.2 Positionnement par rapport aux autres disciplines

La différenciation entre *visualisation d'information* et *visualisation des connaissances* reste délicate. Pour rappel, nous avions défini la visualisation d'information comme « l'utilisation de représentations visuelles et interactives de données abstraites sur un ordinateur pour amplifier la cognition » (section 4.1). Nous pouvons donner trois différences principales :

- **La nature de ce qui est visualisé** : connaissance et information diffèrent sur plusieurs points qui impactent le processus de visualisation :
  - Le volume des connaissances est généralement plus faible que celui des informations.
  - La connaissance est souvent plus qualitative et moins quantitative que l'information.
  - La connaissance évolue suffisamment lentement pour que l'on puisse considérer les connaissances comme stables et immuables durant le temps de la visualisation. Au contraire, certaines informations évoluent à un rythme très rapide et doivent être mises à jour pendant la visualisation, par exemple le rythme cardiaque d'un patient doit être monitoré en temps réel.
- **L'objectif** : les objectifs de la visualisation des connaissances sont la découverte de connaissances, l'extraction, l'acquisition et la structuration des connaissances, la diffusion de la connaissance et de la culture, la compréhension et l'apprentissage, l'aide à la décision et la production artistique. Ces objectifs sont différents de ceux de la visualisation d'information, qui sont l'affichage d'une grande quantité d'information, la recherche d'une information donnée, la détection de motifs, le raisonnement visuel, la surveillance d'événements, et l'exploration du jeu de données (voir section 4.1).

  D'une manière plus générale, la visualisation des connaissances cherche à favoriser l'émergence de nouveaux modèles mentaux chez l'utilisateur, là où au contraire la visualisation d'information essaie de s'appuyer sur les modèles déjà préexistants [195].
- **Le format** : la visualisation d'information est définie comme interactive et faisant appel à un ordinateur ; au contraire nous avons défini la visualisation des connaissances comme pouvant avoir lieu sans ordinateur. Par exemple les boîtes arc-en-ciel du chapitre 4 peuvent être imprimées sur papier. Notons que la quasi-totalité des auteurs s'accorde sur ce point.

  L'une des raisons est que **l'interactivité joue un rôle limité dans la visualisation des connaissances**. La visualisation des connaissances met davantage en avant la vue d'ensemble (qui facilite la compréhension et la mémorisation), tandis que la visualisation d'information utilise l'interactivité pour obtenir davantage d'information sur un sous-ensemble du jeu de données (zoom, filtre et détails à la demande).

Lors du projet L3IM, nous avons obtenu des résultats d'évaluation contradictoires, avec de bons résultats pour les applications aux connaissances médicales (les guides de bonnes pratiques et le moteur de recherche, ainsi que les connaissances sur le médicament lors de ma thèse) et des résultats moins bons pour les applications à l'information (les dossiers patients). Ces résultats contradictoires montrent qu'il existe une différence entre visualisation d'information et visualisation des connaissances, puisqu'une visualisation donnée semble être efficace pour les connaissances mais moins pour l'information. En effet, l'information médicale est souvent plus détaillée, elle est exprimée à un niveau de granularité plus fin, et elle doit être associée à d'autres données. Par exemple, les pathologies du patient sont associées aux dates de début et de fin, que nous n'avons pas fait figurer sur l'icône.

Lors du projet VIIIP, la simple visualisation des propriétés des médicaments (contre-indications, effets indésirables,...) pourrait relever de la visualisation d'information. De fait, l'évaluation réalisée à la section 4.2.8 est davantage l'évaluation d'une technique de visualisation d'information. En revanche, l'objectif de VIIIP n'est pas seulement de faciliter l'accès aux propriétés des médicaments, mais aussi de permettre au médecin de positionner le nouveau médicament dans l'arsenal thérapeutique par rapport aux autres médicaments. Il s'agit donc d'aider le médecin à prendre la décision d'inclure ou non le nouveau médicament dans ses prescriptions futures, de faciliter la mémorisation, et de faire émerger de nouvelles connaissances au-delà des propriétés visualisées (par exemple la similarité des contre-indications au sein de chaque classe thérapeutique que nous mentionnions à la section 4.2.5.1). Dans cette perspective, il s'agit bien de visualisation les connaissances pour l'aide à la décision, ce qui correspond à notre seconde évaluation, section 5.2.3.

Lors du projet de recettes iconiques pour les remèdes en médecine traditionnelle, nos travaux ont clairement porté sur la visualisation des connaissances et non d'information.

Enfin, la visualisation des connaissances doit être distinguée de la *visualisation de domaine de connaissances* (*knowledge domain visualisation*), qui consiste à visualiser les différentes disciplines, leurs structures et leurs classifications, par exemple *via* les références bibliographiques de chacune.

6.2 Évaluation

Tout au long de ce document, et en particulier au chapitre 5, nous avons insisté sur l'importance de l'évaluation des techniques de visualisation et des applications intégrant la visualisation des connaissances. Dans la littérature, les techniques de visualisation sont souvent peu, voire pas, évaluées. Au contraire, dans le domaine médical, la question de l'évaluation est omniprésente depuis l'avènement de la médecine fondée sur les preuves (*Evidence-Based Medicine*, EBM) dans les années 80. Dans le domaine du médicament, de nombreux protocoles et outils statistiques ont été développés pour les essais cliniques. Par analogie, nous proposons une « visualisation fondée sur les preuves ».

Dans le chapitre 5, nous avons vu trois types d'évaluation différents :

1. **L'évaluation subjective** (parfois appelée satisfaction ou utilisabilité perçue), qui consiste à demander l'opinion des évaluateurs, laquelle est forcément subjective.
2. **L'évaluation objective de performances ou « outil visuel - outil non visuel »**, qui consiste à mesurer les performances obtenues par les évaluateurs avec l'outil visuel mis au point et à les comparer avec celles obtenues avec le même outil non visuel.
3. **L'évaluation objective de l'impact ou « avant - après »**, qui consiste à mesurer les décisions prises par les évaluateurs avec l'outil visuel mis au point et à les comparer avec celles prises avant l'utilisation de l'outil.

Dans la littérature, il a été montré que l'efficacité, l'efficience et la satisfaction de l'utilisateur ne sont pas nécessairement corrélées [52] : un utilisateur peut avoir le sentiment d'aller plus vite avec une interface ou de faire moins d'erreurs, sans que cela corresponde à la réalité. Ce problème a notamment été rencontré pour l'aide à la décision médicale [89]. Il est donc essentiel de combiner une évaluation subjective et une évaluation objective.

Le choix du type d'évaluation objective (« outil visuel - outil non visuel » ou « avant » - « après ») peut se faire en fonction de la disponibilité d'un outil non visuel similaire à celui que l'on souhaite évaluer : lorsqu'un tel outil existe, l'évaluation outil visuel - outil non visuel » est préférable, dans le cas contraire on

choisira l'évaluation « avant » - « après ». Les mesures (performances ou impact) doivent inclure une mesure de l'efficacité ou de la qualité du résultat (par exemple taux d'erreur ou décision de prescrire), et une mesure de l'efficience (temps de réponse) lorsque le temps est un facteur primordial. Dans le domaine médical, le temps est un facteur clef pour le médecin lorsqu'il est en consultation (cas du projet L3IM), en revanche, lorsqu'il est en situation de formation le temps est moins limité (cas du projet VIIIP). Le nombre total de cas, qui est égal au nombre d'évaluateurs multiplié par le nombre de scénarios, doit être suffisant pour atteindre une puissance statistique satisfaisante. En pratique, 20 évaluateurs et 10 scénarios (ou l'inverse) donnent un résultat acceptable. Avoir un nombre de scénarios relativement élevé implique de prendre les évaluateurs par groupe (comme lors de l'évaluation du site web VIIIP) plutôt qu'individuellement (comme lors des évaluations du projet L3IM).

Pour l'évaluation subjective, nous avons employé le test SUS (*System Usability Scale*) standard [26]. Ce test comprend 10 questions, chacune se présentant sous la forme d'une affirmation pour laquelle l'évaluateur doit indiquer s'il est pas du tout d'accord, plutôt pas d'accord, neutre, plutôt d'accord, ou tout à fait d'accord. Lors de nos évaluations, certains médecins ont eu des difficultés avec ce test et sont parfois restés perplexes. Certaines questions ne s'appliquent pas à toutes les interfaces (par exemple « Je pense que j'aurai besoin de l'aide d'un technicien pour être capable d'utiliser le système » n'a pas beaucoup de sens pour la simple consultation d'un site web car on imagine mal un médecin faire appel à un technicien dans cette situation). D'autres questions sont très proches et les évaluateurs ne comprennent pas la différence entre elles (par exemple « Je trouve le système inutilement complexe » et « Je trouve le système très lourd à utiliser », « J'ai besoin d'apprendre beaucoup de choses avant de pouvoir utiliser le système », « J'imagine que la plupart des gens seraient capables d'apprendre à utiliser le système très rapidement »). Enfin, le nombre important de questions (10) rend le test assez laborieux en pratique, surtout lorsqu'il est nécessaire de faire passer plusieurs tests SUS pour comparer différentes interfaces. Le test SUS a donc l'avantage d'être standard et de permettre la comparaison de différentes interfaces entre elles, cependant nous pensons qu'il pourrait être amélioré et simplifié.

Par ailleurs, il nous semble important de séparer l'évaluation des techniques de visualisation proprement dite (comme celle réalisée sur les boîtes arc-en-ciel, section 4.2.8) de l'évaluation des applications intégrant des approches visuelles mais ne se limitant pas à celles-ci (comme celle réalisée sur le site web VIIIP, section 5.2.3). Ces deux évaluations auraient très bien pu donner des résultats différents, par exemple les médecins auraient pu ne pas comprendre les boîtes arc-en-ciel mais apprécier malgré tout l'approche comparative du site web VIIIP.

6.3 Multidisciplinarité

Les recherches décrites dans ce document s'inscrivent dans un cadre multidisciplinaire : informatique « pure », informatique médicale, sciences du médicament, sans oublier une dimension artistique lorsqu'il s'agit de dessiner des pictogrammes ou de concevoir les visualisations. J'ai appris l'essentiel de mes connaissances en informatique en autodidacte, ma formation initiale étant en pharmacie. Je pense néanmoins avoir développé et décrit dans ce document des recherches en informatique « pure », dans les domaines de la représentation des connaissances, de la visualisation et de l'aide à la décision.

Dans ce cas, **que m'a apporté ma formation initiale de pharmacien pour mener des recherches en informatique ?** Je pense que celle-ci s'est révélée utile à quatre niveaux :

1. **Mes études de pharmacien m'ont tout d'abord apporté une bonne connaissance du domaine médical, et plus particulièrement du médicament**. Le domaine du médicament est un domaine très complexe, et qui est trop souvent abordé d'un point de vue purement économique. Cela constitue une grave erreur, car le médicament n'est pas un produit comme les autres.

   De plus, maîtriser le domaine médical est intéressant car nous avons vu que ce domaine de connaissance présentait des avantages pour l'informatique, en termes de complexité et d'enjeux sociétaux (voir section 2.1.2).

2. **Mes connaissances médicales ont aussi beaucoup facilité mes contacts avec les médecins et les professionnels de santé**.

3. **Ma formation de professionnel de santé m'a conduit à avoir une approche très ascendante (*bottom-up*) des problèmes en informatique**. Par exemple, le module pour la programmation orientée ontologie décrit en section 2.3 est une réponse aux problèmes pratiques que j'ai rencontrés lors de l'utilisation de l'ontologie du projet VIIIP.
4. **Enfin, mes études médicales m'ont apporté une forme de rigueur, qui est très présente dans la recherche biomédicale**, y compris en informatique médicale, et paradoxalement parfois moins en informatique. Voici trois exemples :
   — **En matière d'évaluation**, nous avons vu à la section 6.2 l'importance de celles-ci dans le monde médical. En particulier, les protocoles mis au point pour les essais cliniques ont souvent été adaptés aux évaluations de logiciels médicaux et nous nous en sommes inspirés pour les évaluations menées au chapitre 5 (par exemple pour le plan d'étude croisé ou *cross-over*). D'autres auteurs [125] ont constaté que les évaluations mesurant les performances des outils visuels ont été influencées par les méthodes issues des sciences naturelles.
   — **En matière de rédaction**, les articles scientifiques médicaux suivent une structure toujours identique : introduction, matériel, méthodes, résultats, discussion et conclusion. La séparation des méthodes d'un côté et des résultats de l'autre n'est pas toujours facile, mais elle est intéressante car elle oblige à s'interroger sur les méthodes et à les décrire noir sur blanc, alors que l'on a souvent tendance à négliger les méthodes au profit des résultats.
   Par exemple, peu d'articles de visualisation décrivent les méthodes qui ont permis de concevoir les techniques de visualisation (à la manière de ce que nous avons fait à la section 4.2.2 pour les boîtes arc-en-ciel). La mise au point des techniques de visualisation est même parfois considérée comme un « art » ; nous pensons malgré tout qu'il est possible de décrire au moins partiellement les méthodes de mise au point, et que ces méthodes pourraient être réutilisées dans d'autres contextes.
   — **En matière de bibliographie**, la base MedLine accessible *via* PubMed est certainement la base bibliographique la plus facile d'accès et la mieux structurée.

6.4 Éthique

De nombreux travaux ont été accomplis dans le domaine de l'aide à la décision médicale, la plupart faisant appel à des avis d'experts ou mettant en œuvre des méthodes d'intelligence artificielle afin d'automatiser le raisonnement médical. Dans ce document, nous avons proposé une approche radicalement différente : la visualisation des connaissances.

Cette approche « visualisation » nous semble plus éthique que l'approche « raisonnement automatique », car elle permet de s'assurer qu'un être humain « reste dans la boucle ». En effet, la visualisation des connaissances nécessite un être humain (pour regarder la visualisation), alors que ce n'est pas le cas des approches automatiques où l'être humain reste passif, son rôle se limitant à recevoir le résultat et éventuellement à « nourrir » le système avec les données nécessaires. Ce rôle, peu gratifiant, est d'ailleurs souvent mal accepté par les professionnels de santé.

De nombreux scénarios de type « science-fiction » décrivent des univers où l'ordinateur pourrait devenir le maître. La question des données de santé, de leur diffusion et de leur réutilisation est devenue cruciale aujourd'hui. Ces données sont importantes pour la prise en charge médicale du patient, mais elles intéressent aussi les chercheurs du domaine biomédical et le secteur assuranciel (ce qui est plus inquiétant). Depuis les révélations d'Edward Snowden, le recueil massif de données personnelles à l'insu des individus concernés n'est plus de la science-fiction. Garder un être humain dans la boucle permet de restreindre les traitements effectués sur les données (car l'intervention d'un humain a un coût) et de s'assurer qu'il y a bien une personne responsable.

De plus, nous pensons que les outils automatiques peuvent présenter un risque sur le long terme. En effet, les outils d'aide à la décision automatique conduisent à un « biais d'automatisation » (*automation bias* [55]) : le médecin devient « passif » et accepte les recommandations du système sans réfléchir. Aussi, si les médecins utilisaient massivement des outils automatiques pour réaliser les diagnostics et choisir les traitements, ils risqueraient de perdre leurs connaissances à force de se reposer sur ces outils. Dans ce cas, des années plus

tard, lorsqu'il faudra effectuer la mise à jour de la base de connaissances du système suite à l'évolution des maladies, des traitements disponibles et des modes de vie, comment ferons-nous pour construire une nouvelle base de connaissances si les médecins ont perdu leur expertise ?

Au contraire, la visualisation des connaissances permet de faciliter le travail du médecin sans courir le risque que celui-ci perde ses connaissances, puisque le raisonnement reste intégralement celui du médecin. L'intelligence artificielle peut alors être utilisée, non pour l'aide à la décision, mais pour préparer les connaissances avant de les visualiser. Par exemple, dans le projet VIIIP, nous avons utilisé une ontologie formelle pour structurer les connaissances et effectuer des raisonnements sur les contre-indications, avant la visualisation.

Les questions que nous soulevons ici sont particulièrement importantes dans le domaine médical, cependant elles ne sont pas limitées à ce domaine. D'une manière générale, **il me semble important que la décision finale reste du ressort de l'humain, mais également le processus de décision**.

6.5 Conclusion

Dans ce document, nous avons proposé une démarche en 4 étapes pour la visualisation des connaissances : la représentation des connaissances, la visualisation iconique, la visualisation structurelle et la conception d'application. Nous avons présenté des contributions méthodologiques aux différentes étapes, et notamment :

- une méthodologie de modélisation du domaine en santé,
- un module pour la programmation orientée ontologie,
- une méthode pour formaliser la sémantique des langages iconiques,
- une nouvelle technique de visualisation d'information pour les ensembles non disjoints,

Ces apports méthodologiques ont ensuite été appliqués dans 3 projets intégrant la visualisation des connaissances dans le domaine biomédical :

- **Dans le projet L3IM**, nous avons appliqué le langage iconique VCM, initialement conçu pour les connaissances sur les médicaments, dans d'autres contextes, et notamment aux guides de bonnes pratiques cliniques (recherche d'un guide puis consultation de celui-ci).
- **Dans le projet sur les recettes en médecine traditionnelle africaine**, nous avons montré que les principes de construction de langage iconique mis au point pour VCM pouvaient être réutilisés pour la conception de nouveaux langages iconiques.
- **Dans le projet VIIIP**, nous avons conçu des outils pour la visualisation des propriétés des nouveaux médicaments et la comparaison visuelle avec les propriétés des médicaments déjà existants dans la même indication. Nous avons ensuite montré que cette visualisation des connaissances permettait au médecin de se forger une opinion sur le nouveau médicament, de le positionner par apport aux médicaments existants et de décider s'il le prescrira ou non à l'avenir.

Nous pouvons donc en tirer les enseignements suivants :

- Il est possible de visualiser des connaissances complexes abstraites telles que les propriétés des médicaments ou les recommandations des guides de bonnes pratiques cliniques, à l'aide d'icônes et/ou de techniques de visualisation d'information.
- Les méthodes mises au point, de même que les outils et langages réalisés, ne sont pas spécifiques d'une application donnée. Ils peuvent être réutilisés dans d'autres applications (cas de VCM par exemple).
- La présentation visuelle des connaissances peut faciliter l'accès aux connaissances et l'aide à la décision, et cette approche est bien acceptée en particulier parce qu'elle permet au médecin de garder le contrôle du raisonnement médical.

Par conséquent, nous pensons que la visualisation des connaissances est une alternative valable pour l'aide à la décision, notamment dans le domaine médical.

Enfin, bien que nous ayons travaillé sur un domaine applicatif médical, les méthodes que nous avons proposées sont d'une portée plus générale. Les connaissances médicales étant parmi les plus complexes aujourd'hui, nous pensons que nos conclusions peuvent être généralisées bien au-delà de ce domaine.

Chapitre 7

Projet de recherches : vers la visualisation du raisonnement

Bel companho, si dormetz o velhatz,
Non dormatz plus, suau vos ressidatz,
Qu'en orïent vei l'estela creguda,
Qu'amena'l jorn, qu'eu l'ai ben conoguda
Et ades sera l'alba
(Reis glorios, chant de troubadour de Guiraut de Bornelh, manuscrit fr 22543 8v, XIII^{ème} *siècle)*

Dans ce chapitre, je souhaite exposer mes perspectives de recherches et les situer dans le cadre de la visualisation des connaissances. Au laboratoire LIMICS, la création d'un groupe consacrée à la visualisation des connaissances a déjà fait l'objet de discussions.

Tout au long de ce document, j'ai montré l'intérêt de la visualisation des connaissances, notamment dans le domaine biomédical et pour l'aide à la décision. J'ai également mis en évidence la **continuité** qui existe entre la *représentation* des connaissances et la *visualisation* des connaissances : d'une part, visualiser les connaissances nécessite de les avoir au préalable représentées avec un formalisme suffisamment expressif pour pouvoir générer automatiquement les présentations visuelles. D'autre part, la visualisation des connaissances est utile en représentation des connaissances, notamment pour aider à l'acquisition et au transfert des connaissances.

Lors de mes recherches futures, je souhaiterai continuer à mener de front des recherches en représentation et visualisation des connaissances. J'envisage de m'intéresser au *raisonnement*, à la fois en représentation des connaissances (raisonnement sur les ontologies) et en visualisation (visualisation du raisonnement).

En ce qui concerne la représentation des connaissances, j'ai présenté en section 2.3 Owlready, un outil pour la programmation orientée ontologie en Python intégrant des constructeurs de haut niveau répondant aux besoins particuliers des ontologies biomédicales. Cet outil répond à un réel besoin dans la communauté Python, car il n'existait aucun autre outil pour accéder à des ontologies OWL dans ce langage. Je souhaiterai aller vers un « éco-système » complet pour la construction et l'utilisation d'ontologies en Python. Owlready est une première étape essentielle.

Pour l'instant, Owlready utilise HermiT pour le raisonnement automatique. **La seconde étape serait de disposer d'un raisonneur OWL en Python**. D'un point de vue purement pratique, cela permettrait d'enlever la dépendance au langage Java (utilisé par hermiT). D'un point de vue scientifique, un raisonneur en Python serait beaucoup plus facile à comprendre, à modifier et à faire évoluer : en effet, le code source d'HermiT est d'une complexité et d'une taille (45 000 lignes) qui le rendent difficile d'accès.

Dans un premier temps, je propose d'implémenter et de tester les algorithmes de raisonnement existant, par exemple à base de tableaux [169] et d'hyper-tableaux [137].

Dans un second temps, j'envisage d'étendre les fonctionnalités de ce raisonneur pour ajouter la prise en charge de nouveaux constructeurs répondant à des situations problématiques fréquemment rencontrées dans

le domaine médical. Par exemple, le langage OWL ne permet pas de définir une relation entre deux classes. Cependant, c'est une situation que l'on rencontre fréquemment dans le domaine biomédical : nous avons vu à la section 2.1.4 que les principaux concepts médicaux (maladies et médicaments notamment) doivent être représentés par des classes à cause des relations d'héritage existantes. Il est alors impossible d'établir une relation directe entre eux. Par exemple, nous ne pouvons pas asserter que « toutes les Aspirines sont contre-indiquées avec toutes les Hémorragies ». OWL permet seulement d'asserter que « toutes les Aspirines sont contre-indiquées avec *au moins* une Hémorragie » (restriction existentielle, *some* dans Protégé) ou que « toutes les Aspirines sont contre-indiquées avec *seulement* des Hémorragies » (restriction universelle, *only* dans Protégé).

En ce qui concerne la visualisation, les travaux ont historiquement commencé par la *visualisation de données* numériques. Ils se sont ensuite poursuivis par la *visualisation d'informations* abstraites, puis plus récemment par la *visualisation des connaissances*. **L'étape suivante me semble être la *visualisation du raisonnement***, c'est-à-dire la traduction visuelle d'un raisonnement qui n'est pas de nature visuelle. L'objectif est de pouvoir reproduire ce raisonnement visuellement, si possible de manière pré-attentive (donc en une fraction de seconde, dans un temps qui est indépendant de la quantité d'information visuelle présentée).

Dans ce domaine, mes travaux préliminaires sur la visualisation du raisonnement d'un perceptron (neurone formel) à l'aide de boîtes arc-en-ciel se sont révélés très prometteurs. Ceux-ci ont abouti à deux « *best papers* » au cours de deux conférences, à la fois en visualisation (International Conference Information Visualization iV2017 [111]) et en informatique médicale (Medical Informatics Europe MIE2017 [181]).

Dans un premier temps, je souhaite proposer un cadre conceptuel pour la visualisation du raisonnement. Une approche de visualisation du raisonnement pourrait se décrire comme la visualisation combinée d'une information I, d'une connaissance C et de la mise en relation R de I avec C. Des travaux déjà existants, notamment sur le raisonnement à partir de cas (RAPC) visuel, relèvent probablement de la visualisation du raisonnement et devront être intégrés. Une typologie des raisonnements visuels possibles de manière pré-attentive devra aussi être associée. Le rôle de l'interactivité sera également à définir.

Je souhaite également développer davantage les boîtes arc-en-ciel [101], une technique de visualisation d'information pour les ensembles non disjoints (ou chevauchant) que j'ai mis au point précédemment. Les développements se feront notamment dans cette direction de la visualisation du raisonnement.

- Une première piste est de poursuivre les travaux préliminaires mentionnés ci-dessus, pour aller vers la visualisation du raisonnement de réseaux de neurones simples : d'abord des réseaux avec plusieurs neurones de sortie, puis avec une couche cachée. Les résultats pourraient trouver une application dans les projets de recherche RaMiPa sur sur l'aide à la prescription en antibiothérapie et PEPS sur l'aide à la prévention cardiovasculaire du laboratoire LIMICS.
- Une seconde piste est l'adaptation des boîtes arc-en-ciel au raisonnement à partir de cas visuel. Il existe déjà des approches de ce type, par exemple s'appuyant sur l'analyse en composantes principales (ACP), le *clustering*, ou les cartes de Kohonen. Ces approches montrent la similarité d'un cas aux autres cas déjà solutionnés, cependant elles n'expliquent pas à l'utilisateur *pourquoi* (ou *en quoi*) ces cas sont similaires. La technique des boîtes arc-en-ciel pourrait être employée pour montrer les caractéristiques partagées par les différents cas. Ceci pourrait se faire dans le cadre du projet européen DESIREE sur le cancer du sein, auquel je participe.
- Une troisième piste est l'utilisation des boîtes arc-en-ciel (et de l'algorithme d'ordonnancement sous-jacent) pour aider à l'établissement de classifications. Lors de travaux antérieurs sur l'utilisation de boîtes arc-en-ciel pour visualiser des extraits de Gene Ontology (voir section 4.2.6), nous avons montré que les boîtes arc-en-ciel permettaient de reconstruire les classifications existantes, par exemple celle des histones. Cette capacité devra être évaluée plus précisément. Un exemple d'application possible est l'établissement de classifications pour des maladies, à partir de leurs signes et de leurs caractéristiques (gènes impliqués notamment), par exemple la classification des spondylarthropathies [184] est encore une question de recherche. La phylogénie est aussi un domaine d'application envisageable.

D'autres applications possibles de la visualisation du raisonnement sont :

- La visualisation du raisonnement d'une ontologie : les raisonneurs existants peuvent produire des inférences à partir d'une ontologie et, pour certains d'entre eux (tel que Pellet), ils peuvent expliquer

ces inférences. Cependant, l'explication reste sommaire et se limite généralement à la liste des axiomes qui ont permis d'aboutir à l'inférence. La visualisation du raisonnement réalisée faciliterait grandement la compréhension par un humain.

— La visualisation du raisonnement médical à destination du patient : il s'agit d'expliquer au patient pourquoi un traitement (par exemple médicamenteux) lui est recommandé et quel est le mécanisme d'action pharmacologique de ce traitement. Ces deux points constituent des éléments importants de l'éducation thérapeutique, qui ont un impact sur l'observance du patient et donc sur l'effet du traitement. Pour cela, nous pouvons envisager une approche visuelle combinant des schémas, un langage diagrammatique et iconique, et des métaphores visuelles.

Enfin, les métaphores visuelles permettent également une forme de raisonnement par analogie, en comparant une situation nouvelle à une « situation prototype » bien connue de tous. Leur utilisation pour la visualisation du raisonnement est un point à étudier.

À plus long terme, je souhaite réaliser **une plate-forme générique intégrée pour la visualisation des connaissances et du raisonnement, et l'évaluation des applications visuelles**. Nous avons vu qu'il existe trois grandes approches pour la visualisation :

1. Les techniques de visualisation d'information, qui permettent de visualiser des structures d'information particulière : tableau, arbres, réseaux, etc.
2. Les langages iconiques, qui représentent la nature de l'information à l'aide d'icônes composées de pictogrammes, de formes géométriques et de couleur, par exemple les panneaux routiers ou le langage iconique médical VCM pour représenter des états du patient et des prises en charge médicale.
3. Les métaphores visuelles, qui permettent un raisonnement par analogie en comparant une situation nouvelle à une « situation prototype » bien connue de tous. Un exemple est la métaphore bien connue de l'iceberg (dont la majeure partie est sous les flots) pour signifier qu'une partie importante d'une chose reste cachée ou inaccessible.

Tout au long de ce document, nous avons décrit différentes méthodes visuelles, issues de la littérature et de nos propres travaux. Ces méthodes ont donné des résultats intéressants lors des évaluations que nous avons réalisées. Cependant, toutes ont un champ d'application restreint. Chaque langage iconique est focalisé sur un domaine précis, chaque technique de visualisation est dédié à un type de structure d'information particulière, et chaque métaphore ne permet qu'un seul raisonnement. Ainsi, le langage VCM est limité à l'expression des états du patient (pathologies, classes d'âge,...) et des prises en charge médicale (traitement, surveillance). De même, les boîtes arc-en-ciel sont limitées à la visualisation d'ensembles non disjoints. La grande majorité des approches visuelles actuelles restent donc spécifiques d'un type de connaissance, caractérisé soit par un domaine précis (pour les langages iconiques) soit par une structure particulière (pour les techniques de visualisation).

Je souhaite donc aller vers une plate-forme générique à même de supporter des connaissances différentes, en termes de domaine et de structure. Voici les principales caractéristiques qu'aurait cette plate-forme :

1. **Elle sera orientée connaissance** (et non information en général). En effet, nous avons vu que la visualisation des connaissances présente plusieurs différences avec la visualisation d'information, telles que la faible importance de l'interactivité ou la part importante de variables qualitatives. Nous avons aussi vu lors des évaluations du projet L3IM qu'une méthode efficace pour les connaissances ne l'était pas nécessairement pour l'information. Il apparaît donc important d'étudier séparément la visualisation de la connaissance et de l'information.
2. **Elle combinera les trois grandes approches visuelles :**
   (a) **les langages iconiques**,
   (b) **la visualisation d'information**,
   (c) **les métaphores visuelles**.

   La complémentarité des métaphores avec les langages iconiques et/ou la visualisation d'information sera un sujet à étudier.

3. **Elle prendra en compte les spécificités du domaine**. Ce point est nécessaire en particulier pour la visualisation iconique. En effet, il existe des outils de visualisations génériques, par exemple pour représenter une ontologie sous la forme d'un graphe, cependant ces outils ne prennent pas en compte le domaine et aboutissent parfois à des visualisations peu adaptées.
4. **Elle sera axée sur les connaissances explicites**, au moins dans un premier temps. Nous avons vu que la visualisation des connaissances pouvait également porter sur des connaissances implicites telles que des œuvres d'art, cependant les méthodes mises en jeu dans ce cas diffèrent sensiblement et mon expérience est très limitée dans ce domaine.
5. **Elle facilitera le développement d'outils visuels**, à la fois en aidant au choix et à la réutilisation de techniques de visualisation existantes, et en aidant à la mise au point de nouvelles techniques.
6. **Elle cherchera à « visualiser le processus de visualisation »**. Nous envisagerons des approches visuelles pour aider l'utilisateur à construire de nouvelles visualisations.
7. **Elle aidera également à l'évaluation des applications de visualisation des connaissances ou du raisonnement**.

Pour faciliter l'élaboration de visualisations iconiques, la plate-forme pourrait proposer un jeu de pictogrammes réutilisables, soit généralistes (par exemple des conventions bien établies comme le signe « ? » pour marquer l'interrogation) soit associés à un domaine donné (par exemple biomédical ; les pictogrammes d'organes de VCM pourraient fournir une première base). Ensuite, un outil automatique pourrait générer les icônes en s'appuyant sur une représentation formelle de la sémantique et de la syntaxe du langage iconique. Cette représentation formelle s'appuierait sur le formalisme à base d'ontologie que nous avons proposée (section 3.3) qui nécessiterait d'être complété sur les aspects syntaxiques (afin de décrire précisément comment les différents pictogrammes sont positionnés les uns par rapport aux autres). Enfin, un outil visuel pourrait faciliter cette étape de formalisation, ce qui éviterait d'avoir à la réaliser manuellement dans un éditeur comme Protégé.

Pour faciliter l'élaboration de visualisations structurelles, la plate-forme pourrait tout d'abord proposer plusieurs techniques de visualisation bien connues (graphes, *treemap*, boîtes arc-en-ciel,...). Le choix de la technique peut se faire en partie selon la structure des connaissances ; nous pouvons envisager un algorithme qui, à partir d'une ontologie, calcule différentes métriques (nombre de classes, niveau de profondeur, présence d'héritage multiple,...) puis recommande une ou plusieurs techniques.

Une autre approche consisterait à faciliter la *création* de nouvelles techniques de visualisation, en s'inspirant de la méthode que nous avons suivie pour la mise au point des boîtes arc-en-ciel (voir Figure 4.7). Il s'agirait de partir d'une visualisation simple, par exemple un tableau, puis d'appliquer différentes étapes, telles que le réordonnancement des lignes ou des colonnes selon un critère donné, l'ajout de couleurs,... À chaque étape, le système pourrait proposer et illustrer visuellement les différentes options possibles. L'utilisateur clique sur celle de son choix, jusqu'à aboutir à une visualisation satisfaisante. Un tel système permettrait de tester très rapidement diverses approches de visualisation ; il pourrait aussi avoir des applications dans le domaine artistique puisqu'il permettrait de construire une visualisation de manière aléatoire (en choisissant au hasard les étapes).

Dans la littérature, des projets ont proposé de structurer et de formaliser la visualisation, sans cependant aller jusqu'à la visualiser. Un exemple est Mondrian [133], qui s'apparente à un langage de programmation dédié à la visualisation.

Pour faciliter l'utilisation des métaphores visuelles, la plate-forme pourrait inclure une ontologie des métaphores visuelles, associée à une librairie d'images et un site web. Actuellement, chaque métaphore est conçue indépendamment pour un usage unique, et dessinée à la main. Cependant, une même métaphore, comme celle du génie d'Aladin en Figure 6.3 ou celle de l'iceberg, peut être réutilisée dans différents contextes.

Le site web proposerait des modalités pour rechercher une métaphore en fonction du message à faire passer, du thème abordé ou des exemples d'utilisation déjà réalisés. La formalisation de la sémantique des

métaphores me paraît complexe et relève certainement d'une question de recherche, elle pourrait s'inspirer des travaux que nous avons présentés ici sur la formalisation des langages iconiques.

Le site web permettrait ensuite à l'utilisateur d'adapter facilement la métaphore choisie, en modifiant les textes mais aussi certains éléments comme les personnages (par exemple transformer le personnage de la Figure 6.3 en enseignant, en médecin ou en policier tout en gardant la position et l'expression du personnage propre à la situation).

Enfin, l'évaluation de l'utilisation et de l'impact des métaphores est à considérer. En particulier, la réutilisation de la même métaphore à plusieurs reprises peut faciliter sa compréhension, mais aussi provoquer une réaction négative de « déjà vu ».

Cette plate-forme se veut générique ; nous pourrons néanmoins la développer au travers d'applications biomédicales et dans un contexte multidisciplinaire. Nous proposons pour cela une démarche en deux étapes :

1. Dans un premier temps, chercher à résoudre un problème de visualisation concret du domaine biomédical.
2. Dans un second temps, chercher à généraliser les résultats obtenus au-delà de la problématique biomédicale de départ.

C'est cette démarche que nous avons suivie pour la mise au point des boîtes arc-en-ciel : premièrement, nous avons cherché à résoudre un problème de visualisation médical : la comparaison des propriétés des médicaments, ensuite nous avons généralisé les résultats obtenus pour aboutir à une technique de visualisation des ensembles non disjoints. Cette technique a ensuite pu être appliquée aux propriétés des acides aminés, à des extraits de Gene Ontology, *etc.*

Chapitre 8

CV détaillé

Et tiens pour vrai, ma tendre belle
Tous les vieux jours des mois heureux
Où les amis, la parentèle,
En voyant l'un, nous pensaient deux
(L'amour est fleur, Glenmor)

Informations générales

État civil

Lamy Jean-Baptiste
Né le 18 juin 1979 à Paris (75016), nationalité française, célibataire.

Coordonnées professionnelles

Courriel : jean-baptiste.lamy@univ-paris13.fr
Téléphone : 01 48 38 73 34
Adresse :
LIMICS, bureau 149
UFR SMBH
74 rue Marcel Cachin
93017 Bobigny

Poste actuel

Maître de Conférences – Université Paris 13, Sorbonne Paris Cité, **titulaire de la PEDR** (Prime d'Encadrement Doctoral et de Recherche).

Composante : **UFR SMBH** (Santé Médecine et Biologie Humaine).

Laboratoire de recherche : **LIMICS** (Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, UMRS 1142, laboratoire de l'université Paris 13, l'université Paris 6 et de l'INSERM).

Domaine de recherche

Mes recherches ont commencé il y a 15 ans en informatique médicale. Elles s'orientent depuis plusieurs années vers l'informatique en général, et plus particulièrement la visualisation d'information et l'ingénierie des connaissances, avec un domaine d'application principalement médical. Mes thématiques portent plus spécifiquement sur la visualisation des connaissances, le web sémantique, l'aide visuelle à la décision, l'aide à la décision médicale et les connaissances sur le médicament. Il s'agit donc de recherches fortement multidisciplinaires.

Mes travaux consistent à mettre au point de nouvelles méthodes de représentation et de visualisation des connaissances, à formaliser et à visualiser les connaissances, notamment médicales, à l'aide de ces nouvelles méthodes ou de méthodes déjà existantes, et à évaluer les applications ainsi obtenues en situation d'aide à la décision médicale. Le domaine d'application principal de mes travaux est la santé, et plus particulièrement les connaissances sur le médicament. Cependant les méthodes développées ne sont pas spécifiques à ce domaine et peuvent être (et ont été pour certaines) appliquées bien au-delà.

Formation et diplômes

Début	Fin	Diplôme
1/10/2003	19/12/2006	**Thèse de doctorat**, université Pierre et Marie Curie - Paris 6, spécialité santé publique et science de l'information biomédicale, titre de la thèse : *Conception et évaluation de méthodes de visualisation des connaissances médicales : Mise au point d'un langage graphique et application aux connaissances sur le médicament.* Soutenue le 19/12/2006 devant le jury ainsi composé : — Président : Alain-Jacques Valleron (professeur, université Pierre et Marie Curie - Paris 6) — Rapporteurs : Stéphan Darmoni (professeur, université de Rouen), Marius Fieschi (professeur, université de la Méditerranée) — Examinateurs : Pierre Zweigenbaum (professeur associé, INALCO) — Directeur : Alain Venot (professeur, université Paris 13) Mention : Très honorable. Financement : Bourse ministérielle (allocataire de recherche).
1/09/1998	30/06/2006	**Doctorat en Pharmacie**, université Joseph Fourier - Grenoble 1, titre de la thèse : *Vers un langage graphique de représentation des connaissances sur les médicaments.* Soutenue le 30/6/2006 devant le jury ainsi composé : — Président : Vincent Danel (médecin hospitalier) — Membres : Vincent Rialle (maître de conférences, université Joseph Fourier - Grenoble 1), Bernard Champon (professeur associé), Jean Calop (professeur, université Joseph Fourier - Grenoble 1) — Directeur : Alain Venot (professeur, université Paris 13)
	2006	**DU de valorisation de la recherche**, université Pierre et Marie Curie - Paris 6.
	2003	**DEA d'informatique biomédicale**, université Pierre et Marie Curie - Paris 6, mention : AB.
	1998	**Baccalauréats** série S option Math, Chambéry, mention : B.

Expérience professionnelle

Début	Fin	Fonction	Durée
1/09/2009	-	**Maître de conférence** en informatique à l'université Paris 13, Sorbonne Paris Cité, UFR SMBH. Titulaire de la PEDR. Laboratoire de recherche : LIMICS, directrice Marie-Christine Jaulent (anciennement LIM&BIO, directeur Alain Venot, qui a par la suite fusionné et donné naissance au LIMICS).	-
1/01/2009	1/09/2009	**Chargé de recherche** (CDD dans le public au laboratoire LIM&BIO, université Paris 13) sur le projet L3IM. Responsable : Alain Venot.	8 mois
1/09/2008	1/01/2009	**Ingénieur de recherche** (CDD dans le privé à la SFTG, Société de Formation Thérapeutique du Généraliste) sur les projets ASTI 2 et 3. Responsable : Hector Falcoff.	4 mois
1/09/2007	1/09/2008	**Demi-ATER** en statistiques à l'université Paris 13, UFR SMBH.	1 an
1/11/2006	1/05/2007	**Chargé de recherche** (CDD dans le public au laboratoire LIM&BIO, université Paris 13) sur le projet EI-xplore. Responsable : Alain Venot.	6 mois
1/10/2006	1/09/2007	**Ingénieur de recherche** (CDD dans le privé à la SFTG) sur les projets ASTI 2 et 3. Responsable : Hector Falcoff.	11 mois
1/10/2003	1/10/2006	**Allocataire de recherche** (Université Pierre et Marie Curie - Paris 6). Directeur de thèse : Alain Venot.	3 ans
1/01/2003	1/07/2003	**Stage de DEA** (Université Joseph Fourier - Grenoble 1). Responsable : Vincent Rialle.	6 mois
1/09/2001	1/03/2002	**Stage de 5ème année de pharmacie option recherche** (Université Joseph Fourier - Grenoble 1). Responsable : Vincent Rialle.	6 mois

8.1 Parcours de recherche

Mon parcours de recherche est atypique car, bien que mes recherches relèvent actuellement de l'informatique, mon cursus commence en pharmacie. J'ai cependant appris l'informatique et en particulier la programmation bien plus tôt, dès l'âge de 10 ans, lorsque je m'étais fixé pour objectif de programmer des jeux vidéos avec mon frère, ce qui m'a amené à découvrir le monde du logiciel libre.

Au cours de mes études universitaires, je me suis assez vite rendu compte que je n'étais pas fait pour être pharmacien d'officine. Après quelques tentatives peu concluantes dans la recherche en chimie au cours de stages, j'ai décidé de faire une 5$^{\text{ème}}$ année de pharmacie option recherche en informatique médicale.

Stage de 5$^{\text{ème}}$ année de pharmacie option recherche (1/09/2001 - 1/03/2002) L'option recherche comprenait un stage de 6 mois que j'ai effectué au laboratoire TIMC-IMAG dans l'équipe AFIRM sous la direction de Vincent Rialle. L'équipe travaillait sur les habitats intelligents en santé et avait reconstitué un prototype d'appartement-témoin fonctionnel pour la surveillance à domicile de patients âgés et/ou atteints de pathologies chroniques telles que la maladie d'Alzheimer. Plusieurs capteurs infrarouges permettaient de suivre les mouvements du patient, cependant il n'était pas possible d'utiliser des caméras pour des raisons évidentes de protection de la vie privée. Au cours de mon stage, j'ai travaillé sur la visualisation à distance du patient au sein de l'habitat intelligent en réalité virtuelle et en réalité augmentée. Il s'agissait de reconstituer une vue en 3D du patient et de l'habitat à partir des données des capteurs de position, et d'enrichir cette visualisation à partir des autres capteurs. Pour cela, j'ai utilisé un moteur 3D en Java et en OpenGL, moteur que j'avais initialement développé pour programmer des jeux vidéos à titre personnel.

Ce stage m'a permis de trouver ma voie ; j'ai ensuite eu envie de poursuivre dans cette direction et je me suis inscrit en DEA d'informatique biomédicale.

Stage de DEA (1/01/2003 - 1/07/2003) L'année suivante, j'ai effectué mon stage de DEA au laboratoire TIMC-IMAG, toujours sous la direction de Vincent Rialle. Cette fois-ci mon travail a porté sur la reconnaissance de scénarios temporels au sein des données des capteurs. Pour cela, je me suis intéressé au raisonnement temporel et aux intervalles d'Allen, ainsi qu'aux réseaux de contraintes temporelles étiquetées de F. Barber.

Des scénarios simples tels que le mouvement d'une pièce à l'autre ou le passage aux toilettes ont été définis. Une dizaine de scénarios médicaux plus complexes, ciblant des patients schizophréniques sous traitement neuroleptique, ont aussi été créés en collaboration avec une équipe québécoise dans laquelle j'ai effectué un stage d'un mois. Par exemple, l'akathisie (« les jambes sans repos ») est définie comme l'impossibilité de rester assis plus de 5 minutes et peut être détectée à l'aide des capteurs de l'appartement intelligent.

Ces scénarios ont pu être modélisés avec le formalisme des réseaux de contraintes temporelles étiquetées et reconnus avec un logiciel que j'ai développé pour l'habitat intelligent. Ils ont pu être testés à l'aide du simulateur en 3D ou (pour certain) en vrai dans le prototype d'habitat.

Ce second stage m'a permis de découvrir plus en profondeur le monde de la recherche, et notamment aux aspects bibliographiques. Par la suite, j'ai souhaité changer de thématique de recherche, car je souhaitais tirer davantage parti de mes connaissances dans le domaine de la pharmacie. De plus, j'avais l'impression que les habitats intelligents seraient difficilement démocratisables à court ou moyen terme, étant donné leur prix très élevé et la difficulté de les faire rembourser par la sécurité sociale.

Thèse (1/10/2003 - 19/12/2006) J'ai obtenu une bourse de thèse ministérielle de l'école doctorale « Santé Publique et Sciences de l'Information Biomédicale » de l'université Paris 6. J'ai effectué ma thèse au laboratoire LIM&BIO de l'université Paris 13, sous la direction d'Alain Venot, co-encadré par Catherine Duclos. Mes travaux ont porté sur la visualisation des connaissances sur le médicament, c'est-à-dire des indications, contre-indications, interactions médicamenteuses, effets indésirables et précautions d'emploi d'un médicament. Ces connaissances sont volumineuses et uniquement présentées sous forme textuelle (papier ou à l'écran), ce qui ne facilite pas leur utilisation en consultation par un médecin. L'objectif de ma thèse était donc d'arriver à une présentation visuelle plus facile d'emploi et plus agréable.

Une recherche bibliographique a montré qu'il existait deux approches pour la visualisation : les langages iconiques et la visualisation d'information ; il nous a semblé intéressant de combiner ces deux approches. Dans un premier temps, j'ai donc cherché à mettre au point un langage iconique pour représenter les principaux concepts intervenants dans les propriétés des médicaments : symptômes, maladies, médicaments, surveillances, catégories de patients (tel que des classes d'âge),... Une analyse d'un corpus de Résumé des Caractéristiques Produit (RCP, document de référence sur le médicament) par des méthodes de Traitement Automatique de la Langue (TAL) a permis d'établir un modèle des concepts à représenter. Je me suis ensuite heurté à l'absence de méthode clairement établie pour concevoir un langage iconique. Cela m'a amené à proposer ma propre méthode, en prenant notamment en compte les relations d'héritage qui peuvent exister entre concepts (par exemple entre « maladie cardiaque » et « insuffisance cardiaque ») et en définissant un « héritage graphique ». En utilisant cette méthode, j'ai pu mettre au point le langage VCM (Visualisation des Connaissances sur le Médicament ; la signification du sigle sera modifiée et étendue par la suite).

Dans un second temps, j'ai mis au point une interface graphique pour présenter les connaissances d'un médicament donné. Cette interface, appelée « Monsieur VCM », rassemble sur un bonhomme anatomique schématique l'ensemble des icônes pour une rubrique donnée (par exemple contre-indications ou effets indésirables). Des règles ont été définies pour combiner les icônes lorsque plusieurs icônes concernent le même élément anatomique (par exemple deux icônes de contre-indications cardiaques), afin de les remplacer par une seule icône plus générale. J'ai implémenté cette interface en langage Python, en utilisant la base de connaissances sur le médicament Thériaque.

Dans un troisième temps, grâce à un financement de la CNAM-TS (Caisse Nationale d'Assurance Maladie - Travailleurs Sociaux), j'ai évalué le langage VCM et l'interface « Monsieur VCM » sur un groupe d'une dizaine de médecins généralistes volontaires. Cette évaluation s'est faite dans des conditions contrôlées (*in vitro*) et après une courte formation des médecins à VCM *via* un didacticiel. Les résultats ont montré que les médecins pouvaient apprendre VCM en 4 à 6 heures, et que l'interface « Monsieur VCM » leur permettait de répondre deux fois plus vite à des questions sur les médicaments, tout en faisant significativement moins d'erreurs, par rapport à une interface textuelle habituelle.

À la fin de ma thèse et dans les années qui ont suivi, ces travaux ont fait l'objet d'un important travail de valorisation aboutissant à un dépôt de brevet américain puis international, et à la signature de contrats avec la société Vidal qui diffuse un livre et une base de connaissances sur le médicament.

Mes travaux de thèse m'ont permis de m'intéresser à la problématique de la visualisation des connaissances, en suivant une approche pleinement multidisciplinaire (ingénierie des connaissances, sciences cognitives, visualisation d'information, informatique médicale). Ils m'ont aussi permis de réaliser que les connaissances que j'avais acquises en pharmacie pouvaient réellement être utiles pour l'informatisation des connaissances sur le médicament. En effet, le domaine d'application du médicament est extrêmement riche et complexe, et il est difficile de le traiter sans connaissance préalable. J'ai acquis la certitude que, si le médicament n'était pas un « produit » comme les autres, il ne pouvait par conséquent pas être informatisé comme les autres.

Postdoc, projet ASTI 2 (Aide à la Stratégie Thérapeutique Informatisée, 1/10/2006 - 31/10/2008) Après ma thèse, je suis entré à la SFTG (Société de Formation Thérapeutique du Généraliste) comme ingénieur de recherche à temps partiel (CDD). La SFTG est une association « loi 1901 » qui effectue de la formation continue auprès des médecins généralistes ; l'association comprend également un pôle « recherche ». Elle dirigeait le projet de recherche ASTI 2 financé par la CNAM-TS (Caisse Nationale d'Assurance Maladie - Travailleurs Sociaux) qui regroupait 6 partenaires publics et privés, dont le LIM&BIO.

Le projet ASTI 2 portait sur l'informatisation des guides de bonnes pratiques. Les guides de bonnes pratiques cliniques (GBP) donnent des recommandations aux médecins pour la prise en charge de maladies ou de conditions patient particulières. Ces guides contiennent notamment une stratégie thérapeutique, c'est-à-dire des recommandations concernant les prescriptions médicamenteuses, par exemple tel médicament sera préféré à tel autre chez telle catégorie de patient. Dans le cas des maladies chroniques (le diabète et l'hypertension dans le cadre du projet), les paramètres à prendre en compte sont nombreux et font appel à un historique de données important. Les médecins ont alors du mal à appliquer la stratégie recommandée.

L'objectif du projet était de proposer des aides informatiques à la stratégie thérapeutique. Le projet comprenait trois modules : un tableau de bord de suivi pour gérer les paramètres à surveiller (par exemple

l'hémoglobine glyquée pour les patients diabétiques), un mode guidé permettant d'appliquer les recommandations du guide sur un patient donné en répondant à des questions, et un mode critique permettant de déclencher automatiquement des alertes lorsque la prescription du médecin ne correspond pas aux recommandations.

Mes travaux ont porté sur le mode critique. Le système existant, hérité du projet ASTI 1, contenait une base de règles décrivant les prescriptions recommandées, de la forme « si conditions patient... alors il faut prescrire... ». Ce système permettait de déterminer le traitement recommandé par le guide, en revanche il ne permettait pas d'expliquer pourquoi un autre traitement n'était pas satisfaisant. Dans ces conditions, il était difficile de convaincre le médecin de son erreur. J'ai alors repensé entièrement le fonctionnement du système, pour arriver à une base de règles de la forme « si conditions patient... et si traitement prescrit... alors afficher la critique... ». Cette base de règle était générée à partir de la précédente, à l'aide d'un traducteur automatique. Le partenaire responsable de l'implémentation du mode critique ayant fait défaut, j'ai été amené à implémenter la totalité du mode critique en Python.

Le projet ASTI 2 m'a permis de découvrir les guides de bonnes pratiques et les systèmes d'aide à la décision, ainsi que de travailler en collaboration étroite avec des médecins généralistes. Je me suis aussi rendu compte que ma formation initiale en pharmacie facilitait grandement mes contacts avec les professionnels de santé, dans un milieu ou « informaticiens » et « médecins » ont souvent du mal à communiquer.

Postdoc, projet ASTI 3 (Aide à la Stratégie Thérapeutique Informatisée, 1/10/2006 - 31/12/2010) J'ai ensuite poursuivi mes travaux à la SFTG lors du projet ASTI 3, financé par la HAS (Haute Autorité de Santé, à hauteur de 689 046, 47 E) et regroupant 6 partenaires publics et privés. L'objectif était à présent de montrer que les méthodes mises au point lors d'ASTI 2 étaient génériques et qu'elles pouvaient s'appliquer à d'autres maladies et s'interfacer avec différents logiciels métier. Pour cela, nous avons implémenté trois nouveaux guides de bonnes pratiques (fibrillation auriculaire, dyslipidémies et arrêt du tabac) et nous avons travaillé avec plusieurs éditeurs de logiciels médicaux (Silk informatique, Cégédim et Alma Pro). Le projet ASTI 3 a aussi permis d'évaluer les trois modules dans des conditions contrôlées.

Cette suite au projet ASTI 2 m'a permis de découvrir le monde « professionnel » des éditeurs de logiciels métiers en santé.

Postdoc, projet EI-xplore (1/11/2006 - 31/04/2007) et VigiTerme (1/01/2008 - 30/06/2010) En parallèle des projets ASTI, j'ai été recruté comme chargé de recherche au LIM&BIO à temps partiel (CDD), sur les projets EI-xplore et VigiTerme, financé respectivement par l'Action Concertée Incitative masse de données et par l'ANR TecSan 2007 à hauteur de 941 658 E. Ces projets regroupaient 9 partenaires publics et privés dont le LIM&BIO. Ils portaient sur la pharmacovigilance, et plus particulièrement sur la détection de nouveaux effets indésirables médicamenteux à partir des notifications réalisées après la commercialisation du médicament. L'objectif était d'améliorer la détection du signal en effectuant des regroupements, soit de médicaments proches (par exemple sulfamides antidiabétiques et sulfamides antibactériens), soit d'effets indésirables proches (par exemple nausée et vomissement). Ces regroupements permettaient d'augmenter le nombre de cas, et donc d'augmenter la puissance des tests statistiques.

Mon travail a consisté à effectuer une première approche exploratoire pour le regroupement de médicaments similaires. Pour cela, je me suis intéressé aux médicaments agissant sur le SNA (Système Nerveux Autonome, c'est-à-dire sympathique et parasympathique) ainsi qu'aux effets indésirables liés au SNA. J'ai modélisé sous forme d'ontologie formelle en langage OWL les différentes composantes du SNA, les organes pilotés par le SNA et les effets indésirables potentiels associés à la stimulation ou à l'inhibition du SNA sur chaque organe (par exemple la constipation pour les intestins). J'ai ensuite intégré dans l'ontologie plusieurs médicaments connus pour avoir un effet sur le SNA, ainsi que la diffusion de ces médicaments. L'utilisation d'un raisonneur (Pellet) a permis de prédire les effets indésirables liés au SNA de ces médicaments. Plusieurs de ces effets étaient déjà connus, et cette approche a été jugée intéressante et prometteuse par le personnel de pharmacovigilance.

Le projet EI-xplore puis VigiTerme a été pour moi l'occasion de découvrir les ontologies et les outils associés (éditeur Protégé, raisonneur Pellet,...). J'ai alors eu l'idée et l'envie d'utiliser ce type de formalisme

pour améliorer la structuration et la formalisation du langage iconique VCM que j'avais conçu au cours de ma thèse. Le projet L3IM allait m'en donner l'occasion.

Postdoc puis maître de conférences, projet L3IM (Langage Iconique et Interfaces Interactives en Médecine, 1/01/2009 - 31/03/2012) Le projet L3IM, financé par l'ANR TecSan 2008 à hauteur de 947 288 E et regroupant 9 partenaires publics et privés, était un prolongement direct de ma thèse. Ce projet était coordonné par Alain Venot et j'ai contribué à son montage et à sa rédaction. J'ai d'abord participé au projet en temps que chargé de recherche (CDD), puis j'ai continué en temps que maître de conférence après ma nomination à l'université Paris 13 en septembre 2009.

L'objectif du projet était d'aboutir à une version 2 du langage VCM, qui ne soit plus limitée aux seules connaissances sur le médicament mais couvrant un domaine d'application plus large, l'ensemble de la médecine. À cette occasion, le sigle VCM devenait « Visualisation des Concepts en Médecine ». Désormais, plusieurs nouvelles applications étaient envisagées pour les icônes : les dossiers patients en médecine générale et à l'hôpital, les moteurs de recherche, les guides de bonnes pratiques cliniques et les réseaux de soin. Les différents partenaires du projet (éditeurs des logiciels éO Médecin et Crossway Hôpital, et de la base de connaissances Vidal Reco, équipe CISMeF,...) permettaient de couvrir ces applications.

Dans un premier temps, j'ai travaillé à l'extension du langage VCM aux besoins des nouvelles applications et à la formalisation du langage. Les travaux sur la formalisation de VCM, mené en collaboration avec Lina Soualmia, ont abouti à la construction d'une ontologie des icônes VCM et la mise au point d'une méthode de formalisation des langages iconiques. Cette formalisation a par la suite permis de déterminer les icônes inconsistantes et de générer automatiquement le lexique des pictogrammes. Dans un second temps, j'ai travaillé avec les différents partenaires du projet pour l'intégration de VCM dans les différentes applications. Les travaux avec l'équipe CISMeF de Rouen ont en particulier donné lieu à la réalisation d'une correspondance entre la CIM10 (Classification Internationale des Maladies 10$^{\text{ème}}$ version) et VCM, établi par Nicolas Griffon avec lequel j'ai collaboré.

Le projet L3IM a été pour moi l'occasion d'encadrer plusieurs stages de M2 lors de travaux exploratoires en amont du projet (Elham Gassemi, Soufiane Cafile) ou durant celui-ci (Christian Simon). Ce projet m'a aussi permis de m'initier au montage et à la rédaction de projet de recherche.

Maître de conférences, projet SiFaDo (Saisie Facile de Données médicales, 1/01/2012 - 30/06/2015) Le projet SiFaDo, financé par l'ANR TecSan 2011 à hauteur de 956 878 E et regroupant 9 partenaires publics et privés, était une suite du projet L3IM. Ce projet était coordonné par Catherine Duclos et j'ai participé à son montage et à sa rédaction.

Le manque de données médicales structurées était un problème récurrent dans les projets de recherche comme ASTI 2 et 3 et L3IM. L'objectif de SiFaDo était donc de proposer des interfaces graphiques pour faciliter la saisie de données médicales structurées (telles que des pathologies) dans les dossiers patient par les médecins en cours de consultation. L'une des pistes envisagées était l'utilisation du langage iconique VCM.

Dans un premier temps, j'ai réalisé une correspondance entre la SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms) et VCM. Contrairement aux correspondances précédentes, celle-ci a été réalisée de manière semi-automatique, en s'appuyant sur les définitions existantes dans la SNOMED CT et sur l'ontologie des icônes VCM. J'ai ensuite commencé à rassembler les différents outils pour la gestion des terminologies dont je disposais en un seul outil, PyMedTermino (Terminologie Médicale en Python). PyMedTermino sera par la suite mis sous licence libre et son développement se poursuivra lors du projet VIIIP. J'ai aussi participé à la conception d'une interface iconique pour l'aide au codage en CIM10, en collaboration avec Adrien Ugon, post-doctorant recruté sur le projet SiFaDo.

Souhaitant revenir vers le « médicament », je me suis en parallèle attelé à la rédaction d'un nouveau projet de recherche.

Maître de conférences, projet VIIIP (Visualisation Intégrée de l'Information sur l'Innovation Pharmaceutique, 1/01/2013 - 15/02/2016) J'ai assuré la rédaction, le montage et la coordination du projet VIIIP, qui faisait suite aux travaux de thèse de Maia Iordatii sur l'innovation pharmaceutique. Ce

projet a été financé par l'ANSM (Agence Nationale de Sécurité du Médicament et des produits de santé) à hauteur de 190 625 E, et réalisé en collaboration avec la SFTG.

L'objectif du projet était de concevoir une plate-forme délivrant une information fiable, indépendante et comparative sur les nouveaux médicaments, à destination des médecins prescripteurs. En effet, leur première source d'information est la visite médicale, directement contrôlée par les laboratoires pharmaceutiques. Des sources indépendantes existent, comme la revue Prescrire, cependant celles-ci proposent des avis d'experts plutôt qu'une information comparative sur les médicaments.

Lors du projet VIIIP, j'ai participé avec les autres participants du projet à la modélisation et la conception de l'ontologie des nouveaux médicaments, la mise au point de méthodes d'extraction des connaissances, la conception de maquette, l'implémentation de la plate-forme et l'évaluation de celle-ci. Plus spécifiquement, j'ai mis en œuvre des méthodes de programmation orientée ontologie, publiées dans le logiciel libre OwlReady par la suite, et j'ai mis au point les « boîtes arc-en-ciel », une nouvelle technique pour la visualisation des ensembles non-disjoints. L'évaluation finale de la plate-forme a montré d'une part l'intérêt très important des médecins pour une information comparative sur le médicament (actuellement inexistante) et d'autre part un gain de temps statistiquement significatif avec les boîtes arc-en-ciel, comparées à des tableaux dynamiques.

L'année 2014 a été très difficile pour moi, à cause d'une maladie rare, longue, handicapante et douloureuse (algodystrophie). Le diagnostic étant compliqué, il m'a de plus fallu attendre près de 6 mois pour obtenir un arrêt de travail. J'ai par suite pu me rétablir en 2015 et reprendre mes activités normalement. J'ai cependant été contraint à réduire mon activité sur 2014 et à me retirer de certaines responsabilités (comme le Yearbook de l'IMIA).

Le projet VIIIP a été pour moi l'occasion de prendre en charge un projet de recherche sur toute la durée de son existence, et d'en assumer la coordination. Il m'a permis d'encadrer de nombreux stagiaires de master (Lilia Kraiem, Camelia Marnia, Mouhssin Gaouar) et un post-doctorant (Adrien Ugon).

Maître de conférences, projet DESIREE (Decision Support and Information Management System for Breast Cancer, 1/05/2016 - ...) Depuis peu, je participe au projet DESIREE, financé par la commission européenne (H2020). Ce projet porte sur la prise en charge du cancer du sein. Mon travail consistera à appliquer et à adapter la technique des boîtes arc-en-ciel (mise au point lors du projet VIIIP) à la visualisation d'une part d'une base de cas de patientes atteintes du cancer du sein, et d'autre part à la comparaison de recommandations en provenance de différents guides de bonnes pratiques et/ou de différents modules d'aide à la décision.

Le tableau suivant montre les thématiques abordées aux différentes étapes de mon parcours :

	Médicament	Recommandations	Dossiers patient	Habitat intelligent	Système de règles	Ontologies	Visualisation	Icônes
SICHIS (5ème année pharma)				X			X	
SICHIS (DEA)				X	X			
VCM (thèse)	X						X	X
EIxplore, Vigitermes (postdoc)	X					X		
ASTI 2 et 3 (postdoc)		X	X		X			
L3IM (MCF)		X	X			X		X
Thèse M. Yasini (MCF)		X					X	
SiFaDo (MCF)			X				X	X
VIIIP (MCF)	X					X	X	
Thèse d'Appoh (MCF)	X					X		X

Animation de la recherche

Comité éditorial de revues

— Co-éditeur de la section « Decision support » de l'IMIA Yearbook of Medical Informatics (2013 à 2014).

Relectures

1. Review d'un article pour la revue Artificial Intelligence in Medicine en 2011
2. Review d'un article pour la revue ACI (Applied Clinical Informatics) en 2011
3. Review d'un article pour la revue IRBM (Ingénierie et Recherche Biomédicale) en 2012
4. Review d'un article pour la revue Expert Opinion on Drug Safety en 2013
5. Review d'un article pour la revue BMC Medical Informatics and Decision Making en 2013
6. Review d'un article pour la revue African Journal of Pharmacy and Pharmacology en 2014
7. Review d'un article pour la revue JIS (Journal of Information Science) en 2014
8. Review d'un article pour la revue Display en 2015
9. Review d'un article pour la revue JVLC (Journal of Visual Language and Computing) en 2016
10. Reviewer pour la conférence MIE (Medical Informatics Europe) 2014, 2016 et 2017
11. Review d'un article pour le Yearbook d'Informatique Médical (IMIA Yearbook) en 2017
12. Review d'un article pour la revue MDPI Informatics en 2017
13. Review d'un article pour la revue Artificial Intelligence in Medicine en 2017

Expertise scientifique

— Évaluateur pour les projets ANR pour l'appel d'offre « Technologies pour la santé » 2015

Sociétés savantes

— Membre de l'AFIA (Association Française pour l'Intelligence Artificielle)
— Membre de l'AIM (Association d'Informatique Médicale)

Primes

— Titulaire de la PEDR (Prime d'Encadrement Doctoral et de Recherche) suite à la campagne 2013 (appelée PES à l'époque)

Jurys de thèse

— Salah Bencherif : *Inertie thérapeutique et hypertension artérielle en médecine générale*, thèse de médecine soutenue le 3/06/2010 à l'université René Descartes - Paris 5. Membre du jury.
— Mondher Cheriet : *Sécurité des données médicales informatiques en cabinet de médecine générale*, thèse de médecine soutenue le 22/06/2012 à l'université Pierre et Marie Curie - Paris 6. Membre du jury.
— Guillaume Grivelet : *Évaluation de la qualité des sites web santé par les médecins généralistes*, thèse de médecine soutenue le 8/3/2013 à l'université Paris 13. Membre du jury.
— Mobin Yasini : *Conception et évaluation de méthodes et outils logiciels pour améliorer la qualité de la prescription et de la réalisation des examens de Biologie*, thèse de science soutenue le 17/12/2013 à l'université Pierre et Marie Curie - Paris 6. Co-encadrant.

Encadrements de travaux

Type	Nombre
Thèse de sciences	
- *soutenue*	1
- *en cours*	1
M2 (6mois)	7
M1 (2 mois)	3
DUT (2 mois)	1

Encadrements de thèses

Mobin Yasini (thèse de science en informatique médicale, taux d'encadrement 70%, 1/11/2010 - 7/12/2013) Mobin Yasini a effectué sa thèse au LIM&BIO encadré par Alain Venot et co-encadré par Catherine Duclos et moi-même, avec une bourse ministérielle de l'école doctorale Epidémiologie et Sciences de l'Information Biomédicale de l'université Paris 6 (ED 393). Ses travaux s'inscrivaient dans le prolongement des projets ASTI 2 et 3, et du stage de M2 de Mobin (voir ci-dessous). Il s'agissait de concevoir et d'évaluer des méthodes et des outils logiciels pour améliorer la qualité de la prescription et de la réalisation des examens de biologie.

Mobin a tout d'abord réuni un corpus de règles de juste prescription des examens de biologie, issues des guides de bonnes pratiques rédigés à l'APHP (Assistance Publique des Hôpitaux de Paris), ainsi que des documents de bonnes pratiques développés par la Haute Autorité de Santé (HAS) et par l'Agence Nationale de Sécurité du Médicament et des produits de santé (ANSM). Un second corpus international a été extrait à partir du *National institute for clinical excellence* (Grande Bretagne) de l'*Association of public health laboratories*, de l'*Ontario association of medical laboratories*, et de l'*ARUP laboratories* (États-Unis). Un modèle conceptuel en UML a ensuite été réalisé à partir du premier corpus, puis validé sur le second. Ce modèle a ensuite été mis en correspondance avec le modèle HL7 RIM (*Health Level 7 Reference Information Model*).

Mobin est ensuite allé très loin dans l'implémentation de ces règles pour la prescription des examens de biologie. Il a tout d'abord évalué leurs implémentabilités en s'appuyant sur le questionnaire *GuideLine Implementability Appraisal* (GLIA). Ce questionnaire a permis de déterminer les règles implémentables et celles qui ne l'étaient pas en l'état, en distinguant deux situations : un logiciel de prescription connecté au dossier patient et un logiciel de prescription non connecté. Mobin a ensuite comparé deux terminologies médicales (CIM10 et SNOMED CT) pour l'implémentation des règles. Enfin, il a travaillé sur le logiciel Orbis de la société Agfa Healthcare, sélectionné par l'APHP, qui intègre le moteur Experter utilisant les MLM (Medical Logic Module) de la syntaxe Arden (un langage de programmation spécifique au domaine médical). Une trentaine de règles ont ainsi pu être implémentées dans Orbis.

Depuis la fin de sa thèse, Mobin a d'abord été consultant « Health & Public service » chez Accenture (octobre 2013 - avril 2014). Il est à présent directeur médical adjoint chez DMD Santé, une PME spécialisée dans l'expertise dans le champ de la santé mobile. Il continue également à enseigner et participe régulièrement à des conférences scientifiques.

Publications communes :

1. **Yasini M**, Duclos C, **Lamy JB**, Venot A. Facilitating access to laboratory guidelines by modeling their contents and designing a computerized user interface. Stud Health Technol Inform 2011 ;169 :487-91. Conférence Medical Informatics Europe MIE 2011, Oslo, Norvège
2. Ebrahiminia V, **Yasini M**, **Lamy JB**. Mapping ASTI patient's therapeutic-data model to virtual Medical Record : can VMR represent therapeutic data elements used by ASTI in clinical guideline implementations ?. AMIA Symposium 2013 ;2013 :372-8. Conférence AMIA, Washington, USA
3. **Yasini M**, Duclos C, Venot A, Lepage E, **Lamy JB**. A guideline-derived model to facilitate the implementation of test-ordering rules within a hospital information system. Stud Health Technol Inform 2013 ;192 :719-23. Conférence MEDINFO 2013, Copenhague, Danemark

4. **Yasini M**, Ebrahiminia V, Duclos C, Venot A, **Lamy JB**. Comparing the use of SNOMED CT and ICD10 for coding clinical conditions to implement laboratory guidelines. Conférence Special Topic Conference STC 2013, Prague, République Tchèque

Valentin Gomoi (thèse de science en informatique, stage européen, 1/06/2012 - 30/06/2012) Valentin Gomoi était un doctorant de l'université Politehnica de Timisoara (Roumanie), sous la direction de Vasile Stoicu-Tivadar. Ses travaux portaient sur la génération automatique de protocoles médicaux en s'appuyant sur des sources d'information hétérogènes. J'ai co-encadré Valentin, avec Sylvie Després et Brigitte Séroussi, lors d'un stage européen d'1 mois en France. Je lui ai en particulier montré les outils mis au point lors des projets ASTI 2 et 3. Valentin a ensuite pu intégrer une partie de ces outils avec les siens.

Appoh Kouame (thèse de science en informatique, taux d'encadrement 50%, 1/12/2015 - ...) Appoh Kouame est doctorant à l'université Gaston Berger (UGB) du Sénégal, sous la direction de Moussa Lo. Sa thèse porte sur la conception d'un système de gestion de la médecine traditionnelle de l'Afrique de l'Ouest dans une plate-forme web sociale et sémantique. Dans ce cadre, il souhaite proposer une approche fondée sur une ontologie visuelle.

J'ai encadré Appoh durant son stage de perfectionnement de 2 mois en France au LIMICS (1/12/2015 - 31/1/2016), financé par une bourse AMRUGE C2D, puis lors de sa thèse. Appoh a conçu une ontologie de la médecine traditionnelle africaine en cours de construction. Je l'ai aidé à mieux cerner son sujet pour se restreindre aux problèmes des recettes des remèdes traditionnels, et à concevoir un langage iconique pour identifier les plantes médicinales intervenant comme ingrédients dans ces recettes.

Les travaux réalisés par Appoh ont ensuite donné lieu deux communications, au Colloque Africain sur la Recherche en Informatique et Mathématiques (CARI) 2016 et à Africomm 2016. Appoh est actuellement en cours de rédaction de sa thèse, pour une soutenance prévue en mars 2017.

Publications communes :

1. **Kouame A**, **Lamy JB**, Brou KM, Lo M. Modelization of recipe in African traditional medicine with visual ontology approach, iconic sketch. Africomm conference 2016, Ouagadougou, Burkina Faso
2. **Kouame A**, **Lamy JB**, Lo M, Brou KM. Vers un système iconique d'aide à la décision pour les praticiens de la médecine traditionnelle. Colloque Africain sur la Recherche en Informatique et Mathématiques CARI 2016

Encadrement de masters et DUT

Elham Gassemi (M2 Informatique biomédiale, 1/01/2007 - 30/06/2007) Elham Gassemi a effectué son stage de M2 sous ma direction et celle de Lina Soualmia (MCF). Ce stage était un travail exploratoire en amont du projet L3IM. L'étudiante a travaillé sur des prolongements de mes travaux de thèse, sur la conception et l'évaluation d'outils graphiques pour la sélection de documents médicaux avec une application aux guides de bonnes pratiques. L'objectif était de mettre au point des outils graphiques et iconiques pour faciliter à un médecin la recherche d'un guide de bonnes pratiques cliniques au sein d'un moteur de recherche tel que CISMeF BP (Catalogue et Index des Sites Médicaux de Langue Française - Bonne Pratique). La démarche a consisté à établir une correspondance entre la terminologie MeSH (Medical Subject Headings) utilisée par CISMeF et les icônes VCM, étudier la couverture du MeSH par les icônes, proposer des enrichissements à VCM et réaliser des maquettes d'interfaces graphiques.

Soufiane Cafile (M2 Informatique biomédiale, 1/01/2009 - 30/06/2009) Soufiane Cafile a effectué son stage de M2 dans le prolongement de mes travaux de thèse et en préparation du projet L3IM. Il a travaillé sur la conception de méthodes pour étudier et visualiser la couverture d'une classification hiérarchique par un langage iconique, en appliquant ses méthodes à la classification CIM10 (Classification Internationale des Maladies $10^{\text{ème}}$ version) et au langage iconique VCM. Le langage VCM étant moins précis que le texte ou les terminologies, la question de la couverture des terminologies par VCM est importante. Soufiane a proposé une méthode consistant à calculer des scores d'alignement pour chaque couple (terme CIM10, icône VCM associée). Il a défini un score « intentionnel » qui dépend de la distance entre le terme CIM10 du couple et le terme CIM10 ancêtre le plus éloigné associé à la même icône, et un score « extensionnel » prenant en compte

le nombre de termes CIM10 associés à la même icône VCM. Il a enfin proposé des méthodes de visualisation de la couverture à l'aide de TreeMap et d'arbre.

Mobin Yasini (M2 Informatique biomédiale, 1/01/2010 - 30/06/2010) Mobin Yasini a travaillé sur la modélisation des connaissances et le développement d'outils informatiques pour aider à la juste prescription des examens de biologie. Ce travail s'inscrit dans le prolongement de ma thèse mais aussi des projets ASTI. L'utilisation des examens de laboratoires (dosages sanguins, *etc*) est délicate et il existe des guides de bonnes pratiques pour la prescription et la réalisation des examens. L'objectif du travail de Mobin était de concevoir des outils pour faciliter l'utilisation de ces guides. Mobin a travaillé sur un corpus de guides rédigés à l'APHP (Assistance Publique des Hôpitaux de Paris) au format textuel. Il a réalisé une analyse détaillée de ce corpus qui a abouti à un modèle UML. Il a ensuite restructuré les différents guides selon un plan unique et homogène. Enfin, Mobin a proposé une interface web pour la consultation des guides. L'évaluation qualitative de cette interface a donné de bons résultats. Par ailleurs, ce travail a par la suite fait l'objet d'une présentation à la conférence MIE 2011 (Medical Informatics Europe) qui a reçu le prix du meilleur article.

Publications communes : voir rubrique « Encadrements de thèses » ci-dessus.

Christian Simon (M2 Informatique biomédiale, 1/01/2010 - 30/06/2010) Christian Simon a effectué un M2 d'informatique biomédicale en parallèle de ses activités dans le privé, et il a effectué son stage sous ma direction et celle de Catherine Duclos. Il a travaillé sur la représentation iconique des données du patient dans le dossier médical informatisé en médecine générale, dans le cadre du projet L3IM (Langage Iconique et Interface Interactive en Médecine). Les dossiers patient contiennent notamment un historique des maladies et des problèmes de santé de chaque patient, cependant la consultation de ces historiques peut être laborieuse, notamment en médecine générale où un même patient peut se retrouver avec des historiques très divers en termes de spécialité (cardiologie, néphrologie,...). L'objectif de Christian était donc de faciliter la consultation des historiques à l'aide du langage iconique VCM. Pour cela, Christian a tout d'abord établi une correspondance entre la terminologie CISP2 (Classification Internationale des Soins Primaires, version 2) et VCM. Il a ensuite intégré les icônes VCM dans le logiciel éO, ainsi qu'une interface « Monsieur VCM » permettant de représenter sur un bonhomme anatomique schématique un résumé des maladies en cours et des antécédents d'un patient.

Publications communes :

1. **Lamy JB**, Duclos C, Hamek S, Beuscart-Zéphir MC, Kerdelhué G, Darmoni S, Favre M, Falcoff H, **Simon C**, Pereira S, Serrot E, Mitouard T, Hardouin E, Kergosien Y, Venot A. Towards iconic language for patient records, drug monographs, guidelines and medical search engines. Stud Health Technol Inform 2010 ;160 :156-60. Conférence MEDINFO 2010, Le Cap, Afrique du Sud
2. **Lamy JB**, Beuscart-Zéphir MC, Boog C, Darmoni S, Favre M, Guigue L, Hamek S, Hassler S, Kerdelhué G, Leroy N, Mitouard T, Pereira S, **Simon C**, Venot A, Duclos C. Langage iconique et interfaces interactives en médecine : application aux dossiers patients, guides de bonnes pratiques et moteurs de recherche médicaux. IRBM 2012 ;33(2) :129-136
3. **Simon C**, Hassler S, Beuscart-Zephir MC, Favre M, Venot A, Duclos C, **Lamy JB**. Using an iconic language to improve access to electronic medical records in general medicine. Stud Health Technol Inform 2014 ;205 :333-7. Conférence Medical Informatics Europe MIE 2014, Istanbul, Turquie

Stéphanie Kréber (M2 Informatique biomédiale, 1/01/2013 - 30/06/2013) Stéphanie Kréber a effectué son stage de M2 sous ma direction et celle de Catherine Duclos. Elle a travaillé sur l'utilisation d'un langage iconique pour optimiser la saisie et la restitution des symptômes et des signes en médecine d'urgence préhospitalière. Les centres de réception et de régulation d'appel du SAMU (Services d'Aide Médicale d'Urgence) reçoivent un nombre important d'appels qui doivent être traités le plus rapidement possible. Dans ce contexte, l'utilisation d'un langage iconique pourrait faciliter et accélérer la saisie des informations au cours de l'appel. Stéphanie a tout d'abord réalisé un modèle du dossier de régulation médicale, et un modèle des signes cliniques dans le domaine cardio-vasculaire. Elle a ensuite proposé une extension du langage VCM pour la représentation iconique de ces signes. Elle a enfin construit une maquette d'interface s'inspirant de « Monsieur VCM » pour permettre la saisie iconique du motif d'appel et pour la régulation en urgence préhospitalière.

Lilia Kraiem (M1 Informatique biomédiale, 1/07/2013 - 30/08/2013) Lilia Kraiem a effectué son stage de M1 sur le calcul d'indicateurs sur l'information sur l'innovation pharmaceutique, dans le cadre du projet VIIIP que je coordonnais. L'objectif de son travail était de réaliser un programme permettant de calculer deux indicateurs pour un médicament donné : (1) la durée entre l'obtention de l'AMM (Autorisation de Mise sur le Marché) et la commercialisation, et (2) la proportion d'essais cliniques portant sur ce médicament dont les résultats sont disponibles publiquement. Le premier indicateur est calculé à partir des informations présentes sur le site de l'ANSM (Agence Nationale de Sécurité du Médicament et des Produits de Santé), et le second à partir des informations disponibles dans les registres d'essais cliniques, les rapports d'évaluation et les publications scientifiques.

Lilia Kraiem (M2 Informatique biomédiale, 1/01/2014 - 30/06/2014) Lilia Kraiem a ensuite effectué son stage de M2 sur l'extraction automatique d'informations sur les nouveaux médicaments, toujours dans le cadre du projet VIIIP. L'objectif de son travail était de mettre au point des méthodes pour extraire automatiquement les connaissances disponibles sur les nouveaux médicaments, à partir de différentes sources d'information : la base Thériaque, les informations présentes sur le site de l'ANSM, les rapports d'évaluation et les avis de la commission de transparence,... Pour cela, Lilia a travaillé conjointement avec Hélène Berthelot, pharmacienne experte, pour formaliser les processus d'extraction qu'utilisait l'expert. Ce travail a abouti à des algorithmes d'extraction. Lilia a ensuite implémenté une partie de ces algorithmes.

Camelia Marnia (M1 Informatique biomédiale, 1/05/2015 - 30/06/2015) Camelia Marnia a effectué son stage de M1 sur l'extraction et la comparaison des précautions d'emploi et des surveillances liées aux médicaments, dans le cadre du projet VIIIP. L'objectif était de développer un module pour extraire les informations sur les surveillances recommandées pour les médicaments (par exemple surveillance rénale, hépatique, *etc*), et de visualiser ces informations de manière à faciliter la comparaison entre plusieurs médicaments similaires. La démarche suivie par Camelia a consisté à extraire les surveillances de la base Thériaque via des requêtes SQL, puis à établir une correspondance entre les codes de Thériaque et les icônes VCM et enfin à produire des tableaux iconiques comparatifs.

Mouhssin Gaouar (M1 Informatique biomédiale, 1/05/2015 - 30/06/2015) Mouhssin Gaouar a effectué son stage de M1 sur la visualisation graphique des résultats d'essais cliniques. Il s'agissait d'un travail exploratoire dans le cadre du projet VIIIP. Les résultats d'essais cliniques sont souvent présentés sous forme de tableaux difficiles à lire ; le travail de Moushin consistait à représenter ces résultats sous forme de diagrammes en bâtons. Pour cela, Moushin est parti d'un modèle ontologique des résumés de résultats d'essais cliniques réalisés au préalable dans le projet VIIIP. Il a ensuite réalisé un programme pour extraire les données nécessaires de cette ontologie et les représenter sous la forme de diagrammes en bâtons.

Coralie Capron (DUT Génie biologique, 2/05/2016 - 8/07/2016) Coralie Capron a effectué son stage de DUT Génie biologique option bioinformatique sous ma direction. Elle a travaillé sur la visualisation de Gene Ontology à l'aide de boîtes arc-en-ciel. Les boîtes arc-en-ciel sont une nouvelle technique pour la visualisation d'ensembles non-disjoints ; cette technique a été mise au point durant le projet VIIIP et appliquée aux connaissances sur le médicament. Elle pourrait cependant être appliquée à bien d'autres informations ou connaissances. L'objectif du travail de Coralie était de réaliser un programme permettant de visualiser des extraits de Gene Ontology avec les boîtes arc-en-ciel.

Publication commune :

1. **Lamy JB**, Berthelot H, Capron C, Favre M. Rainbow boxes : a new technique for overlapping set visualization and two applications in the biomedical domain. Journal of Visual Language and Computing 2017 ;43 :71-82

Marwa Ben Abid (M2 Informatique biomédiale, 1/01/2017 - ...) Marwa Ben Abid a effectué son stage de M2 sous ma direction, dans le cadre du projet européen H2020 DESIREE. Elle a travaillé sur la visualisation de données d'expression génique (transcriptome) et la comparaison des données de plusieurs patientes. La méthode employée a consisté à adapter la technique des boîtes arc-en-ciel et à la compléter avec une visualisation en nuage de points produites par une analyse en composante principale.

Participations à des projets de recherche

Début	Fin	Projet
1/10/2006	31/10/2008	Participation au projet **ASTI2**, financé par la CNAM-TS (Caisse Nationale d'Assurance Maladie - Travailleurs Sociaux) et regroupant 6 partenaires publics et privés. Le projet portait sur l'informatisation des guides de bonnes pratiques cliniques en médecine générale. Contribution : — Reconception du moteur d'ASTI 1 pour rendre possible l'explication des critiques — Rédaction des spécifications détaillées — Implémentation du nouveau moteur — Participation à la définition de la base de règles — Vérification du système sur l'ensemble des cas possibles
1/10/2006	31/12/2010	Participation au projet **ASTI3**, financé par la HAS (Haute Autorité de Santé) à hauteur de 689 046, 47 E et regroupant 6 partenaires publics et privés. Le projet portait sur l'informatisation des guides de bonnes pratiques cliniques en médecine générale. Contribution : — Participation à la définition de 3 nouvelles bases de règles — Collaboration à l'intégration d'ASTI dans plusieurs logiciels métiers — Évaluation du système avec un groupe de médecins
1/11/2006	31/04/2007	Participation au projet **EI-xplore**, financé par l'ACI (Action Concertée Incitative) masse de données Le projet portait sur l'amélioration de la détection du signal en pharmacovigilance, par le regroupement des médicaments et/ou des effets indésirables. Contribution : — Construction d'une ontologie du SNA (Système Nerveux Autonome) — Prédiction d'effets indésirables à partir de cette ontologie
1/01/2008	30/06/2010	Participation au projet **Vigiterme**, financé par l'ANR TecSan 2007 à hauteur de 941 658 E et regroupant 9 partenaires publics et privés. Le projet portait sur l'amélioration de la détection du signal en pharmacovigilance, par le regroupement des médicaments et/ou des effets indésirables. Contribution : — Regroupements de médicaments et d'effets indésirables proches selon l'ontologie du SNA

Début	Fin	Projet
1/01/2009	31/03/2012	Participation au projet **L3IM** (Langage Iconique et Interfaces Interactives en Médecine), financé par l'ANR TecSan 2008 à hauteur de 947 288 E et regroupant 9 partenaires publics et privés. Le projet portait sur l'utilisation du langage iconique VCM dans les interfaces de logiciels médicaux : dossiers patients en médecine générale et à l'hôpital, réseau de soins, guides de bonnes pratiques cliniques, moteurs de recherche. Contribution : — Extension de VCM (développé durant ma thèse) pour la représentation de région anatomique, de professionnels de santé, de documents médicaux — Construction de l'ontologie des icônes VCM — Conception d'un serveur iconique et terminologique — Implémentation du serveur — Collaboration à l'intégration de VCM dans les applications des partenaires — Évaluation des applications en collaboration avec les partenaires
1/01/2012	30/06/2015	Participation au projet **SiFaDo** (Saisie Facile de Données médicales), financé par l'ANR TecSan 2011 à hauteur de 956 878 E et regroupant 9 partenaires publics et privés. Le projet portait sur la conception d'outils pour faciliter la saisie de données médicales structurées par les médecins en consultation. Contribution : — Mise au point d'un premier prototype d'interface iconique pour le codage — Alignement semi-automatique de VCM avec la SNOMED-CT — Conception d'un module Python pour la gestion des terminologie médicale
1/01/2013	15/02/2016	**Coordinateur du projet VIIIP** (Visualisation Intégrée de l'Information sur l'Innovation Pharmaceutique), monopartenaire, financé par l'ANSM (Agence Nationale de Sécurité du Médicament et des produits de santé) à hauteur de 190 625 E. Le projet portait la conception d'une plate-forme pour une information fiable, indépendante et comparative sur les nouveaux médicaments. Contribution : — Rédaction et coordination du projet — Alignement de VCM avec la terminologie Thériaque — Conception d'un module pour la programmation orientée ontologie — Conception de méthodes d'extraction des effets indésirables à partir des textes — Conception de méthode de visualisation pour les ensembles non-disjoints : tableau dynamique et boîtes arc-en-ciel — Implémentation de la plate-forme — Évaluation du site web sur un groupe de médecins
1/05/2016	...	Participation au projet **DESIREE** (Decision Support and Information Management System for Breast Cancer), financé par la commission européenne (H2020) à hauteur de 3 340 720 E et regroupant 11 partenaires publics et privés. Le projet porte sur l'aide à la décision dans la prise en charge médicale du cancer du sein.

Début	Fin	Projet
1/01/2017	...	Participation au projet **PEGASE** (Pharmacovigilance Enrichie par des Groupements Améliorant la détection des Signaux Emergents), financé par l'ANR (Défi 7 Société de l'information et de la communication) à hauteur de 384 000 E et regroupant 7 partenaires universitaires et institutionnels. Le projet porte sur l'aide à la pharmacovigilance par des outils facilitant l'utilisation de la terminologie de référence MedDRA.
1/01/2017	...	Participation au projet **RaMiPA** (Raisonner pour Mieux Prescrire les Antibiotiques), financé par l'ANSM (Agence Nationale de Sécurité du Médicament et des produits de santé) à hauteur de 120 000 E. L'objectif du projet est de concevoir et d'évaluer un système d'aide à la décision pour réduire le mésusage des antibiotiques en antibiothérapie empirique en soins primaires. Ce système implémentera le raisonnement profond que nous aurons explicité, afin de présenter des recommandations fondées sur les preuves, adaptables et extrapolables aux situations cliniques non décrites dans les recommandations, justifiées par des éléments de raisonnement, faciles à mettre à jour et présentées de manière ludique et facile à comprendre.
1/01/2017	...	Participation au projet **PEPS** (Prévention Effective Personnalisée et e-Santé), financé par l'ANR à hauteur de 736 914 E. L'objectif du projet est de développer des outils qui permettront de rendre effective la démarche de prévention du médecin et du patient, et de construire des plans de prévention individualisés.

8.2 Publications

Type	Nombre
Brevet	1
Chapitres de livre et d'ouvrage	5
Revues avec comité de sélection	
- *internationale*	14
- *francophone*	6
Conférences avec comité de sélection	
- *internationale*	23
- *francophone*	13
Rapports, mémoires, thèses	10
Logiciels libres	4

Brevet

1. **Lamy JB**, Duclos C, Venot A. Graphical representation of medical knowledge. Brevet international enregistré auprès de l'Institut Mondial de la Propriété Intellectuelle (OMPI) + brevet américain, WO2007131932, 2007

Chapitres de livre et d'ouvrage divers

1. **Lamy JB**, Ellini A, Nobécourt J, Venot A, Zucker JD. Testing methods for decision support systems. Dans : Decision support systems, Jao CS éd., édition Intech, 2010 ;87-98
2. Duclos C, Burgun A, **Lamy JB**, Landais P, Rodrigues JM, Soualmia L, Zweigenbaum P. Le vocabulaire médical, les ressources terminologiques, le codage de l'information en santé. Dans : Informatique Médicale, e-Santé - Fondements et applications (520 pages), Venot A, Burgun A, Duclos C éd., édition Springer, 2012
3. Bouaud J, **Lamy JB**. A medical informatics perspective on clinical decision support systems. Findings from the yearbook 2013 section on decision support. Yearbook of medical informatics 2013 ;8(1) :128-31
4. Duclos C, Burgun A, **Lamy JB**, Landais P, Rodrigues JM, Soualmia L, Zweigenbaum P. Medical Vocabulary, Terminological Resources and Information Coding in the Health Domain. Dans : Medical Informatics, e-Health - Fundamentals and Applications (490 pages), Venot A, Burgun A, Duclos C éd., édition Springer, 2014 ;11-42
5. Bouaud J, **Lamy JB**. A 2014 medical informatics perspective on clinical decision support systems : do we hit the ceiling of effectiveness ? Yearbook of medical informatics 2014 ;9(1) :163-6

Revues internationales avec comité de sélection

Les facteurs d'impact (IF) sont donnés à la date de l'article. Dans mon laboratoire (LIMICS), l'usage veut que la personne ayant le plus contribué à un article signe en premier auteur, et que l'encadrant principal (le cas échéant, pour les stagiaires et doctorants) signe en dernier auteur.

1. Rialle V, **Lamy JB**, Noury N, Bajolle L. Telemonitoring of patients at home : A Software Agent approach. Comput Methods Programs Biomed 2003 ;72(3) :257-268 (IF 1,1)
2. **Lamy JB**, Duclos C, Bar-Hen A, Ouvrard P, Venot A. An iconic language for the graphical representation of medical concepts. BMC Medical Informatics and Decision Making 2008 ;8 :16 (IF 1,7, article résumé dans Science)
3. **Lamy JB**, Venot A, Bar-Hen A, Ouvrard P, Duclos C. Design of a graphical and interactive interface for facilitating access to drug contraindications, cautions for use, interactions and adverse effects. BMC Medical Informatics and Decision Making 2008 ;8 :21 (IF 1,7)
4. Toussi M, **Lamy JB**, Le Toumelin P, Venot A. Using data mining techniques to explore physicians' therapeutic decisions when clinical guidelines do not provide recommendations : methods and example for type 2 diabetes. BMC medical informatics and decision making 2009 ;9 :28 (IF 1,9)

5. **Lamy JB**, Ebrahiminia V, Riou C, Séroussi B, Bouaud J, Simon C, Dubois S, Butti A, Simon G, Favre M, Falcoff H, Venot A. How to translate therapeutic recommendations in clinical practice guidelines into rules for critiquing physician prescriptions ? Methods and application to five guidelines. BMC Medical Informatics and Decision Making 2010 ;10 :31 (IF 1,9)
6. **Lamy JB**, Soualmia LF, Kerdelhué G, Venot A, Duclos C. Validating the semantics of a medical iconic language using ontological reasoning. Journal of Biomedical Informatics 2013 ;46(1) :56-67 (IF 2,2)
7. Griffon N, Kerdelhué G, Soualmia LF, Merabti T, Grosjean J, **Lamy JB**, Venot A, Duclos C, Darmoni SJ. Evaluating alignment quality between iconic language and reference terminologies using similarity metrics. BMC medical informatics and decision making 2014 ;14 :17 (IF 1,6)
8. Pereira S, Hassler S, Hamek S, Boog C, Leroy N, Beuscart-Zéphir MC, Favre M, Venot A, Duclos C, **Lamy JB**. Improving access to clinical practice guidelines with an interactive graphical interface using an iconic language. BMC medical informatics and decision making 2014 ;14(1) :77 (IF 1,6)
9. Griffon N, Kerdelhué G, Hamek S, Hassler S, Boog C, **Lamy JB**, Duclos C, Venot A, Darmoni SJ. Design and usability study of an iconic user interface to ease information retrieval of medical guidelines. J Am Med Inform Assoc 2014 ;21(e2) :e270-7 (IF 3,6)
10. **Lamy JB**, Séroussi B, Griffon N, Kerdelhué G, Jaulent MC, Bouaud J. Toward a Formalization of the Process to Select IMIA Yearbook Best Papers. Methods Inf Med 2015 ;54(2) :135-44 (IF 2,2)
11. **Lamy JB**, Berthelot H, Favre M, Ugon A, Duclos C, Venot A. Using visual analytics for presenting comparative information on new drugs. Journal of Biomedical Informatics 2017 ;71 :58-69 (IF 2,8)
12. **Lamy JB**. Owlready : Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. Artificial Intelligence in Medicine, 2017 ;80 :11-28 (IF 2,1)
13. **Lamy JB**, Berthelot H, Capron C, Favre M. Rainbow boxes : a new technique for overlapping set visualization and two applications in the biomedical domain. Journal of Visual Language and Computing 2017 ;43 :71-82 (IF 1,2)
14. **Lamy JB**, Soualmia LF. Formalization of the semantics of iconic languages : an ontology-based method and four semantic-powered applications. Knowledge-Based System 2017 ;135 :159-179 (IF 4,5)

Revues francophones avec comité de sélection

1. Rialle V, Noury N, Bajolle L, **Lamy JB**, Virone G, Duchêne F, Moha N, Demongeot J. Le concept d'Habitat Intelligent pour la Santé : considérations techniques et scientifiques pour un service médico-social. Revue de Gériatrie 2003 ;28(5) :403-416
2. **Lamy JB**, Duclos C, Rialle V, Venot A. Quelle méthodologie tenant compte des sciences cognitives pour la conception des langages graphiques ? Revue d'intelligence artificielle (RIA) 2008 ;22(3-4) :265-280
3. Falcoff H, Benainous O, Gillaizeau F, Favre M, Simon C, Desfontaines E, **Lamy JB**, Venot A, Seroussi B, Bouaud J, Durieux P. Développement et étude d'impact d'un système informatique de tableaux de bord pour le suivi des pathologies chroniques en médecine générale. Pratiques et organisation des soins 2009 ;40(3) :177-189
4. Bencherif S, Khau S, Bloede F, **Lamy JB**, Falcoff H. Adaptation du traitement chez les patients hypertendus. Exercer 2011 ;22(97) :88
5. **Lamy JB**, Beuscart-Zéphir MC, Boog C, Darmoni S, Favre M, Guigue L, Hamek S, Hassler S, Kerdelhué G, Leroy N, Mitouard T, Pereira S, Simon C, Venot A, Duclos C. Langage iconique et interfaces interactives en médecine : application aux dossiers patients, guides de bonnes pratiques et moteurs de recherche médicaux. IRBM 2012 ;33(2) :129-136 (IF 0,5)
6. **Lamy JB**, Soualmia LF, Duclos C, Venot A. Formalisation de la sémantique des langages iconiques : méthode à base d'ontologie et applications. Revue d'intelligence artificielle (RIA) 2016 ;30(5) :579-606

Conférences internationales avec comité de sélection

1. **Lamy JB**, Duclos C, Rialle V, Venot A. Which graphical approaches should be used to represent medical knowledge ?. Stud Health Technol Inform 2005 ;116 :719-724. Conférence Medical Informatics Europe MIE 2005, Genève, Suisse
2. **Lamy JB**, Ellini A, Ebrahiminia V, Zucker JD, Falcoff H, Venot A. Use of the C4.5 machine learning algorithm to test a clinical guideline-based decision support system. Stud Health Technol Inform 2008 ;136 :223-228. Conférence Medical Informatics Europe MIE 2008, Göteborg, Sweden
3. **Lamy JB**, Duclos C, Hamek S, Beuscart-Zéphir MC, Kerdelhué G, Darmoni S, Favre M, Falcoff H, Simon C, Pereira S, Serrot E, Mitouard T, Hardouin E, Kergosien Y, Venot A. Towards iconic language for patient records, drug monographs, guidelines and medical search engines. Stud Health Technol Inform 2010 ;160 :156-60. Conférence MEDINFO 2010, Le Cap, Afrique du Sud
4. **Lamy JB**, Ebrahiminia V, Seroussi B, Bouaud J, Simon C, Favre M, Falcoff H, Venot A. A generic system for critiquing physicians' prescriptions : usability, satisfaction and lessons learnt. Stud Health Technol Inform 2011 ;169 :125-129. Conférence Medical Informatics Europe MIE 2011, Oslo, Norvège

5. Yasini M, Duclos C, **Lamy JB**, Venot A. Facilitating access to laboratory guidelines by modeling their contents and designing a computerized user interface. Stud Health Technol Inform 2011 ;169 :487-91. Conférence Medical Informatics Europe MIE 2011, Oslo, Norvège, **best paper**
6. Tsopra R, **Lamy JB**, Venot A, Duclos C. Design of an original interface that facilitates the use of clinical practice guidelines of infection by physicians in primary care. Stud Health Technol Inform 2012 ;180 :93-7. Conférence Medical Informatics Europe MIE 2012, Pise, Italie
7. Ebrahiminia V, Yasini M, **Lamy JB**. Mapping ASTI patient's therapeutic-data model to virtual Medical Record : can VMR represent therapeutic data elements used by ASTI in clinical guideline implementations ?. AMIA Symposium 2013 ;2013 :372-8. Conférence AMIA, Washington, USA
8. **Lamy JB**, Tsopra R, Venot A, Duclos C. A Semi-automatic Semantic Method for Mapping SNOMED CT Concepts to VCM Icons. Stud Health Technol Inform 2013 ;192 :42-6. Conférence MEDINFO 2013, Copenhague, Danemark
9. Yasini M, Duclos C, Venot A, Lepage E, **Lamy JB**. A guideline-derived model to facilitate the implementation of test-ordering rules within a hospital information system. Stud Health Technol Inform 2013 ;192 :719-23. Conférence MEDINFO 2013, Copenhague, Danemark (taux de sélection 36%)
10. Yasini M, Ebrahiminia V, Duclos C, Venot A, **Lamy JB**. Comparing the use of SNOMED CT and ICD10 for coding clinical conditions to implement laboratory guidelines. Conférence Special Topic Conference STC 2013, Prague, République Tchèque
11. Sedki K, Duclos C, **Lamy JB**. A Preference-based framework for medical decision making. Stud Health Technol Inform 2014 ;205 :63-7. Conférence Medical Informatics Europe MIE 2014, Istanbul, Turquie
12. Simon C, Hassler S, Beuscart-Zephir MC, Favre M, Venot A, Duclos C, **Lamy JB**. Using an iconic language to improve access to electronic medical records in general medicine. Stud Health Technol Inform 2014 ;205 :333-7. Conférence Medical Informatics Europe MIE 2014, Istanbul, Turquie
13. Vaillant P, **Lamy JB**. Using graph transformation algorithms to generate natural language equivalents of icons expressing medical concepts. Lecture Notes in Computer Sciences 2014 ;LNAI 8655 :353–362. Conférence Text, Speech and Dialogue, Brno, République Tchèque
14. **Lamy JB**, Venot A, Duclos C. PyMedTermino : an open-source generic API for advanced terminology services. Stud Health Technol Inform 2015 ;210 :924-928. Conférence Medical Informatics Europe MIE 2015, Madrid, Espagne
15. Ugon A, Berthelot H, Venot A, Favre M, Duclos C, **Lamy JB**. Abstractive Summarization of Drug Dosage Regimens for Supporting Drug Comparison. Stud Health Technol Inform 2015 ;213 :79-82. Conférence Informatics, Management, and Technology in Healthcare IMTH 2015, Athène, Grèce
16. Kouamé A, **Lamy JB**, Brou KM, Lo M. Modelization of recipe in African traditional medicine with visual ontology approach, iconic sketch. Africomm conference 2016, Ouagadougou, Burkina Faso
17. **Lamy JB**. Ontology-Oriented Programming for Biomedical Informatics. Stud Health Technol Inform 2016 ;221 :64-68. Special Topic Conference STC 2016, Paris, France
18. **Lamy JB**, Ugon A, Berthelot H. Automatic extraction of drug adverse effects from product characteristics (SPCs) : a text versus table comparison. Stud Health Technol Inform 2016 ;228 :339-343. Conférence Medical Informatics Europe MIE 2016, Munich, Allemagne (taux de sélection 35%)
19. **Lamy JB**, Berthelot H, Favre M. Rainbow boxes. A technique for visualizing overlapping sets and an application to the comparison of drugs properties. Information Visualization 2016, Lisbonne, Portugal
20. Tsopra R, **Lamy JB**. Design of a visual interface for comparing antibiotics using rainbow boxes. Stud Health Technol Inform. 2017 ;235 :529-533. Conférence Medical Informatics Europe MIE 2017, Manchester, Angleterre (taux de sélection 28%), **prix Peter L. Reichertz du meilleur papier présenté par un jeune chercheur**
21. **Lamy JB**, Ugon A, Berthelot H, Favre M. A visual decision support system for helping physicians to make a decision on new drugs, Decision Support Systems VII. Data, information and knowledge visualization in decision support systems, Lecture Notes in Business Information Processing, International Conference on Decision Support System Technology ICDSST 2017, Namur, Belgique (taux de sélection en article long 25%)
22. **Lamy JB**, Tsopra R. Translating visually the reasoning of a perceptron : the weighted rainbow boxes technique and an application in antibiotherapy. Information Visualization 2017, Londres, Angleterre (taux de sélection 45%), **best paper**
23. Séroussi B, Guézennec G, **Lamy JB**, Muro N, Larburu N, Sekar BD, Prebet C, Bouaud J. Reconciliation of multiple guidelines for decision support : a case study on the multidisciplinary management of breast cancer within the DESIREE project. Proc AMIA Annual Symposium 2017

Conférences francophones avec comité de sélection

1. **Lamy JB**, Duclos C, Rialle V, Venot A. Quelles approches graphiques pour faciliter la diffusion des connaissances médicales ?. 5ème congrès Internet et Pédagogie Médicale IPM 2004 ;39-40, Grenoble
2. **Lamy JB**, Duclos C, Rialle V, Venot A. Classification et analyse critique des approches graphiques appliquées à la visualisation des connaissances médicales. Journées Francophones d'Informatique Médicale JFIM 2005 ;1-9, Lille

3. **Lamy JB**, Duclos C, Rialle V, Venot A. Vers une méthode rigoureuse de conception des langages graphiques s'appuyant sur les sciences cognitives. 4ème atelier visualisation et extraction de connaissances, 6èmes journées francophones d'Extraction et de Gestion des Connaissances EGC 2006, Paris.
4. **Lamy JB**, Duclos C, Venot A. Quelles applications pédagogiques pour un langage graphique médical ? Congrès Internet et Pédagogie pour les Sciences de la Santé et du Sport IP3S 2008, Marakech, Maroc
5. **Lamy JB**, Duclos C, Venot A. De l'analyse d'un corpus de texte à la conception d'une interface graphique facilitant l'accès aux connaissances sur le médicament. Actes du congrès d'Ingénierie des Connaissances (IC) 2009 ;265-276, Hammamet, Tunisie, **best paper**
6. **Lamy JB**, Beuscart-Zéphir MC, Boog C, Darmoni S, Favre M, Guigue L, Pinaud N, Hamek S, Hassler S, Kerdelhué G, Iordatii M, Kergosien Y, Leroy N, Mitouard T, Pereira S, Simon C, Duclos C, Venot A. Langage Iconique et Interfaces Interactives pour la Médecine. Colloque ANR TecSan 2012
7. **Lamy JB**, Soualmia LF, Venot A, Duclos C. Validation de la sémantique d'un langage iconique médical à l'aide d'une ontologie : méthodes et applications. Actes du congrès d'Ingénierie des Connaissances (IC) 2014, Clermont-Ferrand (taux de sélection 50%)
8. **Lamy JB**, Berthelot H. Ontopy : programmation orientée ontologie en Python. Actes du congrès d'Ingénierie des Connaissances IC 2015, Rennes
9. **Lamy JB**, Ugon A, Favre M, Duclos C, Venot A, Berthelot H. Comparaison et visualisation des contre-indications des médicaments. Actes du 3ème Symposium Ingénierie de l'Information Médicale SIIM 2015, Rennes
10. **Lamy JB**, Berthelot H, Favre M. Les boîtes arc-en-ciel : une nouvelle technique pour visualiser les relations hiérarchiques dans les ontologies. Actes du congrès Reconnaissance des Formes et Intelligence Artificielle RFIA 2016, Clermont-Ferrand
11. Kouame A, **Lamy JB**, Lo M, Brou KM. Vers un système iconique d'aide à la décision pour les praticiens de la médecine traditionnelle. Colloque Africain sur la Recherche en Informatique et Mathématiques CARI 2016
12. **Lamy JB**, Ugon A, Duclos C, Venot A, Berthelot H, Favre M. Une plate-forme visuelle pour une information comparative sur les nouveaux médicaments. Actes du congrès d'Ingénierie des Connaissances IC 2017, Caen
13. **Lamy JB**. Les oiseaux picorant artificiels : une nouvelle méta-heuristique inspirée du comportement des pigeons. Actes des Journées d'Intelligence Artificielle Fondamentale JIAF 2017, Caen

Rapports, mémoires et thèses

1. **Lamy JB**. Télé-surveillance médicale : conception d'un système d'analyse temporelle, validation par simulation et reconstitution par réalité virtuelle. Mémoire de 5ème année de pharmacie option recherche, 2002.
2. **Lamy JB**. Reconnaissance de scénarios, application à la télé-surveillance médicale à domicile. Mémoire de DEA, 2003.
3. **Lamy JB**. Vers un langage graphique de représentation des connaissances sur les médicaments. Thèse de pharmacie, 2006.
4. **Lamy JB**. Conception et évaluation de méthodes de visualisation des connaissances médicales : Mise au point d'un langage graphique et application aux connaissances sur le médicament. Thèse de sciences, 2006.
5. Duclos C, Hamek S, Hardouin E, **Lamy JB**, Mitouard T, Simon C, Venot A. Fonctionnalités du dossier médical pouvant bénéficier de l'approche graphique. Rapport (43 pages), 2009
6. **Lamy JB**, Venot A, Duclos C, Iordatii M, Kerdelhué G, Darmoni S. Complémentation du langage VCM. Rapport (25 pages), 2010
7. Mitouard T, Simon C, Hardouin E, Venot A, **Lamy JB**, Duclos C. Spécifications détaillées des fonctionnalités à implémenter pour le dossier médical. Rapport (47 pages), 2010
8. Beuscart MC, Dahamna B, Darmoni S, Duclos D, Hamek S, Hassler S, Kerdelhué G, **Lamy JB**, Leroy N, Morel C, Venot A. Conception d'un environnement graphique pour représenter les résultats d'une requête dans CISMeF-Bonnes Pratiques. Rapport (30 pages), 2010
9. Beuscart MC, Duclos C, Favre M, Hamek S, Hassler S, **Lamy JB**, Leroy N, Pereira S, Serrot E, Venot A. Conception d'une interface graphique de consultation du contenu de bonnes pratiques cliniques et d'un outil d'annotation des GBP. Rapport (28 pages), 2010
10. **Lamy JB**, Berthelot H, Ugon A, Kraiem L. Rapport scientifique du projet VIIIP (Visualisation Intégrée de l'Information sur l'Innovation Thérapeutique). Rapport (152 pages), 2016

Valorisation

— **Langage iconique VCM (Visualisation des Connaissances Médicales) :**
Ce langage iconique breveté (US, international) a conduit au développement d'un serveur iconique et à son intégration dans plusieurs produits commercialisés par différentes sociétés en France et à l'étranger :

- Intégration dans Vidal Reco de la société Vidal (http://www.vidal.fr/recommandations/1585/angor_stable/la_maladie/),
- Intégration dans le logiciel de gestion de dossier patient éO Médecin de Silk Informatique (http://silk-info.com/eo-medecin.html),
- Intégration dans le logiciel de gestion de dossier hospitalier de MainCare (ex-McKesson) (http://www.maincare.fr),
- Intégration dans le portail multi-terminologique de l'équipe CISMeF de l'hôpital de Rouen (http://www.ehtop.eu/hetop/)
- Intégration dans un logiciel d'e-learning du centre d'expertise en santé de Sherbrooke (http://expertise-sante.com)

- **Logiciels libres :**
  - **Module OwlReady :**
    OwlReady est un module Python pour accéder et opérationnaliser des ontologies formelles OWL (Web Ontology Language). Ce module permet la programmation orientée ontologie en Python, et la classification automatique via le raisonneur HermiT. Ce module est diffusé en logiciel libre (GNU LGPL v3). Il a été réutilisé par de nombreux autres chercheurs, y compris au-delà du domaine médical, par exemple en physique à l'institut Fraunhofer en Allemagne, ou pour la conception d'hélicoptère en Chine. Owlready a également fait l'objet d'un article sur le site LinuxFR (« Owlready : un module Python pour manipuler les ontologies OWL », http://linuxfr.org/news/owlready-un-module-python-pour-manipuler-les-ontologies-owl)
    (http://bitbucket.org/jibalamy/owlready2)
  - **Logiciel BibReview :**
    BibReview est un logiciel de gestion de bases de données bibliographiques qui propose des fonctions avancées pour les revues de la littérature. J'ai initialement réalisé ce logiciel lors de ma participation en tant que responsable de section du Yearbook of Medical Informatics, et il a été utilisé dans ce cadre. Ce logiciel est diffusé en logiciel libre (GNU GPL v3).
    (http://www.lesfleursdunormal.fr/static/informatique/bibreview/index_fr.html)
  - **Module PyMedTermino :**
    PyMedTermino est un module Python permettant d'utiliser les principales terminologies médicales en Python. Ce module permet notamment l'intégration de terminologies post-coordonnées (tel que les icônes VCM), et il propose aussi des fonctions avancées pour manipuler les terminologies. J'ai développé PyMedTermino au cours des projets L3IM, SiFaDo et VIIIP. Ce module est diffusé en logiciel libre (GNU LGPL v3).
    (http://www.lesfleursdunormal.fr/static/informatique/pymedtermino/index_fr.html)
  - **Module RainbowBox :**
    RainbowBox est un module Python pour générer des « boîtes arc-en-ciel », une technique de visualisation d'information pour les ensembles non-disjoints que j'ai développée. Ce module est diffusé en logiciel libre (GNU LGPL v3).
    (http://bitbucket.org/jibalamy/rainbowbox)
  - **Outil PDFTableExtract2 :**
    Cet outil en ligne de commande permet d'extraire les tableaux dans les fichiers PDF et de les transformer en HTML. Il s'agit d'une version améliorer de PDF-table-extract, qui est capable de reconnaître les espaces blancs comme des séparateurs de ligne ou de colonne. J'ai développé cet outil lors du projet VIIIP. Il est diffusé en logiciel libre (GNU LGPL v3).
    (https://bitbucket.org/jibalamy/pdftableextract2)

8.3 Enseignements

Les enseignements que je réalise correspondent à ceux de ma « fiche de poste » sur l'UFR SMBH, qui est une UFR de médecine, santé et biologie. Les enseignements comprendre des bases de données, de la modélisation, de la bioinformatique et de la bureautique.

Conscient des limites de ses enseignements d'informatique qui ne visent pas un public d'informaticiens, j'ai cherché à les diversifier, notamment en allant enseigner la programmation orientée objet et le développement web en master au Sénégal.

Début	Fin	Niveau	Intitulé	Taille	H éq TD/an	Statut
2015	2016	M1 SR	Programmation orientée objet en Python et développement web	30	48	MCF
2007	...	M1 TTV	Bioinformatique : analyse de séquence	35	27	vacataire puis MCF
2007	...	M1 TTV	Bioinformatique : transcriptomique	35	29	vacataire puis MCF
2014	...	M2 IBM	Terminologies médicales en Python	20	3	MCF
2014	...	M1 IBM	Modélisation UML en santé	12	30	MCF
2009	...	L3 SV	Base de données en biologie et bioinformatique	66	33	MCF
2011	...	L2 SSS	Traitement de texte basique	102	28	MCF
2011	...	L3 SSS	C2I niveau 1 - Traitement de texte avancé	65	14	MCF
2011	...	L3 SSS	C2I niveau 1 - Diaporama	65	14	MCF
2010	2014	M1 IBM	C2I santé	15	30	MCF
2008	2009	L1 SV	HTML		32	vacataire
2007	2008	L3 SV	Initiation à R		36	demi-ATER
2007	2008	L3 SSS	Statistiques descriptives		40	demi-ATER
2005	2007	L2 SV	Initiation à la programmation		53	vacataire
2004	2005	SRC 1	Algorithmique en Java		60	vacataire

SRC 1 : 1ère année Services et Réseaux de Communication (Université Paris 13, IUT de Bobigny)

Lx SSS : Licence Science Sanitaire et Sociale (Université Paris 13, UFR SMBH)

Lx SV : Licence Science de la vie (Université Paris 13, UFR SMBH)

Mx TTV : Master Biologie - Santé spécialité Thérapies et Technologies du Vivant (Université Paris 13, UFR SMBH)

Mx IBM : Master Informatique BioMédicale (Université Paris 5 et Paris 13)

M1 SR : Master 1 Système et Réseaux (Université Alioune Diop de Bambey, Sénégal)

Parcours

J'ai commencé à enseigner en deuxième année de thèse en tant que vacataire, pour diriger des TP de programmation Java à l'IUT en première année SRC. J'ai ensuite poursuivi en troisième année, cette fois-ci à l'UFR SMBH car je préférais combiner informatique et santé. J'ai alors effectué un enseignement (CM + TP) d'initiation à la programmation en Python en deuxième année de licence de biologie.

Après ma thèse, j'ai eu l'occasion d'être demi-ATER en statistiques pendant un an. J'ai enseigné les statistiques descriptives (CM + TD) ainsi que l'utilisation du logiciel R (TP).

Ensuite, j'ai repris l'enseignement de l'informatique en biologie en tant que vacataire, cette fois-ci en HTML (TP) en première année de licence, et en bioinformatique (TP) en Master 1.

Une fois nommé maître de conférence, j'ai pris la responsabilité des enseignements d'informatique et de bioinformatique en L3 de biologie (second semestre, CM + TP) et en M1 (CM + TP). J'ai par la suite fait évoluer ces enseignements, notamment en accordant une place plus importante aux banques de gènes et de protéines en L3, en basculant les enseignements de M1 de R vers Python et en introduisant les progrès récents de la bioinformatique comme les NGS (*Next Generation Sequencing*, séquençage à très haut débit).

Depuis 2010, j'ai été impliqué dans le M1 IBM, d'abord pour l'enseignement de l'informatique médicale et du C2I niveau 2 - métier de la santé, puis pour l'enseignement de la modélisation UML.

En 2011, j'ai pris en charge les enseignements de bureautique en L2 et L3 SSS, ainsi que l'organisation du C2I niveau 1.

En 2016, j'ai obtenu un financement de l'AUF (Agence Universitaire de la Francophonie) pour une mission d'enseignement (48h sur 14 jours) à l'Université Alioune Diop de Bambey. J'ai effectué cette mission du 24/07/2016 au 6/08/2016. Il s'agissait d'enseigner le langage Python, la programmation orientée objet et le développement web (serveur HTTP, WSGI, XML) aux étudiants de master 1 Système et Réseaux. Ceux-ci découvraient Python mais avaient déjà des bases de programmation objet dans d'autres langages (Java, PHP).

Responsabilités

Je suis correspondant C2I niveau 1 sur l'UFR SMBH depuis 2011.

J'ai aussi été correspondant C2I niveau 2 - métier de la santé sur l'UFR de 2010 à 2012.

Loisirs

Écriture (roman fantastique) ; musique (chant, lyre et harpe) ; reconstitution historique (XII - XIII$^{\text{ème}}$ siècle) ; jeux de société ; observation des oiseaux.

Chapitre 9

Bibliographie

« Besnik était penché sur la phrase qui le tourmentait depuis un long moment. Camarade Khrouchtchev, vous avez dit aux camarades chinois : "Nous avons perdu une Albanie et vous en avez gagné une". Il était impossible de rendre correctement les mots "une Albanie" en russe. »
(L'hiver de la grande solitude, Ismaïl Kadaré, éditions Fayard, 1999)

[1] USP Pictograms [http ://www.usp.org/audiences/consumers/pictograms/form.html].

[2] Abraham A. *Handbook of Measuring System Design*, chapter Artificial neural networks. John Wiley & Sons, 2005.

[3] Ackoff RL. From data to wisdom. *Journal of applies systems analysis*, 16 :3–9, 1989.

[4] Ake AL. *Abrégé de Médecine et pharmacopée africaines.* 2011.

[5] I Alecu, C Bousquet, P Degoulet, and M C Jaulent. PharmARTS : terminology web services for drug safety data coding and retrieval. *Stud Health Technol Inform*, 129(Pt 1) :699–704, 2007.

[6] B Alsallakh, W Aigner, S Miksch, and H Hauser. Radial Sets : Interactive Visual Analysis of Large Overlapping Sets. In *IEEE Transactions on Visualization and Computer Graphics (Proceedings Information Visualization 2013)*, volume 19, pages 2496–2505, 2013.

[7] B Alsallakh, L Micallef, W Aigner, H Hauser, S Miksch, and P Rodgers. Visualizing Sets and Set-typed Data : State-of-the-Art and Future Challenges. In *Eurographics Conference on Visualization (EuroVis)*, 2014.

[8] P Arapov, M Buffa, and A Ben Othmane. Programmer le web de données avec un "Wiki-based IDE". In *Actes du congrès d'Ingénierie des Connaissances (IC2014)*, pages 249–260, Clermont-Ferrant, France, 2014.

[9] Arbonnier M. *Trees, shrubs and lianas of West African dry zones.* 2004.

[10] N Aussenac-Gilles, B Biébow, and S Szulman. Modélisation du domaine par une méthode fondée sur l'analyse de corpus, 2000.

[11] N Aussenac-Gilles, S Despres, and S Szulman. *Ontology learning and population : Bridging the gap between text and knowledge*, chapter The terminae method and platform for ontology engineering from texts. IOSPress, 2008.

[12] H Azzag, D Da Costa, C Guinot, and G Venturini. Un aperçu de la fouille visuelle de données, 2008.

[13] F Baader, D Calvanese, D L McGuinness, D Nardi, and P L Patel-Schneider. *The description logic handbook : theory, implementation and applications.* Cambridge University Press, 2007.

[14] Baar T. Correctly defined concrete syntax for visual modeling languages. In *Model driven engineering languages and systems, Lecture Notes in Computer Science*, volume 4199, pages 111–125. Springer, Berlin, Heidelberg, 2006.

[15] M Babik and L Hluchy. Deep Integration of Python with Web Ontology Language. In *Proceedings of the 2nd workshop on scripting for the semantic web*, pages 1–5, Budva, Montenegro, 2006.

[16] E H Baehrecke, N Dang, K Babaria, and B Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *BMC bioinformatics*, 5 :84, 2004.

[17] Ballantyne PJ. Understanding users in the 'field' of medications. *Pharmacy*, 4(2) :19, 2016.

[18] Bertin J. *Sémiologie graphique : Les diagrammes - Les réseaux - Les cartes.* Editions de l'Ecole des Hautes Etudes en Sciences, Paris-La Haye, 1999.

[19] S Bertschi, S Bresciani, T Crawford, R Goebel, W Kienreich, M Lindner, V Sabol, and A Vande Moere. *Knowledge visualisation currents : from text to art to culture*, volume 13, chapter What is knowledge visualisation ? Eight reflections on an evolving discipline, page 32. Springer-Verlag, London, United Kindom, 2013.

[20] M.C. Beuscart-Zéphir, P. Elkin, S. Pelayo, and R. Beuscart. The human factors engineering approach to biomedical informatics projects : state of the art, results, benefits and challenges. *Yearb Med Inform*, 109-27, 2007.

[21] Blackwell AL. Pictorial representation and metaphor in visual language design. *Journal of visual languages and computing*, 12 :223–252, 2001.

[22] P A Bonatti, C Lutz, and F Wolter. The complexity of circumscription in description logic. *Journal of Artificial Intelligence Research*, 35 :717–773, 2009.

[23] F Bouali, S Devaux, and G Venturini. Visual mining of time series using a tubular visualization. *The visual computer*, 32(1) :15–30, 2016.

[24] F Bouali, F Plantard, A Bouséba, and G Venturini. Fouille visuelle de données temporelles avec DataTube2. *Journal d'Interaction Personne-Système*, 2(1) :4, 2011.

[25] J Bouaud and J B Lamy. *Yearbook of medical informatics*, volume 9, chapter A 2014 medical informatics perspective on clinical decision support systems : do we hit the ceiling of effectiveness ?, pages 163–6. 2014.

[26] Brooke J. *Usability evaluation in industry*, chapter SUS : a "quick and dirty" usability scale. Taylor and Francis, London, 1996.

[27] Burch M. A user study on judging the target node in partial link drawings. In *International Conference Information Visualisation (iV)*, pages 199–204, London, United Kingdom, 2017.

[28] Burkhard R. *Knowledge visualization - The use of complementary visual representations for the transfer of knowledge. A model, a framework, and four new approaches.* PhD thesis, Zürich, 2005.

[29] Burkhard RA. Learning from architects : The difference between knowledge visualization and information visualization. In *Proceeding of Information Visualisation*, London, United Kindom, 2004.

[30] Burkhard RA. *Knowledge and information visualization*, volume 226-243, chapter Towards a framework and a model for knowledge visualization : Synergies between information and knowledge visualization. Berlin Heidelberg, 2005.

[31] S K Card, J D Mackinlay, and B Shneiderman. *Readings in information visualization : Using vision to think.* Morgan Kaufmann.

[32] Charlet J. *L'ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales.* PhD thesis, Université Pierre et Marie Curie, Paris, 2002.

[33] K L Clark and F G McCabe. Ontology oriented programming in Go. *Applied Intelligence*, 24 :3–37, 2006.

[34] Cornet R. Definitions and qualifiers in SNOMED CT. *Methods Inf Med*, 48(2) :178–183, 2009.

[35] G Costagliola, A De Lucia, S Orefice, and G Tortora. A parsing methodology for the implementation of visual systems. In *IEEE Transactions on Software Engineering*, volume 23, pages 777–799, 1997.

[36] G Costagliola, M De Rosa, and V Fuccella. Extending local context-based specifications of visual languages. *Journal of Visual Languages and Computing*, 31 :184–195, 2015.

[37] A S Dadzie and A Burger. Providing visualisation support for the analysis of anatomy ontology data. *BMC Bioinformatics*, 6(74), 2005.

[38] Denekamp Y. Clinical decision support systems for addressing information needs of physicians. *Isr Med Assoc J*, 9(11) :771–776, 2007.

[39] Dewally S. Transcription des précautions d'emploi en pictogrammes, évaluation par le public, 1998.

[40] R Dowse and M Ehlers. Medicine labels incorporating pictograms : do they influence understanding and adherence ? *Patient Educ Couns*, 58(2005) :63–70, 2004.

[41] C Duclos-Cartolano and A Venot. Building and evaluation of a structured representation of pharmacokinetics information presented in SPCs : from existing conceptual views of pharmacokinetics associated with natural language processing to object-oriented design. *J Am Med Inform Assoc*, 10(3) :271–80, 2003.

[42] M Dudás, O Zamazal, and V Svátek. *Lecture Notes in Computer Science*, volume 8876, chapter Roadmapping and navigating in the ontology visualization landscape, pages 137–152. 2014.

[43] E Eden, R Navon, I Steinfeld, D Lipson, and Z Yakhini. GOrilla : a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC bioinformatics*, 10 :48, 2009.

[44] S Ellner and W Taha. The Semantics of Graphical Languages. In *Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, 2007.

[45] J Elouni, H Ltifi, M Ben Ayed, and M Masmoudi. Visual knowledge generation from data mining patterns for decision-making, 2016.

[46] J W Ely, J A Osheroff, M H Ebell, G R Bergus, B T Levy, M L Chambliss, and E R Evans. Analysis of questions asked by family doctors regarding patient care. *BMJ*, 319 :358–361, 1999.

[47] M J Eppler and R A Burkhard. Visual representations in knowledge management : Framework and cases. *Journal of knowledge management*, 2007.

[48] Eppler MJ. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information Visualization*, 5(3) :202–210, 2006.

[49] Erwig M. Semantics of Visual Languages, 1997.

[50] M J Field and K N Lohr. *Clinical practice guidelines : directions of a new program.* National Academy Press, Washington, DC, 1990.

[51] E Frank, M Hall, G Holmes, R Kirkby, B Pfahringer, I H Witten, and L Trigg. *Data mining and knowledge discovery handbook*, volume 1-10, chapter Weka : A machine learning workbench for data mining. Springer, US, 2005.

[52] E Frøkjær, M Hertzum, and K Hornbæk. Measuring Usability : Are Effectiveness, Efficiency, and Satisfaction Really Correlated ? In *Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems*, pages 345–352, The Hague, The Netherlands, 2000.

[53] A X Garg, N K Adhikari, H McDonald, M P Rosas-Arellano, P J Devereaux, J Beyene, J Sam, and R B Haynes. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes : a systematic review. *JAMA*, 293(10) :1223–1238, 2005.

[54] S Ghayas, S Sulaiman, M Khan, and J Jaafar. The effects of icon characteristics on users' perception. In *International Visual Informatics Conference (IVIC2013), Lecture Notes in Computer Science*, volume 8237, pages 652–663, 2013.

[55] K Goddard, A Roudsari, and J C Wyatt. Automation bias : empirical results assessing influencing factors. *Int J Med Inf*, 83(5) :368–75, 2014.

[56] Goldman NM. Ontology-oriented programming : static typing for the inconsistent programmer. In *Lecture notes in computer science : the SemanticWeb, ISWC*, volume 2870, pages 850–865, 2003.

[57] Graham MJ. *Visualising Multiple Overlapping Classification Hierarchies.* PhD thesis, 2001.

[58] C Grasso, A Joshi, and E Siegel. Visualization of pain severity events in clinical records using semantic structures. In *IEEE Tenth International Conference on Semantic Computing (ICSC)*, volume 321-324, 2016.

[59] N Griffon, G Kerdelhué, S Hamek, S Hassler, C Boog, J B Lamy, C Duclos, A Venot, and S J Darmoni. Design and usability study of an iconic user interface to ease information retrieval of medical guidelines. *J Am Med Inform Assoc*, 21(e2) :e270–7, 2014.

[60] N Griffon, G Kerdelhué, L F Soualmia, T Merabti, J Grosjean, J B Lamy, A Venot, C Duclos, and S J Darmoni. Evaluating alignment quality between iconic language and reference terminologies using similarity metrics. *BMC medical informatics and decision making*, 14 :17, 2014.

[61] S Grimm and B Motik. Closed world reasoning in the semantic web through epistemic operators. In *Proceedings of the OWLED 2005 workshop on OWL : Experiences and directions*, Galway, Ireland, 2005.

[62] S Grimm, B Motik, and C Preist. Matching semantic service descriptions with local closed-world reasoning. In *European Semantic Web Conference*, volume 575-589. Springer Berlin Heidelberg, 2006.

[63] J M Grimshaw and I T Russel. Effect of clinical guidelines on medical practice : a systematic review of rigorous evaluations. *Lancet*, 342(8883) :1317–22, 1993.

[64] Haarslev V. *Visual Language Theory*, volume 261-292, chapter A fully formalized theory for describing visual notations. Springer, New York, 1998.

[65] A Hayashi, T Itoh, and M Matsubara. *Knowledge visualisation currents : from text to art to culture*, volume 113, chapter Colorscore : Visualization and condensation of classical music, page 128. Springer-Verlag, London, United Kindom, 2013.

[66] M Hazman, S R El-Beltagy, and A Rafea. A survey of ontology learning approaches. *International Journal of Computer Applications*, 22(9) :36–43, 2011.

[67] M Horridge and S Bechhofer. The OWL API : A Java API for OWL ontologies. *Semantic Web 2*, pages 11–21, 2011.

[68] I Horrocks and U Sattler. Decidability of SHIQ with complex role inclusion axioms. *Artificial Intelligence*, 160 :2004, 2003.

[69] Hripcsak G. Writing Arden Syntax Medical Logic Modules. *Comput Biol Med*, 24(5) :331–63, 1994.

[70] Z Hu, J H Hung, Y Wang, Y C Chang, C L Huang, M Huyck, and C DeLisi. VisANT 3.5 : multi-scale network visualization, analysis and inference based on the gene ontology. *Nucleic Acids Res*, 37(Web Server issue) :W115–21, 2009.

[71] H Huang and H H Lai. Factors influencing the usability of icons in the LCD touchscreen. *Displays*, 29(4) :339–344, 2008.

[72] S C Huang, R G Bias, and D Schnyer. How are icons processed by the brain ? Neuroimaging measures of four types of visual stimuli used in information systems. *Journal of the association for information science and technology*, 66(4) :702–720, 2015.

[73] Hustadt U. Do we need the closed-world assumption in knowledge representation. In *Working notes of the KI'94 workshop : Reasoning about structured objects, knowledge representation meets databases (KRDB'94)*, volume D-94-11, pages 24–26, 1994.

[74] S W Hwang, N Tram CQ, and N Knarr. The effect of illustrations on patient comprehension of medication instruction labels. *BMC Family Practice*, 6 :26, 2005.

[75] K Hämeen-Anttila, K Kemppainen, H Enlund, J B Patricia, and A Marja. Do pictograms improve children's understanding of medicine leaflet information ? *Patient Educ Couns*, 55 :371–378, 2004.

[76] Inselberg A. *Parallel coordinates*. Springer, 2009.

[77] M Iordatii, A Venot, and C Duclos. Designing concept maps for a precise and objective description of pharmaceutical innovations. *BMC medical informatics and decision making*, 13 :10, 2013.

[78] M Iordatii, A Venot, and C Duclos. Design and evaluation of a software for the objective and easy-to-read presentation of new drug properties to physicians. *BMC medical informatics and decision making*, 15 :42, 2015.

[79] C M Johnson, T R Johnson, and J Zhang. A user-centered framework for redesigning health care interfaces. *J Biomed Inform*, 38(1) :75–87, 2005.

[80] A Kalyanpur, D Pastor, S Battle, and J Padget. Automatic mapping of OWL ontologies into Java. In *Proceedings of the Sixteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2004)*, pages 98–103, 2004.

[81] L Kascak, C B Rébola, R Braunstein, and J A Sanford. Icon design for user interface of remote patient monitoring mobile devices. In *Proceedings of the 31st ACM international conference on design of communication*, volume 77-84, 2013.

[82] V Kashyap and A Borgida. Representing the UMLS Semantic Network in OWL. In *Proceedings of ISWC 2003 (International Semantic Web Conference)*, volume 1-16, 2003.

[83] A Katifori and C Halatsis. Ontology visualization methods - A survey. *ACM Computing Surveys*, 39(4) :10, 2007.

[84] A Katifori, E Torou, C Vassilakis, G Lepouras, and C Halatsis. Selected results of a comparative study of four ontology visualization methods for information retrieval tasks. In *Research Challenges in Information Science*, volume 133-140, 2008.

[85] C M Keet, M Xakaza, and L Khumalo. Verbalising OWL ontologies in isiZulu with Python. In *Demo at the 14th Extended Semantic Web Conference (ESWC17)*, 2017.

[86] Keet CM. Representing and aligning similar relations : Parts and wholes in isiZulu vs. English. In *Proceedings of the first International Conference on Language, Data and Knowledge, Lecture Notes in Artificial Intelligence*, volume 10318, pages 58–73, Galway, Ireland, 2017. Springer.

[87] J Kerharo and J G Adam. *Pharmacopée traditionnelle sénégalaise : plantes médicinales et toxique.* Paris, 1975.

[88] A A Keshtkar, S Semnani, A Pourshams, H Khademi, G Roshandel, P Boffetta, and R Malekzadeh. Pictogram use was validated for estimating individual's body mass index. *J Clin Epidemiol*, 63(6) :655–9, 2010.

[89] S Khodambashi, H Gilstad, and Ø Nytrø. Usability evaluation of clinical guidelines on the web using eye-tracker. *Stud Health Technol Inform*, 228 :95–99, 2016.

[90] M Knorr, J J Alferes, and P Hitzler. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 175 :1528–1554, 2011.

[91] H Knublauch, D Oberle, P Tetlow, and E Wallace. A Semantic Web Primer for Object-Oriented Software Developers. *W3C Working Group Note*, 2006.

[92] S Koide, J Aasman, and S Haflich. OWL vs. Object Oriented Programming. In *the 4th International Semantic Web Conference (ISWC 2005), Workshop on Semantic Web Enabled Software Engineering (SWESE)*, pages 1–15, 2005.

[93] S Koide and T Takeda. OWL-Full Reasoning from an Object Oriented Perspective. *Lecture Notes in Computer Science, ASWC 2006*, 4185 :263–277, 2006.

[94] Konan A. *Place de la médecine traditionnelle dans les soins de santé primaires à Abidjan, Côte d'Ivoire.* PhD thesis, Université Toulouse III-Paul Sabatier, 2012.

[95] A Krishnadhi, K Sengupta, and P Hitzler. Local closed world semantics : Keep it simple, stupid ! In *Proceedings of the 2011 International Workshop on Description Logics (DL2011)*, volume 745-756, 2011.

[96] N C Kuicheu, N Wang, G N Fanzou Tchuissang, F Siewe, and D Xu. Description logic based icons semantics : An ontology for icons. In *International Conference on Signal Processing (ICSP 2012)*, volume 1260-1263, Beijing, China, 2012.

[97] G J Kuperman, A Bobb, T H Payne, A J Avery, T K Gandhi, G Burns, D C Classen, and D W Bates. Medication-related clinical decision support in computerized provider order entry systems : a review. *J Am Med Inform Assoc*, 14(1) :29–40, 2007.

[98] C E Kuziemsky and F Lau. A four stage approach for ontology-based health information system design. *Artif Intell Med*, 50(3) :133–48, 2010.

[99] J B Lamy, H Berthelot, C Capron, and M Favre. Rainbow boxes : a new technique for overlapping set visualization and two applications in the biomedical domain. *Journal of Visual Language and Computing*, 43 :71–82, 2017.

[100] J B Lamy, H Berthelot, and M Favre. Les boîtes arc-en-ciel : une nouvelle technique pour visualiser les relations hiérarchiques dans les ontologies. In *Conférence Nationale en Intelligence Artificielle (CNIA)*, volume 75-82, Clermont-Ferrand, France, 2016.

[101] J B Lamy, H Berthelot, and M Favre. Rainbow boxes : a technique for visualizing overlapping sets and an application to the comparison of drugs properties. In *International Conference Information Visualisation (iV)*, pages 253–260, Lisboa, Portugal, 2016.

[102] J B Lamy, H Berthelot, M Favre, A Ugon, C Duclos, and A Venot. Using visual analytics for presenting comparative information on new drugs. *J Biomed Inform*, 71 :58–69, 2017.

[103] J B Lamy, H Berthelot, A Ugon, L Kraiem, F Badra, C Duclos, and A Venot. *Rapport scientifique du projet VIIIP (Visualisation Intégrée de l'Information sur l'Innovation Thérapeutique)*. 2016.

[104] J B Lamy, C Duclos, A Bar-Hen, P Ouvrard, and A Venot. An iconic language for the graphical representation of medical concepts. *BMC Medical Informatics and Decision Making*, 8 :16, 2008.

[105] J B Lamy, C Duclos, V Rialle, and A Venot. Vers une méthode rigoureuse de conception des langages graphiques s'appuyant sur les sciences cognitives. In Poulet, F. and Le Grand, B., editor, *4ème atelier visualisation et extraction de connaissances, 6èmes journées francophones d'Extraction et de Gestion des Connaissances*, Lille, France, 2006.

[106] J B Lamy, C Duclos, and A Venot. De l'analyse d'un corpus de texte à la conception d'une interface graphique facilitant l'accès aux connaissances sur le médicament. In *Actes des 20es Journées Francophones d'Ingénierie des Connaissances*, volume 265-276, Hammamet, Tunisie, 2009. PUG.

[107] J B Lamy, V Ebrahiminia, C Riou, B Séroussi, J Bouaud, C Simon, S Dubois, A Butti, G Simon, M Favre, H Falcoff, and A Venot. How to translate therapeutic recommendations in clinical practice guidelines into rules for critiquing physician prescriptions ? Methods and application to five guidelines. *BMC Medical Informatics and Decision Making*, 10 :31, 2010.

[108] J B Lamy and L F Soualmia. Formalization of the semantics of iconic languages : An ontology-based method and four semantic-powered applications. *Knowledge-Based System*, 135 :159–179, 2017.

[109] J B Lamy, L F Soualmia, C Duclos, and A Venot. Formalisation de la sémantique des langages iconiques : méthode à base d'ontologie et applications. *Revue d'intelligence artificielle (RIA)*, 30(5) :579–606, 2016.

[110] J B Lamy, L F Soualmia, G Kerdelhué, A Venot, and C Duclos. Validating the semantics of a medical iconic language using ontological reasoning. *J Biomed Inform*, 46(1) :56–67, 2013.

[111] J B Lamy and R Tsopra. Translating visually the reasoning of a perceptron : the weighted rainbow boxes technique and an application in antibiotherapy. In *International Conference Information Visualisation (iV)*, pages 256–261, London, United Kingdom, 2017.

[112] J B Lamy, A Ugon, and H Berthelot. Automatic extraction of drug adverse effects from product characteristics (SPCs) : A text versus table comparison. *Stud Health Technol Inform*, 228 :339–343, 2016.

[113] J B Lamy, A Ugon, H Berthelot, and M Favre. A visual decision support system for helping physicians to make a decision on new drugs. In *Decision Support Systems VII. Data, information and knowledge visualization in decision support systems, Lecture Notes in Business Information Processing*, volume 282. Springer, 2017.

[114] J B Lamy, A Ugon, C Duclos, A Venot, H Berthelot, and M Favre. Une plate-forme visuelle pour une information comparative sur les nouveaux médicaments. In *Actes du congrès d'Ingénierie des Connaissances (IC)*, Caen, 2017.

[115] J B Lamy, A Ugon, M Favre, C Duclos, A Venot, and H Berthelot. Comparaison et visualisation des contre-indications des médicaments. In *Actes du 3ème Symposium Ingénierie de l'Information Médicale (SIIM)*, 2015.

[116] J B Lamy, A Venot, A Bar-Hen, P Ouvrard, and C Duclos. Design of a graphical and interactive interface for facilitating access to drug contraindications, cautions for use, interactions and adverse effects. *BMC Medical Informatics and Decision Making*, 8 :21, 2008.

[117] J B Lamy, A Venot, and C Duclos. PyMedTermino : an open-source generic API for advanced terminology services. *Stud Health Technol Inform*, 210 :924–928, 2015.

[118] Lamy JB. *Conception et évaluation de méthodes de visualisation des connaissances médicales : Mise au point d'un langage graphique et application aux connaissances sur le médicament*. PhD thesis, 2006.

[119] Lamy JB. Owlready : Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artif Intell Med*, 80 :11–28, 2017.

[120] T Laska, I Tsimbal, S Golubkov, and Y A Petrova. *Knowledge visualisation currents : from text to art to culture*, volume 147, chapter Practice of using virtual reconstruction in the restoration of monumental painting of the church of the transfiguration or our savior on Nereditsa hill, page 164. Springer-Verlag, London, United Kindom, 2013.

[121] C L Lassen, K Fragemann, T Klier, N Meyer, B M Graf, and C H Wiese. Knowledge levels of pharmaceutical sales representatives in pain therapy : a descriptive questionnaire-based study. *Eur J Clin Pharmacol*, 68(2), 2012.

[122] Lexchin J. What information do physicians receive from pharmaceutical representatives ? *Can Fam Physician*, 43 :941–945, 1997.

[123] R Lintern and M A Storey. Jambalaya Express : Ontology visualization-on-demand. In *International Protégé Conference*, 2005.

[124] T Liu, D Bangash Ahmed, F Bouali, and G Venturini. Visualizing a large collection of open datasets : an experiment with proximity graphs. In *Proceedings of the 2nd International Workshop on Open Data*, volume 7, New York, NY, USA, 2013.

[125] Z Liu, N Nersessian, and J Stasko. Distributed cognition as a theoretical framework for information visualization. In *IEEE Transactions on Visualization and Computer Graphics*, volume 14, 2008.

[126] X Ma, N Matta, J P Cahier, C Qin, and Y Cheng. From action icon to knowledge icon : Objective-oriented icon taxonomy in computer science. *Displays*, 39 :68–79, 2015.

[127] G Mancia, R Fagard, K Narkiewicz, J Redon, A Zanchetti, M Böhm, T Christiaens, R Cifkova, G De Backer, A Dominiczak, M Galderisi, D E Grobbee, T Jaarsma, P Kirchhof, S E Kjeldsen, S Laurent, A J Manolis, P M Nilsson, L M Ruilope, R E Schmieder, P A Sirnes, P Sleight, M Viigimaa, B Waeber, and F Zannad. 2013 ESH/ESC Practice Guidelines for the Management of Arterial Hypertension. *Blood Press*, 23(1) :3–16, 2014.

[128] Marchese FT. *Knowledge visualisation currents : from text to art to culture*, chapter Tables and early information visualization. Springer-Verlag, London, United Kindom, 2013.

[129] Marriott K. Constraint multiset grammars. In *Proceedings of IEEE symposium on Visual Languages*, 1994.

[130] P McGettigan, J Golden, J Fryer, R Chan, and J Feely. Prescribers prefer people : the sources of information used by doctors for prescribing suggest that the medium is more important than the message. *Br J Clin Pharmacol*, 51 :184–189, 2000.

[131] Meunier JG. La structure générique des systèmes sémiotiques. *Recherche sémiotique / Semiotic inquiries (RSSI)*, 8 :75–107, 1988.

[132] Meunier JG. The categorial structure of iconic languages. *Theory&Psychology*, 8(6) :805–825, 1998.

[133] M Meyer, T Gîrba, and M Lungu. Mondrian : an agile information visualization framework. In *SOFTVIS conference*, pages 135–144, Brighton, United Kingdom, 2006.

[134] Meyer R. Knowledge visualization. In *Media informatics advanced seminar on information visualization*, 2008.

[135] B Motik, I Horrocks, and U Sattler. Adding Integrity Constraints to OWL. In *Proceedings of the OWLED 2007 workshop on OWL : Experiences and directions*, Innsbruck, Austria, 2007.

[136] B Motik and R Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5) :93–154, 2010.

[137] B Motik, R Shearer, and I Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36 :165–228, 2009.

[138] A Moxey, J Robertson, D Newby, I Hains, M Williamson, and S A Pearson. Computerized clinical decision support for prescribing : provision does not guarantee uptake. *J Am Med Inform Assoc*, 17(1) :25–33, 2010.

[139] Moynihan R. Key opinion leaders : independent experts or drug representatives in disguise ? *BMJ (Clinical research ed.)*, 336(7658) :1402–3, 2008.

[140] P E Muijrers, R P Grol, J Sijbrandij, R Janknegt, and J A Knottnerus. Differences in prescribing between GPs : impact of the cooperation with pharmacists and impact of visits from pharmaceutical industry representatives. *Fam Pract*, 22(6) :624–630, 2005.

[141] H Müller, R Reihs, A E Posch, A Kremer, D Ulrich, and K Zatloukal. Data driven GUI design and visualization for a NGS based clinical decision support system, 2016.

[142] C Nakamura and Q Zeng-Treitler. A taxonomy of representation strategies in iconic communication. *Int. J. Human-Computer Studies*, 70(2012) :535–551, 2012.

[143] Nake F. Data, Information, and Knowledge. In Liu, K. and Clarke, R.J. and Andersen, P.B. and Stamper, R.K., editor, *Organizational Semiotics : Evolving a Science of Information Systems*, volume 41-50, Montréal, Québec, Canada, 2001. Kluwer.

[144] K N'Guessan, B Kadja, G N Zihiri, D Traoré, and L Aké-Assi. Screening phytochimique de quelques plantes médicinales ivoiriennes utilisées en pays Krobou (Agboville, Côte-d'Ivoire), 2009.

[145] K N'Guessan, F H Tra Bi, and M W Koné. Étude ethnopharmacologique de plantes antipaludiques utilisées en médecine traditionnelle chez les Abbey et Krobou d'Agboville (Côte d'Ivoire), 2009.

[146] Paivio A. *Mental representations - A dual coding approach*. Oxford University Press, New York, 1986.

[147] Paul G. *La psychologie de la forme*. Flammarion, Paris, 1979.

[148] S Pereira, S Hassler, S Hamek, C Boog, N Leroy, M C Beuscart-Zéphir, M Favre, A Venot, C Duclos, and J B Lamy. Improving access to clinical practice guidelines with an interactive graphical interface using an iconic language. *BMC medical informatics and decision making*, 14(1) :77, 2014.

[149] D Pfitzner, V Hobbs, and D Powers. A unified taxonomic framework for information visualization. In *Australian Institute of Computer Ethics conference*, Canberra, Australia, 2000.

[150] B Preiss, R Black, C Caron, and D Shapcott. Graphic summaries of expert knowledge for the medical curriculum : an experiment in second-year nephrology. *Methods Inf Med*, 31 :303–310, 1992.

[151] B Preiss, M Kaltenbach, J Zanazaka, and V Echave. Concept graphics : a language for medical knowledge. In Firsse, M., editor, *Proceedings of the 16th annual symposium on computer applications in medical care*, volume 515-519, Baltimore, USA, 1992. McGraw-Hill.

[152] B Preiss, V Échavé, S F Preiss, and M Kaltenbach. UVAL-MED a universal visual associative language for medicine. In *proceedings of the 18th annual symposium on computer applications in medical care*, volume 262-266, 1994.

[153] H Prosser, S Almond, and T Walley. Influences on GPs'decision to prescribe new drugs-the importance of who says what. *Fam Pract*, 20(1) :61–68, 2003.

[154] L Rajaonarivo, E Maisel, and P De Loor. An enactive based realtime 3D self-organization system for the exploration of a cultural heritage data base. In *20th International Conference Information Visualisation*, volume 100, page 105, Lisboa, Portugal, 2016.

[155] A Rector, M Horridge, L Iannone, and N Drummond. Use Cases for Building OWL Ontologies as Modules : Localizing, Ontology and Programming Interfaces & Extensions. In *4th Int Workshop on Semantic Web enabled software engineering (SWESE-08)*, 2008.

[156] Y Ren, J Z Pan, and Y Zhao. Closed world reasoning for OWL2 with NBox. *Tsinghua Science & Technology*, 15(6) :692–701, 2010.

[157] P Rodgers, G Stapleton, and P Chapman. Visualizing sets with linear diagrams. *ACM Transactions on Computer-Human Interaction*, 22(6) :27, 2015.

[158] A Sallaberry, N Pecheur, S Bringay, M Roche, and M Teisseire. Sequential patterns mining and gene sequence visualization to discover novelty from microarray data. *J Biomed Inform*, 44(5) :760–74, 2011.

[159] Y B Salman, H I Cheng, and P E Patterson. Icon and user interface design for emergency medical information systems : a case study. *Int J Med Inf*, 81(1) :29–35, 2012.

[160] M Samwald, K Fehre, J de Bruin, and K P Adlassnig. The Arden Syntax standard for clinical decision support : experiences and directions. *J Biomed Inform*, 45(4) :711–8, 2012.

[161] M Schaaf, F Jahn, K Tahar, C Kücherer, A Winter, and B Paech. Visualization of large ontologies in university education from a tool point of view. *Stud Health Technol Inform*, 228 :349–353, 2016.

[162] S Schulz and L Jansen. Formal ontologies in biomedical knowledge representation. *Yearbook of medical informatics*, 8 :132–46, 2013.

[163] L M Schwartz, S Woloshin, and H G Welch. The Drug Facts Box : Improving the communication of prescription drug information. *Ann Intern Med.*, 150 :516–27, 2009.

[164] R N Shiffman, J Dixon, C Brandt, A Essaihi, A Hsiao, G Michel, and R O'Connell. The GuideLine Implementability Appraisal (GLIA) : development of an instrument to identify obstacles to guideline implementation. *BMC medical informatics and decision making*, 5 :23, 2005.

[165] Shneiderman B. The eyes have it : a task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, volume 336-343, Washington, DC, USA, 1996. IEEE Computer Society.

[166] H Shortliffe E, R Davis, G Axline S, G Buchanan B, C Green C, and N Cohen S. Computer-based consultations in clinical therapeutics : explanation and rule acquisition capabilities of the MYCIN system. *Computers and biomedical research, an international journal*, 8(4) :303–20, 1975.

[167] C Simon, S Hassler, M C Beuscart-Zephir, M Favre, A Venot, C Duclos, and J B Lamy. Using an iconic language to improve access to electronic medical records in general medicine. *Stud Health Technol Inform*, 205 :333–7, 2014.

[168] E Simperl and M Luczak-Rösch. Collaborative ontology engineering : a survey, 2013.

[169] E Sirin, B Parsia, B C Grau, A Kalyanpur, and Y Katz. Pellet : A practical OWL-DL reasoner. *Web Semantics : Science, Services and Agents on the World Wide Web*, 5(2) :51–53, 2007.

[170] R J Sojourner and M S Wogalter. The influence of pictorials on evaluations of prescription medication instructions. *Drug Inf J*, 31 :963–972, 1997.

[171] G K Spurling, P R Mansfield, B D Montgomery, J Lexchin, J Doust, N Othman, and A I Vitry. Information from pharmaceutical companies and the quality, quantity, and cost of physicians' prescribing : a systematic review. *PLoS medicine*, 7(10) :e1000352, 2010.

[172] G Stevenson and S Dobson. Sapphire : Generating Java Runtime Artefacts from OWL Ontologies. In *Lecture Notes in Business Information Processing, Advanced Information Systems Engineering Workshops* , volume 83, pages 425–436, 2011.

[173] B Séroussi, J Bouaud, and E C Antoine. ONCODOC : a successful experiment of computer-supported guideline development and implementation in the treatment of breast cancer. *Artif Intell Med*, 22(1) :43–64, 2001.

[174] B Séroussi, J Bouaud, H Dreau, H Falcoff, C Riou, M Joubert, C Simon, G Simon, and A Venot. ASTI : a guideline-based drug-ordering system for primary care. In *Medinfo*, volume 10, pages 528–32, the Netherlands, 2001.

[175] B Séroussi, J Bouaud, D Sauquet, P Giral, P Cornet, H Falcoff, and J Julien. Why GPs do not follow computerized guidelines : an attempt of explanation involving usability with ASTI guiding mode. *Stud Health Technol Inform*, 160(Pt 2) :1236–40, 2010.

[176] R Tamblyn, A Huang, L Taylor, Y Kawasumi, G Bartlett, R Grad, A Jacques, M Dawes, M Abrahamowicz, R Perreault, N Winslade, L Poissant, and A Pinsonneault. A randomized trial of the effectiveness of on-demand versus computer-triggered drug decision support in primary care. *J Am Med Inform Assoc*, 15(4) :430–438, 2008.

[177] A R Tate, A G R Martin, A Ali, and J A Cassell. Using free text information to explore how and when GPs code a diagnosis of ovarian cancer : an observational study using primary care records of patients with ovarian cancer. *BMJ open*, 1(1) :e000025, 2011.

[178] S O Tergan, T Keller, and R A Burkard. Integrating knowledge and information : digital concept maps as a bridging technology. *Information Visualization*, 5(3) :167–174, 2006.

[179] F Thiessard, F Mougin, G Diallo, V Jouhet, S Cossin, N Garcelon, B Campillo, W Jouini, J Grosjean, P Massari, N Griffon, M Dupuch, F Tayalati, E Dugas, A Balvet, N Grabar, S Pereira, B Frandji, S Darmoni, and M Cuggia. RAVEL : retrieval and visualization in ELectronic health records. *Stud Health Technol Inform*, 180 :194–8, 2012.

[180] M Tory and T Möller. Rethinking visualization : A high-level taxonomy. In *IEEE Symposium on information visualization*, pages 151–158, Austin, Texas, USA, 2004.

[181] R Tsopra, S Kinouani, A Venot, M C Jaulent, C Duclos, and J B Lamy. Design of a visual interface for comparing antibiotics using rainbow boxes. *Stud Health Technol Inform*, 235 :529–533, 2017.

[182] R Tsopra, A Venot, and C Duclos. An algorithm using twelve properties of antibiotics to find the recommended antibiotics, as in CPGs . In *AMIA Annu Symp Proc*, volume 1115 ?24, 2014.

[183] R Tsopra, A Venot, and C Duclos. Towards evidence-based CDSSs implementing the medical reasoning contained in CPGs : application to antibiotic prescription. In *Stud Health Technol Inform*, volume 205, pages 13–7, 2014.

[184] van Tubergen A. The changing clinical picture and epidemiology of spondyloarthritis. *Nature reviews. Rheumatology*, 11(2) :110–8, 2015.

[185] J Vigneron, I Gindre, M Daouphars, P Monfort, S Georget, B Demoré, E Chenot, V Noirez, N Commun, F Laurelli, A Perrin, M Lux, M A Hoffman, and M Hoffman. Stabilis 3 : a European database on the stability and compatibility of injectable drugs. *European journal of hospital pharmacists*, 6 :77–78, 2006.

[186] T D Wang and B Parsia. CropCircles : Topology sensitive visualization of OWL class hierarchies. In *International Semantic Web Conference (ISWC)*, 2006.

[187] Ware C. *Visual thinking for design*, chapter Structuring two-dimensional space. Morgan Kaufmann, Burlington, USA, 2008.

[188] Ware C. *Visual thinking for design*. Morgan Kaufmann, Burlington, USA, 2008.

[189] Whitley KN. Visual programming languages and the empirical evidence for and against. *Journal of visual languages and computing*, 8(1) :109–142, 1997.

[190] W Wong, W Liu, and M Bennamoun. Ontology learning from text : A look back and into the future, 2012.

[191] C Wroe, W Solomon, A Rector, and J Rogers. DOPAMINE : a tool for visualizing clinical properties of generic drugs. In *Proceedings of the Fifth Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP)*, volume 61-65, 2000.

[192] Wyeld T. *Knowledge visualisation currents : from text to art to culture*, volume 129, chapter The implication of David Hockney's thesis for 3D computer graphics, page 145. Springer-Verlag, London, United Kindom, 2013.

[193] B Yang and J G Ganascia. Cartographie des connaissances dans les humanités numériques par l'iles de mémoires - une démonstration. In *IHM'14, 26e conférence francophone sur l'Interaction Homme-Machine*, volume 76-78, Lille, France, 2014.

[194] M Yasini, C Duclos, A Venot, E Lepage, and J B Lamy. A guideline-derived model to facilitate the implementation of test-ordering rules within a hospital information system. *Stud Health Technol Inform*, 192 :719–23, 2013.

[195] J S Yi, Y Kang, and J A Jacko. Understanding and characterizing insights : how do people gain insights using information visualization ? In *Proceeding of BELIV'08 BEyond time and errors : novel evaLuation methods for Information Visualization*, 2008.

[196] Yu AC. Methods in biomedical ontology. *J Biomed Inform*, 39(3) :252–266, 2006.

[197] Q Zeng-Treitler, S Perri, C Nakamura, J Kuang, B Hill, D Duc An Bui, G J Stoddard, and B E Bray. Evaluation of a pictograph enhancement system for patient instruction : a recall study. *J Am Med Inform Assoc*, 21 :1026–1031, 2014.

[198] P Zerbo, J Millogo-Rasolodimby, O G Nacoulma-Ouedraogo, and P Van Damme. Plantes médicinales et pratiques médicales au Burkina Faso : cas des Sanan, 2011.

[199] J C Zhao, L P Huang, L Tian, and K Q Wu. Knowledge system for helicopter transmission design based on ontology. In *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, volume 321-325, Wuhan, China, 2017.

[200] Zimmermann M. OWL2Java : a Java Code Generator for OWL, website. http ://www.incunabulum.de/projects/it/owl2java, 2009.

Représentation, iconisation et visualisation des connaissances :
Principes et applications à l'aide à la décision médicale

HDR de Jean-Baptiste Lamy

Résumé

Les connaissances sont de plus en plus nombreuses et elles évoluent rapidement, tandis que le temps disponible pour se les approprier est limité. Dans le domaine médical, ce problème est très important, car les connaissances sont particulièrement complexes et les erreurs peuvent être dramatiques.

Le format textuel reste la référence pour décrire les connaissances médicales avec précision. Mais les médecins n'ont guère le temps de se référer aux textes durant leurs consultations. Les systèmes d'aide à la décision clinique ont souvent été présentés comme une solution pour résoudre le problème de l'explosion des connaissances médicales. Cependant, la plupart de ces systèmes demandent au médecin un gros travail de saisie de données pour automatiser le raisonnement médical, ce qui conduit à une mauvaise acceptation par les utilisateurs. Ces systèmes n'ont donc pas abouti au succès initialement espéré.

Dans ce document, nous proposons une approche différente pour l'aide à la décision : la visualisation des connaissances. Il s'agit de présenter les connaissances de manière visuelle à l'utilisateur, afin de faciliter la prise de décision. Nous étudierons cette nouvelle approche selon une démarche multidisciplinaire intégrant différentes branches de l'informatique : la représentation des connaissances, la visualisation d'information et l'aide à la décision, avec un champ d'application médical.

Nous présenterons une démarche en 4 étapes pour la visualisation des connaissances :

1. L'étape de **représentation des connaissances** consiste à structurer et à formaliser les connaissances que l'on souhaite visualiser. Nous présenterons une méthodologie pour la modélisation du domaine et la construction d'ontologies dans un contexte médical, et un outil pour la programmation orientée ontologie en langage Python.
2. L'étape de **visualisation iconique** consiste à représenter la nature des connaissances à l'aide d'icônes et de langages iconiques. Nous décrirons une méthode pour formaliser la sémantique des langages iconiques, ainsi que les applications rendues possibles par cette formalisation.
3. L'étape de **visualisation structurelle** consiste à représenter la structure des connaissances à l'aide de technique de visualisation d'information. Nous présenterons les boîtes arc-en-ciel, une technique nouvelle que nous avons développée pour visualiser les ensembles non disjoints, qui peut s'appliquer à la visualisation des relations d'instanciation multiples dans les ontologies.
4. L'étape **applicative** consiste à intégrer la visualisation d'information au sein d'une application ou d'un site web, et à évaluer auprès des utilisateurs les performances et l'utilisabilité des outils obtenus. Ces évaluations permettront de valider notre approche.

Cette démarche sera illustrée au travers de **trois projets** auxquels j'ai participé en tant que postdoc puis maître de conférences au laboratoire LIMICS, portant sur l'utilisation d'un langage iconique médical dans les dossiers patients et les guides de bonnes pratiques cliniques, l'information comparative sur les nouveaux médicaments, et la conception de recettes iconiques pour des remèdes en médecine traditionnelle d'Afrique de l'Ouest.

Nous montrerons qu'il est possible de visualiser des connaissances abstraites complexes à l'aide d'icônes et de techniques de visualisation, mais également que cette approche conduit à de bonnes performances et une bonne acceptation lorsqu'elle est mise en œuvre pour l'aide à la décision.