



HAL
open science

Un portail sémantique pour la gestion des connaissances en cancérologie

Mathieu d'Aquin

► **To cite this version:**

Mathieu d'Aquin. Un portail sémantique pour la gestion des connaissances en cancérologie. Interface homme-machine [cs.HC]. Université Henri Poincaré - Nancy I, 2005. Français. NNT : . tel-00011698

HAL Id: tel-00011698

<https://theses.hal.science/tel-00011698v1>

Submitted on 28 Feb 2006

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.

Un portail sémantique pour la gestion des connaissances en cancérologie

THÈSE

présentée et soutenue publiquement le 14 décembre 2005

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1
(spécialité informatique)

par

Mathieu d'Aquin

Composition du jury

<i>Président :</i>	Jean-Paul Haton	Professeur à l'Université Henri Poincaré – Nancy 1
<i>Rapporteurs :</i>	Rose Dieng-Kuntz	Directeur de recherche INRIA, Sophia Antipolis
	Marie-Christine Jaulent	Directeur de recherche INSERM, Paris
<i>Examineurs :</i>	Jean Lieber	Maître de conférence à l'Université Henri Poincaré – Nancy 1
	Amedeo Napoli	Directeur de recherche CNRS, Nancy
	Michaël Rusinowitch	Directeur de recherche INRIA, Nancy

Remerciements

Je remercie en premier lieu Amedeo Napoli pour avoir accepté d'être mon directeur de thèse et pour la confiance qu'il m'a accordé. Merci aussi de m'avoir parfois traité d'ignorant et/ou de prétentieux, et de (parfois) s'en être excusé.

Je remercie chaleureusement Jean Lieber, qui a autant été un encadrant qu'un ami au cours de cette thèse. Travailler avec Jean m'a permis d'apprendre énormément, sur le plan scientifique, sur le plan humain, ainsi que sur le métier de chercheur. Il est de coutume de mentionner aussi ses chapeaux et ses fromages, qui n'ont pourtant eu aucune influence sur le déroulement de mon travail.

Merci à Jean-Paul Haton de m'avoir fait l'honneur de présider mon jury de thèse, ainsi que pour sa gentillesse, ses conseils et ses encouragements.

Un grand merci à Rose Dieng-Kuntz et Marie-Christine Jaulent d'avoir accepté de rapporter cette thèse, pour l'intérêt et pour l'enthousiasme avec lesquels elles ont évalué mon travail, dans des délais particulièrement courts.

Je ne pourrais oublier de remercier Sandrine Lafrogne, pour ses contributions au projet Kasimir, pour sa sympathie et pour m'avoir déchargé du choix de mon alimentation pendant un an. Je remercie aussi Sébastien Brachais pour sa très enrichissante collaboration aux travaux de l'équipe et pour avoir été, pendant deux ans, un camarade de bureau agréable.

Merci aussi à ceux, informaticiens, médecins, ergonomes ou autres, qui ont participé au projet Kasimir, parmi lesquels, chez HERMES, Olivier Croissant, Julien Lévêque et Fabien Palomares, chez Oncoolor, Isabelle Klein, au Centre Alexis Vautrin, Pierre Bey et Maria Rios, et au laboratoire d'ergonomie du CNAM de Paris, Pierre Falzon, Vanina Mollo et Catherine Sauvagnac.

Je souhaite remercier tous les membres de l'équipe Orpailleur pour leur accueil, leur soutien, leurs conseils et pour tout ce qui fait de cette équipe un lieu où il est agréable et profitable de travailler. Merci en particulier à mes camarades de bureau, László et Sandy, pour m'avoir supporté pendant tout ce temps.

Je souhaite également remercier Nadine Beurné, la très efficace et très sympathique assistante du DFD informatique.

Merci aux autres thésards du LORIA avec qui j'ai partagé tant de discussions et tant de cafés (Clara, Cyril, Fadi, Fred, Guillaume, Hacène, Jean-Luc, Laurent, Sylvain, Yves et tous les autres).

Je tiens également à remercier ma famille et tout particulièrement mon père pour m'avoir toujours fait confiance et toujours soutenu dans la conduite de mes études.

Merci à mes amis, Cat et Pedje, Céline et Pat, Maryline, Nat, Xavier, et Yasmin qui me rappellent de temps en temps que le « monde réel » existe aussi.

Je remercie aussi ceux que j'ai oublié de remercier, en particulier de ne pas m'en vouloir ;-).

Enfin, je remercie Marylène, ce mémoire, ma soutenance et ma vie ne seraient pas les mêmes sans elle, son gaspacho et sa Fiat Uno bleue.

Table des matières

Introduction	1
1 Le projet Kasimir pour la gestion des connaissances décisionnelles en cancérologie	5
1.1 Le contexte médical	5
1.1.1 Les référentiels Oncolor	6
1.1.2 Cycle de vie des référentiels : construction, application, adaptation, évolution . .	7
1.2 Objectifs du projet Kasimir	8
1.2.1 Gestion des connaissances et aide à la décision en médecine	8
1.2.2 Mise en œuvre de la gestion des connaissances pour la cancérologie	9
2 Le portail sémantique KASIMIR pour la cancérologie	11
2.1 Le Web sémantique	11
2.1.1 Tentative de définition et introduction aux principes du Web sémantique	11
2.1.2 Langages pour le Web sémantique	14
2.1.3 OWL et les logiques de descriptions	16
2.1.4 Outils pour le Web sémantique	22
2.1.5 Services Web et Web sémantique	24
2.1.6 Le Web sémantique en médecine	25
2.2 Un portail sémantique pour la cancérologie	25
2.2.1 Les portails sémantiques	25
2.2.2 Le portail sémantique KASIMIR : vue d'ensemble	26
2.3 Représentation des référentiels au sein du Web sémantique	28
2.3.1 Principes de la représentation des référentiels en OWL	28
2.3.2 Outils pour l'édition et la maintenance des connaissances	32
2.4 Services Web pour le raisonnement	35
2.5 Interfaces pour l'accès aux connaissances et l'aide à la décision	36
2.5.1 ÉDHIBOU : éditeur d'instances OWL	36
2.5.2 NAVHIBOU : navigation dans une ontologie OWL	37
2.6 Étendre OWL pour répondre aux objectifs de Kasimir	38

3	Raisonnement à partir de cas dans l'infrastructure du Web sémantique	39
3.1	Le RàPC comme élément d'un système à base de connaissances	39
3.1.1	Principes du RàPC	40
3.1.2	Le RàPC guidé par les connaissances	41
3.1.3	Acquisition et représentation des connaissances pour le RàPC	43
3.1.4	Le RàPC au sein du Web sémantique	43
3.2	Représentation des connaissances d'adaptation dans le cadre du Web sémantique	44
3.2.1	Le modèle des reformulations	44
3.2.2	Les reformulations en OWL	46
3.2.3	Raisonnements en OWL pour le RàPC	49
3.3	Un service de raisonnement à partir de cas pour le Web sémantique	50
3.3.1	Architecture et algorithmes	51
3.3.2	Opérations de transformation et services Web pour la mise en œuvre des reformulations	52
3.3.3	Notes sur l'implémentation	54
3.4	Le RàPC comme mécanisme de raisonnement pour le Web sémantique	55
4	Représentation explicite de points de vue	57
4.1	Les points de vue en représentation des connaissances	58
4.1.1	Points de vue en représentation des connaissances par objets	58
4.1.2	Points de vue en logique	60
4.1.3	Points de vue et graphes	61
4.1.4	Points de vue et contextes	61
4.1.5	Points de vue et ontologies	62
4.2	C-OWL : ontologies contextualisées	63
4.2.1	Ontologies, contextes et points de vue	63
4.2.2	Syntaxe et sémantique de C-OWL	64
4.2.3	Raisonnement avec C-OWL	67
4.2.4	Application de C-OWL	70
4.3	Points de vue pour la cancérologie : représentation des référentiels en C-OWL	70
4.3.1	Principes de la représentation	71
4.3.2	Représentation du référentiel pour le traitement du cancer du sein en C-OWL	72
4.3.3	Raisonnement multi-points de vue pour l'application des référentiels	77
4.3.4	Apports de la représentation multi-points de vue	79
4.4	De l'approche ascendante de la construction d'ontologies	80

5	Raisonnement à partir de cas décentralisé selon plusieurs points de vue	83
5.1	Pourquoi prendre en compte les points de vue en RÀPC	83
5.1.1	Raisonnement à partir de plusieurs cas	84
5.1.2	Raisonnement avec plusieurs vues sur la similarité	84
5.1.3	Raisonnement décentralisé	84
5.1.4	RÀPC dans le cadre du Web sémantique	85
5.2	RÀPC décentralisé avec C-OWL	85
5.2.1	Principe général	86
5.2.2	RÀPC local à un point de vue	86
5.2.3	Combinaison des points de vue selon les passerelles	87
5.3	Exemple d'application dans le cadre du traitement du cancer du sein	89
5.4	Note sur l'implémentation	91
5.5	D'autres approches pour la prise en compte de points de vue au sein du RÀPC	91
6	Travaux en cours au sein du projet KASIMIR	93
6.1	Prise en compte de types de données flous en OWL	93
6.1.1	Logiques de descriptions floues	94
6.1.2	Extension de OWL pour la prise en compte de types de données flous	95
6.2	Acquisition de connaissances d'adaptation (ACA)	96
6.2.1	ACA auprès d'experts	96
6.2.2	ACA semi-automatique	97
6.2.3	Vers une approche mixte de l'ACA	99
	Conclusion	101
	Bibliographie	105
	Annexes	117
A	Sources des modèles utilisés pour le RÀPC et algorithmes	117
A.1	Code OWL FULL du modèle des reformulations	117
A.2	Code OWL de l'exemple pour le traitement du cancer du sein	118
A.3	Algorithme pour la remémoration	120
A.4	Algorithme pour l'adaptation	121
A.5	Code OWL du modèle des opérations de transformation	121

Table des figures

1.1	Extrait du référentiel pour le traitement du cancer du sein (adapté de [(RSO), 2005e]). . .	6
2.1	Page web (HTML) décrivant une recette de gaspacho.	12
2.2	Description XML d'une recette de gaspacho.	13
2.3	Hierarchie de concepts d'une ontologie pour les recettes de cuisine.	14
2.4	Le « gâteau » du Web sémantique (traduit et adapté de [Koivunen et Miller, 2002]). . . .	15
2.5	Les trois variantes de OWL.	20
2.6	Vue d'ensemble de l'architecture du portail sémantique KASIMIR.	27
2.7	Extrait du référentiel pour le traitement du cancer du sein, repris de la figure 1.1.	28
2.8	Extrait de l'ontologie de domaine utile à la représentation du diagramme de la figure 1.1.	30
2.9	Les deux modules de visualisation de connaissances intégrés à PROTÉGÉ : PALÉTU- VIER(a) et HYPERTREE(b).	33
2.10	Le greffon KILT pour la visualisation des modifications effectuées sur une ontologie. . .	34
2.11	L'éditeur d'instances pour OWL ÉDHIBOU.	37
2.12	Utilisation de NAVHIBOU pour parcourir les classes d'une ontologie en OWL.	38
3.1	Remémoration et adaptation en RÀPC.	40
3.2	Les étapes du RÀPC guidé par les connaissances.	41
3.3	Un chemin de similarité entre <i>source</i> et <i>cible</i> (première ligne) et le chemin d'adapta- tion correspondant (deuxième ligne).	45
3.4	Exemple d'utilisation des reformulations.	45
3.5	Schéma du modèle des reformulations en OWL.	46
3.6	Architecture du service de RÀPC.	51
3.7	Modèle OWL pour la représentation d'opérations de transformation.	53
4.1	Un exemple de représentation en TROEPS.	59
4.2	Points de vue en RCO.	60
4.3	Exemple d'appariement entre deux contextes avec C-OWL.	65
4.4	Exemple de représentation d'appariements dans la syntaxe RDF/XML de C-OWL. . . .	68
4.5	Exemple d'application du principe de propagation de l'instanciation.	70
4.6	Démarche pour la construction d'une représentation multi-points de vue des connaissances.	71
4.7	Appariements mis en œuvre dans la représentation du référentiel pour le traitement du cancer du sein.	76

5.1	Un exemple de RÀPC décentralisé. Les première et deuxième lignes introduisent des problèmes sources. Les troisième et quatrième lignes définissent des appariements associés aux contextes. De la cinquième à la septième ligne sont décrites des assertions concernant le problème cible. Les différentes étapes du RÀPC décentralisé sur cet exemple sont finalement décrites entre la huitième et la onzième ligne.	88
5.2	Le chemin de similarité et le chemin d'adaptation du RÀPC local à O_2	88
6.1	Fonctions d'appartenance pour l'ensemble classique des réels supérieurs à 4 (a), pour l'ensemble flou des réels supérieurs à environ 4 (b) et pour l'ensemble flou des réels entre environ 3 et environ 5 (c).	95
6.2	Un couple de cas sources considéré comme un cas d'adaptation.	97

Introduction

Cette thèse présente à la fois une application et des extensions des principes et des technologies du Web sémantique afin de répondre aux besoins du projet KASIMIR pour la gestion des connaissances décisionnelles en cancérologie.

Contexte

Le Web sémantique. L'idée du Web sémantique est de rendre le *contenu* des ressources diffusées sur le Web accessible aux programmes informatiques de façon à ce que ceux-ci soient mieux à même d'aider les utilisateurs humains. Il s'agit de décrire ces ressources selon une représentation formelle, c'est-à-dire en lien avec une sémantique clairement définie et qui soit conçue pour une interprétation par des programmes. En termes plus imagés, le Web sémantique est une extension du Web actuel dans laquelle les machines « comprennent » les informations auxquelles elles accèdent et sont ainsi en mesure de les manipuler en tant que connaissances, au sein de mécanismes de raisonnements automatiques. À la base de l'infrastructure du Web sémantique, se trouvent les *ontologies*. Une ontologie est une description formelle des concepts relatifs à un domaine et des relations entre ces concepts. Elle apporte ainsi les éléments essentiels à la description sémantique des informations dans le domaine concerné. OWL (*Web Ontology Language*) est le langage standard pour la représentation des ontologies sur le Web et constitue en ce sens la principale technologie sur laquelle repose le Web sémantique. OWL est issu à la fois des langages du Web et de formalismes de représentation des connaissances, tels que les logiques de descriptions.

Le projet KASIMIR. Le but du projet KASIMIR est de permettre la gestion, la diffusion et l'évolution des connaissances utiles à la pratique de la cancérologie en Lorraine. Un des objectifs principaux de cette démarche est la mise à disposition, à destination des médecins, de connaissances « standard » sur la conduite à tenir en cancérologie, afin d'homogénéiser les pratiques et ainsi de les améliorer. La pratique de la cancérologie en Lorraine repose sur des documents appelés *référentiels*, qui s'apparentent à des protocoles de décision concernant des tâches comme le diagnostic, le traitement ou la surveillance post-thérapeutique pour certains types de cancers. Ces référentiels contiennent les informations utiles à la prise de décision dans les cas les plus courants et selon les connaissances actuelles en médecine. Ainsi, dans la majorité des cas médicaux, le référentiel concerné est *appliqué*, c'est-à-dire que les décisions concernant la prise en charge du patient sont établies en s'appuyant sur les recommandations du référentiel. Pour les cas où le référentiel n'apporte pas de réponse satisfaisante, la décision est prise par un comité d'experts, au cours d'une RCP (réunion de concertation pluridisciplinaire). Un tel comité regroupe des experts des différentes disciplines de la cancérologie dans le but d'*adapter* les recommandations d'un référentiel aux cas particuliers pour lesquels il ne répond pas, ces adaptations pouvant conduire à des évolutions du référentiel concerné.

Un portail sémantique pour la gestion des connaissances en cancérologie

La gestion des connaissances consiste en l'acquisition et la formalisation des connaissances utiles à un domaine particulier, afin d'en permettre la diffusion, l'utilisation et l'évolution. Dans le cadre du projet KASIMIR, il s'agit de fournir un accès intelligent aux connaissances utiles à la prise de décision en cancérologie, parmi lesquelles celles contenues dans les référentiels. L'infrastructure du Web sémantique, les principes et les technologies sur lesquels il repose, apportent pour cela un environnement adapté. Nous nous attachons dans cette thèse à l'exploiter pour la construction d'un *portail sémantique* dédié à la gestion des connaissances décisionnelles en cancérologie. Un portail sémantique est un portail du Web voué à la diffusion de connaissances utiles à une communauté particulière, selon les principes et les technologies du Web sémantique. Cela suppose la mise en œuvre d'une représentation formelle de ces connaissances au sein d'ontologies, de services d'accès reposant sur des mécanismes de raisonnements, ainsi que d'interfaces permettant l'interrogation de ces services par des utilisateurs, humains ou non. La construction du portail sémantique KASIMIR apporte ainsi une expérience concernant l'application des technologies du Web sémantique, en particulier de OWL, dans le domaine de la cancérologie, par nature vaste et complexe. L'approche suivie dans le développement de ce portail nous conduit par ailleurs à concevoir une architecture, des outils et des techniques génériques, indépendants du domaine de la cancérologie, et ainsi réutilisables pour la construction de portails sémantiques dans d'autres domaines, qu'ils soient médicaux ou non.

Mais si les technologies du Web sémantique fournissent une base adéquate pour la mise en œuvre de ce portail sémantique et des outils sur lesquels il repose, elles ne permettent pas de répondre à l'ensemble des problématiques liées à la gestion des connaissances et à l'aide à la décision dans le cadre du projet KASIMIR. Nous nous attachons ainsi dans un deuxième temps à les étendre dans le but de répondre à certaines de ces problématiques. En particulier, l'aide à la décision dans le cadre de l'adaptation des référentiels engendre des besoins en représentation des connaissances et en raisonnement pour lesquels les technologies actuelles du Web sémantique ne sont pas prévues. Le raisonnement à partir de cas (RÀPC) est un paradigme de raisonnement de l'intelligence artificielle qui repose sur l'adaptation des solutions de problèmes déjà résolus, jugés similaires au problème à résoudre. Il s'agit, dans le cadre de KASIMIR, d'établir une recommandation pour un cas auquel un référentiel ne répond pas, en adaptant les solutions fournies par le référentiel pour des cas similaires. Une étape importante dans la construction du portail sémantique est ainsi la mise en œuvre d'un service de RÀPC s'intégrant à l'infrastructure du Web sémantique et exploitant les connaissances diffusées au travers de celui-ci. Par ailleurs, plusieurs disciplines, telles que la chirurgie, la radiothérapie ou la chimiothérapie, interviennent lors de la prise en charge d'un patient, que ce soit dans le cadre de l'application des référentiels ou de leur adaptation au cours de RCP. Chacune de ces disciplines apporte son propre point de vue sur la cancérologie et collabore avec les autres dans la prise de décision. Pour cette raison, une part importante du travail présenté dans ce mémoire concerne la mise en œuvre explicite de la notion de points de vue pour la représentation des connaissances. Un mécanisme de RÀPC décentralisé, tirant parti de la multiplicité des points de vue et des relations qui les lient, est développé sur la base de cette représentation, afin d'être appliqué dans le cadre de l'aide à la décision pour l'adaptation des référentiels.

Plan du Mémoire

Ce mémoire n'est pas structuré conformément à la coutume qui veut que les premiers chapitres soient dédiés à l'état de l'art et les suivants aux contributions. Chaque chapitre concerne un des points clés de nos travaux, — le Web sémantique, le RÀPC, la représentation multi-points de vue, etc. — et comporte

sa propre part d'état de l'art et de contributions. Le premier chapitre est particulier puisqu'il s'agit d'un chapitre introductif concernant le projet KASIMIR. Il présente le contexte médical du projet et introduit les problématiques concernant la gestion des connaissances et l'aide à la décision qui y sont liées.

La chapitre 2 concerne l'application des principes et des technologies du Web sémantique pour la mise en œuvre d'un portail sémantique pour la gestion des connaissances en cancérologie. Pour cela, le domaine du Web sémantique est en premier lieu expliqué et les technologies sur lesquelles il repose sont détaillées. Une part importante du développement de ce portail sémantique concerne la représentation formelle des connaissances contenues dans les référentiels au sein du langage OWL. Une grande partie du chapitre 2 y est donc consacrée, ainsi qu'aux outils développés pour construire cette représentation, pour y donner accès et pour l'exploiter dans le cadre de l'aide à la décision.

La mise en œuvre d'un service de RÀPC intégré à l'infrastructure du Web sémantique est présentée dans le chapitre 3. Ce chapitre inclut une introduction au domaine du RÀPC et en particulier à l'approche adoptée dans le cadre du projet KASIMIR : le RÀPC *guidé par les connaissances*¹. Un modèle des connaissances utiles au processus de RÀPC (les *connaissances d'adaptation*) est développé au sein du langage OWL pour être utilisé comme cadre à la formalisation des connaissances relatives à la similarité et à l'adaptation dans le domaine d'application. Les connaissances d'adaptation ainsi formalisées sont opérationnalisées au sein d'un service de RÀPC générique, applicable aux ressources diffusées au travers du Web sémantique.

Dans le chapitre 4, la représentation multi-points de vue des connaissances est étudiée et différents formalismes, issus de différents paradigmes de représentation des connaissances, sont présentés. Parmi ceux-ci, C-OWL est une extension de OWL pour la représentation d'ontologies regroupant plusieurs contextes. C-OWL est utilisé pour mettre en œuvre les multiples points de vue correspondant aux disciplines de la cancérologie. La représentation multi-points de vue avec C-OWL est alors illustrée sur l'exemple du référentiel pour le traitement du cancer du sein.

Sur la base de la représentation multi-points de vue présentée dans le chapitre 4, le chapitre 5 s'intéresse au développement d'un mécanisme de RÀPC décentralisé. Celui-ci exploite la multiplicité des points de vue pour distribuer le processus de raisonnement et obtenir plusieurs points de vue sur la résolution du problème, ainsi que plusieurs points de vue sur la solution. Ces points de vue sont combinés au travers des échanges de connaissances rendus possibles par les relations reliant des éléments de différents points de vue.

Finalement, le chapitre 6 présente deux travaux en cours dans le cadre du projet KASIMIR. Il s'agit en premier lieu d'une extension du formalisme OWL pour l'introduction du *fou* dans la représentation et de la mise en œuvre de mécanismes de raisonnement tenant compte de cette représentation floue. D'autre part, comme pour tout autre type de connaissances, l'acquisition des connaissances d'adaptation pour le RÀPC est une tâche difficile, nécessitant la mise en œuvre de méthodes et d'outils dédiés. Le chapitre 6 présente les approches en cours de développement pour cela et se focalise en particulier sur la conception d'un outil de fouille de bases de cas pour l'acquisition semi-automatique de connaissances d'adaptation.

¹ce terme étant une traduction propre et imparfaite de *knowledge-intensive case-based reasoning*.

Chapitre 1

Le projet Kasimir pour la gestion des connaissances décisionnelles en cancérologie

Le projet KASIMIR est un projet pluridisciplinaire regroupant des experts en cancérologie du centre hautement spécialisé en cancérologie Alexis Vautrin (le CAV, à Vandœuvre-lès-Nancy), des membres du réseau de santé en cancérologie de Lorraine Oncolor, des chercheurs en psycho-ergonomie du laboratoire d'ergonomie du CNAM de Paris et des chercheurs en informatique de l'équipe Orpailleur du LORIA de Nancy. Il a pour objectif la gestion de connaissances décisionnelles en cancérologie. Les connaissances standard concernant la prise en charge d'un patient atteint de cancer en Lorraine sont contenues dans des documents appelés référentiels. Un référentiel est un protocole de décision, ou guide de bonnes pratiques, concernant par exemple le diagnostic, le traitement ou la surveillance post-thérapeutique d'un type de cancer particulier (cancer du sein, de la prostate, etc.). Dans ce cadre, la gestion des connaissances doit permettre, au travers d'outils d'aide à la décision, un accès intelligent aux connaissances contenues dans ces référentiels pour des médecins de niveaux d'expertise variés et géographiquement dispersés (des cancérologues de centres hautement spécialisés aux médecins généralistes). D'un point de vue médical, il s'agit d'amener, par la diffusion de connaissances standard et actualisées, à une homogénéisation des pratiques et ainsi, à une amélioration de la qualité des soins.

Ce chapitre présente le contexte médical du projet Kasimir, la pratique de la cancérologie en Lorraine et la mise en œuvre de la gestion des connaissances dans ce contexte.

1.1 Le contexte médical

Les connaissances standard concernant la prise de décision en cancérologie sont contenues dans des documents construits par des experts en cancérologie et qui doivent évoluer au fur et à mesure des avancées de la médecine. Cette section présente ces documents, les *référentiels*, leur forme et leur « cycle de vie », en se concentrant en particulier sur l'utilisation qui en est faite comme support d'aide à la décision dans le cadre de la prise en charge de patients atteints de cancer. Le référentiel pour le traitement du cancer du sein est, du fait de sa taille et de sa complexité, le premier à avoir été étudié au sein du projet KASIMIR et sera utilisé pour illustrer les exemples de ce chapitre et des suivants.

1.1.1 Les référentiels Oncolor

De nombreux référentiels ont été construits pour différents types de décisions (diagnostic, traitement, surveillance post-thérapeutique, inclusion de patients dans les protocoles d'essais thérapeutiques, etc.) et pour différentes localisations cancéreuses (sein, prostate, colon, etc.). Ceux-ci sont édités et diffusés par le réseau de santé Oncolor, au travers de son site internet [(RSO), 2005e]. À l'heure actuelle, ils sont principalement publiés sous la forme de documents électroniques. Ces documents sont généralement structurés selon les différentes disciplines impliquées dans différentes phases de la décision. Par exemple, le référentiel pour le traitement du cancer du sein est divisé en plusieurs parties, chacune supportant les décisions qui concernent une phase du traitement particulière, relevant de disciplines comme la chimiothérapie, la chirurgie, la radiothérapie ou l'hormonothérapie.

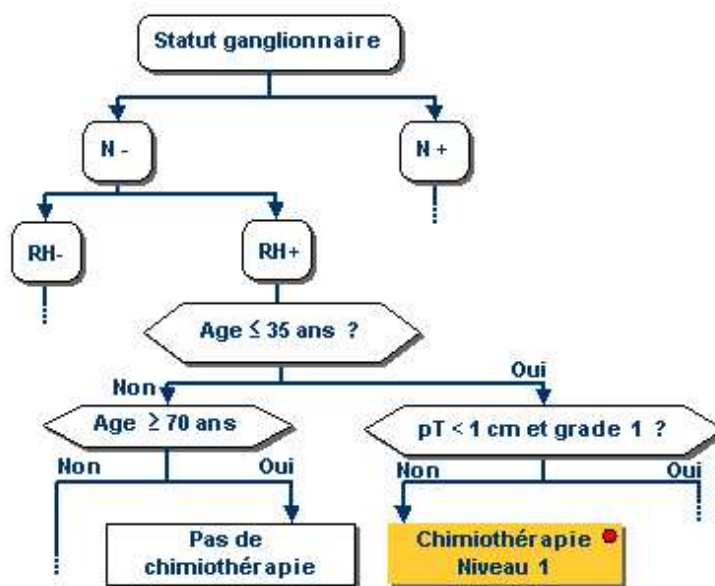


FIG. 1.1 – Extrait du référentiel pour le traitement du cancer du sein (adapté de [(RSO), 2005e]).

Un référentiel est supposé contenir toute les informations nécessaires pour permettre à un médecin de prendre les décisions les mieux adaptées dans les situations standard, c'est-à-dire les plus courantes et les plus étudiées. Dans chacune des parties structurant un référentiel, le vocabulaire nécessaire est d'abord introduit succinctement, sous une forme textuelle. La partie concernant la chimiothérapie du référentiel pour le traitement du cancer du sein indique par exemple que la « chimiothérapie de niveau 1 » est une cure de 4 à 6 séances d'injections selon les protocoles FEC 50 ou FEC 60. Ces protocoles correspondent à la combinaison de trois produits (fluoriuracile, épirubicine et cyclophosphamide) selon des dosages particuliers. Les éléments permettant d'associer les caractéristiques d'un patient et de sa maladie aux décisions les mieux adaptées sont ensuite introduits sous la forme de diagrammes, similaires à des arbres de décision. Chaque nœud d'un tel diagramme correspond à un test, à une question, concernant une ou plusieurs caractéristiques du patient. Un arc indique la branche à suivre en fonction des valeurs des caractéristiques testées. Parcourir le diagramme à partir de la racine permet d'aboutir à un ou plusieurs nœuds (le plus souvent aux nœuds feuilles) indiquant les décisions à prendre en fonction du chemin

suivi. Un exemple de ce type de diagramme, extrait de la partie chimiothérapie du référentiel pour le traitement du cancer du sein, est présenté à la figure 1.1. Les caractéristiques prises en compte dans cet exemple sont : le statut ganglionnaire, positif (N+) si les ganglions sont envahis, le statut des récepteurs hormonaux (RH+ ou RH-), l'âge du patient, la taille de la tumeur (pT) et le grade de la tumeur (I, II ou III). Ce diagramme indique par exemple que pour un patient dont les ganglions ne sont pas envahis (N-), dont les récepteurs hormonaux sont positifs (RH+), d'un âge inférieur à 35 ans et avec une tumeur de plus de 1 cm ou de grade différent de I, une chimiothérapie de niveau 1 sera recommandée.

1.1.2 Cycle de vie des référentiels : construction, application, adaptation, évolution

Construction

Les référentiels sont construits par des experts en cancérologie, selon les principes de la *médecine factuelle* (ou médecine fondée sur les faits, en anglais *evidence-based medicine* [(EBMWG), 1992]). Cela signifie qu'ils contiennent les éléments permettant de prendre les décisions les mieux adaptées, pour les situations les plus courantes, selon les connaissances actuelles en médecine. En ce sens, ils peuvent être vus comme une synthèse validée de la littérature récente en cancérologie.

Application

Appliquer un référentiel consiste à établir une recommandation pour un patient particulier, en fonction de ses caractéristiques propres et selon les indications du référentiel. Un référentiel est conçu pour contenir les connaissances nécessaires à la prise de décision dans les situations les plus courantes. Son application par des médecins, experts ou non, sera ainsi possible pour une majorité de patients. Le référentiel pour le traitement du cancer du sein, par exemple, permet de prendre en compte environ 60 à 70% des cas médicaux [Lieber *et al.*, 2002]. Il s'agit en particulier de patients atteints uniquement de cancer du sein au stade locorégional (non métastatique), de sexe féminin, sans contre-indication aux traitements recommandés et pour lesquels toutes les informations nécessaires sont disponibles.

Adaptation

Les cas qui ne sont pas pris en compte par le référentiel (cancer du sein chez l'homme, présence d'autres maladies, contre-indication, information manquante, etc.) sont discutés lors de *réunions de concertation pluridisciplinaires* (RCP²). Ces réunions regroupent des experts de différentes disciplines de la cancérologie (chimiothérapie, chirurgie, radiothérapie, hormonothérapie, etc.) dans le but de décider d'une recommandation pour les cas hors référentiel. Pour cela, les participants d'une RCP *adaptent* les recommandations du référentiel pour des cas similaires afin qu'elles correspondent au cas traité [Sauvagnac, 2000].

Évolution

Du fait qu'il contiennent des éléments conformes aux connaissances *standard* et *actuelles* en médecine, les référentiels sont amenés à évoluer régulièrement. Ces évolutions sont plus ou moins fréquentes en fonction du référentiel. Elles ont lieu selon un rythme d'environ 2 fois par an pour le traitement du cancer du sein. La première raison pour laquelle un référentiel peut nécessiter d'être mis à jour est l'évolution de la cancérologie elle-même. De nouvelles techniques apparaissent régulièrement, des études remettent en cause certaines pratiques ou en favorisent de nouvelles. Les référentiels doivent être mis à jour rapidement pour prendre en compte ces nouveaux éléments. La confrontation des référentiels avec

²D'abord appelées CDT, comités de décision thérapeutique, puis CCP pour comités de concertation pluridisciplinaires.

la pratique est une autre raison d'évolution de leur contenu. En effet, certaines situations non prévues par un référentiel peuvent s'avérer fréquentes en réalité et certaines recommandations issues d'un référentiel peuvent être difficiles à appliquer [Sauvagnac *et al.*, 1999].

1.2 Objectifs du projet Kasimir

L'objectif du projet KASIMIR, en particulier du point de vue informatique, concerne la mise en œuvre d'outils permettant la gestion des connaissances décisionnelles en cancérologie, notamment celles contenues dans les référentiels. La démarche adoptée pour le développement de ces outils est ainsi fortement guidée par les besoins liés à l'application en cancérologie, mais est également mise en œuvre dans une optique de réutilisabilité, s'appuyant sur des techniques et des technologies *génériques*, de façon à être applicables pour la gestion de connaissances dans d'autres domaines, médicales ou non.

1.2.1 Gestion des connaissances et aide à la décision en médecine

La *gestion des connaissances* (en anglais, *knowledge management*) consiste à acquérir et représenter les connaissances utiles à un domaine, une tâche ou une organisation particulière dans le but d'en favoriser l'accès, la réutilisation et l'évolution [Dieng *et al.*, 2000]. Ayant détecté les connaissances cruciales et les besoins, cela revient généralement à construire, maintenir et faire évoluer une représentation explicite de ces connaissances (appelée *mémoire d'entreprise* dans [Dieng *et al.*, 2000]). Il s'agit ensuite de fournir un accès à ces connaissances, c'est-à-dire de les diffuser dans le but d'en permettre une utilisation efficace. Nous nous intéressons plus particulièrement ici à la mise en œuvre de connaissances utiles à la décision en cancérologie afin d'en fournir un accès intelligent au travers d'outils d'aide à la décision.

La cancérologie, ou plus généralement la médecine, est un domaine d'application privilégié pour les systèmes d'aide à la décision. Un système d'aide à la décision médicale (souvent appelé *clinical decision support system* en anglais [Pusic et Ansermino, 2004]) peut être défini comme un programme informatique ayant pour entrée la description d'une situation médicale particulière et pour sortie des informations permettant d'aider un médecin dans la prise de décision concernant cette situation [Musen, 1997]. Un tel programme se présente le plus souvent comme un système à base de connaissances, intégrant les connaissances médicales nécessaires à la décision dans le cadre du domaine d'application et implémentant une ou plusieurs méthodes de résolution de problèmes [Musen, 1997].

Un précurseur dans le domaine des systèmes d'aide à la décision médicaux a été MYCIN, un *système expert* développé dans les années 70 à l'université de Stanford [Shortliffe, 1986]. MYCIN s'appuie sur une représentation à base de *règles* de type SI-ALORS et sur le *chaînage avant* comme technique d'inférence. Originellement, MYCIN était conçu pour diagnostiquer et soigner des maladies infectieuses du sang. Par la suite, son moteur d'inférence (EMYCIN : *Essential MYCIN*) ayant été extrait pour pouvoir être utilisé séparément, de nombreux autres systèmes ont été développés selon le même modèle dans différents domaines [Musen, 1997].

ONCOCIN est un autre système à base de règles, dédié plus spécifiquement à la cancérologie et développé lui aussi à l'université de Stanford [Shortliffe, 1986]. ONCOCIN a la particularité de représenter les décisions et les actions dans le temps, comme des séquences d'évènements. Comme indiqué dans [Peleg *et al.*, 2003], la majorité des systèmes actuels s'intéressant à la mise en œuvre de protocoles médicaux et de guide de bonnes pratiques cliniques (ASBRU, GLIF, etc.) s'appuient sur une représentation de ce type, fondée sur un modèle de « réseaux de tâches ».

ONCODOC [Séroussi *et al.*, 2001] est un système d'aide à la décision dédié à la diffusion et l'utilisation de *guides de bonnes pratiques* en cancérologie. Ces guides de bonnes pratiques se présentent sous la forme de documents textuels ayant un rôle très similaire à celui des référentiels du réseau Oncolor. Le

système s'appuie sur une représentation en *arbre de décisions* de ces documents, où un nœud correspond à un paramètre de décision et un arc à l'attribution d'une valeur au paramètre. Aux feuilles d'un tel arbre sont associés des plans de traitements recommandés. Une particularité d'ONCODOC vis-à-vis d'autres systèmes d'aide à la décision est que son utilisation repose sur une navigation hypertextuelle dans les arbres de décision. Aucun processus automatique n'est en effet utilisé pour associer la description d'un patient à la recommandation qui lui convient, laissant de cette façon toute liberté à l'utilisateur d'interpréter les données du patient et les connaissances contenues dans le guide de bonnes pratiques. En ce sens, ONCODOC se situe à mi-chemin entre la « formalisation » des connaissances médicales et une mise en œuvre documentaire.

1.2.2 Mise en œuvre de la gestion des connaissances pour la cancérologie

La mise en œuvre de la gestion des connaissances dans un domaine particulier amène à se poser un certain nombre de questions, concernant notamment le but de la gestion des connaissances, le type de connaissances mis en œuvre et les utilisateurs [Dieng *et al.*, 1999]. Les réponses à ces questions doivent guider le choix des méthodes, des techniques et des technologies employées. Concernant le projet KASIMIR, l'objectif poursuivi est d'amener à une homogénéisation des pratiques médicales par la diffusion des connaissances pour la prise de décision et ainsi de permettre l'amélioration de ces pratiques [Lieber *et al.*, 2002].

Les connaissances prises en compte dans ce cadre sont, en premier lieu, celles contenues dans les référentiels, mais aussi toute connaissance impliquée dans la prise de décision en général. L'adaptation des référentiels lors de RCP est une tâche particulière qui nécessite des connaissances d'un type particulier (appelées par la suite *connaissances d'adaptation*). La mise en œuvre du projet KASIMIR suppose ainsi le développement d'outils et de méthodes pour l'acquisition, la représentation et l'opérationnalisation de ces connaissances d'adaptation, avec pour objectif l'aide à la décision dans le cadre des RCP. La cancérologie est de plus un domaine fortement multi-disciplinaire, faisant intervenir des disciplines telles que la chirurgie, la radiothérapie, la chimiothérapie et l'hormonothérapie. Ces disciplines intègrent des connaissances différentes, considèrent les problèmes selon différents angles et apportent ainsi chacune leur propre part aux décisions. En d'autres termes, chaque discipline considère la cancérologie selon un point de vue particulier, qui coexiste et collabore avec plusieurs autres. La gestion des connaissances dans le cadre du projet KASIMIR doit prendre en compte et tirer parti de cette multiplicité de points de vue.

Enfin, les utilisateurs envisagés ici sont principalement des médecins, membres du réseau de santé Oncolor, géographiquement repartis dans la région Lorraine et de niveaux d'expertise variés. Il peut en effet s'agir de cancérologues experts dans des centres hautement spécialisés lors de RCP ou de médecins généralistes en consultation. La diffusion des connaissances décisionnelles dans le cadre de KASIMIR doit ainsi être adaptée à la diversité et à la répartition des utilisateurs. Il n'est d'ailleurs pas question de restreindre les possibilités de diffusion des connaissances à ces seuls scénarios, de nombreuses autres utilisations possibles pouvant être envisagées (information auprès des patients, enseignement, etc.).

Outre la détection des besoins, les différentes tâches à accomplir pour la mise en œuvre de la gestion des connaissances concernent, selon [Dieng *et al.*, 2000], l'acquisition, la représentation, la diffusion, l'utilisation, l'évaluation, la maintenance et l'évolution des connaissances. Ainsi, l'étude et le développement des techniques et des outils employés dans le cadre du projet KASIMIR pour certaines de ces tâches sont détaillés dans les chapitres suivants. Néanmoins, l'acquisition des connaissances contenues dans les référentiels est une part du projet KASIMIR entamée avant le début de cette thèse [Bresson et Lieber, 1999] et qui sort du cadre de ce mémoire. La tâche d'évaluation est par ailleurs réalisée par les partenaires du domaine médical du projet [Rios *et al.*, 2003] et ne sera pas non plus

traitée dans ces pages. Enfin, l'évolution des connaissances est un des objectifs à long terme du projet KASIMIR [Lieber *et al.*, 2002]. Cette tâche a été étudiée du point de vue des ergonomes impliqués dans le projet [Sauvagnac, 2000] et la mise en œuvre d'outils informatiques permettant de proposer des évolutions des référentiels fait partie des perspectives de cette thèse.

Chapitre 2

Le portail sémantique KASIMIR pour la cancérologie

Le Web actuel peut être vu comme un ensemble de ressources inter-connectées (textes, images, services, etc.) et conçues pour être manipulées et interprétées par des utilisateurs humains. Mais avec l'augmentation constante de la quantité d'informations disponibles sur le Web, les outils censés aider l'utilisateur à y accéder montrent leurs limites. Forcés de les manipuler à un niveau purement syntaxique (mots clés, co-occurrence de termes, etc.) ils ne peuvent fournir des résultats pertinents vis-à-vis du *contenu* de ces ressources. Les tâches fastidieuses de tri, de sélection et de combinaison d'informations, sont laissées à l'utilisateur, seul en mesure de les interpréter et donc de les comprendre. C'est ce constat qui est à l'origine du *Web sémantique*. Le Web sémantique selon Tim Berners-Lee désigne « une extension du Web dans laquelle les informations sont fournies avec une signification bien définie, permettant aux ordinateurs et aux personnes de mieux travailler en coopération » [Berners-Lee *et al.*, 2001]. Cela suppose que les ressources du Web soient associées à une sémantique formelle, c'est-à-dire standard et conçue pour les machines, de façon à ce que leur contenu devienne manipulable par des programmes informatiques, dotés de mécanismes de raisonnements automatiques.

Le projet KASIMIR vise à la gestion des connaissances utiles à la pratique de la cancérologie en Lorraine. Il s'agit en particulier de formaliser les connaissances décisionnelles contenues dans les référentiels et d'en fournir un accès intelligent au travers d'outils d'aide à la décision et de raisonnement à destination des médecins lorrains. Ainsi, les méthodes, les technologies et les outils proposés dans le cadre du Web sémantique apportent une infrastructure adaptée pour la mise en œuvre du projet KASIMIR. C'est l'objectif de ce chapitre que de montrer de quelle façon cette infrastructure est exploitée au sein d'un *portail sémantique* pour la cancérologie.

2.1 Le Web sémantique

Cela peut paraître paradoxal, mais il est assez difficile de donner une définition claire et consensuelle du Web sémantique. Nous nous attacherons donc ici à apporter une vision particulière de ce domaine et des technologies sur lesquelles il repose.

2.1.1 Tentative de définition et introduction aux principes du Web sémantique

En premier lieu, il convient de rappeler que le Web sémantique se définit comme une *extension* du Web [Berners-Lee *et al.*, 2001]. Le Web peut être considéré comme un réseau de documents multimédias (textes, images, sons, vidéos, etc.) reliés entre eux par des liens *hyper-*

textes [Berners-Lee et Fischetti, 2000]. Il repose principalement sur le protocole de communication HTTP (*HyperText Transfer Protocol*) permettant d'accéder à des documents identifiés et localisés grâce à leurs URL (*Unified Resource Locator*). HTML (*HyperText Markup Language*) est le langage de présentation généralement utilisé, permettant une mise en forme graphique des documents diffusés sur le Web. Le mot sémantique, quant à lui, relève de la signification associée aux expressions d'un langage. En informatique, il s'agit généralement des principes régissant l'interprétation des constructions permises par la syntaxe d'un langage particulier. La sémantique d'un langage de programmation est, par exemple, encodée au sein d'un interpréteur ou d'un compilateur, chargé de mettre en œuvre les instructions décrites. En intelligence artificielle, est généralement appelé sémantique d'un langage de représentation des connaissances l'ensemble des règles permettant à un moteur d'inférences d'interpréter les éléments décrits et ainsi, d'en inférer de nouvelles connaissances au travers de mécanismes de raisonnements [Kayser, 1997]. C'est à cette notion de sémantique que nous nous rattachons ici, dans le cadre du Web sémantique.

Pour comprendre de quelle façon et sous quelle forme le Web intègre la notion de sémantique prenons l'exemple d'une recette de gaspacho. Un gaspacho est une soupe froide, d'origine espagnole, généralement à base de tomates, et servie le plus souvent en entrée. Comme tout document décrivant une recette, une page Web dédiée à la recette du gaspacho (voir figure 2.1) contiendra une liste d'ingrédients, une suite d'étapes de préparation, une photographie, etc. Il est assez facile pour un utilisateur connaissant la langue française de comprendre qu'il s'agit d'une recette, que cette recette est celle du gaspacho et que la photographie représente le résultat attendue de la recette. Néanmoins, le langage HTML utilisé ne s'intéresse qu'à la mise en forme graphique du texte et à l'inclusion d'éléments multimédias. Le contenu de ce document n'est ainsi compris par un programme informatique que comme un ensemble de données, une séquence de mots.



FIG. 2.1 – Page web (HTML) décrivant une recette de gaspacho.

```

<recette>
  <titre>Gaspacho</titre>
  <tempsprep unite="minutes">5</tempsprep>
  <niveau>facile</niveau>
  <tempscuisson>0</tempscuisson>
  <cout>faible</cout>
  <nombre-personne>4</nombre-personne>
  <photo>http://www.loria.fr/~daquin/gaspacho/photo.jpg</photo>
  <listingredients>
    <ingredient quantite="1" unite="grosses boîtes">tomates</ingredient>
    <ingredient quantite="1/2">poivron vert</ingredient>
    <ingredient quantite="1/2">concombre</ingredient>
    <ingredient quantite="1">échalote grise</ingredient>
    <ingredient quantite="1">gousse d'ail entière</ingredient>
    <ingredient quantite="2" unite="cuillères à soupe">huile d'olive</ingredient>
    <ingredient quantite="1/2" unite="cuillères à café">vinaigre balsamique</ingredient>
    <ingredient>tabasco ou cayenne (au goût) </ingredient>
    <ingredient quantite="1/2" unite="cuillères à café">sucre</ingredient>
    <ingredient>sel et poivre</ingredient>
    <ingredient quantite="15" unite="cl">eau</ingredient>
  </listingredients>
  <preparation>
    <etape ordre="1">Mettre tous les ingrédients coupés grossièrement dans un mixer,
      démarrer l'appareil à petite vitesse et terminer à grande vitesse</etape>
    <etape ordre="2">Laisser refroidir au réfrigérateur</etape>
    <etape ordre="3">Ajouter des glaçons au moment de servir</etape>
  </preparation>
</recette>

```

FIG. 2.2 – Description XML d'une recette de gaspacho.

Une première étape dans la *description sémantique* de cette recette serait d'en délimiter et d'en identifier les différentes parties, c'est-à-dire d'en représenter la structure. Il s'agit pour cela d'inclure à la description du document des informations concernant le contenu même des informations ou, en d'autres termes, d'y adjoindre des *méta-données*. Le langage actuellement le plus utilisé pour structurer les informations dans cette optique est XML (*eXtensible Markup Language*), qui permet de définir des *balises* servant à marquer les éléments de contenu des documents. La figure 2.2 montre un exemple de représentation possible de notre recette de gaspacho dans ce langage. Sur la base de cette structure XML, il sera par exemple possible de rechercher parmi un ensemble de recettes celles qui contiennent une grande quantité de tomates, peu d'étapes de préparation ou un faible temps de cuisson, et de retrouver alors la recette du gaspacho. Les méta-données sont ainsi utilisées comme base pour l'accès aux informations, ainsi que pour leurs échanges, au sein de programmes informatiques.

Néanmoins, XML ne s'intéresse qu'à la structure des informations, à la syntaxe selon laquelle elles doivent être représentées. Il ne permet pas d'en exploiter la sémantique. En effet, chercher une recette dont la liste des ingrédients contient le mot « tomate » ne suppose pas d'appréhender la signification des mots « ingrédient » et « tomate ». Un programme informatique s'appuyant sur cette structure ne sera ainsi pas en mesure de reconnaître la recette du gaspacho comme une recette ne contenant aucune viande et ne nécessitant que peu de matériel. Il ne pourra pas non plus établir que le plat obtenu sera froid et peu calorique. Ce qui nous permet, à nous utilisateurs humains, de *déduire* ces informations à partir de celles décrites dans la recette provient de notre *connaissance* du monde, et en particulier, de la cuisine, des soupes et des tomates. Ainsi, dans le but de permettre aux programmes informatiques de comprendre et de manipuler les informations selon la sémantique qu'elles transportent, il est nécessaire qu'ils disposent de connaissances, définissant de façon formelle les concepts mis en œuvre et les relations entre ces concepts. C'est le rôle des *ontologies* au sein du Web sémantique.

Une ontologie, selon la définition la plus répandue en informatique, est « la spécification explicite

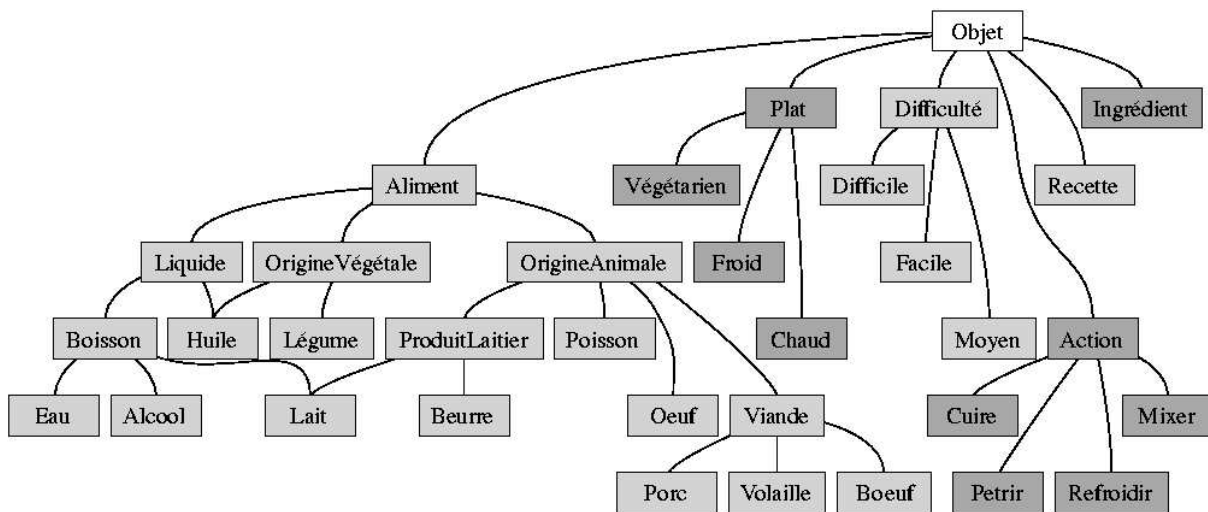


FIG. 2.3 – Hiérarchie de concepts d'une ontologie pour les recettes de cuisine.

d'une conceptualisation » [Gruber, 1993]. Plus concrètement, il est habituel de considérer une ontologie comme la définition formelle des concepts relatifs à un domaine et des relations entre ces concepts. Elle doit de plus être conçue comme un modèle réutilisable et partagé des connaissances afin de servir de base commune à l'ensemble du domaine pour l'interprétation et l'échange d'informations. La forme la plus courante selon laquelle les ontologies sont présentées consiste en une hiérarchie de concepts, organisés des plus généraux aux plus spécifiques. Une ontologie relative aux recettes de cuisine contiendra par exemple les concepts d'aliments et de plats. La figure 2.3 présente un extrait de la hiérarchie de concepts d'une telle ontologie. Chacun de ces concepts doit de plus être associé à une *définition formelle*, c'est-à-dire à une expression permettant de reconnaître un objet comme relatif au concept et d'en inférer de nouvelles informations. Sur l'exemple, le concept de recette pourra être défini comme un procédé permettant de réaliser un plat. Un plat froid pourra quant à lui être défini comme un plat dont la recette a un temps de cuisson nul. L'expression de ces définitions, c'est-à-dire la syntaxe utilisée et la sémantique associée, dépend du langage de représentation dans lequel l'ontologie est formalisée.

Les ontologies, en tant que définition formelle des concepts employés dans la représentation des informations, permettent de mettre en œuvre des mécanismes de raisonnements automatiques pour un accès intelligent au contenu de ces informations. Au travers de l'ontologie décrite plus haut, un programme informatique sera en mesure d'inférer par exemple que le gaspacho est un plat froid, le temps de cuisson de sa recette étant nul, ou encore que les ingrédients décrits ne sont pas des viandes. Les informations présentes au sein de la recette du gaspacho ne sont ainsi plus accessibles seulement par les mots employés, par la syntaxe, mais aussi selon la sémantique qu'elles transportent, telle qu'elle peut être perçue au travers de l'ontologie. En ce sens, le Web sémantique peut être défini comme une infrastructure dont les piliers sont les ontologies. Au travers des ontologies, on passe en effet d'un Web de données, dédié aux seuls utilisateurs humains, à un Web de connaissances, où les machines se comprennent dans le but de mieux servir ces utilisateurs.

2.1.2 Langages pour le Web sémantique

Les ontologies étant à la base de l'infrastructure du Web sémantique, la première tâche à accomplir dans sa mise en œuvre a été le développement d'un langage pour la représentation des ontologies. Ce langage devait être standard, dans un but d'interopérabilité, expressif, pour répondre aux besoins de do-

maines variés, et, bien sûr, associé à une sémantique formelle clairement définie. Le W3C (*World Wide Web Consortium*) est l'organisme chargé d'établir les standards concernant les technologies relatives au Web. Ainsi, un langage de représentation d'ontologies pour le Web sémantique ne pouvait être développé qu'en intégrant les technologies établies au sein du W3C et sous le contrôle du W3C. Le langage de représentation d'ontologies OWL (*Web Ontology Language*³) est ainsi décrit au sein de recommandations du W3C (en particulier, [(WOWG), 2004a], [(WOWG), 2004b], [(WOWG), 2004c] et [(WOWG), 2004d]) et s'inscrit comme une des couches d'une *pyramide des langages* (voir figure 2.4), reposant sur d'autres standard du W3C et ouvrant la voie à d'autres problématiques pour le Web sémantique.

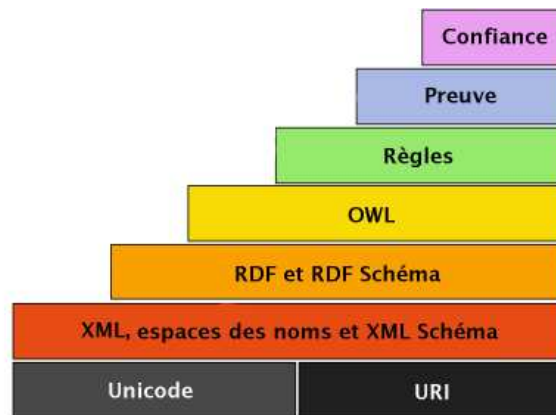


FIG. 2.4 – Le « gâteau » du Web sémantique (traduit et adapté de [Koivunen et Miller, 2002]).

À la base de cette pyramide se trouve UNICODE et les URI. UNICODE est une norme de codage des caractères [The Unicode consortium, 2003]. Elle est vouée à remplacer les précédentes, telles que la norme ASCII, qu'elle étend en particulier dans le sens de l'internationalisation. Les URI (*Unified Resource Identifier*, [(NWG), 2005]) sont utilisés comme identificateurs des ressources diffusées sur le Web et en particulier au sein du Web sémantique. Dans le langage OWL, ainsi qu'en RDF dont il est issu, chaque ressource est supposée posséder un URI, l'identifiant de façon unique et non ambigu. Un URI dans ce cadre est composé d'un *espace des noms* et d'un *fragment*. L'URI de la recette du gaspacho pourra par exemple être :

`http://www.cuisine.com/recettes#gaspacho`

où `http://www.cuisine.com/recettes` est l'espace des noms, et `gaspacho` le fragment.

XML (*eXtensible Markup Language* [(XML-CWG), 2004]) est à l'heure actuelle le standard incontournable pour l'échange d'informations sur le Web. Les DTD (*Document Type Definition*) et les schémas XML permettent de contraindre l'utilisation des balises et de leurs imbrications pour la représentation des informations. De cette façon, XML permet de déclarer, et donc de partager, une syntaxe propre à la représentation d'informations utiles à un domaine ou une application.

RDF (*Resource Description Framework* [(RDF-CWG), 2004c]) est décrit comme un langage pour la définition de méta-données sur les documents du Web. Il permet de déclarer des triplets (*objet*, *prédicat*, *sujet*) où les objets et les sujets sont des ressources (identifiées par des URI) et les prédicats des relations entre ces ressources. RDF possède une syntaxe XML [(RDF-CWG), 2004b].

³L'acronyme WOL ayant été jugé par le groupe chargé du développement de ce langage comme peu satisfaisant, il a été choisi de changer l'ordre des lettres afin d'obtenir OWL, qui signifie hibou en anglais. Le personnage du hibou du roman d'Alan Alexander Milne *Winnie l'ourson* [Milne, 1926] épelant son nom WOL au lieu de OWL, ce choix apparaissait alors comme parfaitement justifié [Hendler et W3C Communications Team, 2004].

RDF SCHEMA [(RDF-CWG), 2004a] est le langage de définition de vocabulaires pour RDF. Il permet de définir des hiérarchies de classes et de propriétés (relations entre classes), ainsi que certaines contraintes simples sur les propriétés. Les ressources d'une description RDF sont ainsi les instances des classes du schéma RDF SCHEMA utilisé pour la description. De plus, RDF SCHEMA possède certaines fonctionnalités intéressantes du point de vue de la représentation des connaissances. En particulier, il est réflexif : les ressources décrivant le méta-modèle de RDF SCHEMA, comme `rdfs:Class` ou `rdf:Property`, sont manipulables au sein d'expressions du langage. Les triplets, éléments de base des expressions en RDF, sont eux aussi considérés comme des ressources et peuvent ainsi être utilisés dans la constitution d'autres triplets. RDF(S)⁴ possède de plus une sémantique définie de façon standard [(RDF-CWG), 2004b], indiquant les inférences possibles sur ce langage et contribuant à en faire un premier formalisme de représentation des connaissances pour le Web sémantique [Staab *et al.*, 2001].

Par sa simplicité et la souplesse de son modèle, RDF(S) est conçu pour être étendu et servir de base à des langages plus expressifs. OWL est une de ces extensions, apportant au langage du Web qu'est RDF(S) des capacités provenant des langages de représentation des connaissances, notamment des logiques de descriptions.

2.1.3 OWL et les logiques de descriptions

De part leur nature, les recherches sur les ontologies intègrent plusieurs domaines de l'informatique tels que l'ingénierie des connaissances, la représentation des connaissances, la recherche d'informations ou les bases de données [Staab et Studer, 2004]. Nous nous intéressons plus spécifiquement ici au langage de représentation d'ontologies OWL dont les origines proviennent de plusieurs langages de représentation des connaissances [Horrocks *et al.*, 2003]. En tant que langage de représentation d'ontologies pour le Web sémantique, OWL doit être doté de certaines caractéristiques, en particulier :

- Il doit être associé à une sémantique standard et formellement définie, permettant la mise en œuvre de mécanismes de raisonnements bien maîtrisés.
- Il doit être très expressif, pour prendre en compte la variété des utilisateurs, des domaines et des applications présents sur le Web et le Web sémantique.

Pour ces raisons, une des influences les plus importantes ayant conduit à OWL provient des logiques de descriptions.

OWL comme une logique de descriptions

Les logiques de descriptions (LD, voir [Baader *et al.*, 2003]) sont des langages de représentation des connaissances s'appuyant à la fois sur une représentation structurée, comme dans les langages à objets ou de *frames* [Ducournau *et al.*, 1998], et sur une sémantique formellement et précisément définie, à la manière des logiques.

Une base de connaissances en LD contient la description des concepts, des rôles et des individus du domaine d'application. En OWL, on dira plutôt qu'une *ontologie* contient la description de *classes*, de *propriétés* et d'*individus*, respectivement. Pour simplifier, on adoptera dans la suite le vocabulaire associé à OWL. Les *individus* sont utilisés pour représenter les objets du domaine. Les *propriétés* correspondent à des relations binaires entre ces objets. Les *classes* représentent des ensembles d'objets, possédant des caractéristiques communes. Par exemple, au sein de l'ontologie pour les recettes, la classe Légume, représentant l'ensemble des légumes, peut être associée à des propriétés telles que la couleur, la taille ou la teneur en calories. Un individu concombre, représentant un concombre, sera une *instance* de la classe Légume, c'est-à-dire un membre de l'ensemble des légumes représentés par Légume, avec par exemple la propriété d'être de couleur verte. Les classes d'une ontologie sont organisées en une hiérarchie selon

⁴Il est habituel de noter RDF(S) les technologies RDF et RDF SCHEMA utilisées conjointement.

la relation de *subsumption*, qui est un ordre partiel. La subsumption correspond à un lien de spécialisation entre classes. On pourra ainsi exprimer le fait que la classe des cucurbitacées est plus spécifique que la classe des légumes, c'est-à-dire que Cucurbitacée est subsumée par Légume. Cela signifie que toute instance de Cucurbitacée, comme concombre par exemple, est nécessairement une instance de Légume.

La sémantique associée à une base de connaissances en LD s'exprime généralement sous une forme inspirée de la théorie des modèles (on parle en anglais de *model theoretic semantics*). Une *interprétation* dans ce cadre est un couple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. $\Delta^{\mathcal{I}}$ est un ensemble non vide appelé le *domaine d'interprétation* et contient l'ensemble des objets du domaine. $\cdot^{\mathcal{I}}$ est appelée *fonction d'interprétation* et associe à une classe \mathcal{C} un ensemble d'objet du domaine, c'est-à-dire un sous-ensemble de $\Delta^{\mathcal{I}}$ ($\mathcal{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$), à une propriété p une relation binaire, sous-ensemble de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ ($p^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$) et à un individu a un objet du domaine, élément de $\Delta^{\mathcal{I}}$ ($a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$).

On peut distinguer différents types de classes en LD et en OWL. Les deux classes \top et \perp sont pré-définies dans le langage et représentent respectivement l'ensemble des objets du domaine ($\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$) et l'ensemble vide ($\perp^{\mathcal{I}} = \emptyset$). Une classe peut être décrite uniquement par son nom, on parlera dans ce cas de classe atomique, ou construite sur la base de *constructeurs*, pour une classe définie. Par exemple, la *conjonction* de deux classes \mathcal{C} et \mathcal{D} est un constructeur de classe qui s'écrit $\mathcal{C} \sqcap \mathcal{D}$ et s'interprète comme l'intersection des ensembles représentés par ces deux classes : $(\mathcal{C} \sqcap \mathcal{D})^{\mathcal{I}} = \mathcal{C}^{\mathcal{I}} \cap \mathcal{D}^{\mathcal{I}}$. Avec \mathcal{C} une classe et p une propriété, la classe construite sur la base du *quantificateur existentiel*, $\exists p.\mathcal{C}$, correspond à l'ensemble des objets liés par la propriété p à au moins un objet appartenant à la classe \mathcal{C} : $(\exists p.\mathcal{C})^{\mathcal{I}} = \{x \mid \exists y, (x, y) \in p^{\mathcal{I}} \wedge y \in \mathcal{C}^{\mathcal{I}}\}$. Beaucoup d'autres constructeurs ($\sqcup, \forall, \neg, \leq n$, etc.) permettent la définition de classes en OWL (voir tableau 2.1) et il est par ailleurs aussi possible d'associer certaines caractéristiques aux propriétés (la transitivité, la propriété inverse, le domaine (*domain*) et le co-domaine (*range*), etc.). Cela contribue à faire de OWL un langage très expressif, c'est-à-dire permettant de définir des éléments de connaissances ayant des structures complexes, comme par exemple la classe des « recettes d'au plus 3 étapes, dont aucune n'est une étape de cuisson ou de pétrissage » :

$$\text{Recette} \sqcap \leq 3 \text{ étape} \sqcap \forall \text{étape} . \neg (\text{Cuire} \sqcup \text{Pétrir})$$

Le tableau 2.1 contient la liste des constructeurs de classes utilisables en OWL et la sémantique associée à chacun d'eux.

Une *ontologie* en OWL est un ensemble d'axiomes et d'assertions introduisant respectivement des classes et des individus. L'inclusion et l'équivalence de classes sont les deux types d'*axiomes* classiquement utilisés en LD. L'axiome d'inclusion entre deux classes \mathcal{C} et \mathcal{D} se note $\mathcal{C} \sqsubseteq \mathcal{D}$ et indique que l'ensemble représenté par la classe \mathcal{C} doit nécessairement être inclus dans l'ensemble représenté par la classe \mathcal{D} . Plus formellement, on dit qu'une interprétation \mathcal{I} *satisfait* l'axiome $\mathcal{C} \sqsubseteq \mathcal{D}$ si $\mathcal{C}^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}}$. Intuitivement, il s'agit de déclarer qu'une classe est plus spécifique qu'une autre ou, inversement, que cette dernière est plus générale que la première, comme par exemple dans : $\text{Beurre} \sqsubseteq \text{ProduitLaitier}$. L'axiome d'équivalence $\mathcal{C} \equiv \mathcal{D}$ permet d'indiquer que \mathcal{C} et \mathcal{D} représentent les mêmes ensembles : \mathcal{I} *satisfait* $\mathcal{C} \equiv \mathcal{D}$ si $\mathcal{C}^{\mathcal{I}} = \mathcal{D}^{\mathcal{I}}$. OWL ajoute à l'inclusion et l'équivalence d'autres axiomes comme celui d'incompatibilité (*disjoint classes*) permettant d'exprimer qu'aucun objet d'une classe \mathcal{C} ne peut être membre d'une autre classe \mathcal{D} . Ce dernier peut s'écrire, sur la base de l'axiome d'équivalence des LD, $\mathcal{C} \sqcap \mathcal{D} \equiv \perp$. Par exemple, au sein de l'ontologie des recettes, les trois axiomes :

$$[\text{Ax1}] \text{PlatVégétarien} \equiv \text{Plat} \sqcap \forall \text{recette} . (\forall \text{ingrédient} . (\neg \exists \text{aliment} . \text{Viande}))$$

$$[\text{Ax2}] \text{Recette} \sqcap \ni \text{tempsCuisson}.0 \sqcap \leq 3 \text{ étape} \sqcap \neg \exists \text{étape} . \text{Pétrir} \sqsubseteq \exists \text{niveau} . \text{Facile}$$

$$[\text{Ax3}] \text{Viande} \sqcap (\text{Légume} \sqcup \text{EpiceAromate}) \equiv \perp$$

nom	syntaxe LD	syntaxe abstraite	sémantique
Nom de classe	\mathcal{C}	C (URI)	$\mathcal{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Top, Thing	\top	owl:Thing	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Bottom, Nothing	\perp	owl:Nothing	$\perp^{\mathcal{I}} = \emptyset$
Intersection	$\mathcal{C} \sqcap \mathcal{D}$	intersectionOf($C D$)	$(\mathcal{C} \sqcap \mathcal{D})^{\mathcal{I}} = \mathcal{C}^{\mathcal{I}} \cap \mathcal{D}^{\mathcal{I}}$
Union	$\mathcal{C} \sqcup \mathcal{D}$	unionOf($C D$)	$(\mathcal{C} \sqcup \mathcal{D})^{\mathcal{I}} = \mathcal{C}^{\mathcal{I}} \cup \mathcal{D}^{\mathcal{I}}$
Négation	$\neg \mathcal{C}$	complementOf(C)	$(\neg \mathcal{C})^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \mathcal{C}^{\mathcal{I}}$
Énumération	$\{a, b, \dots\}$	oneOf($a b \dots$)	$\{a, b, \dots\}^{\mathcal{I}} = \{a^{\mathcal{I}}, b^{\mathcal{I}}, \dots\}$
Quantificateur existentiel	$\exists p.C$	restriction(p someValuesFrom(C))	$(\exists p.C)^{\mathcal{I}} = \{x \mid \exists y, (x, y) \in p^{\mathcal{I}} \wedge y \in \mathcal{C}^{\mathcal{I}}\}$
Quantificateur universel	$\forall p.C$	restriction(p allValuesFrom(C))	$(\forall p.C)^{\mathcal{I}} = \{x \mid \forall y, (x, y) \in p^{\mathcal{I}} \rightarrow y \in \mathcal{C}^{\mathcal{I}}\}$
Restriction à une valeur	$\exists p.a$	restriction(p hasValue(a))	$(\exists p.a)^{\mathcal{I}} = \{x \mid (x, a^{\mathcal{I}}) \in p^{\mathcal{I}}\}$
Restriction non qualifiée de cardinalité	$= n p$	restriction(p cardinality(C))	$(= n p)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in p^{\mathcal{I}}\} = n\}$
	$\geq n p$	restriction(p minCardinality(C))	$(\geq n p)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in p^{\mathcal{I}}\} \geq n\}$
	$\leq n p$	restriction(p maxCardinality(C))	$(\leq n p)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in p^{\mathcal{I}}\} \leq n\}$
Quantificateur existentiel	$\exists d.T$	restriction(d someValuesFrom(T))	$(\exists d.T)^{\mathcal{I}} = \{x \mid \exists y, (x, y) \in d^{\mathcal{I}} \wedge y \in T^{\mathcal{D}}\}$
Quantificateur universel	$\forall d.T$	restriction(d allValuesFrom(T))	$(\forall d.T)^{\mathcal{I}} = \{x \mid \forall y, (x, y) \in d^{\mathcal{I}} \rightarrow y \in T^{\mathcal{D}}\}$
Restriction à une valeur	$\exists d.a$	restriction(d hasValue(a))	$(\exists d.a)^{\mathcal{I}} = \{x \mid (x, a^{\mathcal{D}}) \in d^{\mathcal{I}}\}$
Restriction non qualifiée de cardinalité	$= n d$	restriction(d cardinality(T))	$(= n d)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in d^{\mathcal{I}}\} = n\}$
	$\geq n d$	restriction(d minCardinality(T))	$(\geq n d)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in d^{\mathcal{I}}\} \geq n\}$
	$\leq n d$	restriction(d maxCardinality(T))	$(\leq n d)^{\mathcal{I}} = \{x \mid \{y \mid (x, y) \in d^{\mathcal{I}}\} \leq n\}$

TAB. 2.1 – Constructeurs de classes en OWL, où \mathcal{C} et \mathcal{D} sont des classes, T est un type de donnée, v est un littéral (valeur d'un type de données), n est un nombre, a et b sont des individus, p une propriété d'objets (*ObjectProperty*) et d une propriété de données (*DatatypeProperty*).

indiquent respectivement qu'un plat végétarien se définit comme un plat sans viande ([Ax1]), que les recettes de plats froids de moins de 3 étapes dont aucune de pétrissage sont des recettes faciles ([Ax2]) et qu'une viande ne peut être un légume ni un élément de la classe des épices et aromates ([Ax3]).

Il existe généralement deux types d'*assertions* en LD. Une assertion de la forme $\mathcal{C}(a)$, où \mathcal{C} est une classe et a un individu, indique que a est une instance de \mathcal{C} , c'est-à-dire qu'il représente un élément de la classe \mathcal{C} . Formellement, l'interprétation \mathcal{I} satisfait $\mathcal{C}(a)$ si $a^{\mathcal{I}} \in \mathcal{C}^{\mathcal{I}}$. Une assertion de la forme $p(a, b)$, permet d'indiquer que l'individu a est en relation avec l'individu b par la propriété p , c'est-à-dire que \mathcal{I} satisfait $p(a, b)$ si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in p^{\mathcal{I}}$. Les assertions :

Recette(rgaspacho) Plat(gaspacho) réalise(rgaspacho, gaspacho) tempsCuisson(rgaspacho, 0)

indiquent par exemple que `rgaspacho` est un individu représentant une recette, que `gaspacho` est un plat, que `rgaspacho` est la recette permettant de réaliser `gaspacho` et que cette recette a un temps de cuisson de 0.

Finalement, on dira qu'une interprétation \mathcal{I} est un *modèle* d'une ontologie O , si elle satisfait tous les axiomes et toutes les assertions de O .

OWL comme langage du Web

Les LD, comme langages de représentation des connaissances structurés, très expressifs et disposant d'une sémantique formellement définie, semblent bien adaptées pour la représentation d'ontologies. Mais en plus d'être une LD, OWL doit aussi être un langage du Web sémantique, et donc du Web. Intégrer OWL à l'environnement du Web implique des contraintes supplémentaires telles que :

- être reconnu comme standard et s'intégrer aux standards existants sur le Web,
- être simple et utilisable par une importante variété d'utilisateurs, rarement experts en LD,
- être fiable, robuste et rapide.

OWL est une *recommandation* du W3C, l'organisme chargé de mettre en œuvre les standard du Web, depuis février 2004. Il a depuis été adopté par de nombreux utilisateurs comme le principal langage de représentation d'ontologies, ainsi qu'en attestent les nombreux outils développés pour OWL et les nombreux projets de développement ou de traduction d'ontologies en OWL. Il est ainsi reconnu comme standard, aussi bien d'un point de vue institutionnel que d'un point de vue pratique. De plus, OWL s'intègre aux standards existants du Web, selon le schéma décrit en figure 2.4. Il peut en effet être vu comme une extension de RDF(S) dans le sens où le langage est lui même représenté sous la forme d'un schéma RDF et qu'il est supposé être compatible avec celui-ci (compatibilité descendante ou *backward compatibility* : toute ontologie en OWL est une description RDF valide). Ainsi, OWL bénéficie de la sérialisation de RDF(S) dans une syntaxe XML et les ressources manipulées en OWL (classes, propriétés, individus) sont nommées et identifiées par des URI. Dans cette syntaxe, dite XML/RDF, la classe des plats végétariens, telle que précédemment définie, sera par exemple représentée par :

```
<owl:Class rdf:ID="PlatVegetarien">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Plat" />
    <owl:Restriction><owl:onProperty rdf:resource="#recette" />
      <owl:allValuesFrom>
        <owl:Restriction><owl:onProperty rdf:resource="#ingredient" />
          <owl:allValuesFrom><owl:complementOf>
            <owl:Restriction><owl:onProperty rdf:resource="#aliment" />
              <owl:someValuesFrom rdf:resource="#Viande" /></owl:Restriction>
            </owl:complementOf></owl:allValuesFrom></owl:Restriction>
          </owl:allValuesFrom>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
```

Mais l'intégration de OWL dans les langages du Web tels que XML et RDF(S) contredit certaines des contraintes indiquées plus haut. Il semble en effet évident d'après la description de `PlatVégétarien` que la présentation de OWL dans la syntaxe XML/RDF n'a rien de simple et ne pourra être manipulée par les utilisateurs habituels du Web. Les langages de frames [Lassila et McGuinness, 2001] sont en revanche reconnus pour posséder une syntaxe facile à lire et à comprendre. Pour cette raison, une syntaxe, appelée *syntaxe abstraite* [(WOWG), 2004d] et inspirée des langages de frames a été définie pour OWL [Horrocks *et al.*, 2003]. Dans cette syntaxe, l'exemple de la classe `PlatVégétarien` devient :

```
Class(PlatVegetarien complete restriction(recette allValuesFrom(  
  restriction(ingredient allValuesFrom(complementOf(  
    restriction(aliment someValuesFrom(Viande)))))))
```

La correspondance entre les constructeurs, axiomes et assertions des LD et la syntaxe abstraite de OWL est donnée dans le tableau 2.1, page 18.

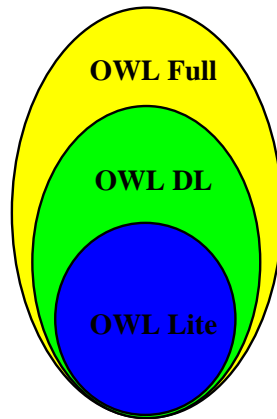


FIG. 2.5 – Les trois variantes de OWL.

La compatibilité avec RDF(S) et l'importante expressivité de OWL posent par ailleurs d'autres problèmes. Tout d'abord, certaines constructions de RDF(S) impliquent que les mécanismes de raisonnements associés au langage soient indécidables. L'aspect réflexif de RDF(S), qui permet de redéfinir les éléments du langage, empêche de plus la mise en œuvre d'une sémantique standard et fixée de manière définitive [Horrocks *et al.*, 2003]. En outre, même si ces constructions particulières à RDF(S) ne sont pas utilisées, la LD sous-jacente à OWL (une $SHOIN(\mathcal{D})$, voir [Napoli, 1997] et [Baader *et al.*, 2003]) reste très expressive et les inférences associées sont de fait très complexes. Il est ainsi peu envisageable que la majorité des utilisateurs exploitent l'intégralité des constructeurs proposés par OWL. Pour ces raisons, le langage OWL est dérivé en trois sous-langages — OWL FULL, OWL DL et OWL LITE — qui sont conçus comme trois couches successives, du plus simple (OWL LITE) au plus complexe (OWL FULL) (voir figure 2.5).

- **OWL FULL** correspond au langage OWL dans son ensemble, c'est-à-dire qu'il inclut tous les constructeurs de OWL et toutes les possibilités de RDF(S). L'inférence en OWL FULL constitue ainsi un problème indécidable.
- **OWL DL** est un sous-langage de OWL FULL correspondant à la LD $SHOIN(\mathcal{D})$. OWL DL ne permet pas l'utilisation de certaines constructions particulières de RDF(S) (réflexivité, etc.).
- **OWL LITE** est un sous-langage de OWL DL, correspondant lui aussi à une LD ($SHIF(\mathcal{D})$), mais moins expressive. OWL LITE ne reprend de OWL DL que les constructeurs jugés les plus utiles et les plus faciles à mettre en œuvre.

OWL et les types de données

Un langage de représentation des connaissances expressif ne peut se contenter de manipuler des éléments de connaissances abstraits, c'est-à-dire des classes et des individus, mais doit aussi inclure la possibilité d'intégrer des données concrètes telles que des nombres ou des chaînes de caractères. En LD, les approches fondées sur la représentation de domaines concrets [Baader et Hanschke, 1991, Lutz, 2003] et la représentation de types de données [Horrocks et Sattler, 2001, Pan et Horrocks, 2003]

sont toutes deux conçues dans ce sens et sont très similaires en terme d'expressivité. Un *domaine concret* D est un couple (Δ_D, Φ_D) où Δ_D est un ensemble de valeurs et Φ_D est un ensemble de noms de prédicats sur Δ_D . On ne s'intéressera par la suite qu'aux noms de prédicats unaires. L'interprétation d'un nom de prédicat unaire φ de Φ_D , noté φ^D , est un sous-ensemble de Δ_D .

OWL permet de gérer des données concrètes selon l'approche à base de types de données et réutilise pour cela les types de données de XML SCHEMA [(XML-SWG), 2004]. XML SCHEMA contient en effet un ensemble de types de données prédéfinis — `xsd:integer`, `xsd:float`, `xsd:string`, `xsd:date`, etc. — ainsi que des mécanismes permettant la définition de types de données dérivés de ces types prédéfinis (*derived simple types*). Un type de données représente un ensemble de valeurs, adressées par des littéraux. Par exemple, `"40"^^xsd:integer` est un littéral, représentant l'entier 40. On dira qu'un type de données T possède un espace de valeurs $V(T)$, un espace lexical $L(T)$ et une fonction $L2V(T)$ qui associe à un littéral de $L(T)$ la valeur de $V(T)$ qui lui correspond [(SWBPDWG), 2005]. Par exemple, le littéral `"40"^^xsd:integer` appartient à l'espace lexical $L(\text{xsd:integer})$ de `xsd:integer`. L'espace de valeurs de `xsd:integer` correspond aux entiers (\mathbb{Z}) et ainsi, $L2V(\text{xsd:integer})(\text{"40"})$ est égal à l'entier 40.

OWL intègre les types de données et les littéraux dans la construction de classes, mais seuls les types de données prédéfinis sont utilisables de façon standard [(WOWG), 2004c]. Par ce biais, il est par exemple possible de spécifier que le temps de cuisson d'une recette est un entier positif (`Recette \sqsubseteq \forall tempsCuisson.xsd:PositiveInteger`) ou qu'un plat dont le temps de cuisson est nul est nécessairement froid (`\forall recette. \exists tempsCuisson. "0"^^xsd:integer \sqsubseteq PlatFroid`).

Pour l'interprétation des types de données, [Pan et Horrocks, 2003] et [Horrocks et Sattler, 2001] introduisent un *domaine concret universel* $\mathcal{D} = (\Delta_{\mathcal{D}}, \Phi^1)$, où $\Delta_{\mathcal{D}}$ est le domaine de tous les types de données et où Φ^1 est l'ensemble des noms de type de données, correspondant ainsi à un ensemble de prédicats unaires sur $\Delta_{\mathcal{D}}$. L'espace de valeurs $V(T)$ d'un type de données T correspond ainsi à l'interprétation de T dans ce domaine concret universel : $T^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}$.

En OWL DL, le domaine d'interprétation « abstrait » $\Delta^{\mathcal{I}}$ est supposé disjoint du domaine $\Delta_{\mathcal{D}}$ des types de données. En particulier, deux types distincts de propriétés existent : les propriétés liant des individus à d'autres individus (les *propriétés d'objets*, appelées *ObjectProperty* en OWL) et celles liant des individus à des littéraux (les *propriétés de données* ou *DatatypeProperty*). Une propriété particulière ne peut être à la fois une propriété d'objets et une propriété de données. Grâce à cette distinction entre les deux domaines, abstrait et concret, l'implémentation d'un moteur d'inférences sur OWL DL est simplifiée. Un tel système pourra en effet contenir deux modules distincts et découplés, l'un étant chargé du raisonnement sur les classes, les propriétés et les individus, et l'autre du raisonnement sur les types de données et les littéraux [Pan et Horrocks, 2003].

Raisonnements avec OWL

Un avantage important que tire OWL de sa filiation avec les LD est la possibilité de profiter des inférences et des mécanismes de raisonnements associés à ces formalismes. Parmi ceux-ci, il convient en premier lieu de citer le *test de subsomption*. Étant donné deux classes C et D d'une ontologie O , on dit que C subsume D , c'est-à-dire que C est plus générale que D , si pour toute interprétation \mathcal{I} modèle de O , $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. Dans ce cas, on dira aussi que D est subsumée par C ou que D est plus spécifique que C . Ce test permettra par exemple de déduire que la classe des *plats végétaliens*, définie par

$$[\text{Ax4}] \text{PlatVégétalien} \equiv \text{Plat} \sqcap \forall \text{recette}. (\forall \text{ingrédient}. (\neg \exists \text{aliment}. \text{OrigineAnimale}))$$

est subsumée par *PlatVégétarien*, la classe des plats végétariens (voir axiome [Ax1]). La *classification* est un mécanisme de raisonnement construit sur la base du test de subsomption. Il s'agit, pour une classe C de retrouver toutes les classes de l'ontologie qui la subsume.

Le test d'instanciation est un autre type d'inférences hérité des LD. Son objectif est de tester si un individu, appelé a , peut être reconnu comme instance d'une classe C dans une ontologie O , c'est-à-dire, en termes formels, si pour toute interprétation \mathcal{I} modèle de O , $a^{\mathcal{I}} \in C^{\mathcal{I}}$. D'après les définitions vues plus haut, il sera par exemple possible de reconnaître que le gaspacho est un plat froid (gaspacho est une instance de `PlatFroid`) ou que la recette du gaspacho est facile (`rGaspacho` est une instance de `∃niveau.Facile`). Le test d'instanciation est à la base du mécanisme d'instanciation, aussi appelé *classification d'instances*, qui consiste à retrouver pour un individu a toutes les classes de l'ontologie O dont a est instance. Il est par ailleurs possible, sur la base de l'instanciation, d'inférer de nouvelles informations sur un individu [Brachman *et al.*, 1991]. En effet, l'instanciation de `rGaspacho` l'ayant reconnue comme une instance de la classe des recettes faciles, `rGaspacho` doit nécessairement être reliée par la propriété `niveau` à un objet de `Facileℐ` : $\exists x, (\text{rGaspacho}^{\mathcal{I}}, x) \in \text{niveau}^{\mathcal{I}} \wedge x \in \text{Facile}^{\mathcal{I}}$. Les informations concernant l'individu `rGaspacho` sont ainsi automatiquement complétées selon son instanciation au sein de l'ontologie. Ce mécanisme de raisonnement, appelé par la suite *complétion d'instance*, sera à la base de plusieurs applications dans le cadre de KASIMIR.

Enfin, les tests de *satisfiabilité* d'une classe et de *consistance* d'une ontologie permettent une certaine forme de validation des connaissances représentées. Une classe C est dite *satisfiable* s'il existe un modèle \mathcal{I} de O dans lequel $C^{\mathcal{I}} \neq \emptyset$. C'est-à-dire qu'aucun objet ne pourra être reconnu comme instance d'une classe insatisfiable. Au sein d'une ontologie des recettes, la classe des plats végétariens contenant de la viande (`PlatVégétarien` \sqcap `∃recette.(∃ingrédient.(∃aliment.Viande))`) pourra par exemple être détectée comme insatisfiable. On dit qu'une ontologie O est *consistante* s'il existe une interprétation qui soit un modèle pour O . Par exemple, une ontologie contenant une instance de plat végétarien avec de la viande sera inconsistante, c'est-à-dire que les assertions :

`PlatVégétarien(a) recette(a,r) ingrédient(r,i) aliment(i,v) Viande(v)`

amèneront à la conclusion que a est un élément de l'ensemble vide : $a^{\mathcal{I}} \in \perp^{\mathcal{I}}$. Dans la mesure où une ontologie inconsistante ne permet pas de raisonnement valide, ce test est particulièrement utile durant la phase de construction d'une ontologie. Par ailleurs, il peut être démontré que les inférences habituellement utilisées en LD, en particulier le test de subsumption, peuvent être implantées par un test de satisfiabilité de classe. En effet, pour tester que C subsume D il suffit de démontrer que la classe $D \sqcap \neg C$ est insatisfiable.

2.1.4 Outils pour le Web sémantique

Le langage à la base du Web sémantique étant défini, cette section s'intéresse aux outils utiles à la mise en œuvre d'ontologies en OWL, c'est-à-dire à leur utilisation et à leur construction.

Implémentations de OWL

Les principaux moteurs d'inférences utilisés pour la mise en œuvre des mécanismes de raisonnements décrits plus haut pour OWL sont ceux issus des LD. Ces systèmes reposent dans la plupart des cas sur la méthode des tableaux sémantiques pour tester la satisfiabilité d'une classe [Horrocks et Patel-Schneider, 1998], test sur lequel reposent les autres types d'inférences en LD. Parmi les nombreux systèmes existants, les deux plus utilisés pour OWL sont FACT [Horrocks, 1998] et RACER [Haarslev et Möller, 2001]. Ils sont tous deux développés en Lisp et permettent de raisonner sur des LD de type $\mathcal{SHOIQ}(\mathcal{D})$. Le système Pellet [Parsia et Sirin, 2004] est quant à lui spécifiquement conçu pour le raisonnement sur OWL DL et est développé en Java. Notons que les systèmes de LD sont dotés d'un langage de communication commun, sous la forme de l'interface DIG (*DL Implementation Group*, [Bechhofer, 2003]).

Il existe principalement deux librairies pour manipuler des ontologies en OWL dans le langage de programmation Java, JENA et OWL API. OWL API [Bechhofer *et al.*, 2003] est une librairie Java développée dans le cadre du projet WONDERWEB pour la manipulation et le raisonnement sur des ontologies en OWL DL. Les capacités de raisonnement associées à cette librairie sont fournies par un moteur d'inférences en LD, intégré au travers de l'interface DIG. JENA [McBride, 2002] est une librairie, ou « boîte à outils », permettant l'analyse de fichiers OWL, ainsi que la manipulation, l'interrogation et la gestion de la persistance d'ontologies. JENA s'appuie pour cela sur un modèle centré sur les triplets RDF, au dessus duquel est implanté le modèle de représentation des connaissances de OWL. JENA inclut de plus des fonctionnalités de raisonnement, au travers d'un moteur d'inférences propre fondé sur un ensemble de règles d'inférences. Ces règles sont par ailleurs paramétrables, permettant ainsi de choisir quelle part de la sémantique de OWL on souhaite mettre en œuvre, voire de l'étendre pour des ontologies spécifiques à un domaine ou une application. Un moteur d'inférences en LD peut aussi être utilisé au sein de JENA, grâce à l'interface DIG, et PELLET s'intègre naturellement avec JENA, en Java. De plus, JENA reposant sur le modèle de RDF, il est possible de manipuler des ontologies en OWL FULL, exploitant ainsi toutes les possibilités expressives de OWL et RDF.

Construction d'ontologies

Acquérir des connaissances et en construire une représentation formelle sont des tâches complexes, nécessitant généralement l'intervention d'experts du domaine, ainsi que l'aide de méthodes et d'outils facilitant la démarche de conceptualisation et d'explicitation des connaissances que cela suppose. Cette problématique est ainsi souvent considérée comme primordiale dans les domaines de l'ingénierie et de la gestion des connaissances. De nombreuses méthodes d'acquisition et de construction de connaissances ont été proposées dans ce cadre, parmi lesquelles COMMONKADS [Schreiber *et al.*, 1999] est la plus fréquemment citée. D'autres, comme ON-TO-KNOWLEDGE [Sure *et al.*, 2004] ou METHONTOLOGY [López *et al.*, 1999], ont été plus récemment mises en œuvre afin de répondre à la problématique du développement d'ontologies dans le cadre du Web sémantique. METHONTOLOGY est par exemple plus orientée sur la réutilisation d'ontologies existantes.

Les *environnements de développement d'ontologies* contiennent généralement un ensemble d'outils, centrés autour d'un éditeur de connaissances et supposés aider à la construction d'une ontologie ainsi qu'à sa formalisation dans un langage tel que OWL. PROTÉGÉ [Noy *et al.*, 2001] est l'éditeur de connaissances jouissant de la plus grande notoriété à l'heure actuelle. Cette notoriété provient, au moins en partie, de son architecture extensible, permettant d'intégrer des greffons (*plugins*) afin d'étendre l'éditeur pour de nouvelles fonctionnalités et la prise en charge de nouveaux langages. Un greffon dédié à l'édition d'ontologies en OWL [Knublauch *et al.*, 2004] a par ce biais été développé. Une particularité de celui-ci est qu'il permet, au travers de l'interface DIG, l'utilisation d'un moteur d'inférences pour les LD à des fins de validation des éléments de connaissances édités. Ce type de fonctionnalité est aussi présent au sein de l'éditeur de connaissances OILED [Bechhofer *et al.*, 2001], originellement développé autour du moteur d'inférences FACT. ONTOEDIT [Sure *et al.*, 2002] est un éditeur de connaissances fondé sur un processus de développement d'ontologies suivant les différentes étapes de la méthode de construction ON-TO-KNOWLEDGE. WEBODE [Arpérez *et al.*, 2003] est quant à lui un environnement de développement d'ontologies fondé sur une architecture client-serveur et offrant des outils couvrant l'ensemble du cycle de vie d'une ontologie, selon la méthode METHONTOLOGY.

Même en disposant de méthodes et d'environnements de développement d'ontologies efficaces, la mise en œuvre d'une ontologie reste dans beaucoup de cas une tâche d'ingénierie difficile à réaliser manuellement. Cela est d'autant plus vrai que les éléments à modéliser sont souvent présents sous forme de documents ou de données, qu'il est nécessaire d'analyser dans le but d'en extraire des connaissances. L'objectif des outils d'*apprentissage d'ontologies* [Maedche et Staab, 2004] est de servir de

support à cette tâche par l'automatisation de certaines parties. Ils reposent généralement sur des techniques telles que la fouille de textes (voir par exemple [Cherfi, 2004]) ou l'extraction de connaissances de bases de données (voir par exemple [Simon, 2000]). Par exemple, KAON TEXT-TO-ONTO [Maedche et Staab, 2000] est un outil d'apprentissage d'ontologies qui s'appuie sur un ensemble d'algorithmes pour l'analyse de textes et la fouille (extraction de règles d'association, etc.), afin de construire une ontologie à partir d'un corpus de textes relatifs au domaine.

Alignement, fusion et versions d'ontologies

Le Web est par essence un environnement peu structuré et hétérogène. Du fait de cette nature particulière du Web, de multiples ontologies, recouvrant plus ou moins les mêmes domaines, coexistent au sein du Web sémantique. L'*alignement d'ontologies* ayant pour objectif de permettre une utilisation conjointe de plusieurs ontologies est ainsi une problématique importante dans ce cadre. Il s'agit généralement de construire, de façon plus ou moins automatique, des *appariements* entre les éléments décrits dans différentes ontologies. Différentes techniques sont utilisées pour cela, s'appuyant sur l'analyse des étiquettes (*labels*) associés aux éléments des ontologies, sur leur structure ou sur la sémantique associée. [Kalfoglou et Schorlemmer, 2003] et [Euzenat (coordinator), 2004] présentent chacun une vue d'ensemble des techniques et des outils d'alignement existants. Un exemple parmi ces outils est PROMPT [Noy et Musen, 2003], développé dans le cadre de l'éditeur d'ontologies PROTÉGÉ et qui repose principalement sur l'analyse de la structure des ontologies pour détecter des appariement.

Deux problématiques ayant un fort lien avec l'alignement d'ontologies sont la fusion d'ontologies et la détection des modifications au sein de différentes versions d'une même ontologie. En effet, la fusion d'ontologies consiste à construire une ontologie à partir de plusieurs autres. Il est souvent nécessaire pour cela d'apparier en premier lieu les éléments contenus dans les ontologies, afin de fusionner les éléments mis en correspondance. Un exemple d'outils de fusion d'ontologies est FCA-MERGE [Stumme et Maedche, 2001], qui s'appuie sur des techniques issues de l'analyse de concepts formels (*formal concept analysis* en anglais) afin d'extraire des correspondances entre deux ontologies et ainsi de les fusionner. La détection de changements dans les versions d'une même ontologie peut être vue comme un processus d'alignement entre ontologies, les éléments mis en correspondance étant alors considérés comme inchangés. Selon ce principe, l'outil PROMPTDIFF [Noy *et al.*, 2004] est par exemple développé sur la base de PROMPT.

2.1.5 Services Web et Web sémantique

Les services Web sont des programmes informatiques, disposant d'une interface Web et ainsi accessibles par d'autres programmes au travers du Web. L'idée d'intégrer les principes du Web sémantique aux services Web s'est très vite imposée dans le domaine. En effet, un service Web, au même titre qu'un document, est une ressource. Décrire les services que l'on diffuse selon une sémantique précise, en relation avec une ontologie du domaine, doit ainsi permettre de mettre en œuvre des mécanismes de raisonnement rendant possible la découverte, l'invocation et la composition automatique de ces services. Ce nouvel axe de recherche, intégrant services Web et Web sémantique est celui des *services Web sémantiques* (en anglais, *Semantic Web services*).

Deux technologies standard (ou en passe de l'être), toutes deux fondées sur XML, sont utilisées pour la mise en œuvre des services Web : SOAP et WSDL. SOAP [(XPWG), 2003], pour *Simple Objects Access Protocol*, est un protocole permettant la communication entre un service Web et ses clients. SOAP permet notamment à un client de faire appel à des procédures distantes (*Remote procedure call*). WSDL [(WQSWG), 2005], pour *Web Service Description Language*, est utilisé pour la description des services. Une description en WSDL contient les informations essentielles permettant à un client de faire

appel au service décrit, comme par exemple, son adresse Web, les méthodes qu'il publie et les paramètres de ces méthodes.

WSDL permet une représentation très simple d'un service afin de permettre son invocation. Dans le cadre des services Web sémantiques, plusieurs langages et technologies sont actuellement en développement pour la description sémantique de services, afin d'en permettre la découverte et la composition automatique. Parmi ceux-ci, OWL-S et WSML semblent les plus prometteurs. OWL-S [Martin *et al.*, 2004] se présente comme une ontologie en OWL pour la description de services Web. En OWL-S, un service est principalement décrit par *ce qu'il fait* (*service profile*), par *comment il le fait* (*process model*) et par les informations nécessaires à la mise en œuvre de ce processus (*service grounding*, généralement exprimé en WSDL). OWL-S étant fondé sur OWL, il permet d'intégrer naturellement les éléments de l'ontologie du domaine à la description du service. WSML, pour *Web Service Modeling Language*, est un langage permettant la description de services Web sémantiques selon différents aspects tels que leurs buts, les ontologies utilisées et les médiateurs entre services. La syntaxe de WSML est inspirée des langages de *frames*, mais peut être stockée et échangée sous une forme XML et RDF. Une transformation en OWL est aussi possible.

2.1.6 Le Web sémantique en médecine

Quelle que soit la discipline concernée, la pratique de la médecine fait appel à de grandes quantités de connaissances que l'on peut qualifier de « critiques » dans le sens où elles doivent être disponibles rapidement, dans leur version la plus actuelle, en des lieux dispersés et pour des personnes de niveaux d'expertises variés. Le Web sémantique apporte un environnement approprié à la diffusion et au partage intelligent de connaissances. Il existe ainsi un nombre important de projets visant à la construction d'ontologies dans le domaine médical (voir par exemple [Rector et Nowlan, 1993] ou [Gibaud *et al.*, 2004]), ainsi qu'à la traduction des nombreux thésaurus, terminologies et classifications existants dans le langage OWL (voir par exemple [Hahn et Schulz, 2004] et [Golbeck *et al.*, 2003]). Au-delà de la construction d'ontologies, certains projets tendent à exploiter le potentiel du Web sémantique dans des applications médicales, touchant au dossier patient informatisé [Bringay *et al.*, 2005] ou à la gestion de connaissances au sein d'un réseau de soin [Dieng-Kuntz *et al.*, 2004], par exemple.

2.2 Un portail sémantique pour la cancérologie

Un *portail sémantique* est un portail dédié à la diffusion de connaissances et de services utiles à une communauté particulière selon les principes et les technologies du Web sémantique. L'idée générale est de fournir un accès intelligent à des ressources en s'appuyant sur les ontologies du domaine et sur les mécanismes de raisonnement qu'elles permettent. Dans le cadre du projet KASIMIR, il s'agit de construire une architecture permettant aux médecins lorrains de disposer des connaissances contenues dans les référentiels et de leur apporter par ce biais un support pour la prise de décision. Cette section a pour but de présenter une vue d'ensemble de cette architecture (voir figure 2.6), dont les détails sont fournis dans les sections suivantes.

2.2.1 Les portails sémantiques

Sur le Web actuel, un portail est un site regroupant et agrégeant en un point d'accès unique des documents, des informations et des services utiles à une communauté particulière et provenant le plus souvent de sources multiples et hétérogènes [Stollberg *et al.*, 2004]. Ils se sont ainsi imposés comme un moyen efficace de diffusion de l'information et ont de cette façon contribué au succès du Web. Sur le Web sémantique, les informations sont désormais interprétables par les machines et reposent sur des

connaissances formelles vouées à être diffusées et échangées. L'accès à ces connaissances s'appuie sur des processus de raisonnements, offrant la possibilité d'une diffusion intelligente au sein de ce qui est appelé un portail sémantique. Nous nous intéressons ici au développement d'un environnement générique pour la mise en œuvre de portails sémantiques qui soit en particulier adapté et appliqué au projet KASIMIR. Par générique on entend que, même si leurs développements sont guidés par les besoins de l'application en cancérologie, les outils employés doivent en être indépendants et sont conçus pour être applicables dans d'autres domaines, médicaux ou non.

SEAL (pour *SEmantic portAL* [Maedche *et al.*, 2002]) se présente comme une approche générique pour le développement de portails sémantiques. Les ontologies sont, comme il se doit, à la base de l'architecture de SEAL. Les connaissances diffusées au travers d'un portail SEAL sont accessibles via ONTOBROKER [Decker *et al.*, 1999], qui est le principal mécanisme d'inférence au sein de cette architecture. Sur la base d'ONTOBROKER, différents modules sont développés, offrant des services, tels que le traitement de requêtes et la navigation, pour différents type d'utilisateurs, incluant les agents logiciels. Il faut remarquer que les ontologies au sein d'un portail SEAL sont décrites selon un modèle spécifique, qui sort des standards du W3C. Un module est par ailleurs disponible pour la communication avec des agents logiciels manipulant le langage RDF.

Dans [Stollberg *et al.*, 2004], les auteurs tentent de construire un cadre général pour la description de portails sémantiques et se concentrent en particulier sur l'exploitation des technologies du Web sémantique pour cela. Ce cadre est ensuite utilisé pour évaluer et faire un état des lieux des portails sémantiques existants, tels que ceux des projets ESPERONTO (www.esperonto.net) et ONTOWEB (www.ontoweb.org).

2.2.2 Le portail sémantique KASIMIR : vue d'ensemble

Le portail sémantique KASIMIR ayant pour objectif de fournir un accès intelligent aux connaissances contenues dans les référentiels, la clef de voûte de son architecture correspond à une représentation formelle de ces connaissances en OWL (partie haute de la figure 2.6). Deux niveaux se distinguent dans cette représentation. En effet, selon [Guarino, 1998], il existe différents type d'ontologies, parmi lesquels les *ontologies de domaines*, décrivant les concepts relatifs au domaine en général, et les *ontologies d'applications*, décrivant les concepts relatifs à une tâche particulière dans le domaine. Une ontologie du domaine de la cancérologie permet de définir les concepts utiles à la pratique de la cancérologie, tels que ceux de patient, de traitement ou de tumeur, ainsi que les relations entre ces concepts. Elle constitue un vocabulaire élémentaire pour la représentation des référentiels. La représentation d'un référentiel correspond, selon cette distinction, à une ontologie d'application, contenant les connaissances nécessaires pour l'aide à la décision dans une situation médicale particulière (traitement du cancer du sein, surveillance du cancer de la prostate, etc.). Chacune de ces ontologies d'applications intègre les connaissances contenues dans un référentiel particulier, c'est-à-dire qu'elle constitue une représentation formelle des diagrammes (les « arbres de décision ») et des définitions textuelles contenues dans ce référentiel (voir chapitre 1). Le choix du langage OWL pour la représentation des connaissances dans ce cadre s'impose naturellement. En effet, OWL est en premier lieu le langage standard pour la représentation d'ontologies du Web sémantique. Il constitue de plus un langage de représentation des connaissances très expressif permettant la mise en œuvre des mécanismes de raisonnements qui sont à la base de l'aide à la décision dans le cadre de l'application des référentiels.

Du point de vue logiciel, la partie centrale de l'architecture présentée figure 2.6 est constituée d'un *serveur de connaissances*, implémenté sous la forme de services Web. Ces services s'appuient sur un moteur d'inférences pour LD, permettant de mettre en œuvre les mécanismes de raisonnements associés à OWL et, par ce biais, d'interroger et de manipuler les connaissances diffusées. L'utilisation de services Web permet de disposer de protocoles standard (SOAP) pour la communication entre le portail et les

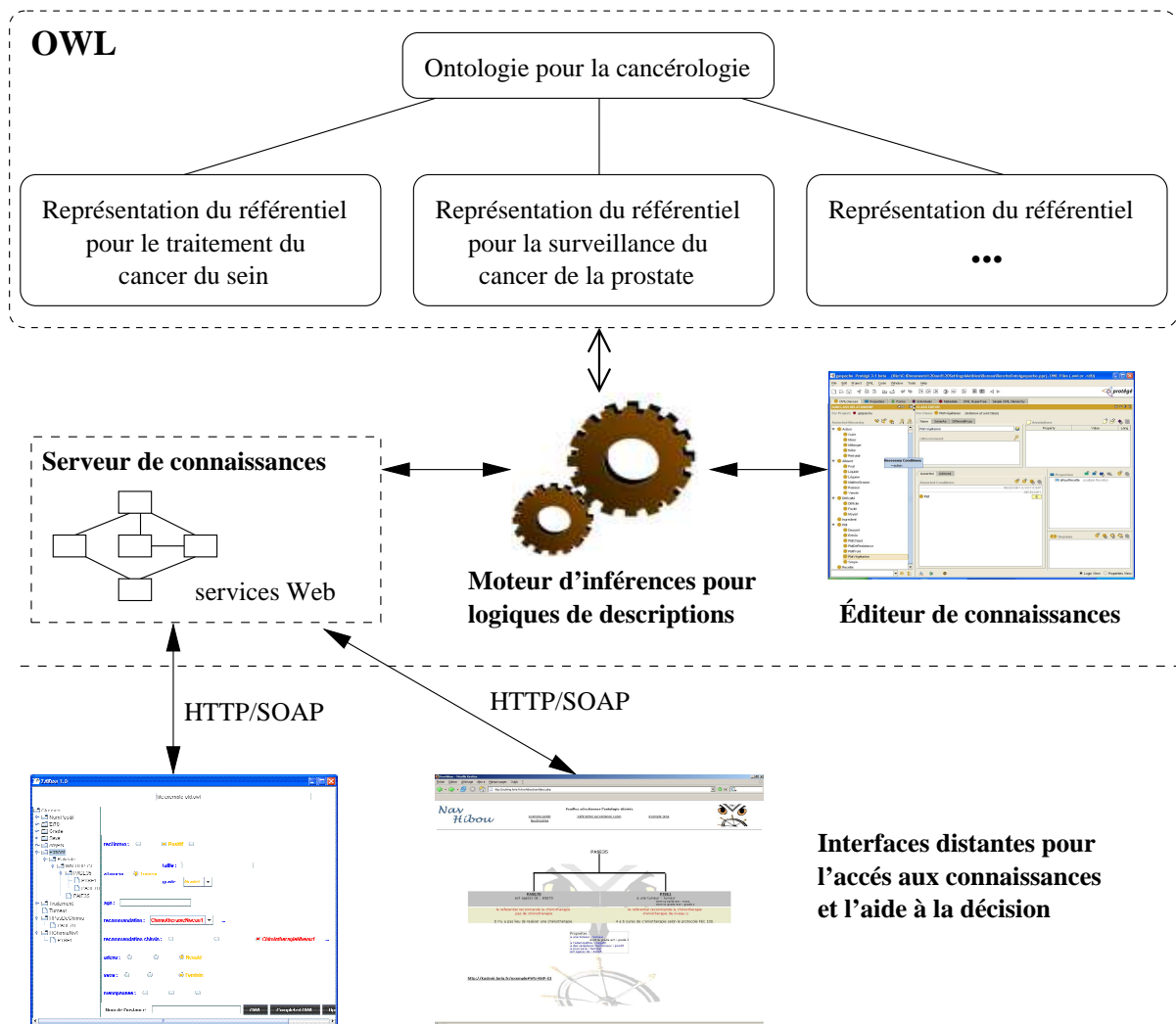


FIG. 2.6 – Vue d'ensemble de l'architecture du portail sémantique KASIMIR.

clients distants, éventuellement développés sur des plates-formes différentes, dans des langages différents. Ceux-ci peuvent de plus être peu gourmands en ressources, les mécanismes de raisonnements et de manipulation des ontologies étant implémentés au niveau du serveur. Centraliser de cette façon l'accès au moteur d'inférences permet par ailleurs de mettre en œuvre des mécanismes de *cache*, réutilisant les inférences réalisées par un utilisateur pour d'autres.

Deux types de clients ont à l'heure actuelle été développés pour permettre l'accès aux connaissances et l'aide à la décision sur la base des ontologies et des services disponibles sur le portail KASIMIR. Il s'agit pour le premier (ÉDHIBOU) d'une interface permettant la saisie d'une instance d'une classe définie dans une ontologie en OWL. Cette interface permet en effet de construire un individu en indiquant sa classe d'appartenance et de saisir les valeurs des propriétés pertinentes pour cet individu. Les mécanismes d'inférences fournis par le serveur de connaissances permettent de compléter cette description, spécifiant par exemple la classe d'appartenance de l'instance et déduisant d'autres valeurs de propriétés d'après les connaissances définies dans l'ontologie utilisée. Le deuxième (NAVHIBOU) est une interface de navigation au sein d'ontologies en OWL, indiquant à chaque étape les éléments inférés au niveau

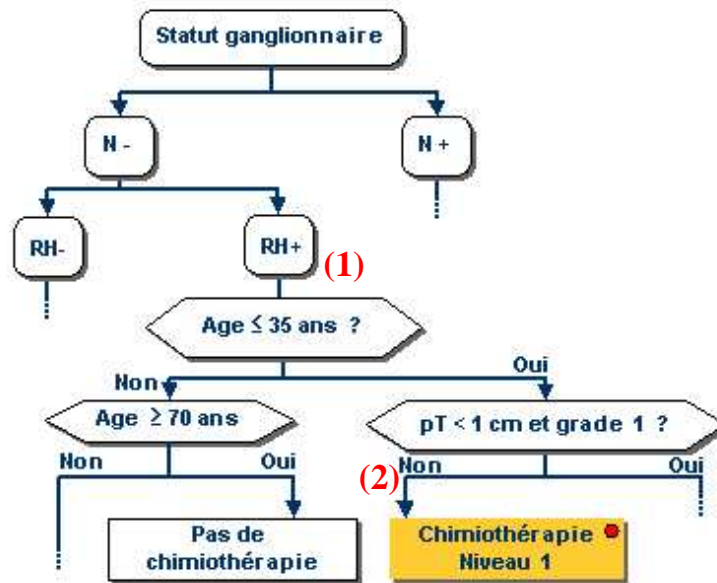


FIG. 2.7 – Extrait du référentiel pour le traitement du cancer du sein, repris de la figure 1.1.

courant de la navigation.

Un exemple de scénario d'utilisation de ce portail est celui d'un médecin en consultation souhaitant connaître le traitement recommandé pour un patient atteint de cancer du sein. Utilisant ÉDHIBOU, il s'agira ici de construire une instance de la classe Patient de l'ontologie OWL correspondant au référentiel pour le traitement du cancer du sein. ÉDHIBOU permet alors la saisie des propriétés correspondant aux caractéristiques pertinentes dans ce cadre, selon ce qui est indiqué dans l'ontologie. À partir des connaissances contenues dans la représentation du référentiel et des caractéristiques indiquées pour le patient, les mécanismes de raisonnement du portail permettent alors de déduire le traitement recommandé et il ne restera plus à ÉDHIBOU qu'à l'afficher comme une propriété inférée de l'instance considérée.

2.3 Représentation des référentiels au sein du Web sémantique

Le portail sémantique KASIMIR est conçu pour permettre la gestion des connaissances décisionnelles en cancérologie. Il est ainsi centré autour d'une représentation explicite et formelle de ces connaissances. L'idée est ici de formaliser le contenu des référentiels dans le langage OWL et de profiter ainsi de sa double nature de langage de représentation des connaissances et de langage du Web sémantique pour permettre un accès intelligent et distribué à ce contenu.

2.3.1 Principes de la représentation des référentiels en OWL

Les référentiels, tels que décrits au chapitre 1, contiennent les connaissances décisionnelles standard et partagées concernant la prise en charge de patients atteints de cancer. Ils se présentent principalement

sous la forme de diagrammes, similaires à des arbres de décision, permettant d'associer aux caractéristiques d'un patient les recommandations à appliquer (voir l'exemple de la figure 2.7). Un référentiel inclut par ailleurs la définition d'une part du vocabulaire employé dans ces diagrammes.

Des diagrammes aux classes de OWL

Les diagrammes contenus dans les référentiels constituent une représentation que l'on pourrait qualifier de semi-formelle. En effet, ils se rapprochent des *arbres de décision* étudiés par exemple en apprentissage [Quinlan, 1986] dans le sens où une condition sur le patient est testée à chaque étape et aboutit à une recommandation de décision. Néanmoins, il n'existe pas nécessairement de branches pour toutes les valeurs possibles des caractéristiques testées (en d'autres termes, les arbres ne sont pas exhaustifs). De plus, afin d'en faciliter la lecture, certaines conditions sont testées en dehors du cadre normal des nœuds du diagramme, comme des notes associées aux recommandations ou ajoutées aux arcs par exemple. Enfin, plus important encore, le vocabulaire employé pour désigner les caractéristiques des patients et les recommandations proposées n'est pas formalisé. Certains termes disposent au mieux d'une définition textuelle succincte. Ainsi, ce qui est appelé par les médecins « les arbres de décision » des référentiels laisse une part importante à l'ambiguïté et à l'interprétation de l'utilisateur. Pour cette raison, générer de façon automatique la représentation des référentiels dans le langage OWL ne peut être envisagé pour le moment. Nous nous intéressons donc ici aux principes qui sous-tendent cette représentation, réalisées manuellement et en concertation avec les médecins, en nous appuyant sur la structure des diagrammes contenus dans les référentiels.

Chaque branche d'un diagramme correspond à un ensemble de conditions sur les patients. L'ensemble des patients pour lesquels un chemin particulier est suivi constitue donc une classe de patients ayant des caractéristiques communes. Un tel chemin peut alors être représenté par une classe OWL, définie par des restrictions sur les caractéristiques testées dans les nœuds du chemin. Par exemple, la classe NM_RHP des patients dont les ganglions ne sont pas infectés et dont les récepteurs hormonaux sont positifs, décrite par :

$$\text{NM_RHP} \equiv \text{Patient} \sqcap \exists \text{ganglion.NonInfecté} \sqcap \exists \text{recHormo.Positif}$$

correspond au parcours de la racine au nœud marqué (1) du diagramme de la figure 2.7. De cette façon, toute instance de Patient dont les propriétés respectent les conditions indiquées pour aboutir au nœud (1) sera reconnu comme une instance de la classe NM_RHP. Cette représentation permet par ailleurs de s'appuyer sur la subsomption pour conserver une part de la structure présente dans les diagrammes. En effet, le chemin menant à l'arc marqué (2) (non) sur la figure 2.7 pourra être représenté par une classe nommée NM_RHP_AIE35_TS1oG2oG3⁵ et définie par :

$$\begin{aligned} \text{NM_RHP_AIE35_TS1oG2oG3} \equiv \text{NM_RHP} \sqcap \exists \text{âge.inf}35 \sqcap \\ \exists \text{tumeur} . (\exists \text{taille.supe}1 \sqcup \exists \text{grade} . (\text{Grade}2 \sqcup \text{Grade}3)) \end{aligned}$$

Selon cette définition, NM_RHP_AIE35_TS1oG2oG3 est subsumée par NM_RHP, tout comme le chemin menant au nœud (1) est inclus dans celui menant à l'arc (2). On peut remarquer que plusieurs représentations différentes auraient été possibles pour la classe NM_RHP_AIE35_TS1oG2oG3 et que celle-ci s'appuie sur le fait qu'il n'y a que trois valeurs possibles pour le grade d'une tumeur.

Les nœuds correspondant aux recommandations doivent être interprétés comme les décisions à prendre pour les patients respectant les conditions amenant à ces nœuds. En d'autres termes, un diagramme extrait d'un référentiel peut être vu comme un ensemble structuré de règles de la forme $CP \rightarrow RD$, où

⁵La convention de nommage des classes de patients pour la représentation des référentiels au sein du projet KASIMIR consiste en une suite d'abréviations correspondant aux caractéristiques des membres de la classe. NM, par exemple, signifie « pas d'envahissement ganglionnaire » (N-).

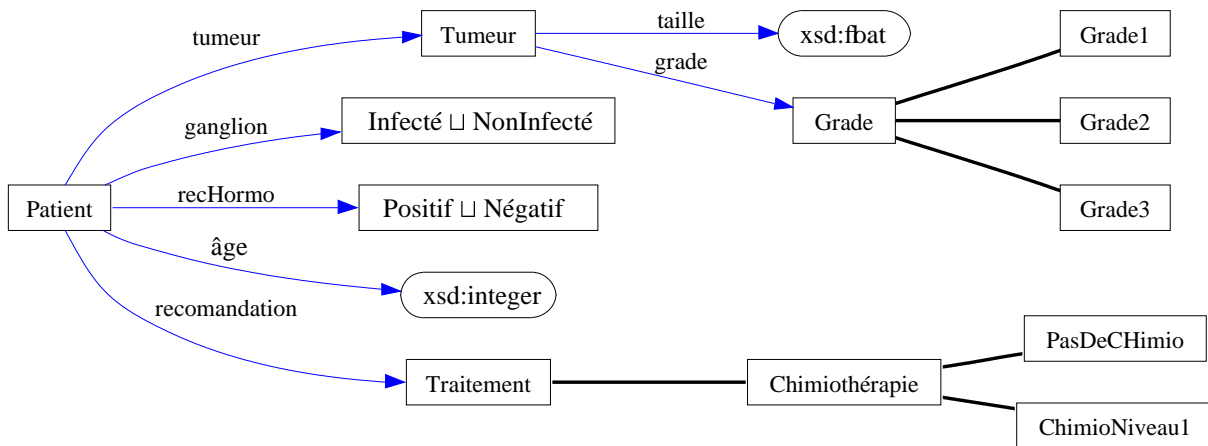


FIG. 2.8 – Extrait de l’ontologie de domaine utile à la représentation du diagramme de la figure 1.1.

CP correspond à un ensemble de conditions sur le patient, un chemin dans le diagramme, et RD correspond à une recommandation de décision. Comme vu ci-dessus, CP est représenté par une classe de patients. Les recommandations RD sont elles aussi représentées par des classes. Pour le traitement du cancer du sein par exemple, les recommandations correspondent aux traitements à appliquer et sont représentées par des sous-classes de la classe *Traitement*. Finalement, une propriété particulière, appelée *recommandation*, est utilisée pour lier les patients aux recommandations correspondantes. Une règle $CP \rightarrow RD$ sera ainsi représentée par un axiome de la forme :

$$P \sqsubseteq \exists \text{recommandation}.D$$

où P est une classe de patients traduisant les conditions de CP et D est une classe traduisant la recommandation décrite dans RD . De cette façon toute instance de *Patient* reconnue comme instance de la classe P sera automatiquement considérée comme instance de la classe $\exists \text{recommandation}.D$ et ainsi associée par la propriété *recommandation* à un objet de la classe D . Par exemple, l’axiome :

$$NM_RHP_AIE35_TS1oG2oG3 \sqsubseteq \exists \text{recommandation}.ChimioNiveau1$$

traduit le lien qui existe dans le diagramme de la figure 2.7 entre le chemin aboutissant à (2) et la recommandation de chimiothérapie de niveau 1. De cette façon, un individu *pat* introduit par les assertions :

```

Patient(pat)   NonInfecté(i)   Positif(p)       Tumeur(t)
Grade1(g)     ganglion(pat, i)  recHormo(pat, p)  âge(pat, 27)
tumeur(pat, t)  taille(t, 1.2)   grade(t, g)
    
```

sera reconnu comme une instance de la classe $NM_RHP_AIE35_TS1oG2oG3$ et, selon l’axiome précédant, se verra associé à un objet de la classe *ChimioNiveau1* par la propriété *recommandation*. En termes plus formels, on pourra conclure que $\exists x \in \text{ChimioNiveau1}^{\mathcal{I}}, (\text{pat}^{\mathcal{I}}, x) \in \text{recommandation}^{\mathcal{I}}$.

Nécessité de la construction d’une ontologie du domaine

La représentation des diagrammes extraits des référentiels, telle qu’elle est vue ci-dessus, nécessite de disposer d’un ensemble de classes et de propriétés élémentaires, définies formellement, et servant de briques de base à la représentation. Ces classes et ces propriétés relèvent de notions partagées entre plusieurs référentiels, telles que celles de patient, de tumeur, de traitement, et sont ainsi définies au sein d’une ontologie de domaine. Un extrait de l’ontologie de domaine, décrivant les classes et les propriétés

utiles à la représentation du diagramme de la figure 2.7 est présenté à la figure 2.8. La réalisation de cette étape passe soit par la réutilisation d'une ontologie existante, soit par la construction d'une nouvelle. Il existe de nombreuses ontologies et terminologies, plus ou moins standard et plus ou moins formelles, en médecine. Néanmoins, pour une question de simplicité, nous avons choisi la deuxième solution. En effet, même si la construction d'une ontologie suppose un lourd travail d'acquisition et de « désambiguïation » auprès des experts, il est plus facile de représenter le contenu des référentiels en s'appuyant sur les concepts et la terminologie qu'ils emploient. L'utilisation de cette représentation sera aussi de cette façon plus aisée pour les médecins coutumiers des référentiels. De plus, pour permettre la mise en œuvre d'inférences sur la représentation des référentiels, les classes et propriétés doivent être définies de façon précise et complète. On a vu par exemple que la représentation de la classe `NM_RHP_AIE35_TS1oG2oG3` repose sur le fait que si une tumeur n'est pas de grade 1, elle doit être soit de grade 2, soit de grade 3. L'ontologie doit donc inclure ce type de connaissances et cela est rarement le cas dans les terminologies standard en médecine.

Le problème des données concrètes et des types de données

Selon les spécifications actuelles [(WOWG), 2004c], la manipulation de données concrètes en OWL, telles que des nombres ou des chaînes de caractères, ne peut se faire qu'au travers des types de données *prédéfinis* de XML SCHÉMA [(XML-SWG), 2004]. Un moteur d'inférences est par ailleurs supposé prendre en compte au moins les types chaîne de caractères (`xsd:string`) et entier (`xsd:integer`). Mais ces types prédéfinis ne peuvent suffire dans le cadre de la représentation des référentiels. En effet, afin de représenter par exemple la classe ASE70 des patients de plus de 70 ans, il est nécessaire de disposer d'un type de données `supe70` représentant les entiers supérieurs ou égaux à 70 :

$$\text{ASE70} \equiv \text{Patient} \sqcap \exists \text{âge.supe70}$$

Pour cela, nous nous appuyons sur ce qui est appelé en XML SCHÉMA les *types dérivés* (*XML Schema derived simple types*). Les constructeurs de types dérivés permettent en effet de construire de nouveaux types de données en utilisant notamment des restrictions sur des types prédéfinis. Par exemple, `supe70`, le type des entiers supérieurs à 70 pourra s'écrire en XML SCHÉMA :

```
<xsd:simpleType name="supe70">
  <xsd:restriction base="integer">
    <xsd:minInclusive value="70"/>
  </xsd:restriction>
</xsd:simpleType>
```

Mais, contrairement à ce qu'on pourrait croire, inclure ces types de données dérivés dans les spécifications de OWL n'est pas une tâche facile. En effet, OWL étant fondé sur RDF(S), tout élément d'une description en OWL est supposé être une ressource identifiable par un URI. Or les types de données de XML SCHÉMA n'en possèdent pas. Néanmoins, disposer de types de données définis par l'utilisateur étant une fonctionnalité essentielle dont OWL ne saurait se passer, un groupe de travail du W3C cherche actuellement à recenser les solutions potentielles à ce problème [(SWBPDWG), 2005]. Par ailleurs, le travail de Jeff Z. Pan et de Ian Horrocks concernant l'extension appelée OWL-EU [Pan et Horrocks, 2005] de OWL pour la représentation et le raisonnement sur des *groupes de types de données* (*datatype groups*) doit être souligné comme une solution prometteuse au problème de la représentation expressive de type de données en OWL.

Pour les besoins du projet KASIMIR et en attendant que ces travaux aboutissent à une extension des spécifications actuelles de OWL, nous avons modifié le moteur d'inférences pour OWL DL PELLET

afin qu'il soit en mesure de raisonner sur des types de données dérivés de types numériques prédéfinis, du genre de `supe70`. En OWL DL, le domaine abstrait, c'est-à-dire celui des objets ($\Delta^{\mathcal{I}}$), est distinct du domaine des valeurs concrètes ($\Delta^{\mathcal{D}}$). Cela signifie qu'un littéral ne pourra être considéré comme un objet, un type de données comme une classe et une *DatatypeProperty* ne pourra être aussi une *ObjectProperty*. Comme indiqué dans [Pan et Horrocks, 2003], cela a l'avantage de faciliter l'implémentation de moteurs d'inférences dans le sens où ceux-ci peuvent ainsi être composés de deux modules découplés, un chargé du raisonnement sur les classes et l'autre sur les type de données. Le module de raisonnement sur les classes doit faire appel au module de raisonnement sur les types de données pour les calculs de subsomption et d'instanciation. En effet, afin de pouvoir déduire par exemple que la classe des patients âgés de plus de 35 ans (`Patient \sqsupseteq \exists âge.supe35`) subsume la classe des patients de plus de 70 ans, le module de raisonnement sur les type de données devra d'abord calculer que `supe70` est un sous-type de `supe35`. De la même façon, pour reconnaître qu'un individu décrivant un patient âgé de 78 ans est une instance de `ASE70`, il faudra pour cela calculer que le littéral 78 (ou `"78"^^xsd:integer`) relève du type de données `supe70`.

2.3.2 Outils pour l'édition et la maintenance des connaissances

Cette section est un résumé de l'article [d'Aquin *et al.*, 2005a] qui présente un ensemble d'outils développés dans le cadre du projet KASIMIR pour l'édition et la maintenance de connaissances, sur la base de l'éditeur PROTÉGÉ (voir section 2.1.4). Ces outils étaient originellement dédiés à un langage propre au projet KASIMIR. Ils sont présentés ici tels qu'ils ont été adaptés au langage OWL et utilisés conjointement au greffon de PROTÉGÉ pour la construction d'ontologies en OWL.

Éditeur de connaissances

Le choix de l'éditeur PROTÉGÉ pour l'édition et la maintenance de connaissances dans le cadre du portail sémantique KASIMIR relève de plusieurs raisons. Tout d'abord, celui-ci est fondé sur une architecture à base de greffons qui facilite l'intégration d'outils propres au projet et adaptés au domaine d'application. C'est d'ailleurs au travers d'un greffon que PROTÉGÉ a été étendu pour l'édition de connaissances dans le langage OWL. De plus, lorsque l'on utilise ce greffon, la construction d'ontologies en OWL peut être assistée par l'utilisation d'un moteur d'inférences pour LD, couplé à l'éditeur grâce à l'interface DIG (voir section 2.1.4). Cela permet une forme de validation des connaissances au cours de l'édition, notamment par l'utilisation des tests de consistance de l'ontologie et de satisfiabilité des classes qu'elle contient. Il est de plus possible grâce à cela de détecter certaines « erreurs » d'édition. On pourra par exemple vérifier que l'ontologie ne contient pas de redondance, deux classes inférées comme équivalentes étant considérées comme deux définitions redondantes d'un même ensemble. De la même façon, il sera possible de vérifier que la hiérarchie de classes *inférée* par le moteur sur la base de la subsomption, coïncide avec celle *déclarée* sous PROTÉGÉ. Ces deux hiérarchies ne sont pas supposées coïncider en théorie, mais, en pratique, leurs différences sont souvent le signe d'erreurs, la hiérarchie inférée n'étant pas celle attendue par la concepteur de l'ontologie.

Visualisation des connaissances

La visualisation des connaissances est une part importante d'un environnement de développement d'ontologies car elle permet à l'utilisateur d'évaluer et de valider visuellement le résultat du processus d'édition. Pour être efficace, un module de visualisation doit en particulier faciliter la *navigation* au sein des connaissances éditées, en apporter une *vue globale*, pour en évaluer les principales caractéristiques, et refléter *précisément* le contenu de ces connaissances, sous une forme facile à appréhender

par la plupart des utilisateurs. PROTÉGÉ affiche la hiérarchie des classes d'une ontologie sous la forme d'un arbre, proche de ceux utilisés pour représenter l'arborescence des fichiers dans la plupart des systèmes d'exploitation. Si cette forme de présentation a l'avantage d'être familière pour la majorité des utilisateurs, elle apparaît comme insuffisante au regard des besoins en visualisation mentionnés plus haut. En particulier, lorsqu'une classe de la hiérarchie possède plusieurs super-classes, le nœud de l'affichage lui correspondant est dupliqué sous chacun de ses nœuds parents, afin de conserver la structure d'arbre de la présentation. De plus, lorsque l'ontologie contient plus de quelques dizaines de classes, ce qui est souvent le cas, l'ensemble de la hiérarchie ne peut être affiché en même temps à l'écran. Enfin, ce type de présentation est connu pour ne pas être particulièrement efficace concernant la navigation [Pirolli *et al.*, 2001]. Pour ces raisons, deux modules de visualisation de connaissances ont été développés et intégrés sous forme de greffons à PROTÉGÉ afin que leur utilisation combinée réponde à l'ensemble des besoins. Le premier, baptisé PALÉTUVIER (voir figure 2.9(a)), permet la visualisation des classes sous une forme simple et qui respecte la structure de la hiérarchie. Cela signifie en particulier que les nœuds possédant plusieurs parents ne sont pas dupliqués. PALÉTUVIER inclut par ailleurs une fonctionnalité de *zoom* permettant de choisir entre une vue globale de la hiérarchie ou une vue plus détaillée d'une partie de celle-ci. Le deuxième (voir figure 2.9(b)) repose sur l'API HYPERTREE développée par Christophe Bouthier [Bouthier, 2005] et permet de visualiser la hiérarchie des classes sous la forme d'un arbre hyperbolique. Ce type de visualisation est reconnu comme étant le plus efficace concernant la navigation [Pirolli *et al.*, 2001]. Il s'agit ici de projeter l'ensemble de la hiérarchie sur un disque, selon les principes de la géométrie hyperbolique et du modèle de Poincaré, résultant en un effet d'« œil de poisson ». Cette technique permet de focaliser l'affichage sur un élément particulier de la hiérarchie, tout en conservant une vue du contexte dans lequel il se trouve, c'est-à-dire de son voisinage dans la hiérarchie.

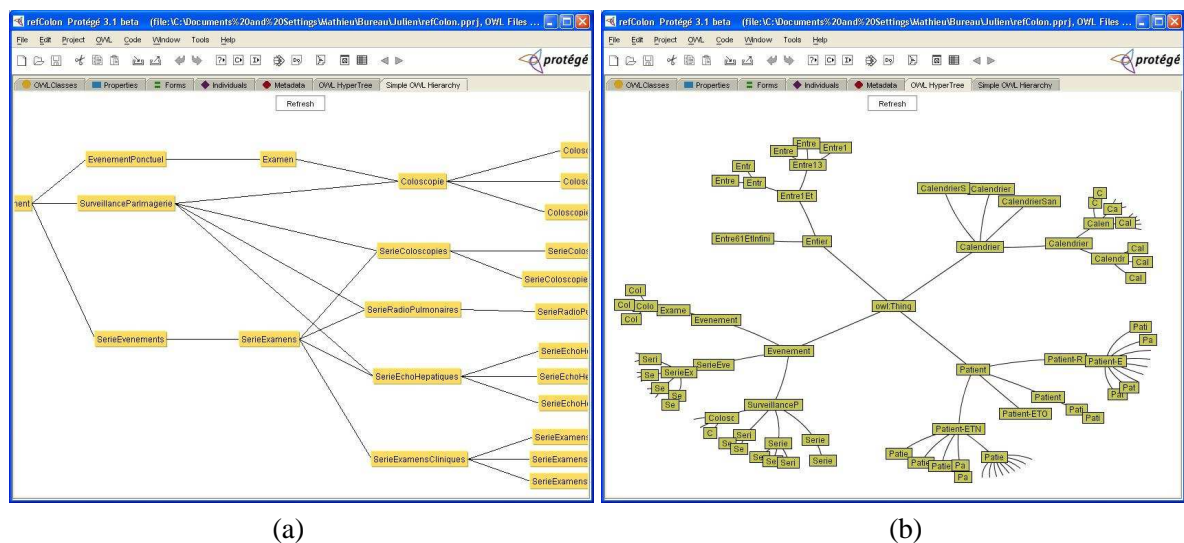


FIG. 2.9 – Les deux modules de visualisation de connaissances intégrés à PROTÉGÉ : PALÉTUVIER (a) et HYPERTREE (b).

Comparaison de versions de bases de connaissances

Lors de la maintenance ou de la mise à jour d'une ontologie, pouvoir visualiser la portée des modifications effectuées peut s'avérer très utile. Pour cette raison, un outil de comparaison de versions d'ontologies, appelé KILT, a été développé dans le cadre de KASIMIR et intégré en tant que greffon

à l'éditeur PROTÉGÉ. Cet outil a été conçu pour s'appuyer sur un langage et un moteur d'inférences propres à KASIMIR et, contrairement aux outils présentés précédemment, n'a pas encore été adapté pour OWL. Nous détaillons néanmoins ici les principes sous-jacents à KILT tels qu'ils se traduisent dans ce langage.

De façon générale, KILT cherche à comparer le contenu de la version courante d'une ontologie, appelée $O_{\text{après}}$, avec celui d'une version antérieure, appelée O_{avant} . Plus précisément, il s'agit de retrouver les classes qui ont été ajoutées, celles qui ont été supprimées et celles qui sont restées inchangées. Pour cela, une distinction est faite entre les *classes atomiques*, c'est-à-dire celles qui ne possèdent pas d'autre définition que leur nom, et les *classes définies*, c'est-à-dire celles possédant une définition sous la forme d'une expression construite. À partir de cette distinction, l'algorithme sur lequel repose KILT est très simple : il s'agit d'apparier les éléments de $O_{\text{après}}$ et O_{avant} en établissant des équivalences entre ces éléments. Les classes atomiques et les propriétés sont appariées selon leur nom. Par exemple, s'il se trouve une classe atomique nommée GRADE1 dans $O_{\text{après}}$, avec comme URI <http://www.ontologies.org/apres#Grade1>, et une classe atomique également nommée GRADE1 dans O_{avant} , avec comme URI <http://www.ontologies.org/avant#Grade1>, alors KILT établira une relation d'équivalence entre ces deux classes, sous la forme :

```
<owl:Class rdf:about="http://www.ontologies.org/apres#Grade1">
  <owl:equivalentClass rdf:resource="http://www.ontologies.org/avant#Grade1"/>
</owl:Class>
```

Les classes définies étant construites sur la base des classes atomiques et des propriétés, l'algorithme de KILT utilise par la suite un moteur d'inférences pour LD afin de *classifier les classes définies de $O_{\text{après}}$ dans la hiérarchie de O_{avant}* . Toutes paires de classes possédant des définitions équivalentes seront alors détectées comme équivalentes. Il sera ainsi possible de distinguer les classes inchangées, c'est-à-dire les classes de $O_{\text{après}}$ possédant un équivalent dans O_{avant} , les classes ajoutées, c'est-à-dire celles de $O_{\text{après}}$ ne possédant pas d'équivalent dans O_{avant} , et enfin, les classes supprimées, c'est-à-dire celles de O_{avant} ne possédant pas d'équivalent dans $O_{\text{après}}$.

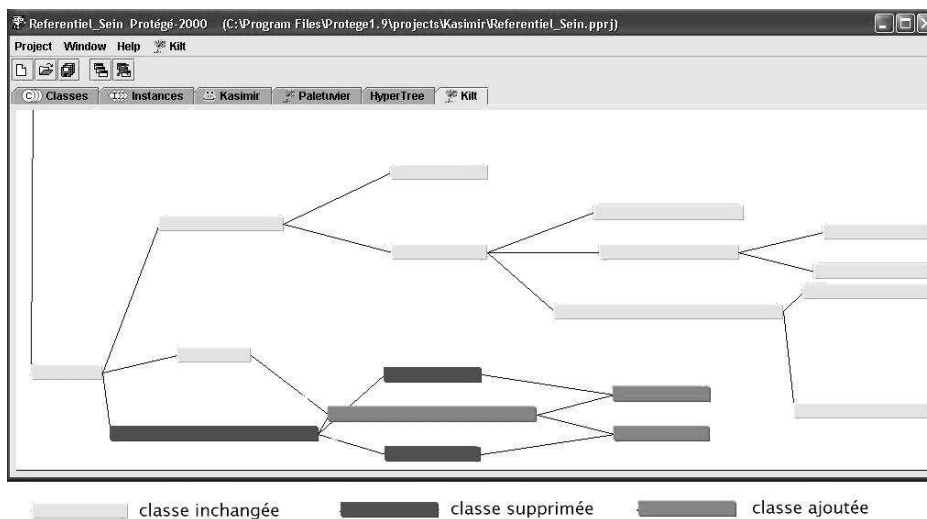


FIG. 2.10 – Le greffon KILT pour la visualisation des modifications effectuées sur une ontologie.

KILT est intégré à l'éditeur PROTÉGÉ sous la forme d'un greffon permettant de visualiser les modifications effectuées au cours d'une session d'édition. Dans ce cadre, O_{avant} est l'ontologie telle qu'elle

était à son ouverture dans PROTÉGÉ et $O_{\text{après}}$ est l'ontologie courante, ayant subie des modifications. Le greffon KILT affiche le résultat de l'algorithme présenté ci-dessus sous la forme d'une hiérarchie de classes, avec différentes couleurs pour les différentes catégories de classes : les classes inchangées sont affichées en jaune, celles ajoutées sont affichées en vert et celles supprimées sont affichées en rouge (voir figure 2.10). Du fait de l'utilisation intensive des mécanismes de raisonnement liés à OWL, connus pour être très complexes, le temps de calcul pour dérouler l'algorithme de KILT peut être important dans le cas d'ontologies de grandes tailles, limitant ainsi son utilisation pratique. La mise en œuvre de mécanismes permettant le *passage à l'échelle* de ce type d'inférences (voir par exemple [Groot *et al.*, 2005]) fait partie des développements envisagés pour améliorer ce point.

2.4 Services Web pour le raisonnement

D'un point de vue logiciel, l'architecture du portail sémantique KASIMIR repose essentiellement sur un *serveur de connaissances* implémenté sous la forme de services Web. Celui-ci a pour rôle de permettre l'interrogation des ontologies diffusées au sein du portail (retrouver les instances d'une classe, ses sous-classes, etc.) sur la base des mécanismes de raisonnement sous-jacents au langage OWL. Pour cela, le serveur de connaissances est implémenté grâce à la *boîte à outils* JENA. Deux moteurs d'inférences différents peuvent être intégrés à ce serveur, celui inclus dans JENA et le système de LD PELLET. Le moteur d'inférences présent au sein de JENA est un moteur à base de règles, où les règles traduisent les inférences possibles sur le langage OWL. Par exemple, dans la syntaxe dédiée, les règles :

```
[scot: (?a rdfs:subClassOf ?b), (?b rdfs:subClassOf ?c) -> (?a rdfs:subClassOf ?c)]
[scor: (?a rdf:type owl:Class) -> (?a rdfs:subClassOf ?a)]
[scoa: (?a rdfs:subClassOf ?b), (?b rdfs:subClassOf ?a) -> (?a owl:equivalentClass ?b)]
```

indiquent les inférences réalisables sur la base des propriétés de transitivité, de réflexivité et d'antisymétrie de la subsomption en OWL. Cela a l'avantage de permettre la construction d'une base de règles propre à une application, ne prenant en compte que les constructeurs d'OWL jugés utiles et en intégrant de nouveaux. De cette façon, une base de règles pour un sous-ensemble très simple de OWL, mais intégrant la possibilité de raisonner sur des *types de données dérivés* décrits en XML SCHÉMA, a été développée pour les besoins du projet KASIMIR. Le principal désavantage de cette approche est que, même si l'expressivité du langage pris en compte est très restreinte, ce moteur d'inférences à base de règles est particulièrement gourmand en ressources. Ainsi, une autre version du serveur de connaissances a aussi été développée sur la base du moteur d'inférences pour LD PELLET, qui prend en compte l'ensemble de OWL DL. Celui-ci a nécessité d'être étendu pour pouvoir raisonner sur des types de données dérivés.

Les services Web peuvent être vus comme des composants logiciels dont l'interface est accessible au travers du Web. Ils sont ainsi utilisables à distance pour un grand nombre d'applications, clientes des services et communiquant avec ceux-ci au travers du protocole standard SOAP. On pourrait par exemple imaginer que l'application en charge de la gestion des *dossiers patients informatisés* au sein d'un centre hospitalier fasse appel au serveur de connaissances du portail pour inclure automatiquement les recommandations des référentiels dans le dossier d'un patient. De plus, centraliser comme c'est le cas ici l'accès aux connaissances et au moteur d'inférences permet de mettre en œuvre des mécanismes de *cache*, réutilisant les résultats des inférences réalisées pour un client afin de répondre aux requêtes d'autres clients. Trois services *intermédiaires* jouent ce rôle au sein du portail KASIMIR : un pour l'accès aux éléments relatifs aux classes, un pour les éléments relatifs aux propriétés et un pour ceux relatifs aux instances. Celui pour les classes, par exemple, conserve dans des structures à accès rapide les résultats des inférences concernant la subsomption. Lorsqu'un client demande au serveur les sous-classes d'une classe d'une ontologie, ce service de cache est appelé en premier lieu. S'il contient l'information recherchée, alors il la renvoie au client, sinon il fait appel au moteur d'inférences et stocke le résultat pour une

prochaine utilisation. Il faut remarquer que l'utilisation de ce type de technique impose de réinitialiser les caches à chaque modification d'une ontologie.

2.5 Interfaces pour l'accès aux connaissances et l'aide à la décision

Deux applications clientes du serveur de connaissances décrit à la section précédente ont été développées. Il s'agit toutes deux d'interfaces graphiques permettant l'accès aux connaissances contenues dans les référentiels et, par ce biais, l'aide à la décision. La première, ÉDHIBOU, se présente comme une interface graphique pour la saisie, guidée par une ontologie en OWL, d'une instance d'une classe de cette ontologie. Les mécanismes de raisonnement fournis par le serveur sont utilisés, au sein de cette interface, pour compléter les informations saisies par des informations inférées sur la base de l'ontologie. Dans le cadre du projet KASIMIR, ÉDHIBOU permet de construire une instance de la classe *Patient*, indiquant les caractéristiques du patient correspondant, et ainsi, d'obtenir une recommandation déduite à partir de la représentation en OWL d'un référentiel pour ce patient. La deuxième interface, appelée NAVHIBOU, permet de naviguer au sein d'une ontologie en OWL au travers de liens *hypertextes*. À chaque étape de la navigation NAVHIBOU indique les informations qui peuvent être inférées à partir des éléments du niveau courant dans l'ontologie. Cette interface permet ainsi de parcourir la représentation d'un référentiel selon les caractéristiques d'un patient particulier, afin d'établir la décision recommandée pour ce patient.

2.5.1 ÉDHIBOU : éditeur d'instances OWL

De façon générale, ÉDHIBOU peut être vu comme un éditeur d'instances pour OWL. Cela signifie que, à partir d'une ontologie et d'une classe de cette ontologie, ÉDHIBOU va *générer* dynamiquement une interface graphique permettant de saisir une instance de cette classe et d'en indiquer les propriétés. Les mécanismes de raisonnements associés à OWL sont utilisés dans ce cadre, afin de retrouver les propriétés applicables à l'instance considérée, ainsi que les valeurs possibles pour ces propriétés. ÉDHIBOU va ensuite générer le code OWL correspondant à l'instance éditée et le transmettre au serveur de connaissances, qui aura la charge de spécifier sa classe d'appartenance (mécanisme d'instanciation) et d'en inférer de nouvelles propriétés (mécanisme de complétion d'instances).

Un exemple d'utilisation de ÉDHIBOU dans le cadre du projet KASIMIR est présenté à la figure 2.11. L'utilisateur a en premier lieu entré l'URL du fichier OWL contenant l'ontologie (partie haute de la figure), permettant à ÉDHIBOU d'afficher la hiérarchie des classes de cette ontologie (partie gauche de la figure). La classe *Patient* a ensuite été sélectionnée et l'ensemble des propriétés applicables aux instances de cette classe, ainsi que les valeurs possibles pour ces propriétés, ont été affichées par ÉDHIBOU. L'utilisateur a par la suite saisi certaines de ces valeurs : l'âge, le sexe, le grade et la taille de la tumeur, l'adénopathie et le statut des récepteurs hormonaux. À partir de ces informations, l'instance éditée a été reconnue par le mécanisme d'instanciation comme appartenant à la classe *PTSE1*. Le mécanisme de complétion d'instances a alors inféré une valeur provenant de la classe *ChimiotherapieNiveau1* pour la propriété *recommandation* (car $PTSE1 \sqsubseteq \exists recommandation.ChimiotherapieNiveau1$). Il faut remarquer que la propriété *menopausee* a été ajoutée à la liste des propriétés applicables après que l'instance éditée ait été reconnue comme appartenant à la classe *Patiente* (patient de sexe féminin).

Concernant l'implémentation d'ÉDHIBOU, celui-ci est développé en langage Java et s'appuie sur la librairie APACHE SOAP [(ASC), 2005] pour la communication avec le serveur de connaissances du portail sémantique KASIMIR. Un important travail de développement est en cours concernant ÉDHIBOU, afin de permettre la *personnalisation* de l'interface générée (choix des couleurs, de l'ordre des propriétés, séparation en différents panneaux, greffons spécifiques à une application, etc.).

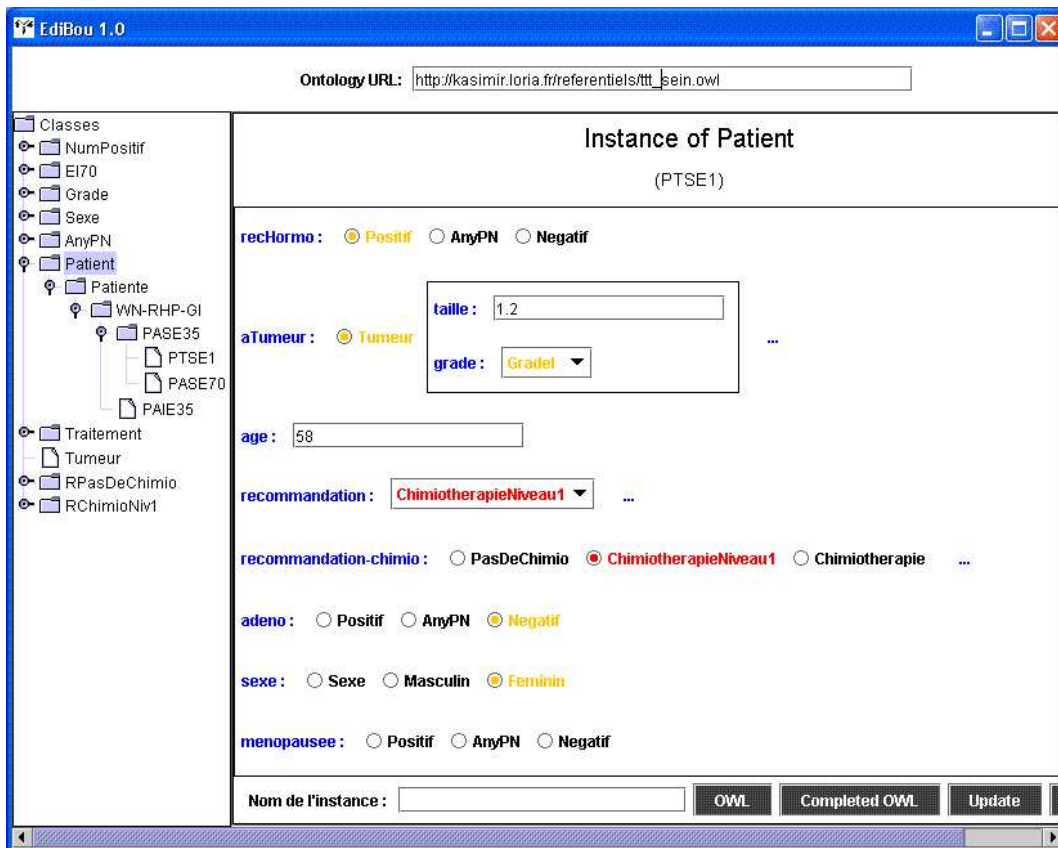


FIG. 2.11 – L'éditeur d'instances pour OWL ÉDHIBOU.

2.5.2 NAVHIBOU : navigation dans une ontologie OWL

NAVHIBOU est une interface permettant de naviguer dans la hiérarchie des classes d'une ontologie en OWL en suivant des liens *hypertextes* d'une page Web classique (en HTML). Pour chacune des classes parcourues, NAVHIBOU génère une page contenant des informations sur la classe, ainsi que des liens vers ses sous-classes directes, sous la forme d'un arbre de profondeur 1 (voir figure 2.12). Les informations attachées aux classes sont inférées à l'aide du serveur de connaissances. Une instance de la classe courante est d'abord créée et son code OWL transmis au serveur. Au travers du mécanisme de complétion d'instances, des propriétés peuvent ainsi être inférées pour cette instance. Par exemple, il peut être déduit par ce biais que toute instance de la classe *Patiente* ($\text{Patient} \sqcap \exists \text{sexe.Feminin}$), prend un objet de la classe *Feminin* pour valeur de la propriété *sexe*. Cette information est affichée par NAVHIBOU et attachée à la classe *Patiente*, sous la forme « *sexe* : *Feminin* » (en fait, sur la figure, les *étiquettes* des classes et des propriétés sont utilisées, ce qui donne « a pour *sexe* : *sexe Feminin* »).

Dans le cadre du projet KASIMIR, NAVHIBOU est utilisé pour parcourir la hiérarchie des classes contenues dans la représentation en OWL d'un référentiel, en choisissant à chaque étape la classe correspondant aux caractéristiques du patient considéré. Ce parcours permet, *in fine*, d'aboutir à une classe pour laquelle une décision est recommandée par le référentiel. En ce sens, l'utilisation de NAVHIBOU se rapproche de l'utilisation des diagrammes contenus dans les référentiels, à la différence que NAVHIBOU génère ces « arbres » dynamiquement, à partir d'une représentation formelle du contenu du référentiel.

Concernant l'implémentation, NAVHIBOU est développé dans le langage PHP et repose sur la li-

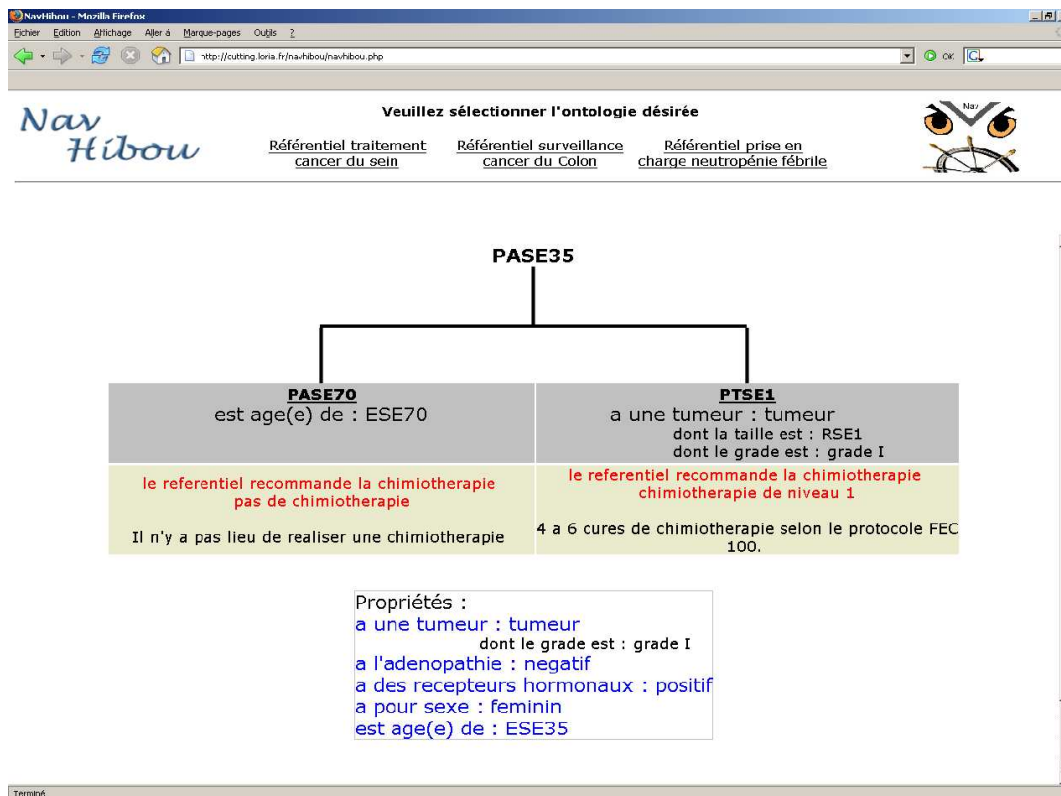


FIG. 2.12 – Utilisation de NAVHIBOU pour parcourir les classes d’une ontologie en OWL.

brairie PHP nuSOAP [Ayala et Nichol, 2005] pour la communication avec le serveur de connaissances du portail sémantique KASIMIR. De plus, NAVHIBOU génère des pages HTML. Il est donc possible de récupérer l’ensemble des pages correspondant à un référentiel, grâce à un *aspirateur Web*, et ainsi, de les consulter par la suite sans avoir à être connecté au serveur.

2.6 Étendre OWL pour répondre aux objectifs de Kasimir

Nous avons vu dans ce chapitre que le Web sémantique apporte une base technologique adaptée à la mise en œuvre de la gestion des connaissances décisionnelles en cancérologie, incluant l’aide à la décision pour l’application des référentiels. En particulier, le langage de représentation d’ontologies OWL fournit les mécanismes nécessaires à la représentation formelle des référentiels et aux raisonnements, fondements du portail sémantique KASIMIR. Mais les technologies du Web sémantique, OWL en tête, s’avèrent insuffisantes au regard de certaines des problématiques du projet KASIMIR. En particulier, les mécanismes de représentation et de raisonnements associés à OWL ne permettent pas de répondre aux besoins engendrés par l’aide à la décision dans le cadre de l’adaptation des référentiels. La notion d’ontologies semble de plus inadaptée lorsqu’il s’agit de prendre en compte plusieurs points de vue sur le domaine, comme les multiples disciplines de la cancérologie. Pour cette raison, les chapitres suivants s’intéressent à étendre les principes et les technologies du Web sémantique, principalement ceux liés à OWL, afin de répondre aux besoins du projet KASIMIR en termes de représentation des connaissances et de raisonnements.

Chapitre 3

Raisonnement à partir de cas dans l'infrastructure du Web sémantique

Le Web sémantique repose sur la représentation formelle des connaissances diffusées sur le Web afin de permettre l'utilisation de mécanismes de raisonnements automatiques pour l'accès à ces connaissances. Dans le chapitre précédent, une architecture fondée sur ces technologies est mise en œuvre pour permettre l'aide à la décision par l'application des référentiels dans le cadre du projet KASIMIR. OWL, le formalisme de représentation des connaissances employé, a été conçu pour l'utilisation de raisonnements déductifs, tels que la classification et l'instanciation, bien adaptés à cette tâche. Mais, lorsque pour un patient particulier un référentiel n'est pas applicable, la décision est prise par *adaptation* du référentiel. Le *raisonnement à partir de cas* (RÀPC) est un paradigme de raisonnement pour la résolution de problèmes qui repose sur l'adaptation des solutions de problèmes déjà résolus, stockés dans une *base de cas*, pour résoudre de nouveaux problèmes. L'aide à la décision pour l'adaptation des référentiels peut ainsi être mise en œuvre au travers d'un mécanisme de raisonnement fondé sur les principes du RÀPC. Il s'agit en effet ici, pour un patient sortant du cadre d'un référentiel, d'adapter les décisions recommandées par ce référentiel pour d'autres patients, jugés similaires au premier et pour lesquels le référentiel s'applique. Au-delà de l'application au projet KASIMIR, ce chapitre s'intéresse à la mise en œuvre du RÀPC dans l'infrastructure du Web sémantique. On cherchera en effet à montrer comment les technologies telles que OWL peuvent être étendues pour le RÀPC, apportant ainsi au Web sémantique un mode de raisonnement relevant non plus de la déduction, mais de l'analogie. Ce mode de raisonnement est alors opérationnalisé au sein d'un service de RÀPC générique, applicable aux connaissances diffusées au travers du Web sémantique et conçu pour être intégré au portail sémantique KASIMIR.

3.1 Le RàPC comme élément d'un système à base de connaissances

On peut distinguer quatre principaux paradigmes de raisonnement en intelligence artificielle : la déduction, l'induction, l'abduction et l'analogie [Chouraqui, 1986]. Le mécanisme de déduction est celui sur lequel repose la majorité des logiques, en particulier les LD et donc aussi OWL, le langage de représentation d'ontologies du Web sémantique. Le RÀPC est vu dans [Haton *et al.*, 1991] comme un cas particulier de raisonnement par analogie. L'analogie consiste à inférer des éléments concernant une situation particulière à partir des éléments connus d'une situation jugée analogue. Un exemple classique concerne l'analogie qui met en correspondance l'attraction gravitationnelle s'exerçant au sein du système solaire, entre les planètes et le soleil, et l'attraction électrostatique s'exerçant au sein d'un atome, entre les électrons et le noyau. Le RÀPC est alors considéré comme un mode de raisonnement par analogie appliqué au sein d'un seul et même domaine, sur des situations de même nature [Aamodt et Plaza, 1994].

Nous nous plaçons plus particulièrement ici dans le cadre de la *résolution de problèmes*, où le RÀPC se définit comme le processus permettant de résoudre un problème cible en réutilisant et en adaptant les solutions connues de problèmes sources similaires.

3.1.1 Principes du RÀPC

Nous considérons un *cas*, noté $(pb, Sol(pb))$, comme la description d'un problème pb associé à sa solution $Sol(pb)$. L'objectif principal du processus de RÀPC est d'établir une solution $Sol(cible)$ d'un problème cible, noté $cible$, en réutilisant la solution $Sol(source)$ contenue dans un cas source $(source, Sol(source))$ connu. Les deux étapes fondamentales pour cela sont la remémoration et l'adaptation (voir figure 3.1). La *remémoration* consiste à retrouver, parmi un ensemble de cas sources stockés dans une *base de cas*, un problème source jugé similaire au problème cible à résoudre. La solution du problème cible est ensuite construite par *adaptation* de la solution $Sol(source)$ du problème source remémoré.

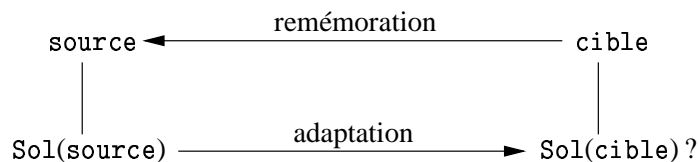


FIG. 3.1 – Remémoration et adaptation en RÀPC.

Outre les deux étapes fondamentales de remémoration et d'adaptation qui caractérisent le RÀPC, d'autres opérations peuvent être considérées, en fonction des besoins du domaine d'application. En effet, l'article [Aamodt et Plaza, 1994] jugé comme une référence dans le domaine, définit un *cycle du RÀPC* composé de quatre étapes. À la suite de la remémoration et de l'adaptation (*retrieve* et *reuse*) s'ajoutent les étapes de révision de la solution (*revise*) et de mémorisation du cas (*retain*). La *révision* suppose d'être en mesure de tester la validité de la solution $Sol(cible)$ construite et éventuellement de la modifier pour la rendre valide. La *mémorisation* correspond à une étape d'apprentissage au sein du processus de RÀPC. Il s'agit, si cela est jugé utile et une fois la solution $Sol(cible)$ validée, d'ajouter le cas $(cible, Sol(cible))$ à la base de cas pour une éventuelle utilisation future. Enfin, certains travaux, comme par exemple [Mille, 1998], considèrent une étape précédant la remémoration et ayant pour objectif l'*élaboration* du problème cible. Cette opération vise à guider la formulation du problème cible dans une représentation utilisable par le système de RÀPC, à partir d'une description éventuellement informelle et incomplète de ce problème.

De nombreux systèmes de RÀPC ont été implémentés depuis les premiers travaux dans le domaine [Riesbeck et Schank, 1989]. Ils se concentrent généralement sur des domaines d'application particuliers. Le système DÉJÀ VU [Smyth, 1996] se focalise par exemple sur des problèmes de planification en conception. Le système CHEF [Hammond, 1990] s'intéresse quant à lui à la mise en œuvre de recettes de cuisine. Ces différents systèmes reposent sur une très grande variété de techniques et de représentations pour la mise en œuvre des opérations mentionnées plus haut. Nous nous concentrerons donc dans la suite sur une approche particulière du RÀPC : le RÀPC *guidé par les connaissances*, appelé en anglais *knowledge-intensive case-based reasoning* [Aamodt, 1990, Aamodt, 2004]. Les différentes étapes du cycle du RÀPC sont ainsi détaillées dans la section suivante, telles qu'elles sont considérées par cette approche.

3.1.2 Le RÀPC guidé par les connaissances

Le RÀPC *guidé par les connaissances* est une approche particulière du RÀPC, décrite en premier lieu dans [Aamodt, 1990] et appliquée de nombreuses fois par la suite, dans différents systèmes et différents formalismes (par exemple dans [Lieber, 1997], [Salotti et Ventos, 1999], [Gómez-Albarrán *et al.*, 1999] et [Aamodt, 2004]). Dans cette approche, les connaissances utiles à la mise en œuvre des étapes du cycle du RÀPC sont décrites au sein d'un modèle formel, dans un langage de représentation des connaissances. Nous distinguons ici trois types de connaissances utiles au processus de RÀPC : les cas, les connaissances du domaine et les connaissances d'adaptation. Un modèle des *connaissances du domaine* est en effet nécessaire afin de définir les notions relatives au domaine d'application que le processus de RÀPC aura à manipuler. Parmi les *conteneurs de connaissances* (en anglais *knowledge containers*) décrits dans [Richter, 1998], le modèle des connaissances du domaine est proche de ce que Michael M. Richter appelle le *vocabulaire utilisé*. Ce que nous appelons les *connaissances d'adaptation* correspond aux connaissances relatives à la similarité entre problèmes et à l'adaptation des solutions, essentielles aux étapes de remémoration et d'adaptation du RÀPC. Dans la terminologie de [Richter, 1998], le modèle de connaissances d'adaptation se rapproche ainsi des conteneurs de connaissances relatifs aux *mesures de similarité* et aux *transformations de solutions*.

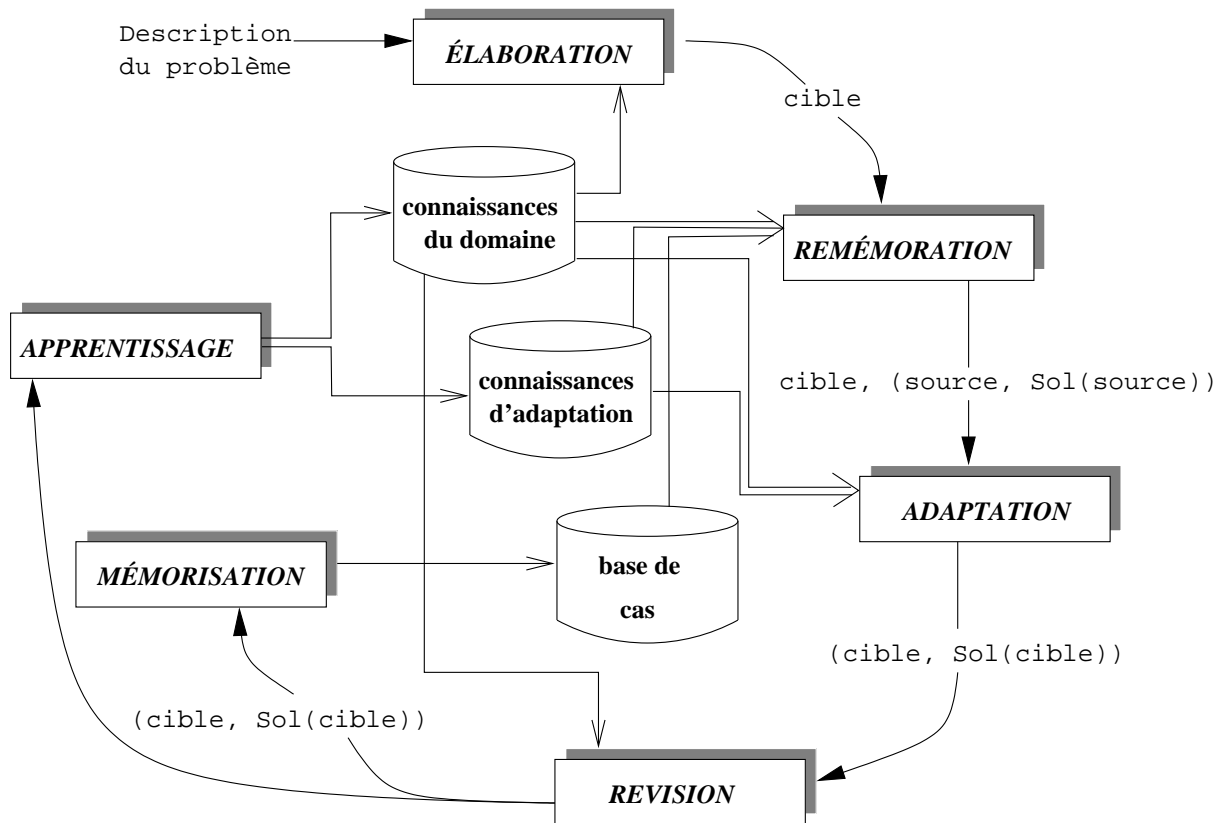


FIG. 3.2 – Les étapes du RÀPC guidé par les connaissances.

La figure 3.2 résume le cycle du RÀPC selon l'approche considérée ici. Les différentes opérations de ce cycle reposent sur des connaissances décrites au sein d'une base de cas, d'un modèle des connaissances du domaine et d'un modèle des connaissances d'adaptation.

L'élaboration. L'élaboration est l'étape où la représentation du problème `cible` est construite. Il s'agit ici d'aider l'utilisateur à formuler le problème de façon à ce qu'il soit manipulable par le système. Le modèle des connaissances du domaine fournit ici un cadre, un vocabulaire, pour cette description. Par ailleurs, des informations supplémentaires, venant s'ajouter à la description éventuellement incomplète du problème, pourront être inférées lors de cette étape, à partir des connaissances du domaine [Mille, 1998].

La remémoration. Une fois le problème `cible` construit, la remémoration a pour rôle de retrouver, au sein de la base de cas, un ou plusieurs cas sources réutilisables pour résoudre le problème `cible`. Nous suivons ici une approche particulière de la remémoration, appelée *remémoration guidée par l'adaptation* [Smyth, 1996]. Celle-ci repose sur le principe selon lequel un cas source n'est remémoré que s'il est adaptable en une solution du problème `cible`. Selon cette approche, l'étape de remémoration s'appuie donc essentiellement sur les connaissances d'adaptation, qui contiennent les éléments permettant de juger de la similarité entre un problème source et le problème `cible`, ainsi que de l'adaptabilité de sa solution. Les connaissances du domaine sont aussi utilisées du fait qu'elles sont à la base de la description des connaissances d'adaptation. Elles peuvent par ailleurs être utilisées pour guider la remémoration en jouant un rôle dans l'indexation de la base de cas [Lieber, 1997, Salotti et Ventos, 1999].

L'adaptation. L'adaptation est souvent considérée comme l'étape la plus importante et la plus complexe du RÀPC [Fuchs *et al.*, 1999]. C'est en effet lors de cette opération qu'est construite la solution du problème `cible`. Il s'agit de réutiliser la solution du problème source remémoré et de l'*adapter* afin qu'elle corresponde à une solution du problème `cible`. Les connaissances d'adaptation sont dans ce cadre essentielles pour déterminer les adaptations nécessaires, en fonction du cas source, du problème `cible` et des éléments de similarité entre source et `cible` utilisés lors de la remémoration. Certains systèmes s'appuient par ailleurs sur plusieurs cas sources et nécessitent ainsi l'adaptation et la combinaison de plusieurs solutions `Sol(source)` (voir par exemple [Netten et Vingerhoeds, 1996] et [Smyth, 1996]).

La révision. La révision consiste à tester la validité de la solution `Sol(cible)` construite par adaptation et éventuellement à la réparer si elle n'est pas valide. La révision est une opération essentielle par exemple pour le système CHEF [Hammond, 1990], dans lequel une recette de cuisine construite par adaptation devra être validée et éventuellement réparée pour donner le résultat escompté. Dans le cadre du RÀPC guidé par les connaissances, cette opération repose principalement sur les connaissances du domaine, qui contiennent les éléments permettant de reconnaître une solution comme valide. Néanmoins, les connaissances nécessaires pour cela n'étant que rarement formalisées, la révision est dans la plupart des cas réalisée par l'utilisateur.

La mémorisation et l'apprentissage. Dans l'approche considérée ici, une distinction doit être faite entre deux formes d'apprentissage : la mémorisation de cas et l'ajout de nouvelles connaissances. La mémorisation consiste à inclure le cas (`cible`, `Sol(cible)`) validé dans la base de cas pour une éventuelle utilisation future. L'autre forme d'apprentissage consiste à extraire de nouvelles connaissances, de domaine ou d'adaptation, à partir de l'épisode de résolution de problèmes ayant mené à la solution `Sol(cible)`. Par exemple, dans [Hammond, 1989], en plus de la mémorisation des cas validés et réparés, des connaissances sont extraites à partir de la révision des cas. Celles-ci concernent principalement les caractéristiques permettant de reconnaître une solution comme valide, ainsi que des éléments pour la réparation.

3.1.3 Acquisition et représentation des connaissances pour le RÀPC

Le RÀPC a souvent été cité comme une alternative aux systèmes à base de règles dans le sens où il permet de réduire, dans une certaine mesure, le problème de l'acquisition des connaissances, les cas étant considérés comme plus faciles à acquérir que des règles [Kolodner, 1993]. Cependant, l'approche présentée ici défend l'idée que le RÀPC, au même titre que tout système à base de connaissances, repose sur des connaissances qu'il faut acquérir et formaliser. Les domaines de l'ingénierie et de la gestion des connaissances se sont longtemps concentrés sur le développement de méthodes et d'outils répondant à cette problématique [Dieng *et al.*, 2000]. Ceux-ci sont ainsi applicables à la mise en œuvre des connaissances du domaine sur lesquelles repose le RÀPC. L'acquisition des connaissances d'adaptation est plus particulièrement étudiée dans le domaine du RÀPC. Plusieurs approches, résumées dans [Lieber *et al.*, 2004], peuvent être employées pour cela, selon les sources de connaissances disponibles, le type des connaissances à acquérir ou le formalisme de représentation des cas utilisé.

Le domaine de la représentation des connaissances se trouve au centre des problématiques liées à la construction d'un système de RÀPC guidé par les connaissances. Il apporte en effet non seulement les langages selon lesquels les cas, les connaissances du domaine et les connaissances d'adaptation peuvent être formalisés, mais aussi les mécanismes d'inférences utiles à la mise en œuvre des différentes opérations du cycle du RÀPC. Le système RESYN/RÀPC [Lieber, 1997], par exemple, s'appuie sur un langage de représentation des connaissances par objets (RCO) et étend le mécanisme de raisonnement par classification pour une application du RÀPC dans le domaine de la synthèse en chimie organique. Toujours en RCO, les travaux de Béatrice Fuchs [Fuchs, 1997] ont amené au développement du système ROCADE, un système de représentation des connaissances dédié au RÀPC. De nombreux systèmes se sont intéressés à la mise en œuvre du RÀPC dans le cadre des LD, comme par exemple [Salotti et Ventos, 1999], [Koehler, 1994], [Kamp, 1996], [Gómez-Albarrán *et al.*, 1999] et [Metzger, 2002]. Néanmoins, les formalismes tels que les LD et les langages de la RCO n'ont pas été conçus dans l'objectif d'exprimer des notions relatives à l'adaptation, nécessaires au RÀPC et présentes au sein des connaissances d'adaptation. La mise en œuvre d'un système de RÀPC guidé par les connaissances nécessite ainsi de développer, conjointement au formalisme utilisé, des capacités de représentation de ces connaissances d'un genre particulier.

3.1.4 Le RÀPC au sein du Web sémantique

Nous nous intéressons ici à la mise en œuvre du RÀPC guidé par des connaissances formalisées et diffusées selon les principes et les technologies du Web sémantique. Le Web sémantique vise en effet à la diffusion de connaissances permettant la mise en œuvre de mécanismes de raisonnements automatiques, qui sont, à l'heure actuelle, principalement déductifs. Le RÀPC est souvent considéré comme une bonne alternative à ce type de raisonnements dans des domaines incomplètement formalisés [Kolodner, 1993], comme c'est le cas de la cancérologie, ainsi que de beaucoup d'autres domaines d'applications du Web sémantique. Néanmoins, très peu de travaux mêlent les deux domaines. Quelques-uns ont pour objectif de décrire un langage de représentation des cas dans des technologies du Web, telles que XML [Coyle *et al.*, 2004] et RDF [Chen et Wu, 2003]. Dans [Champin, 2002], Pierre-Antoine Champin part quant à lui de l'idée que les technologies du Web sémantique visent à assister l'utilisation des ressources diffusées sur le Web et que, conjointement aux connaissances définies au sein d'ontologies, il peut être profitable de s'appuyer pour cela sur des expériences d'utilisation. Il propose ainsi d'annoter ces ressources selon leurs usages, en s'appuyant sur un modèle de *trace d'utilisation* en RDF, et d'exploiter ces expériences selon les principes du RÀPC. Enfin, les travaux décrits dans [Díaz-Agudo et González-Calero, 2002], bien que n'étant pas explicitement liés au Web sémantique, visent à permettre la réutilisation de connaissances représentées sous la forme d'ontologies au sein

du RÀPC. En s'appuyant sur les LD, ils décrivent une *ontologie du RÀPC* (CBRONTO) servant d'intermédiaire entre des ontologies de domaine et un système de RÀPC appelé COLIBRI. [Bello-Tomás *et al.*, 2004] montre l'évolution de COLIBRI en JCOLIBRI, développé dans le langage Java. Dans ce dernier, les avantages du RÀPC guidé par les ontologies sont laissés de côté, au profit d'une configuration et d'une implémentation simplifiées de systèmes de RÀPC à partir de la description de *tâches et de méthodes*. On remarque par ailleurs que l'ontologie du RÀPC, CBRONTO, n'est plus un composant à part entière, isolable du système et réutilisable, mais « est contenue dans le modèle UML de l'application » [Bello-Tomás *et al.*, 2004].

3.2 Représentation des connaissances d'adaptation dans le cadre du Web sémantique

Au sein du Web sémantique, une ontologie a pour rôle de contenir une représentation formelle des connaissances relatives à un domaine particulier. Un des avantages d'intégrer le RÀPC aux technologies du Web sémantique est la possibilité de s'appuyer sur ces ontologies comme modèles des connaissances du domaine, ainsi que comme cadre à la représentation des cas. En ce sens, OWL, le langage de représentation des ontologies, fournit la base technologique sur laquelle le RÀPC pourra s'appuyer. Néanmoins, des mécanismes permettant la formalisation et l'opérationnalisation de connaissances d'adaptation selon les principes et les technologies du Web sémantique restaient à développer. En premier lieu, contrairement à ce qui est fait dans beaucoup de systèmes de RÀPC où les notions de similarité et d'adaptation sont « codées » au sein du moteur, les connaissances d'adaptation doivent être représentées explicitement, découplées du moteur qui les manipule, et, par ce biais, être réutilisables, partageables, dans d'autres contextes et pour d'autres applications. Elles doivent de plus permettre la description des notions spécifiques au RÀPC relativement aux ontologies diffusées sur le Web sémantique, représentées en OWL, établissant de cette façon un lien entre les éléments du domaine d'application et le processus de RÀPC, indépendant du domaine. Cette section s'intéresse ainsi à la mise en œuvre d'un *modèle de représentation des connaissances d'adaptation* en OWL ou, en d'autres termes, d'une ontologie des connaissances utiles au RÀPC. Nous nous plaçons pour cela dans le cadre des *reformulations*, telles qu'elles sont définies dans [Lieber, 1997] et [Melis *et al.*, 1998].

3.2.1 Le modèle des reformulations

Le modèle des reformulations a été à l'origine développé pour la comparaison et l'adaptation de structures complexes (formules logiques dans [Melis, 1995] et graphes moléculaires dans [Lieber, 1997]). Une *reformulation* est définie comme une unité élémentaire pour la modélisation des connaissances d'adaptation [Melis *et al.*, 1998]. Elle est notée sous la forme d'un couple (r, \mathcal{A}_r) où r est une *relation entre problèmes* et \mathcal{A}_r , une *fonction d'adaptation de solutions*. Une telle reformulation s'interprète comme une règle indiquant que si deux problèmes pb_1 et pb_2 sont en relation par r (noté $pb_1 \ r \ pb_2$), alors la solution $Sol(pb_1)$ de pb_1 peut être adaptée par \mathcal{A}_r en une solution $Sol(pb_2)$ de pb_2 .

Dans le modèle des reformulations, la remémoration consiste à établir une séquence de relations entre problèmes r_i , formant ce qui est appelé un *chemin de similarité* et reliant un problème source de la base de cas, au problème cible à résoudre :

$$\text{source } r_1 \ pb_1 \ r_2 \ pb_2 \ \dots \ pb_{q-1} \ r_q \ \text{cible}$$

où les pb_i sont des problèmes intermédiaires, c'est-à-dire des problèmes introduits pour les besoins du raisonnement. Chacune des relations entre problèmes r_i qui composent un chemin de similarité est reliée par une reformulation à une fonction d'adaptation \mathcal{A}_{r_i} . L'adaptation consiste donc dans ce cadre à suivre

le chemin de similarité pour construire ce qui est appelé un *chemin d'adaptation*, appliquant les fonctions d'adaptation \mathcal{A}_{r_i} à chaque étape comme montré à la figure 3.3.

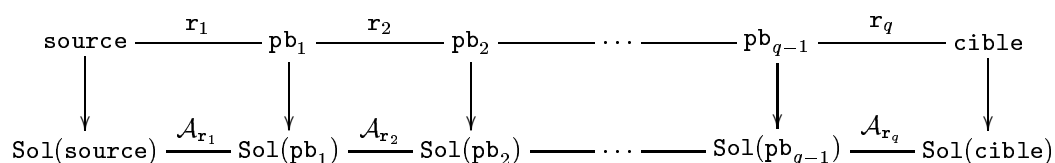


FIG. 3.3 – Un chemin de similarité entre *source* et *cible* (première ligne) et le chemin d'adaptation correspondant (deuxième ligne).

Le modèle des reformulations suit le principe de la *remémoration guidée par l'adaptation* (en anglais *adaptation-guided retrieval*, voir [Smyth, 1996]), garantissant qu'un cas source est remémoré seulement si sa solution est adaptable, c'est-à-dire si les connaissances nécessaires à cette adaptation sont disponibles. En ce sens, les chemins de similarité sont une forme de représentation symbolique de la similarité entre problèmes, permettant au système de RÀPC de construire des solutions explicables et compréhensibles. Il s'agit par ailleurs d'un modèle simple et général, conçu pour être spécifié pour différentes approches du RÀPC, dans différents domaines d'application. Il peut ainsi être utilisé comme une base commune à la modélisation de connaissances d'adaptation pour différentes applications.

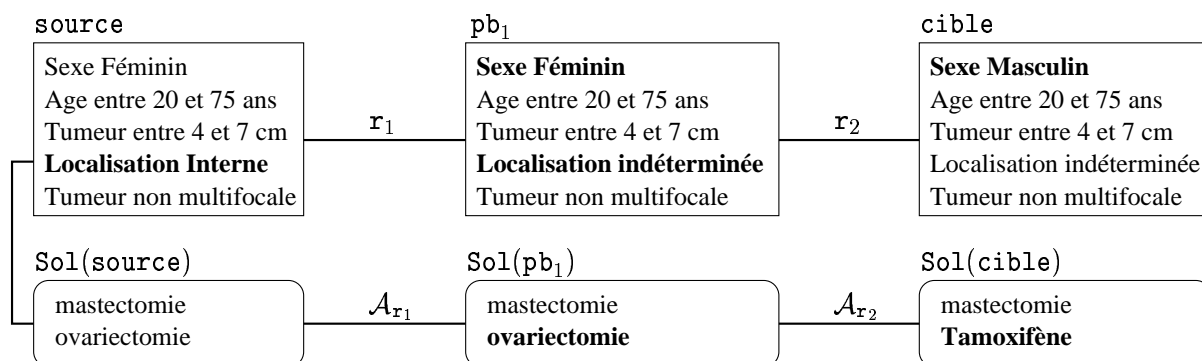


FIG. 3.4 – Exemple d'utilisation des reformulations.

Exemple d'application au traitement du cancer du sein. Au sein du projet KASIMIR, le RÀPC est utilisé afin d'établir une recommandation pour les patients pour lesquels un référentiel ne donne pas de réponse satisfaisante. Il s'agit d'adapter les solutions fournies par ce référentiel dans des situations similaires où il s'applique. Ainsi, dans le cadre du traitement du cancer du sein par exemple, les problèmes sont des descriptions de patients atteints de cancer du sein et les solutions des descriptions des traitements à appliquer. La figure 3.4 présente de façon informelle un exemple simplifié d'application des reformulations dans ce cadre. Sur cet exemple, le problème *cible* correspond à la description d'un patient pour lequel le référentiel ne donne pas de réponse. Il y a deux raisons à cela. Tout d'abord, le patient est un homme et le référentiel pour le traitement du cancer du sein est conçu pour des patients de sexe féminin. Par ailleurs, pour ce patient, la localisation de la tumeur dans le sein ne peut être déterminée (cela n'est pas sans lien avec le fait qu'il soit de sexe masculin). Les autres caractéristiques utilisées concernent l'âge

du patient, la taille de la tumeur et le fait que la tumeur ne soit pas multifocale. Deux reformulations, (r_1, \mathcal{A}_{r_1}) et (r_2, \mathcal{A}_{r_2}) , sont appliquées pour la résolution du problème `cible`. La première indique que, dans le cas où la localisation de la tumeur est indéterminée, on considère le patient selon la situation la plus pessimiste, c'est-à-dire celle d'une localisation interne. La fonction d'adaptation \mathcal{A}_{r_1} correspond alors à une simple copie de la solution. La deuxième reformulation indique que, comme le problème `pb` diffère du problème `cible` par la valeur de la caractéristique `sexe` (féminin pour le premier, masculin pour le deuxième), l'adaptation de `Sol(pb)` en `Sol(cible)` consiste à remplacer l'ovariectomie, impossible dans le cas d'un homme, par un traitement hormonothérapeutique ayant un bénéfice thérapeutique similaire : une cure de Tamoxifène.

3.2.2 Les reformulations en OWL

Le processus de RÀPC manipule des notions telles que celles de problèmes, de solutions, de similarité et d'adaptation. Au sein du modèle des reformulations, la similarité et l'adaptation sont réifiées au travers des relations `r` entre problèmes, des fonctions d'adaptation \mathcal{A}_r , des chemins de similarité et des chemins d'adaptation. Nous cherchons ici à permettre la représentation de ces notions au sein du Web sémantique, en lien avec les connaissances de domaines diffusées sous la forme d'ontologies en OWL. Il s'agit en quelque sorte de construire une « ontologie » des connaissances utiles au RÀPC, de façon à ce que le processus de RÀPC puisse s'appuyer sur les technologies du Web sémantique et s'appliquer sur les connaissances diffusées par ce biais. Il faut remarquer que mettre en œuvre le modèle des reformulations en OWL, même si celui-ci peut sembler très simple, amène à l'utilisation de fonctionnalités de représentation des connaissances présentes en RDF(S) mais hors des capacités d'OWL DL, comme la réification de propriétés par exemple. Pour cette raison, le modèle présenté ici a précédemment été formalisé en RDF(S) dans [d'Aquin, 2003]. Nous le détaillons dans cette section tel qu'il a été mis en œuvre dans le langage OWL FULL, cumulant les constructions d'OWL DL et la flexibilité de RDF(S).

Le modèle des reformulations formalisé en OWL est schématisé à la figure 3.5. Le détail de cette représentation (son code OWL) se trouve à l'annexe A.

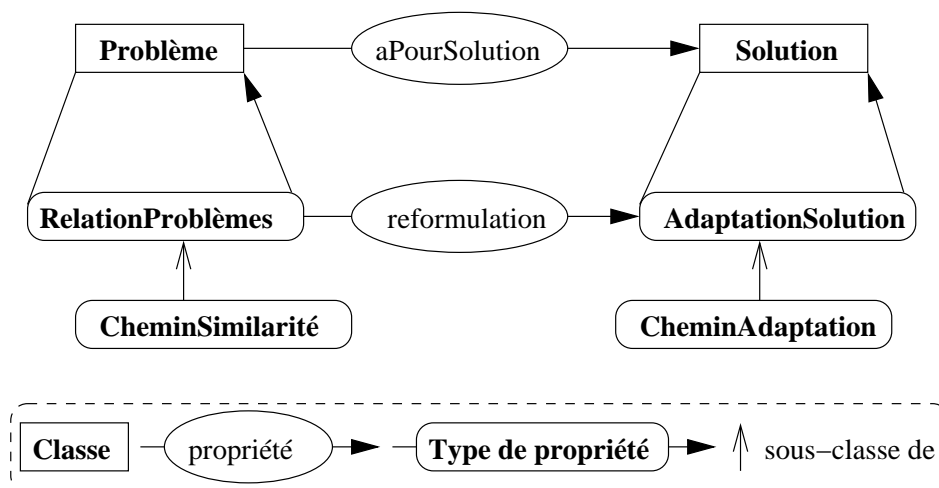


FIG. 3.5 – Schéma du modèle des reformulations en OWL.

Problèmes et solutions. Le RÀPC manipule des cas $(pb, Sol(pb))$ correspondant à des problèmes `pb` associés à leurs solutions `Sol(pb)`. Nous représentons les problèmes par des instances de la classe

Problème et les solutions par des instances de la classe `Solution`. L'association entre un problème et sa (ou ses) solution(s) est réalisée au travers d'une propriété particulière nommée `aPourSolution`. De cette façon, un service de RÀPC reposant sur le modèle présenté en figure 3.5 pourra manipuler des bases de cas représentés en OWL par un ensemble d'assertions de la forme :

$$\text{Problème}(p) \quad \text{Solution}(s) \quad \text{aPourSolution}(p, s)$$

Le problème cible est quant à lui simplement déclaré comme une instance de la classe `Problème` :

$$\text{Problème}(\text{cible})$$

et l'objectif du processus de RÀPC est d'associer à `cible` une ou plusieurs instances de `Solution` par la propriété `aPourSolution`.

Les classes `Problème` et `Solution` jouent de plus un rôle central dans la mise en relation du processus de RÀPC avec les éléments du modèle des connaissances du domaine, représentés au sein d'ontologies en OWL. Dans l'application pour le traitement du cancer du sein par exemple, les problèmes sont des descriptions de patients et les solutions les recommandations de traitements. Les connaissances du domaine ainsi que la base de cas sont ici contenues dans la représentation du référentiel pour le traitement du cancer du sein. Il suffira donc de lier la classe `Patient` du référentiel à `Problème`, la classe `Traitement` à `Solution` et la propriété `recommandation` à la propriété `aPourSolution` au travers des axiomes :

$$\text{Patient} \sqsubseteq \text{Problème} \quad \text{Traitement} \sqsubseteq \text{Solution} \quad \text{recommandation} \sqsubseteq \text{aPourSolution}$$

pour que le processus de RÀPC ait accès aux connaissances contenues dans le référentiel et puisse considérer celui-ci comme une base de cas. L'adaptation des référentiels est néanmoins un cas particulier d'application du RÀPC. Les cas sur lesquels s'appuie le raisonnement ne sont en effet pas des unités spécifiques, représentables par des instances, mais s'apparentent plus à des règles, représentées par des axiomes de la forme $P \sqsubseteq \exists \text{recommandation}.S$, où P et S sont des classes, respectivement sous-classes de `Problème` et `Solution`. Il sera montré dans la suite que le modèle s'adapte aussi bien à des cas spécifiques sous forme d'instances, qu'à des cas généraux (appelés *ossified cases* dans [Riesbeck et Schank, 1989]) comme ceux contenus dans les référentiels.

Similarité et adaptation. Dans le modèle des reformulations, une relation r forme un lien entre deux problèmes et constitue une unité élémentaire de modélisation de la similarité au sein du domaine d'application. Une telle relation est ainsi représentée en OWL par une propriété s'appliquant entre deux classes de problèmes. Par exemple, la relation entre problèmes correspondant au changement de sexe dans l'exemple pour le traitement du cancer du sein (r_2 dans la figure 3.4) correspond en OWL à une propriété s'appliquant à la classe des patients de sexe féminin et prenant ses valeurs dans celle des patients de sexe masculin. OWL permet d'indiquer sur quelle classe s'applique une propriété, son *domaine*, au travers de la propriété `domain` : on note `domain(p, C)` pour déclarer que la classe C représente le domaine de la propriété p . De la même façon, le *co-domaine* de la propriété, la classe dans laquelle elle va prendre ses valeurs, est indiqué par la propriété `range` et la notation associée est de la forme `range(p, C)`. Les deux relations r_1 et r_2 utilisées dans l'exemple de la figure 3.4 pourront ainsi être introduites de la façon suivante :

$$\begin{array}{ll} \text{domain}(r_1, P\text{-LInt}) & \text{range}(r_1, P\text{-LInd}) \\ \text{domain}(r_2, \text{Patiente}) & \text{range}(r_2, \text{Patient-M}) \end{array}$$

les quatre classes utilisées dans cette description étant définies au sein de la représentation du référentiel pour le traitement du cancer du sein par les axiomes :

$$\begin{aligned} P\text{-LInd} &\equiv \text{Patient} \sqcap \exists a\text{Tumeur} . (\exists \text{localisation} . \text{Indéterminée}) \\ P\text{-LInt} &\equiv \text{Patient} \sqcap \exists a\text{Tumeur} . (\exists \text{localisation} . \text{Interne}) \\ \text{Patient-M} &\equiv \text{Patient} \sqcap \exists \text{sexe} . \text{Masculin} \\ \text{Patiente} &\equiv \text{Patient} \sqcap \exists \text{sexe} . \text{Féminin} \end{aligned}$$

De la même façon, les fonctions d'adaptation \mathcal{A}_r sont représentées par des propriétés ayant pour domaines et co-domaines des sous-classes de `Solution`.

Une description précisant le domaine et le co-domaine ne permet néanmoins pas de définir complètement les relations entre problèmes et les fonctions d'adaptation : ces indications, concernant par exemple une relation r , forment des conditions nécessaires mais non suffisantes pour reconnaître que deux problèmes sont liés par r . Une définition plus complète serait dépendante de l'approche utilisée pour l'implémentation des mécanismes sous-jacents à la construction des chemins de similarité et des chemins d'adaptation. Elle sortirait ainsi du cadre général du modèle des reformulations. Il sera montré dans la suite comment ces mécanismes ont été mis en œuvre sur la base d'*opérateurs de transformation*.

Afin de distinguer les relations r entre problèmes et les fonctions d'adaptation \mathcal{A}_r d'autres propriétés, deux *types de propriétés*, `RelationProblèmes` et `AdaptationSolution`, sont introduits au sein du modèle en OWL. Il faut savoir que le modèle de représentation de OWL, son *méta-modèle*, est construit comme une extension de RDF(S) et est manipulable en tant que tel en OWL FULL. En d'autres termes, OWL FULL est réflexif. Un type de propriété peut ainsi être représenté dans ce langage comme une sous-classe de la classe des propriétés dans le méta-modèle d'OWL (`Property`) :

$$\begin{aligned} \text{RelationProblèmes} &\sqsubseteq \text{Property} \\ \text{AdaptationSolution} &\sqsubseteq \text{Property} \end{aligned}$$

Au sein du modèle des connaissances d'adaptation, les relations r_1 et r_2 ainsi que les fonction d'adaptation \mathcal{A}_{r_1} et \mathcal{A}_{r_2} seront alors déclarées au travers des assertions :

$$\begin{array}{ll} \text{RelationProblèmes}(r1) & \text{RelationProblèmes}(r2) \\ \text{AdaptationSolution}(Ar1) & \text{AdaptationSolution}(Ar2) \end{array}$$

c'est-à-dire comme des propriétés, respectivement instances des types de propriétés `RelationProblèmes` et `AdaptationSolution`.

Un chemin de similarité est défini dans le modèle des reformulations comme une séquence de relations entre problèmes r_i , reliant deux problèmes et introduisant des problèmes intermédiaires pb_i . En ce sens, un chemin de similarité peut lui-même être considéré comme une relation entre problèmes. Le type de propriétés `CheminSimilarité` est donc déclaré comme une sous-classe de `RelationProblèmes`. La représentation de l'enchaînement des relations r_i entre problèmes au sein d'un chemin de similarité s'appuie sur la définition récursive d'une liste. Les trois propriétés `relationPrécédente`, `pbi` et `relationSuivante` sont utilisées pour cela. Par exemple le chemin de similarité,

$$\text{source } r_1 \text{ } pb_1 \text{ } r_2 \text{ } pb_2 \text{ } r_3 \text{ } \text{cible}$$

pourra être décrit par les assertions :

$$\begin{array}{lll} r1(\text{source}, pb1) & \text{CheminSimilarité}(cs1) & \text{CheminSimilarité}(cs2) \\ r2(pb1, pb2) & \text{relationPrécédente}(cs1, r1) & \text{relationPrécédente}(cs2, r2) \\ r3(pb2, cible) & pbi(cs1, pb1) & pbi(cs2, pb2) \\ cs1(\text{source}, cible) & \text{relationSuivante}(cs1, cs2) & \text{relationSuivante}(cs2, r3) \\ cs2(pb1, cible) & & \end{array}$$

De la même façon, les chemins d'adaptation sont représentés au travers d'un type de propriétés `CheminAdaptation`, sous-classe de `AdaptationSolution`, et correspondent à des séquences de fonctions d'adaptation.

Reformulations. Une reformulation est un couple (r, \mathcal{A}_r) , liant une relation entre problèmes r à une fonction d'adaptation \mathcal{A}_r . Il semble ainsi naturel de représenter les reformulations au travers d'une propriété de `RelationProblèmes` vers `AdaptationSolution` (faisant une fois de plus appel à la réification de propriétés en OWL FULL). Les deux reformulations (r_1, \mathcal{A}_{r_1}) et (r_2, \mathcal{A}_{r_2}) de l'exemple sont ainsi représentées par les assertions :

$$\text{reformulation}(r_1, \mathcal{A}_{r_1}) \quad \text{reformulation}(r_2, \mathcal{A}_{r_2})$$

De plus, afin de garantir le respect du principe de la remémoration guidée par l'adaptation, c'est-à-dire qu'à chaque relation entre problèmes soit nécessairement associée une fonction d'adaptation, une contrainte est ajoutée à `RelationProblèmes` et exprimée sous la forme :

$$\text{RelationProblèmes} \sqsubseteq \exists \text{reformulation.AdaptationSolution}$$

Le mécanisme de RÀPC aura ainsi la charge d'associer à chacun des chemins de similarité construits, le chemin d'adaptation qui lui correspond.

Le code OWL correspondant au modèle des connaissances d'adaptation sur lequel repose l'exemple de la figure 3.4 se trouve en annexe A.

3.2.3 Raisonnements en OWL pour le RÀPC

De par sa filiation avec les LD, OWL permet l'utilisation de mécanismes d'inférences, principalement la subsomption et l'instanciation, utiles à la mise en œuvre du processus de RÀPC [Salotti et Ventos, 1999, Gómez-Albarrán *et al.*, 1999]. En effet, le test de subsomption est à la base du mécanisme de classification qui organise les classes d'une ontologie en une hiérarchie. Cette hiérarchie de classes est utilisée pour le RÀPC afin de structurer et d'indexer la base de cas, selon les principes décrits dans [Lieber et Napoli, 1996, Napoli *et al.*, 1997]. Plus précisément, l'index d'un problème source `source` est représenté par une classe, noté `idx(source)`, dont `source` est instance. Un index est considéré comme une abstraction d'un problème, conservant les caractéristiques amenant à sa solution. La solution du problème source, `Sol(source)`, est ainsi réutilisable pour toute instance de son index `idx(source)`. En d'autres termes, si un problème cible est reconnu comme une instance de la classe `idx(source)`, index du problème source, alors la solution `Sol(source)` de `source` peut être considérée comme une solution de cible (`Sol(source) = Sol(cible)`). La remémoration consistera ainsi non plus à retrouver un problème source directement, mais à s'appuyer sur le mécanisme d'instanciation pour retrouver l'index du problème source dont la solution sera adaptée. Il s'agit de construire un chemin de similarité entre un problème `pb0` et le problème cible, tel que `source` et `pb0` soient tous deux instances d'une même classe index `idx(source)` de `source` :

$$\text{source} \xrightarrow{\text{isa}} \text{idx}(\text{source}) \xleftarrow{\text{isa}} \text{pb}_0 \text{ r}_1 \text{ pb}_1 \text{ r}_2 \dots \text{pb}_{q-1} \text{ r}_q \text{ cible}$$

où la flèche marquée « isa » signifie « est une instance de ». Dans ce cas, `pb0` étant reconnu par le mécanisme d'instanciation comme une instance de `idx(source)`, sa solution correspondra à celle du problème `source`. Un chemin d'adaptation de la forme suivante sera donc construit :

$$\text{Sol}(\text{source}) = \text{Sol}(\text{pb}_0) \mathcal{A}_{r_1} \text{Sol}(\text{pb}_1) \mathcal{A}_{r_2} \dots \text{Sol}(\text{pb}_{q-1}) \mathcal{A}_{r_q} \text{Sol}(\text{cible})$$

De plus, un index $idx(source)$ amenant nécessairement à la solution $Sol(source)$ du problème $source$ qu'il indexe, celle-ci peut être directement rattachée à $idx(source)$. On définit pour cela un *axiome problème-solution* comme un axiome de la forme : $I \sqsubseteq \exists a PourSolution.S$, où I est une classe index, correspondant à $idx(source)$, et S est une classe de solutions, correspondant à la classe dont $Sol(source)$ est instance. De cette façon, au travers du mécanisme de complétion d'instances, toute instance de la classe index I sera automatiquement associée à une solution de la classe S . C'est grâce à ce principe que les « règles » des référentiels KASIMIR pourront être utilisées comme des cas pour l'adaptation. En effet, le référentiel pour le traitement du cancer du sein contient un ensemble d'axiomes de la forme : $P \sqsubseteq \exists recommandation.T$, où P est une sous-classe de *Patient*, donc de *Problème*, et T est une sous-classe de *Traitement*, donc de *Solution*. Ainsi, le problème source utilisé dans la figure 3.4 (page 45) est représenté par une classe index I et relié à la classe S des traitements recommandés de la façon suivante :

$$I \equiv Patiente \sqcap \exists age.supe20 \sqcap \exists age.infe75$$

$$\sqcap \exists aTumeur.(TumeurNonMultifocale \sqcap \exists localisation.Interne \sqcap \exists taille.supe4$$

$$\sqcap \exists taille.infe7)$$

$$S \equiv MastectomieEtOvariectomie$$

$$I \sqsubseteq \exists recommandation.S$$

MastectomieEtOvariectomie étant la classe des traitements composés d'une mastectomie et d'une ovariectomie. En s'appuyant sur les relations entre problèmes r_1 et r_2 , la remémoration utilisant l'instanciation construira le chemin de similarité :

$$I \xleftarrow{isa} pb_0 \ r_1 \ pb_1 \ r_2 \ cible$$

où *cible* est une instance de *Patient* (*Problème*) possédant les propriétés correspondant aux caractéristiques du patient à traiter (sexe masculin, localisation de la tumeur, etc.). pb_0 et pb_1 sont eux aussi des instances de *Patient*, dont les propriétés diffèrent de *cible*, selon les relations r_1 et r_2 (pb_1 est de sexe féminin, pb_0 a une tumeur interne). pb_0 est alors reconnu comme une instance de la classe index I ($I \xleftarrow{isa} pb_0$) et est ainsi associé, par le mécanisme de complétion d'instance, à une solution de la classe S ($S \xleftarrow{isa} Sol(pb_0)$). Finalement, l'adaptation de cette solution se fera au travers d'un chemin d'adaptation de la forme :

$$S \xleftarrow{isa} Sol(pb_0) \ \mathcal{A}_{r_1} \ Sol(pb_1) \ \mathcal{A}_{r_2} \ Sol(cible)$$

où $Sol(pb_1)$ est une copie de $Sol(pb_0)$ (application de \mathcal{A}_{r_1}) et $Sol(cible)$ est construite en substituant l'élément correspondant à l'ovariectomie de $Sol(pb_1)$ par un autre, correspondant à une cure de Tamoxifène (application de \mathcal{A}_{r_2}) : avec *MastectomieEtTamoxifène* la classe des traitements composés d'une mastectomie et d'une cure de Tamoxifène, $Sol(cible) \xrightarrow{isa} MastectomieEtTamoxifène$.

Le mécanisme de complétion d'instances permet par ailleurs de compléter les informations connues sur le problème *cible* en fonction des connaissances du domaine. Ce mécanisme est ainsi particulièrement utile pour la mise en œuvre de l'étape d'élaboration du RÀPC [Gómez-Albarrán *et al.*, 1999].

Les algorithmes et l'architecture sur lesquels repose l'implémentation des étapes de remémoration et d'adaptation selon le modèle des reformulations sont présentés dans la section suivante.

3.3 Un service de raisonnement à partir de cas pour le Web sémantique

Le modèle des reformulations permet de représenter les connaissances utiles au processus de RÀPC, les connaissances d'adaptation, en lien avec les connaissances du domaine contenues dans des ontologies

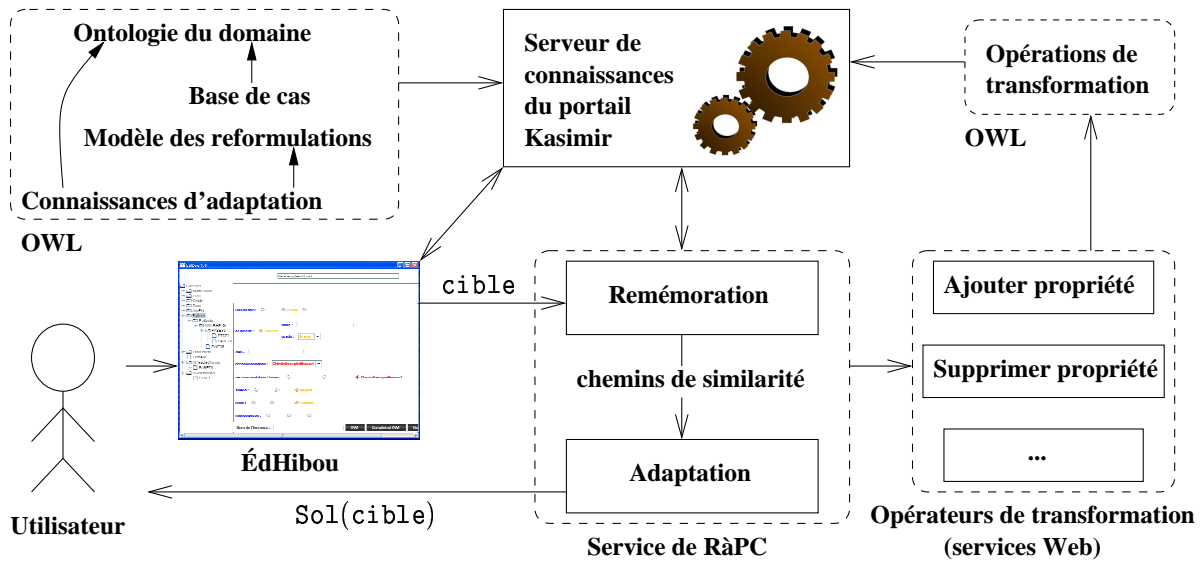


FIG. 3.6 – Architecture du service de RÀPC.

en OWL. Cette section détaille le développement d'un service de RÀPC réalisant les étapes de remémoration et d'adaptation selon ce modèle. Le terme *service* est utilisé ici dans le sens où le processus de RÀPC est implémenté sous la forme d'un service Web, exploitant les connaissances diffusées au travers du Web sémantique. Ce service est voué à être intégré au portail sémantique KASIMIR, décrit dans le chapitre 2, pour être appliqué à l'aide à la décision dans le cadre de l'adaptation des référentiels.

3.3.1 Architecture et algorithmes

L'architecture sur laquelle repose le service de RÀPC est décrite à la figure 3.6. Celle-ci comporte deux modules principaux. Celui en charge de la remémoration prend en entrée un problème *cible* et retourne un ensemble de chemins de similarité liant *cible* à des problèmes sources de la base de cas. Le module d'adaptation s'appuie sur ces chemins de similarité pour construire une ou plusieurs solutions pour le problème *cible*. Ces deux modules manipulent des connaissances formalisées en OWL et reposent sur les mécanismes d'inférences associés à ce langage. Pour cela, ils font appel au serveur de connaissances développé dans le cadre du portail sémantique KASIMIR (voir section 2.4). Les connaissances d'adaptation en entrée du système sont décrites selon le modèle des reformulations présenté à la section précédente, sur la base d'une ontologie du domaine d'application.

Les mécanismes sous-jacents à la détection des relations entre problèmes et à l'application des fonctions d'adaptation sont mis en œuvre grâce à des *opérations de transformation*, décrites elles aussi selon un modèle en OWL (qui sera détaillé en section 3.3.2). Ces opérations permettent de transformer des individus sur la base d'opérateurs comme l'ajout et la suppression de propriétés. Ces opérateurs étant implémentés et décrits sous la forme de services Web, de nouveaux opérateurs pourront facilement être ajoutés pour les besoins d'une application particulière.

Le logiciel ÉDHIBOU (voir section 2.5.1) est utilisé pour l'étape d'élaboration du processus de RÀPC. Il permet en effet de s'appuyer sur les connaissances du domaine comme cadre à la description du problème *cible* à résoudre, ainsi que pour compléter cette description sur la base des inférences réalisées au sein du serveur de connaissances.

Remémoration

Les détails de l'algorithme implémenté pour le module de remémoration sont donnés en annexe A. Celui-ci est conçu comme une simplification de l'algorithme dit de *classification élastique* présenté dans [Lieber, 1997] et [Lieber, 2002]. Il prend en entrée un problème `cible`, décrit sous la forme d'un individu en OWL et instance de la classe `Problème`. Son objectif est d'établir un ensemble de chemins de similarité liant `cible` à des classes `idx(source)` de problèmes sources de la base de cas. Le principe de l'algorithme de classification élastique est présenté dans [Lieber, 1997] par : « On cherche les classes `idx(source)`, une déformation de `idx(source)` et une déformation de l'entité `cible`, telles que `idx(source)` déformée subsume `cible` déformée », les déformations étant représentées par les relations r_i du chemin de similarité. L'algorithme présenté ici consiste en une simplification de ce principe, dans le sens où seul `cible` est « déformé ». Il peut être vu comme un parcours en profondeur d'abord d'un graphe de problèmes liés par des relations r entre problèmes. Les relations r_i et les problèmes intermédiaires pb_i à suivre à chaque étape sont générés grâce à la fonction `génèrePlusProchesProblèmes`. Celle-ci est implémentée à l'aide d'opérations de transformation, indépendantes du modèle des reformulations. Elle sera ainsi détaillée dans la section suivante qui est consacrée à ce point particulier.

Il peut par ailleurs être remarqué que l'algorithme mis en œuvre renvoie l'ensemble des chemins de similarité pouvant être construits à partir de `cible` (hormis ceux contenant plusieurs problèmes de représentations équivalentes, pour éviter les bouclages), et non seulement le plus court. Cela permet à l'adaptation de construire toutes les solutions possibles, laissant le choix à l'utilisateur parmi celles-ci.

Adaptation

De part la nature du modèle des reformulations, conçu selon les principes de la remémoration guidée par l'adaptation, l'algorithme utilisé pour l'adaptation est assez simple. Les détails le concernant se trouvent à l'annexe A. Il prend en entrée un ensemble de chemins de similarité, liant le problème `cible` à des classes `idx(source)`, index de problèmes sources `source`. Son rôle est alors de construire pour chacun de ces chemins de similarité, un chemin d'adaptation partant de la solution `Sol(source)` liée à `idx(source)`, pour aboutir à la solution `Sol(cible)` du problème `cible`. Il s'agit pour cela d'appliquer pour chaque relation r_i du chemin de similarité, la fonction d'adaptation \mathcal{A}_{r_i} correspondante. L'algorithme fait ainsi appel à la fonction `appliquerAdaptation`, chargée de créer une nouvelle solution à partir d'une solution existante et d'une fonction d'adaptation. La fonction `appliquerAdaptation` est elle aussi implémentée sur la base des opérations de transformation détaillées dans la section suivante.

3.3.2 Opérations de transformation et services Web pour la mise en œuvre des reformulations

Le modèle des reformulations est un cadre très général pour la représentation de connaissances d'adaptation. Il doit ainsi être spécialisé afin de permettre l'implémentation des opérations sous-jacentes à la construction des chemins de similarité et d'adaptation. Il s'agit de mettre en œuvre une approche particulière du RAPC sur la base du modèle des reformulations. L'approche adoptée ici repose sur des *opérations de transformation* et se veut réutilisable pour de nombreuses applications. L'objectif est de permettre une description modulaire et extensible des opérations permettant de détecter les relations r entre problèmes et d'appliquer les fonctions d'adaptation \mathcal{A}_r (respectivement `génèrePlusProchesProblèmes` et `appliquerAdaptation` des algorithmes précédents). Pour cette raison, nous proposons un modèle en OWL pour la description d'opérations de transformation. Les opérateurs sur lesquels reposent celles-ci sont décrits et implémentés sous la forme de services Web. L'inclusion de nouveaux opérateurs, utiles à une application particulière, est de cette façon simplifiée.

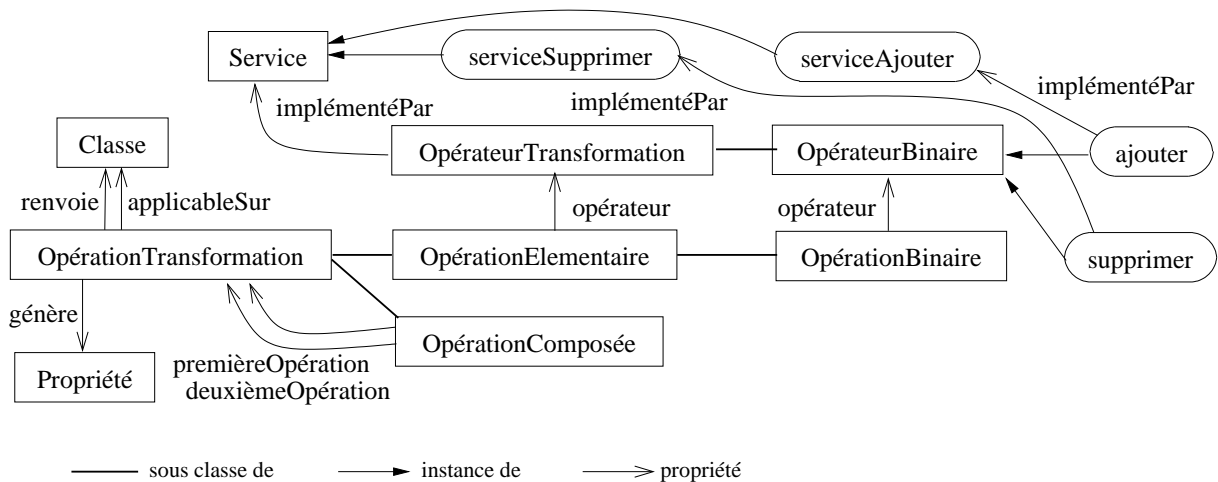


FIG. 3.7 – Modèle OWL pour la représentation d’opérations de transformation.

Le modèle OWL utilisé pour représenter les opérations de transformation est schématisé en figure 3.7. Une opération de transformation prend en entrée un individu origine et génère un nouvel individu transformé, correspondant à une transformation de origine selon un ou plusieurs *opérateurs de transformation*. Une telle opération sera représentée en OWL par une instance de la classe `OpérationTransformation` dont les propriétés sont :

- `applicableSur` indique la classe OWL dont les instances pourront être utilisées en entrée de l’opération.
- `renvoie` indique la classe de l’individu généré.
- `génère` indique la propriété OWL liant l’individu d’origine à celui transformé.

Par exemple, si une opération `o` est déclarée selon les assertions suivantes :

`OpérationTransformation(o) applicableSur(o, C1) renvoie(o, C2) génère(o, p)`

alors l’application de `o` sur une instance origine de `C1` donnera un individu transformé, respectant au minimum les assertions :

`C2(transformé) p(origine, transformé)`

Il peut être remarqué que le fait d’associer aux individus représentant les opérations de transformation des classes et de propriétés de OWL relève de possibilités d’OWL FULL que ne possède pas OWL DL. Dans le cadre du RÀPC, la propriété `p` générée correspondra soit à une relation `r` entre problèmes, soit à une fonction d’adaptation \mathcal{A}_r .

La classe `OpérationTransformation` possède les deux sous-classes suivantes :

- `OpérationComposée` correspond à la composition de deux opérations de transformation, indiquées par les propriétés `premièreOpération` et `deuxièmeOpération`.
- Une instance de `OpérationÉlémentaire` correspond à une opération élémentaire s’appuyant sur un opérateur de transformation, indiqué par la propriété `opérateur`.

Une opération élémentaire utilisant un opérateur binaire (classe `OpérateurBinaire`), c’est-à-dire nécessitant un opérande en plus de l’individu origine à transformer, est une opération binaire (classe `OpérationBinaire`). La valeur de l’opérande supplémentaire sera alors indiquée au travers d’une propriété opérande.

Un opérateur de transformation est représenté par une instance de `OpérateurTransformation`. Il est associé au service Web qui l'implémente au travers d'une propriété `service`. `OpérateurBinaire` est une sous-classe de `OpérateurTransformation` pour représenter les opérateurs binaires. À l'heure actuelle, deux services sont implémentés, pour les deux opérateurs `ajouter` et `supprimer`. Ceux-ci permettent respectivement d'ajouter une propriété à `origine` et d'en supprimer une. Ce sont tous deux des opérateurs binaires, dont la deuxième opérande correspond à un individu, contenant les informations sur la propriété et la valeur à ajouter ou supprimer.

Le modèle OWL utilisé pour la représentation des opérations de transformation schématisé en figure 3.7 est détaillé dans l'annexe A.

Exemple d'utilisation des opérations de transformations. La relation r_2 utilisée dans la figure 3.4 est supposée rendre compte d'une modification de la valeur de la propriété `sexe` entre `cible` et `pb1`. Elle pourra ainsi être mise en œuvre au travers d'une opération de transformation, appelée par exemple `remplaceFH` et décrite par les assertions :

<code>OpérationComposée(remplaceHF)</code>	<code>OpérationBinaire(supprimeH)</code>
<code>génère(remplaceHF, r2)</code>	<code>opérateur(supprimeH, supprimer)</code>
<code>applicableSur(remplaceHF, Patient-M)</code>	<code>opérande(supprimeH, sexeMasculin)</code>
<code>renvoie(remplaceHF, Patiente)</code>	
	<code>OpérationBinaire(ajouteF)</code>
<code>premièreOpération(remplaceHF, supprimeH)</code>	<code>opérateur(ajouteF, ajouter)</code>
<code>deuxièmeOpération(remplaceHF, ajouteF)</code>	<code>opérande(ajouteF, sexeFéminin)</code>

Selon ces assertions, `remplaceHF` est une opération composée de deux autres. La première (`supprimeH`) supprime la propriété `sexe` dont la valeur est une instance de `Masculin`. La deuxième (`ajouteF`) ajoute la propriété `sexe` avec une instance de `Féminin` pour valeur. `sexeMasculin` et `sexeFéminin` sont deux individus utilisés pour représenter les opérandes de `supprimeH` et `ajouteF`. Leur forme dépend des services chargés d'implémenter les opérateurs `ajouter` et `supprimer`. `remplaceHF` prend donc une instance de `Patient-M` en entrée et renvoie une instance de `Patiente` (rappelons que `Patient-M` \equiv `Patient` \sqcap \exists `sexe.Masculin` et que `Patiente` \equiv `Patient` \sqcap \exists `sexe.Féminin`).

On peut remarquer que la construction du chemin de similarité faisant appel aux opérations de transformation devra s'appuyer sur le mécanisme d'instanciation pour retrouver, parmi les opérations existantes, lesquelles sont applicables. Ici, `remplaceHF` est applicable à `cible` du fait que celui-ci est reconnu comme une instance de `Patient-M`. Cette opération sera donc utilisée pour transformer `cible` et créer le problème intermédiaire `pb1`. Cette transformation nécessitera de plus l'appel aux services Web implémentant les opérateurs `ajouter` et `supprimer`. D'autres opérations de transformation seront ensuite utilisées pour la relation r_1 , ainsi que pour l'application des fonctions d'adaptation \mathcal{A}_{r_1} et \mathcal{A}_{r_2} . \mathcal{A}_{r_2} correspond par exemple à une opération composée supprimant l'ovariectomie et ajoutant le traitement par Tamoxifène.

3.3.3 Notes sur l'implémentation

À l'heure actuelle, seul un prototype du service de RÀPC présenté ici a été implémenté, de nombreux points restant à améliorer au sein de l'architecture proposée. Un point critique concerne en particulier la complexité de l'algorithme de remémoration. Celui-ci fait en effet appel de nombreuses fois (à chaque problème généré) au mécanisme d'instanciation de OWL, particulièrement gourmand en temps de calcul

(il peut prendre à lui seul plusieurs minutes). Différentes approches pourraient être envisagées pour améliorer cela. En particulier, l'algorithme proposé pour la remémoration construit l'ensemble des chemins de similarité possibles amenant à cible, sans favoriser les plus courts ou les *moins coûteux en adaptation*. Transformer la recherche en profondeur utilisée dans cet algorithme en une recherche de type A* permettrait par exemple de ne construire que les chemins les plus courts ou de prendre en compte une indication sur le coût des fonctions d'adaptation utilisées lors de la construction du chemin d'adaptation. Par ailleurs, à chaque résolution d'un problème, le service de RÀPC génère une représentation en OWL des chemins de similarité et d'adaptation utilisés, qui sont intégrés aux connaissances d'adaptation. Ceux-ci pourraient être réutilisés, au moins en partie, dans des épisodes de résolution de problèmes ultérieurs, évitant ainsi d'avoir à effectuer de nouveau les inférences déjà réalisées. Notons de plus que les chemins de similarité et d'adaptation mis en œuvre fournissent une base permettant de générer des explications significatives des résultats du processus de RÀPC pour l'utilisateur. Une interface permettant de visualiser ces résultats et les explications associées reste encore à définir.

Un des points intéressants dans l'architecture proposée est qu'elle est applicable et extensible pour différents domaines d'application et différentes approches du RÀPC. Le modèle des reformulations mis en œuvre permet en effet d'utiliser, au sein d'un processus de RÀPC générique, des connaissances et des cas formalisés dans le langage de représentation standard du Web sémantique. De plus, le modèle des opérations de transformation est prévu pour être étendu par de nouveaux opérateurs, mis en œuvre sous la forme de services Web. Dans l'implémentation actuelle, la représentation de ces services est assez simple. Il pourrait néanmoins être intéressant de s'appuyer sur les efforts actuels autour de la description sémantique de services Web, en utilisant par exemple les *profiles de services* d'OWL-S [Martin *et al.*, 2004].

3.4 Le RÀPC comme mécanisme de raisonnement pour le Web sémantique

Nous avons décrit dans ce chapitre le développement d'un service de RÀPC permettant d'exploiter les connaissances diffusées sur le Web sémantique dans un mécanisme de raisonnement par analogie. L'objectif premier est d'intégrer ce service au sein du portail sémantique KASIMIR, dans le but d'appliquer le RÀPC à l'aide à la décision pour l'adaptation des référentiels. Celui-ci est néanmoins conçu de façon générique, indépendamment de l'application en cancérologie, afin d'être exploitable dans d'autres domaines d'application. Contrairement aux approches déductives habituellement préconisées par le Web sémantique, le RÀPC est reconnu comme bien adapté dans des domaines incomplètement formalisés [Kolodner, 1993]. Dans un environnement tel que celui du Web, naturellement ouvert, complexe et peu structuré, ce type de raisonnement pourra s'avérer profitable à de nombreuses applications. La formalisation des connaissances utiles au RÀPC selon les technologies standard du Web sémantique offre de plus un cadre à la diffusion et à la réutilisation de connaissances d'adaptation.

Certains points concernant l'utilisation du RÀPC dans l'environnement du Web sémantique restent cependant à étudier. Dans l'approche proposée, le RÀPC repose notamment sur différents types de connaissances. L'acquisition et la construction d'ontologies sont depuis longtemps des sujets d'études primordiaux en ingénierie et en gestion des connaissances [Dieng *et al.*, 2000, Schreiber *et al.*, 1999]. Ces recherches sont naturellement mises en application dans le cadre du Web sémantique. Mais, peu de travaux se sont intéressés à l'acquisition de connaissances d'adaptation [Lieber *et al.*, 2004]. La conception de méthodes et d'outils pour l'acquisition et la construction de ce type de connaissances, spécifiques au RÀPC, fait ainsi partie des études en cours dans le cadre du projet KASIMIR. Ce point particulier sera détaillé au chapitre 6.

Dans des domaines complexes, tels la cancérologie, les connaissances sont souvent considérées et représentées selon plusieurs points de vue. Cela est d'autant plus vrai dans l'environnement du Web

sémantique où peuvent coexister plusieurs ontologies traitant d'un même domaine. Les notions sur lesquelles s'appuie le RÀPC, telles que la similarité, sont de plus particulièrement dépendantes du point de vue adopté. Un chirurgien ne considérera pas les mêmes patients comme étant similaires qu'un radiothérapeute par exemple. La représentation explicite de ces multiples points de vue et des relations qu'ils entretiennent doit ainsi être profitable au RÀPC. Cela permet de plus de distribuer le mécanisme de RÀPC au sein de différents points de vue, de manière décentralisée, améliorant de cette manière la capacité de ce mécanisme à « passer à l'échelle ». Le chapitre suivant s'intéresse pour cette raison à la notion de points de vue, sous l'angle de la représentation des connaissances. Puis la mise en œuvre d'un mécanisme de RÀPC décentralisé, exploitant une représentation multi-points de vue des connaissances, est détaillée au chapitre 5.

Chapitre 4

Représentation explicite de points de vue

Lorsqu'on parle du *point de vue* de quelqu'un dans le langage courant, il s'agit généralement de la *position* que prend cette personne concernant un sujet particulier, comme dans l'expression « le point de vue de Mathieu sur le Web sémantique » par exemple. Ces deux termes (point de vue et position) relèvent d'ailleurs de la même image : celle du lieu où se trouve la personne vis-à-vis du sujet et ainsi de l'angle selon lequel elle le considère. Mais, au-delà de cette définition très générale, trois façons de voir les points de vue, c'est-à-dire trois types de points de vue différents, peuvent être envisagés :

- pdv1. Un point de vue est une *sélection* parmi les caractéristiques des objets considérés. En effet, selon la personne et le contexte dans lequel elle se trouve, certaines de ces caractéristiques seront considérées comme pertinentes, et d'autres non. Par exemple, du point de vue d'un ébéniste, une chaise est composée d'un certain bois et est assemblée selon une certaine technique. Un décorateur s'intéressera plus à la couleur de la chaise, à son style, etc. Un point de vue est donc ici considéré comme un *filtre* sur les caractéristiques des objets, ne conservant que les plus pertinentes selon le point de vue.
- pdv2. Un point de vue est une *représentation* particulière du monde. Différentes définitions, plus ou moins utiles selon le contexte, peuvent être données à un même concept ou à une même notion. Par exemple, un triangle peut être défini au moins de deux façons différentes, comme un polygone à trois angles ou comme un polygone à trois cotés. En ce sens, un point de vue peut être une façon particulière de représenter et de définir les concepts d'un domaine.
- pdv3. Un point de vue est une *opinion* sur un sujet particulier. En effet, le point de vue d'une personne peut indiquer ce que celle-ci croit être vrai sur un sujet donné. Il peut en ce sens être en contradiction avec un autre point de vue. Par exemple, du point de vue d'un Français, le meilleur vin est le vin français. Ce point de vue pourra ne pas être partagé par un Italien. Dans cet exemple, le Français et l'Italien ont deux points de vue différents, dans le sens où leurs opinions s'opposent.

La représentation des connaissances consiste à modéliser et à formaliser les connaissances relatives à un domaine. Dans le cadre de la représentation d'ontologies, il est de coutume de considérer qu'une telle représentation se doit d'être la plus générale et la plus indépendante du contexte possible [Charlet, 2002]. Elle reste néanmoins nécessairement relative à un point de vue particulier. En effet, construire une base de connaissances ou une ontologie passe par la sélection des objets et des concepts qu'il est pertinent de représenter, ainsi que par une sélection, parmi les caractéristiques des objets, de celles jugées utiles. Une ontologie relève en ce sens des trois définitions d'un point de vue. Il s'agit d'une représentation particulière du monde (pdv2), qui délimite des caractéristiques pertinentes pour les objets représentés (pdv1) et qui indique ce qui doit être considéré comme vrai lors de son utilisation (pdv3).

Le projet KASIMIR s'intéresse à la représentation de connaissances pour l'aide à la décision en cancérologie, c'est-à-dire principalement des connaissances contenues dans les référentiels et des connais-

sances utiles à leur adaptation. Ces connaissances sont structurées selon les différentes disciplines impliquées dans la prise en charge d'un patient (chimiothérapie, chirurgie, hormonothérapie, radiothérapie), chacune s'appuyant sur sa propre représentation du domaine et considérant les patients selon ses propres intérêts. En d'autres termes, chaque discipline apporte son propre point de vue lors de la prise de décision et cette multiplicité des points de vue doit être prise en compte au sein de la représentation. Pour cette raison, nous nous intéressons dans ce chapitre à la représentation explicite de points de vue, c'est-à-dire à l'intégration et la coordination de plusieurs représentations alternatives des objets, des concepts ou plus généralement, des connaissances relatives à un domaine.

4.1 Les points de vue en représentation des connaissances

Il existe généralement plusieurs façons d'appréhender les éléments de connaissances relatifs à un domaine, c'est-à-dire différents points de vue selon lesquels ces connaissances peuvent être représentées. Par exemple, selon la tâche que l'on cherche à accomplir, on pourra considérer une voiture de différentes manières. On s'intéressera à sa consommation et à son prix d'achat d'un point de vue économique, au nombre de places et à la taille du coffre si c'est l'aspect fonctionnel qui l'emporte, ou encore à sa couleur et à sa forme s'il est plus opportun de se focaliser sur l'esthétique de la voiture. Les travaux en représentation des connaissances se sont ainsi souvent intéressés à faire coexister plusieurs représentations alternatives au sein d'une même base de connaissances, en intégrant de façon explicite la notion de point de vue. Cette section a pour rôle de donner un aperçu de ces systèmes et des langages de représentation multi-points de vue qu'ils implémentent.

4.1.1 Points de vue en représentation des connaissances par objets

C'est en représentation des connaissances par objets (RCO, [Ducournau *et al.*, 1998]) que le plus grand nombre de systèmes de représentation de connaissances multi-points de vue a été développé. La représentation des connaissances par objets repose généralement sur la notion de *classe* pour représenter des ensembles d'objets. Les classes sont associées à des *attributs* et organisées en une hiérarchie. Une classe dans cette hiérarchie *hérite* des attributs de tous ses parents. Seuls certains systèmes autorisent qu'une classe ait plusieurs parents directs (héritage multiple). Un *objet* est instance d'une ou plusieurs classes et hérite des attributs de cette ou de ces classes.

KRL (*Knowledge representation language* [Bobrow et Winograd, 1977]) est sûrement le premier système de RCO permettant la représentation de points de vue (appelés *perspectives* en KRL). Comme la plupart des systèmes ultérieurs, la notion de point de vue est intégrée en permettant qu'un objet soit associé à plusieurs représentations différenciées, au travers de plusieurs classes. Les objets de KRL (*individuals*) peuvent ainsi appartenir à plusieurs classes, chacune regroupant les attributs pertinents selon un point de vue particulier. En ce sens, la représentation multi-points de vue dans ce langage relève principalement de l'*instanciation multiple*. On pourra déclarer en KRL une expression traduisant la phrase :

(La Fiat Uno de Mary) est-une (Vieille Fiat de couleur bleue) et est-une (Voiture au super qui coûte peu mais consomme beaucoup)

pour représenter deux points de vue différents (esthétique et économique) sur l'objet *Fiat Uno de Mary*.

Dans le système TROEPS [Mariño, 1993], un *concept* représente, au même titre qu'une classe, un ensemble d'objets. Chaque concept est associé à un ensemble de points de vue. Les *points de vue des concepts* sont de hiérarchies de classes au sens de la RCO classique. Ainsi, TROEPS ne considère pas, comme il est habituel de le faire en RCO et en LD, les termes classe et concept comme des synonymes.

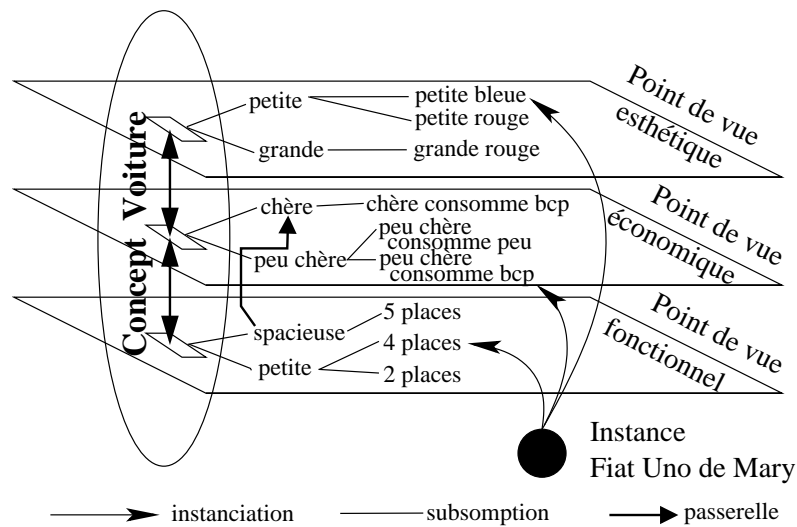


FIG. 4.1 – Un exemple de représentation en TROEPS.

Par principe, TROEPS se limite à des hiérarchies d'héritage simple, c'est-à-dire que chaque classe n'a qu'un seul ascendant direct. Les objets en TROEPS sont instances des concepts et toute instance d'un concept doit nécessairement être directement instance d'une (et une seule) classe de chacun des points de vue du concept. Par exemple, la figure 4.1 montre la représentation du concept de voiture, vu sous les trois points de vue *esthétique*, *économique* et *fonctionnel*. Le point de vue *esthétique* pourra s'intéresser à des attributs comme le modèle, la taille ou la couleur de la voiture, le point de vue *économique* au prix d'achat et à la consommation en carburant par exemple, et le point de vue *fonctionnel* au nombre de places et à la taille du coffre. La *Fiat Uno de Mary* est ainsi décrite comme une instance de voiture, représentée par une classe de chacun des trois points de vue sur les voitures.

TROEPS permet de définir des liens, appelés *passerelles*, entre les points de vue. Une passerelle est une relation entre deux classes de deux points de vue différents d'un même concept. Elle est définie par une classe d'origine et une classe de destination, pour indiquer que toute instance de la classe d'origine est nécessairement instance de la classe de destination. En ce sens, une passerelle correspond à une implication. Elle se rapproche par ailleurs de la relation de *coréférence* définie dans [Ferber, 1989]. Par exemple, on peut spécifier par une passerelle que les voitures *spacieuses* du point de vue *fonctionnel* sont *chères* du point de vue *économique* (voir figure 4.1). Des passerelles sont implicitement déclarées entre les classes racines des hiérarchies qui constituent les points de vue d'un concept. Ces passerelles permettent de contraindre à ce que toute instance d'un concept soit instance d'une classe de chacun des points de vue du concept.

Dans le cadre de la représentation de points de vue, ROME et FROME [Dekker, 1994] distinguent deux relations différentes pour associer un objet à une classe : le lien d'*instanciation* et le lien de *représentation*. Le lien d'instanciation relie l'objet à la classe dont il est instance. Le lien de représentation associe l'objet aux classes qui le représentent dans différents points de vue. Une relation d'*exclusion mutuelle* entre classes permet de spécifier que deux classes ne peuvent représenter un même objet, c'est-à-dire qu'il s'agit de deux points de vue incompatibles. La figure 4.2 montre un exemple de représentation multi-points de vue avec FROME, à comparer avec la même représentation en KRL et en TROEPS.

Les *perspectives* de KRL peuvent être vues comme une sélection des propriétés des objets, en s'ap-

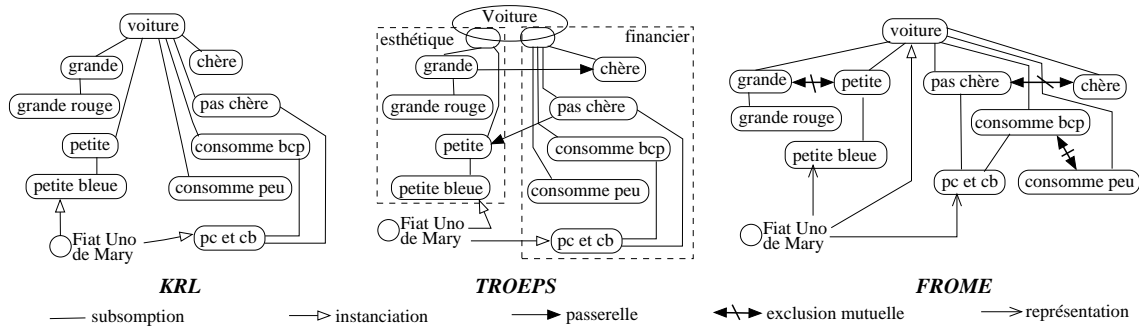


FIG. 4.2 – Points de vue en RCO.

puyant sur l'héritage. En ce sens, elles relèvent de la première définition des points de vue (pdv1 page 57) : le point de vue en tant que filtre sur les caractéristiques des objets. La représentation des points de vue en FROME étant assez similaire à celle de KRL, elle se concentre sur le même aspect. TROEPS permet quant à lui à la fois la sélection des caractéristiques des objets par rapport au point de vue et de disposer de plusieurs représentations pour les concepts du domaine, sous la forme des hiérarchies de classes qui composent les points de vue. En ce sens, il relève aussi de la deuxième façon de voir un point de vue, c'est-à-dire comme une représentation particulière du monde.

4.1.2 Points de vue en logique

Parmi les premiers travaux s'intéressant à la notion de point de vue, [Hautamäki, 1986] vise à la définir dans un formalisme logique. Cette étude réalisée dans le cadre de la philosophie est notamment mise en relation avec le domaine de la représentation des connaissances. L'auteur part de l'idée qu'un point de vue correspond à une vue partielle sur un objet, c'est-à-dire à une représentation incomplète des informations le concernant. Il définit ainsi la syntaxe et la sémantique d'une logique permettant la manipulation de points de vue en tant que sélections parmi les caractéristiques des objets. La définition proposée par [Hautamäki, 1986] relève donc exclusivement de la première des trois façons de considérer les points de vue évoquées page 57.

Dans [Attardi et Simi, 1995], Giuseppe Attardi et Maria Simi suivent l'idée que différents points de vue expriment différentes visions relatives de la « vérité », au sens logique du terme. Ils formalisent cette notion dans une extension de la logique des prédicats. Un point de vue est alors défini comme un ensemble d'assertions de la logique, assimilé à une « micro-théorie ». L'extension consiste en un métalangage et un ensemble de règles d'inférences permettant de raisonner *dans et sur* les points de vue. Un point de vue s'exprime dans ce métalangage au travers de prédicats de la forme $in('A', pdv)$, signifiant intuitivement « A est vraie dans le point de vue pdv », avec A une formule de la logique et pdv un symbole correspondant à un point de vue. De plus, la formule A de l'expression précédente pouvant elle-même contenir des prédicats in , il est possible au sein d'un point de vue de raisonner sur le contenu d'autres points de vue. Par exemple, la propriété dite d'*introspection positive* s'exprime dans ce cadre par :

$$in('A', vp) \implies in('in('A', vp)', vp)$$

De plus, représenter les points de vue par des ensembles d'assertions permet d'en faciliter la manipulation. On peut en effet très simplement grâce à cela définir des opérateurs tels que l'union et l'intersection de points de vue, ainsi qu'une relation de subsomption (dénotée par \succ) permettant de hiérarchiser les

points de vue :

$$pdv2 \succcurlyeq pdv1 \Leftrightarrow \forall A, in('A', pdv1) \Rightarrow in('A', pdv2)$$

Un point de vue étant défini ici comme une vision relative de la vérité, ces travaux s'intéressent principalement aux points de vue en tant qu'opinions (pdv3 page 57).

4.1.3 Points de vue et graphes

Le système VIEWS [Davis, 1987] permet la représentation d'*objets structurés* sous la forme de *vues*. Une vue est un réseau de *parties* (les nœuds du réseau), de *relations* et de *contraintes*. La représentation multi-points de vue, ou multi-perspectives, en VIEWS s'appuie sur la possibilité de décrire plusieurs vues contenant des éléments communs (identifiés par leurs noms). Il est de plus possible d'établir des relations entre les vues. Une vue délimitant les relations que possède un objet particulier, elle peut être considérée comme un point de vue au sens de la première définition. Chaque vue décrivant une structure particulière pour ces objets, elle constitue aussi une représentation particulière du monde, en accord avec la deuxième définition des points de vue.

Dans [Ribière et Dieng, 1997], une extension du formalisme des *graphes conceptuels* (GC) pour la représentation explicite de points de vue est présentée. Une correspondance simple peut être établie entre les GC et la RCO. Ce qui est appelé en RCO un objet correspond à un concept en GC et les classes correspondent aux *types de concepts*. En GC, les concepts forment les nœuds de graphes, dont les arcs sont des *relations*, qui jouent ainsi le rôle des attributs de la RCO. Les types de concepts sont organisés en une hiérarchie, un type de concept étant considéré comme un *sous-type* d'un type de concept plus général. Les points de vue sont intégrés à ce formalisme en permettant d'« étiqueter » une relation de sous-typage entre deux types de concepts selon un point de vue. Par exemple, il peut être établi par ce biais que *VoitureChère* est un sous-type de *Voiture* selon le point de vue économique et que *VoitureSpacieuse* est un sous-type de *Voiture* selon le point de vue fonctionnel. En ce sens, la représentation de points de vue dans ce formalisme est proche de ce qui est réalisé par le système TROEPS (duquel [Ribière et Dieng, 1997] s'inspire explicitement). Elle relève donc des mêmes définitions des points de vue (pdv1 et pdv2 page 57). De plus, des relations d'*inclusion* et d'*exclusion* peuvent être déclarées entre types de concepts, jouant les mêmes rôles que les passerelles de TROEPS et les relations d'exclusions mutuelles de FROME.

4.1.4 Points de vue et contextes

La notion de contexte, telle que considérée en représentation des connaissances, est très proche de la notion de point de vue. En effet, [Benerecetti *et al.*, 2001] indique que dans une représentation multi-contextes, deux contextes se distinguent soit parce qu'ils couvrent différentes « parties » du domaine, soit parce qu'ils se placent à différents niveaux d'approximation ou encore parce qu'ils considèrent le monde selon *différentes perspectives*. En ce sens, il semble que la notion de contexte soit plus générale que celle de point de vue. Un modèle de représentation multi-contextes doit ainsi pouvoir être utilisé pour représenter de multiple points de vue.

La référence en ce qui concerne la représentation et le raisonnement avec les contextes est incontestablement la formalisation réalisée dans le cadre de la logique par John McCarthy dans [McCarthy, 1993]. Dans celle-ci, toute assertion A de la logique doit nécessairement être réalisée au sein d'un contexte, par une expression de la forme $c : A$, où c correspond à un contexte. Les contextes s'expriment par ailleurs au travers d'un prédicat *ist* (pour *is true*) dans des expressions de la forme :

$$c' : ist(c, p)$$

signifiant intuitivement que la proposition p est considérée comme vraie dans le contexte c , cette assertion étant elle-même exprimée dans le contexte c' . Dans cette formule, c' est considéré comme un *contexte englobant* pour c . Dans le contexte c'' englobant c' , la proposition $ist(c', ist(c, p))$ peut alors être vérifiée.

Les *systèmes multi-contextes* (en anglais, MCS, *multi-context systems*) ont en premier lieu été présentés dans [Guinchiglia et Serafini, 1994] comme une alternative aux logiques modales⁶. Il s'agit d'un cadre logique, fondé sur la définition des règles d'inférences pour le raisonnement au sein de multiples contextes (*proof-theoric semantics*). Ces travaux ont par la suite été complétés dans [Ghidini et Guinchiglia, 2001] par la définition d'une sémantique à modèles locaux (en anglais, LMS, *local models semantics*), apportant une formalisation de la notion de contexte dans le cadre de la théorie des modèles (*model-theoric semantics*). En LMS/MCS, un contexte est considéré comme une théorie partielle et approximative du monde, traduisant une perspective particulière. Les raisonnements sont principalement réalisés localement, au sein des contextes, et les contextes sont reliés au travers de *passerelles*, permettant aux raisonnements locaux de collaborer. Le langage C-OWL présenté en détail dans la suite est issu des travaux sur les LMS/MCS. Ces principes seront donc expliqués plus formellement dans ce cadre.

En permettant chacun la mise en œuvre de théories partielles du domaine, ces deux travaux sur la représentation de contextes relèvent à la fois de la définition d'un point de vue comme une représentation particulière du monde (pdv2) et comme une opinion (pdv3).

4.1.5 Points de vue et ontologies

Dans [Falquet et Mottaz, 2001, Falquet et Mottaz, 2002], une ontologie est définie comme un ensemble de concepts, organisés en une hiérarchie et associés à des définitions formelles. Une ontologie multi-points de vue est ainsi une ontologie dans laquelle un concept peut être associé à plusieurs définitions, chacune correspondant à un point de vue particulier sur le concept. La place d'un concept dans la hiérarchie étant dépendante de sa définition, un concept pourra être classifié de différentes façons dans différents points de vue. En permettant de multiples représentations des concepts et de multiples hiérarchies selon les points de vue, ces travaux relèvent de la deuxième définition des points de vue. Par ailleurs, les définitions associées à un même concept dans différents points de vue étant contraintes à être compatibles, ce modèle ne peut être utilisé pour représenter des opinions sur le domaine (pdv3).

Le formalisme C-OWL est conçu pour intégrer la notion de contexte dans les ontologies du Web sémantique. Il se présente comme une extension de OWL pour la représentation *d'ontologies contextualisées* [Bouquet et al., 2004]. Un contexte en C-OWL est une *ontologie locale*, représentée en OWL et possédant son propre langage et sa propre interprétation. Les éléments de chaque contexte, c'est-à-dire les classes, les propriétés et les individus, peuvent être reliés entre eux par des *passerelles*. Par exemple, une passerelle de la forme $i : C \stackrel{\sqsubseteq}{\rightarrow} j : D$ indique que la classe C représentée dans le contexte O_i est vue par le contexte O_j comme plus spécifique que sa classe D . Ces passerelles sont associées à une sémantique clairement définie permettant de réaliser des raisonnements globaux, réutilisant les connaissances d'un contexte à l'autre.

Un contexte de C-OWL permet de représenter une vision particulière du monde, pouvant être reliée à d'autres de ces visions par les passerelles. En ce sens, il relève de la deuxième définition des points

⁶Les points de vue et les contextes semblent en effet entretenir des liens étroits avec les logiques modales. [Attardi et Simi, 1995] montre par exemple comment la formalisation des points de vue qu'il décrit peut permettre la définition de modalités telles que celles de *croyance* et de *connaissance*. Une étude plus approfondie des relations entre points de vue et modalités serait ainsi intéressante, mais sortirait du cadre de ce mémoire.

de vue, comme de multiples représentations. Par ailleurs, comme chaque contexte en C-OWL possède sa propre interprétation, indépendante de celles des autres contextes, il est possible de représenter des éléments logiquement incompatibles. De cette façon, C-OWL peut servir à représenter des points de vue du troisième type, c'est-à-dire des opinions différentes sur le domaine.

C-OWL est particulièrement bien adapté pour répondre à la problématique de la représentation multi-points de vue dans le cadre du projet KASIMIR. En effet, les disciplines impliquées dans la prise en charge de patients atteints de cancers correspondent à autant de sous-domaines de la cancérologie. Chacune organise ses connaissances d'une façon qui lui est propre, ne considérant pas les mêmes éléments comme pertinents. Ainsi, associer à ces différentes disciplines différentes ontologies locales, des contextes, et modéliser la collaboration et le partage de connaissances par des passerelles semble judicieux. Par ailleurs, étant défini comme une extension de OWL, C-OWL et les mécanismes de raisonnements associés s'intègrent naturellement dans l'architecture de portail du Web sémantique élaborée pour KASIMIR.

4.2 C-OWL : ontologies contextualisées

C-OWL, pour *context-OWL*, est un formalisme récemment proposé pour la représentation d'*ontologies contextualisées* [Bouquet *et al.*, 2003b, Bouquet *et al.*, 2004], c'est-à-dire d'ontologies locales appariées entre elles. Un des objectifs de C-OWL est de fournir un langage formel pour l'alignement d'ontologies du Web sémantique, afin de raisonner sur les appariements et ainsi d'utiliser conjointement plusieurs ontologies relatives au même domaine. Mais, au-delà de l'alignement, nous sommes intéressés ici à la mise en œuvre d'ontologies modulaires, où un contexte correspond à un point de vue selon lequel on peut appréhender les connaissances du domaine. C'est de cette façon que nous utilisons C-OWL dans la suite.

4.2.1 Ontologies, contextes et points de vue

Le Web sémantique repose essentiellement sur la notion d'ontologie. Une ontologie définissant de façon formelle et non ambiguë la signification selon laquelle les termes et les concepts d'un domaine doivent être interprétés, elle constitue une base sémantique commune pour la communication entre les différents acteurs de ce domaine. Elle doit pour cette raison être partagée par tous ces acteurs et construite indépendamment de toute application, pour être, le plus possible, réutilisable globalement. La mise en œuvre concrète de cette vision du Web sémantique suppose ainsi l'existence d'ontologies développées sur la base de consensus dans les domaines qu'elles recouvrent. Mais, construire et maintenir un modèle consensuel des connaissances d'un domaine dans un environnement aussi dynamique et hétérogène que le Web peut s'avérer d'une grande complexité.

En opposition avec la notion de modèle *global* des connaissances d'un domaine que représente une ontologie, les travaux concernant la modélisation et la représentation de contextes s'intéressent à des modèles de connaissances *locaux*, c'est-à-dire construits selon une vue particulière du domaine [Ghidini et Guinchiglia, 2001]. Les connaissances sont représentées pour les besoins d'une application, modélisant la vision d'une partie seulement des acteurs du domaine, et n'ont ainsi pas pour vocation d'être partagées. Une telle représentation se focalise uniquement sur la représentation des connaissances pertinentes dans un contexte donné et est de cette façon plus simple à construire et à maintenir qu'une ontologie couvrant le domaine dans sa globalité.

Contextes et ontologies s'opposent dans le sens où ce qui fait la force des uns apparaît comme la faiblesse des autres. L'idée sous-jacente à C-OWL est donc d'intégrer la représentation de contextes à la représentation d'ontologies en OWL, afin de mettre en commun leurs avantages. [Bouquet *et al.*, 2003b]

et [Bouquet *et al.*, 2004] définissent ainsi la notion d'*ontologie contextualisée* (ou ontologie contextuelle) pour désigner une ontologie dont le contenu est décrit dans un contexte particulier et mis en relation avec le contenu d'autres ontologies au travers d'*appariements* (en anglais *mappings*). Une ontologie contextualisée relève ainsi à la fois d'une représentation locale et d'une représentation globale dans le sens où elle est combinée et utilisée conjointement avec d'autres ontologies locales du domaine.

Dans le cadre du projet KASIMIR, on cherche à construire une représentation des connaissances décisionnelles en cancérologie qui soit partagée entre les différents acteurs du domaine et qui intègre plusieurs points de vue, correspondant notamment aux différentes disciplines de la cancérologie. La notion de point de vue sur laquelle on s'appuie ici correspond principalement à la deuxième façon de considérer les points de vue (pdv2 page 57) : il s'agit de mettre en œuvre plusieurs représentations alternatives des connaissances du domaine. Plus précisément, on considérera dans la suite un point de vue comme *une représentation partielle des connaissances d'un domaine, construite dans un but particulier et coexistant avec d'autres points de vue*.

En ce sens, la notion de contexte, telle qu'elle est mise en œuvre dans C-OWL, semble englober celle de point de vue. En effet, comme montré dans [Benerecetti *et al.*, 2001], la distinction entre les contextes d'un système multi-contextes peut être de plusieurs natures. Elle relève en particulier de relations entre des contextes représentant des vues *partielles*, des représentations *approximatives* ou de multiples *perspectives* sur le monde. Une représentation est dite partielle lorsqu'elle ne représente qu'un sous-ensemble des connaissances du domaine. On entend par approximation que la représentation se situe à un certain niveau de granularité et d'abstraction. Une représentation est considérée comme une perspective lorsqu'elle dépend d'un élément extérieur tel que le lieu, le moment, une application, etc. Ainsi, au sein d'un ensemble de contextes, deux contextes peuvent se distinguer du fait qu'ils se focalisent sur différentes parties du domaine, selon différents niveaux d'approximation ou selon différentes perspectives. La notion de point de vue s'intègre alors complètement dans celle de contexte. Un point de vue correspond en effet à une vue partielle, définie relativement à un objectif extérieur en présence d'autres représentations. Il pourrait d'ailleurs être montré que le modèle de représentation de C-OWL est plus général que certains systèmes de représentation multi-points de vue présentés à la section 4.1. La représentation des concepts selon plusieurs points de vue en TROEPS, par exemple, pourrait être traduite en C-OWL.

4.2.2 Syntaxe et sémantique de C-OWL

C-OWL est une extension de OWL fondée sur les principes développés dans le domaine de la modélisation, de la représentation et du raisonnement sur les contextes (voir par exemple [Bouquet *et al.*, 2003a]). Suivant le modèle des *systèmes multi-contextes* [Guinchiglia et Serafini, 1994], le formalisme de C-OWL s'appuie sur deux principes fondamentaux :

- Une *sémantique à modèles locaux* : chaque contexte correspond à une ontologie locale ayant son propre langage et surtout sa propre interprétation. C'est-à-dire en particulier que différents contextes seront interprétés dans des domaines d'interprétation différents.
- Des *relations sémantiques entre domaines locaux* : les appariements sont interprétés par des relations sémantiques entre les domaines locaux de différents contextes. Ils établissent ainsi des correspondances entre les éléments de ces domaines.

De plus, de la même façon que OWL est conçu sur les principes des logiques de descriptions, la logique sous-jacente à C-OWL est une *logique de descriptions distribuée* (LDD [Borgida et Serafini, 2002]). Dans la suite, pour une question de simplicité, il n'est fait de distinction entre C-OWL et les LDD que lorsque cela est nécessaire.

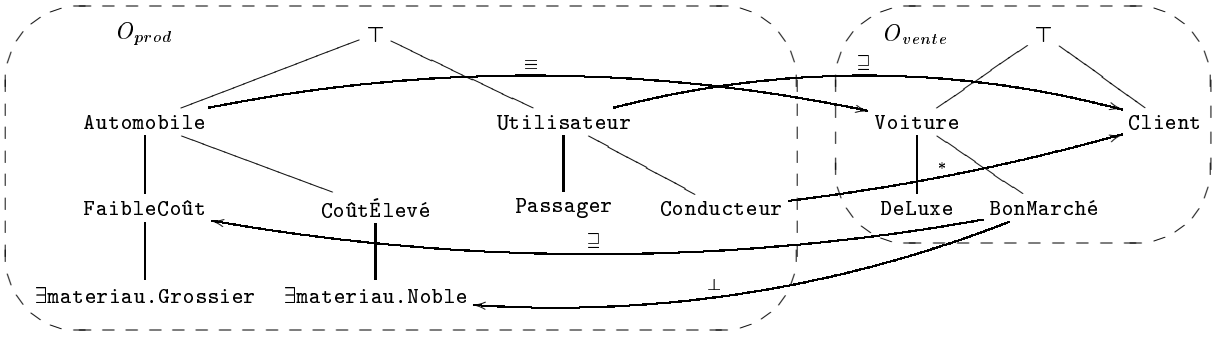


FIG. 4.3 – Exemple d'appariement entre deux contextes avec C-OWL.

Espace de contextes

En C-OWL, les connaissances sont représentées de façon distribuée au sein d'*ontologies contextualisées*, c'est-à-dire de *contextes*, correspondant à des ontologies locales en OWL et reliés entre eux par des appariements. Un *espace de contextes* est ainsi composé d'un ensemble de contextes $\{O_i\}_i$ et d'un ensemble $\{M_{jk}\}_{j,k}$ d'appariements entre contextes, i, j et k étant des index appartenant à un ensemble I d'index de contextes et $j \neq k$.

Chacun des contextes O_i d'un espace de contextes constitue une représentation particulière du domaine et possède son propre *langage local*. Un index de contexte $i \in I$ est utilisé comme préfixe aux expressions du langage local du contexte O_i , pour délimiter les éléments de connaissances définis au sein de ce contexte. Par exemple, les expressions $i:C$, $i:C \sqcap D$, $i:\exists p.C$, $i:C \sqsubseteq D$ ou $i:C(a)$ sont des constructions du langage local de O_i . L'expression $i:C \sqsubseteq D$ par exemple, correspond à la déclaration, au sein du contexte O_i , d'un axiome d'inclusion entre la classe locale $i:C$ et la classe locale $i:D$.

Comme indiqué plus haut, un contexte possède aussi *sa propre interprétation*, indépendamment des autres contextes de l'espace de contextes. Ainsi, la sémantique d'un espace de contextes est donnée selon une *interprétation distribuée* \mathcal{I} , contenant un ensemble $\{\mathcal{I}_i\}_{i \in I}$ d'interprétations locales, avec une interprétation locale \mathcal{I}_i pour chaque contexte O_i de l'espace de contextes [Serafini et Taminin, 2005]. Une interprétation locale \mathcal{I}_i est une interprétation au sens classique des LD, composée d'un domaine d'interprétation local $\Delta^{\mathcal{I}_i}$ et d'une fonction d'interprétation locale $\cdot^{\mathcal{I}_i}$. Les éléments d'un contexte O_i sont interprétés selon l'interprétation locale \mathcal{I}_i correspondante. Formellement, un axiome ou une assertion défini dans un contexte O_i est satisfait par l'interprétation distribuée \mathcal{I} s'il est satisfait par l'interprétation locale \mathcal{I}_i . Cela signifie par exemple que, $\mathcal{I} \models i:C \sqsubseteq D$ si $\mathcal{I}_i \models C \sqsubseteq D$, c'est-à-dire si $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.

Passerelles

L'objectif des appariements est de permettre l'utilisation conjointe de plusieurs contextes, en rendant possible les échanges et la réutilisation de connaissances d'un contexte à l'autre. Il s'agit d'établir des correspondances entre les éléments de connaissances de différents contextes. Un appariement M_{ij} entre les contextes O_i et O_j est composé de *passerelles*, appelées en anglais *bridge rules*, liant des classes, des propriétés et des individus de O_i à des classes, propriétés et individus de O_j . Dans [Bouquet *et al.*, 2004] sont définies cinq formes de passerelles différentes. Celles-ci sont illustrées dans l'exemple de la figure 4.3, entre les classes de deux contextes, celui de la production de voiture (O_{prod}) et celui de la vente de voiture (O_{vente}).

- $i:x \stackrel{\sqsubseteq}{\rightarrow} j:y$ (**dedans**, en anglais *into*) indique que l'élément $i:x$ du contexte O_i est considéré *selon* O_j comme plus spécifique que l'élément $j:y$ qu'il définit. Par exemple, il pourra être consi-

déré, du point de vue de la production, qu'une voiture vendue comme « bon marché » est une voiture à faible coût de production : $\text{vente}:\text{BonMarché} \xrightarrow{\sqsubseteq} \text{prod}:\text{FaibleCoût}$.

- $i:x \xrightarrow{\sqsupseteq} j:y$ (**dessus**, en anglais *onto*) indique que O_j considère l'élément $i:x$ de O_i comme plus général que $j:y$. Ainsi, il pourra être déclaré que ce que la vente considère comme des clients fait partie de ce que la production appelle des utilisateurs : $\text{prod}:\text{Utilisateur} \xrightarrow{\sqsupseteq} \text{vente}:\text{Client}$.
- $i:x \xrightarrow{\equiv} j:y$ (**équivalent**) indique que O_j considère l'élément $i:x$ de O_i comme équivalent à $j:y$. La classe *Voiture* pourra par exemple être considérée d'après la vente comme correspondant à la classe *Automobile* de la production : $\text{prod}:\text{Automobile} \xrightarrow{\equiv} \text{vente}:\text{Voiture}$.
- $i:x \xrightarrow{\perp} j:y$ (**incompatible**) indique que O_j considère l'élément $i:x$ de O_i comme incompatible avec $j:y$, c'est-à-dire qu'ils ne représentent aucun élément commun. On pourra par ce biais indiquer par exemple que, selon la production, construire une voiture « bon marché » est incompatible avec l'utilisation de matériaux nobles : $\text{vente}:\text{BonMarché} \xrightarrow{\perp} \text{prod}:\exists\text{matériau.Noble}$.
- $i:x \xrightarrow{*} j:y$ (**compatible**) indique que O_j considère l'élément $i:x$ de O_i comme compatible avec $j:y$, c'est-à-dire qu'ils représentent au moins un élément commun. Par exemple, ce que la production appelle un conducteur peut être considéré par la vente comme compatible avec la notion de client : $\text{prod}:\text{Conducteur} \xrightarrow{*} \text{vente}:\text{Client}$.

L'interprétation des passerelles est réalisée au travers de ce qui est appelé en C-OWL des *relations de domaines*. Une interprétation distribuée \mathcal{I} inclut en effet un ensemble de relations $\{r_{ij}\}_{i,j \in I}$, avec une relation de domaine $r_{ij} \subseteq \Delta^{I_i} \times \Delta^{I_j}$ pour chaque contextes O_i et O_j ($i \neq j$) de l'espace de contextes. Une telle relation indique, pour chaque objet de Δ^{I_i} , les objets de Δ^{I_j} auxquels il correspond. En d'autres termes, $r_{ij}(\mathcal{C}^{I_i})$ est une notation pour l'interprétation de la classe $i:C$ du contexte O_i telle que considérée dans le domaine d'interprétation de O_j ($r_{ij}(\mathcal{C}^{I_i}) \subseteq \Delta^{I_j}$). On dira alors par exemple que l'interprétation distribuée \mathcal{I} satisfait $i:x \xrightarrow{\sqsubseteq} j:y$ (noté $\mathcal{I} \models i:x \xrightarrow{\sqsubseteq} j:y$) si $r_{ij}(x^{I_i}) \subseteq y^{I_j}$. Cela correspond bien à la description intuitive de la passerelle *dedans* vue plus haut. Les règles de satisfiabilité pour les cinq types de passerelles sont données ci-dessous. On remarque que, pour des raisons d'homogénéité de la notation, l'interprétation d'un individu est ici considérée comme un singleton $\{x\}$ plutôt que comme un objet x .

- $\mathcal{I} \models i:x \xrightarrow{\sqsubseteq} j:y$ si $r_{ij}(x^{I_i}) \subseteq y^{I_j}$
- $\mathcal{I} \models i:x \xrightarrow{\sqsupseteq} j:y$ si $r_{ij}(x^{I_i}) \supseteq y^{I_j}$
- $\mathcal{I} \models i:x \xrightarrow{\equiv} j:y$ si $r_{ij}(x^{I_i}) = y^{I_j}$
- $\mathcal{I} \models i:x \xrightarrow{\perp} j:y$ si $r_{ij}(x^{I_i}) \cap y^{I_j} = \emptyset$
- $\mathcal{I} \models i:x \xrightarrow{*} j:y$ si $r_{ij}(x^{I_i}) \cap y^{I_j} \neq \emptyset$

Il doit être remarqué que, comme la description informelle des passerelles donnée plus haut le laisse entendre, une passerelle est *directionnelle*, dans le sens où elle est considérée du point de vue de son contexte de destination. Elle ne peut ainsi avoir d'influence que sur celui-ci. $i:C \xrightarrow{\sqsubseteq} j:D$, par exemple, signifie que *selon le contexte* O_j , la classe $i:C$ est plus spécifique que la classe $j:D$. Il peut ne pas en être de même du point de vue de O_i . Cette passerelle n'est donc pas équivalente à $j:D \xrightarrow{\sqsupseteq} i:C$. Considérant la sémantique des passerelles, $r_{ij}(\mathcal{C}^{I_i}) \subseteq \mathcal{D}^{I_j}$ n'est pas équivalent à $r_{ji}(\mathcal{D}^{I_j}) \supseteq \mathcal{C}^{I_i}$.

Par ailleurs, les cinq formes de passerelles indiquées ne sont pas toutes utilisées. Les LDD ne s'intéressent qu'aux passerelles de types *dedans* et *dessus*, et cela, uniquement entre classes [Borgida et Serafini, 2002]. Il peut de plus facilement être montré que $i:x \xrightarrow{\equiv} j:y$ est équivalent à la conjonction de $i:x \xrightarrow{\sqsubseteq} j:y$ et $i:x \xrightarrow{\sqsupseteq} j:y$ ou encore, avec $i:C$ et $j:D$ deux classes respectivement de O_i et O_j , que $i:C \xrightarrow{\perp} j:D$ peut se réécrire $i:C \xrightarrow{\sqsubseteq} j:\neg D$. Pour notre part, nous nous restreindrons pour plus de facilité aux passerelles de la forme $i:C \xrightarrow{\sqsubseteq} j:D$, $i:C \xrightarrow{\sqsupseteq} j:D$ et $i:a \xrightarrow{\equiv} j:b$, où $i:C$ et $j:D$ sont des classes et $i:a$ et $j:b$ des individus.

Syntaxe RDF/XML

C-OWL est une extension de OWL et doit à ce titre fournir une syntaxe qui soit, le plus possible, compatible avec celle de OWL. Chaque contexte O_i d'un espace de contextes est représenté sous la forme d'une ontologie classique en OWL. La syntaxe RDF/XML standard de OWL est donc suffisante pour cela. Par ailleurs, dans cette syntaxe, la notion d'*espace des noms* est utilisée pour regrouper les identificateurs selon leur ontologie d'appartenance. Par exemple, dans l'URI

`http://www.voiture.com/production#Automobile`

la partie `http://www.voiture.com/production` fait référence à un espace des noms, celui de l'ontologie utilisée par exemple par les usines de production de voiture. `Automobile` dans ce cas est l'identificateur de la classe des voitures au sein de cet espace des noms. De cette façon, une autre ontologie pourra elle aussi utiliser l'identificateur `Automobile`, dans un autre espace des noms, sans confusion possible. Ce mécanisme est utilisé en C-OWL pour représenter les index des contextes. Par exemple, l'URI `http://www.voiture.com/production#Automobile`, pourra correspondre à la classe `prod:Automobile` du contexte O_{prod} et `http://www.voiture.com/vente#Voiture` à la classe `vente:Voiture` définie dans le contexte O_{vente} .

Les appariements et les passerelles doivent eux aussi être représentés dans une syntaxe RDF/XML, compatible avec celle de OWL. Afin d'être utilisable dans la majorité des situations, cette extension prend pour principe d'être la moins « invasive » possible vis-à-vis de l'existant [Bouquet *et al.*, 2004]. Ainsi, la mise en œuvre d'un espace de contextes en C-OWL consiste à construire les appariements entre des ontologies existantes, décrites en OWL classique. La représentation des appariements se fait selon un schéma RDF, conçu sur la base de CTXML [Bouquet *et al.*, 2002], un langage de spécification d'appariements. Suivant ce schéma, les passerelles de la figure 4.3 sont représentées au sein de deux appariements, comme détaillé à la figure 4.4.

4.2.3 Raisonnement avec C-OWL

Dans [Serafini et Tamin, 2004], une distinction est faite entre les mécanismes de *raisonnements locaux*, c'est-à-dire les inférences classiques des LD réalisées localement dans un contexte, et les mécanismes de *raisonnement globaux*, qui reposent sur les passerelles pour réaliser des inférences dans un contexte sur la base de connaissances contenues dans un autre contexte. Deux approches différentes ont été décrites pour la mise en œuvre de raisonnements globaux en LDD et sont applicables à C-OWL. En effet, [Borgida et Serafini, 2002] montre comment un espace de contextes en LDD peut être transformé en une base de connaissances en LD classique. On obtient une seule base de connaissances, regroupant l'ensemble des contextes de l'espace de contextes et traduisant les passerelles de la façon suivante :

- Une règle *dedans* $i : C \stackrel{\sqsubseteq}{\rightarrow} j : D$ est transformée en un axiome $iC \sqsubseteq \forall Rij.jD$
- Une règle *dessus* $i : C \stackrel{\supseteq}{\rightarrow} j : D$ est transformée en un axiome $iD \sqsubseteq \exists Rij^- .jC$

où iC et jD sont des classes créées pour correspondre aux classes $i : C$ et $j : D$, Rij est une propriété utilisée pour représenter la relation de domaine r_{ij} entre les domaines d'interprétation des deux contextes O_i et O_j et Rij^- est la propriété inverse de Rij . Cette approche a l'avantage de donner une idée simple des raisonnements globaux en LDD en les ramenant à des raisonnements en LD classique. Elle permet par ailleurs la réutilisation des moteurs d'inférences existants. Mais, un des intérêts de C-OWL est que les connaissances soient *localisées* au sein de multiples contextes. Leur utilisation doit ainsi pouvoir être distribuée, gagnant de cette façon en flexibilité et en temps de calcul. En construisant une représentation unique, on perd l'avantage de cette distribution. Par ailleurs, cette représentation globale nécessite de disposer d'un moteur d'inférences capable de mettre en œuvre une LD dont l'expressivité répond aux besoins de l'ensemble des contextes.

```

<owl:Mapping rdf:ID="M-ProdVente">
  <rdfs:comment>Appariement de l'ontologie pour la production de
    voiture à celle pour la vente</rdfs:comment>
  <owl:sourceOntology><owl:Ontology rdf:about="http://www.voiture.com/production"/></owl:sourceOntology>
  <owl:targetOntology><owl:Ontology rdf:about="http://www.voiture.com/vente"/></owl:targetOntology>
  <owl:bridgeRule>
    <owl:Equivalent>
      <owl:source><owl:Class rdf:about="http://www.voiture.com/production#Automobile"/></owl:source>
      <owl:target><owl:Class rdf:about="http://www.voiture.com/vente#Voiture"/></owl:target>
    </owl:Equivalent>
  </owl:bridgeRule>
  <owl:bridgeRule>
    <owl:Onto>
      <owl:source><owl:Class rdf:about="http://www.voiture.com/production#Utilisateur"/></owl:source>
      <owl:target><owl:Class rdf:about="http://www.voiture.com/vente#Client"/></owl:target>
    </owl:Onto>
  </owl:bridgeRule>
  <owl:bridgeRule>
    <owl:Compatible>
      <owl:source><owl:Class rdf:about="http://www.voiture.com/production#Conducteur"/></owl:source>
      <owl:target><owl:Class rdf:about="http://www.voiture.com/vente#Client"/>
    </owl:target>
  </owl:Compatible>
  </owl:bridgeRule>
</owl:Mapping>

<owl:Mapping rdf:ID="M-VenteProd">
  <rdfs:comment>Appariement de l'ontologie pour la vente
    de voiture à celle pour la production</rdfs:comment>
  <owl:sourceOntology><owl:Ontology rdf:about="http://www.voiture.com/vente"/></owl:sourceOntology>
  <owl:targetOntology><owl:Ontology rdf:about="http://www.voiture.com/production" /></owl:targetOntology>
  <owl:bridgeRule>
    <owl:Into>
      <owl:source><owl:Class rdf:about="http://www.voiture.com/vente#BonMarché"/></owl:source>
      <owl:target><owl:Class rdf:about="http://www.voiture.com/production#FaibleCoût"/></owl:target>
    </owl:Into>
  </owl:bridgeRule>
  <owl:bridgeRule>
    <owl:Incompatible>
      <owl:source><owl:Class rdf:about="http://www.voiture.com/vente#BonMarché"/></owl:source>
      <owl:target>
        <owl:Class>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:about="http://www.voiture.com/production#matériau"/>
            </owl:onProperty>
            <owl:someValuesFrom>
              <owl:Class rdf:about="http://www.voiture.com/production#Noble"/>
            </owl:someValuesFrom>
          </owl:Restriction>
        </owl:Class>
      </owl:target>
    </owl:Incompatible>
  </owl:bridgeRule>
</owl:Mapping>

```

FIG. 4.4 – Exemple de représentation d'appariements dans la syntaxe RDF/XML de C-OWL.

En réponse aux problèmes posés par l'approche précédente, [Serafini et Taminin, 2005] décrit une extension de l'algorithme des tableaux, classiquement utilisé dans les moteurs d'inférences en LD, pour le raisonnement global en LDD. Le calcul de la *subsomption globale* repose en effet sur ce qui est appelé dans cet article le *principe de propagation de la subsomption*. Dans sa forme la plus simple, ce principe peut s'écrire :

Si l'appariement M_{ij} entre O_i et O_j contient $i : A \xrightarrow{\sqsubseteq} j : C$ et $i : B \xrightarrow{\sqsubseteq} j : D$ alors
 \mathcal{I} satisfait $i : A \sqsubseteq B$ implique que \mathcal{I} satisfasse $j : C \sqsubseteq D$
 (si $\mathcal{I} \models i : A \sqsubseteq B$ alors $\mathcal{I} \models j : C \sqsubseteq D$)

En d'autres termes, la subsomption dans un contexte O_j peut être déduite de la subsomption dans un contexte O_i et de passerelles de O_i à O_j . L'avantage de cette approche est qu'elle peut être mise en œuvre de façon distribuée. En effet, dans l'architecture proposée par [Serafini et Taminin, 2005], un moteur d'inférences propre est attaché à chaque contexte de l'espace de contextes. Le calcul de subsomption dans un contexte particulier est ainsi réalisé localement par le moteur d'inférences associé, qui fait appel aux moteurs d'inférences d'autres contextes lorsque la présence de passerelles le nécessite. On conserve alors tout l'avantage de la distribution des connaissances dans différents contextes. Les moteurs d'inférences peuvent en effet s'exécuter en parallèle, éventuellement sur différentes machines. De plus, la représentation des connaissances dans les différents contextes pouvant faire appel à des constructeurs différents, chaque moteur d'inférences pourra être spécialisé pour ne prendre en compte que le niveau d'expressivité nécessaire au contexte auquel il est associé, réduisant ainsi la complexité du raisonnement. Cette approche a par ailleurs été implémentée dans un prototype appelé DRAGO, construit au dessus du moteur d'inférences pour OWL PELLET [Serafini et Taminin, 2005]. DRAGO est à l'heure actuelle toujours en développement et souffre de nombreuses limitations. Il n'implante en particulier que la subsomption comme mécanisme de raisonnement global et ne permet pas de prendre en compte des espaces de contextes dont les appariements contiennent des cycles. Ainsi, si l'appariement M_{ij} contient au moins une passerelle alors M_{ji} devra être vide.

D'autres mécanismes de raisonnements que la subsomption globale sont par ailleurs envisageables. Le test de cohérence d'un appariement ou l'inférence de passerelles sont par exemple mentionnés dans [Bouquet *et al.*, 2004]. Nous nous appuyons pour notre part sur un mécanisme d'*instanciation globale* qui repose sur un principe de *propagation de l'instanciation*, fortement inspiré du principe de propagation de la subsomption décrit dans [Serafini et Taminin, 2005] :

Si l'appariement M_{ij} entre O_i et O_j contient $i : C \xrightarrow{\sqsubseteq} j : D$ et $i : a \xrightarrow{\sqsubseteq} j : b$ alors
 \mathcal{I} satisfait $i : C(a)$ implique que \mathcal{I} satisfasse $j : D(b)$
 (si $\mathcal{I} \models i : C(a)$ alors $\mathcal{I} \models j : D(b)$)

En d'autres termes, il est possible de déduire qu'un individu est instance d'une classe dans un contexte en s'appuyant sur l'instanciation dans un autre contexte et sur les passerelles. On peut, pour illustrer ce principe, imaginer un exemple très simple, avec deux contextes (voir figure 4.5) : l'ontologie des êtres vivants (O_v) et celle des entités mythologiques (O_m). O_v décrit notamment la classe des animaux et celle des humains. O_m décrit en particulier les classes d'êtres mortels et d'êtres immortels. Une passerelle *dedans* est déclarée entre la classe des humains dans O_v et la classe des mortels dans O_m : $v : \text{Humain} \xrightarrow{\sqsubseteq} m : \text{Mortel}$ (tout homme est mortel). Un individu $v : \text{socrate}$, instance de la classe $v : \text{Humain}$, est représenté dans O_v (Socrate est un homme). Un autre $m : \text{socrate}$ est représenté dans O_m et une passerelle $v : \text{socrate} \xrightarrow{\sqsubseteq} m : \text{socrate}$ est déclarée afin d'indiquer qu'il s'agit du même individu dans deux contextes différents. Selon le principe de propagation de l'instanciation, il sera ainsi possible de déduire que $m : \text{socrate}$ est une instance de la classe $m : \text{Mortel}$ dans le contexte O_m (donc Socrate est mortel).

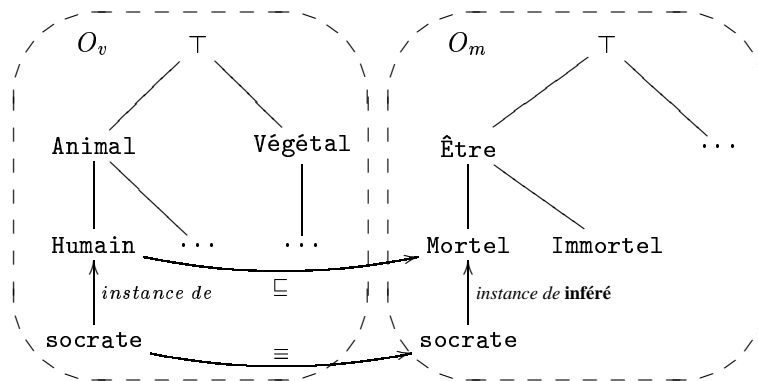


FIG. 4.5 – Exemple d’application du principe de propagation de l’instanciation.

4.2.4 Application de C-OWL

L’article [Bouquet *et al.*, 2004] mentionne une application de C-OWL, précédemment présentée dans [Stuckenschmidt *et al.*, 2004]. Il s’agit de l’alignement, c’est-à-dire la mise en correspondance par des appariements, de trois ontologies du domaine médical : GALEN [Rector et Nowlan, 1993], TAM-BIS [Baker *et al.*, 1999] et le réseau sémantique de l’UMLS [Nelson *et al.*, 2002]. Outre celles présentées ici dans le cadre du projet KASIMIR, cet exemple semble être, à l’heure actuelle, la seule application existante de C-OWL.

4.3 Points de vue pour la cancérologie : représentation des référentiels en C-OWL

La cancérologie, comme beaucoup de domaines en médecine, implique plusieurs disciplines. Les connaissances décisionnelles dans ce domaine, que ce soient les connaissances contenues dans les référentiels ou les connaissances d’adaptation, combinent ainsi plusieurs aspects, plusieurs façons de voir et donc plusieurs points de vue. Chacun de ces points de vue constitue une façon propre à une discipline de représenter, d’organiser et d’utiliser ces connaissances. En effet, un cancérologue s’appuiera sur des éléments de connaissances différents et sur différentes représentations du patient selon qu’il s’intéresse à établir un traitement chirurgical ou une chimiothérapie. Ces multiples points de vue sont d’ailleurs visibles dans la structure même des référentiels. Le référentiel pour le traitement du cancer du sein par exemple est divisé en plusieurs parties, chacune se focalisant sur une phase de traitement particulière (chimiothérapie préopératoire, chirurgie, radiothérapie et traitement complémentaire), impliquant une à deux disciplines de la cancérologie, parmi la chimiothérapie, la chirurgie, la radiothérapie et l’hormonothérapie. Une partie définit son propre vocabulaire. Elle s’intéresse à fournir un type de recommandation particulier, dans une phase particulière de la prise en charge du patient, et n’utilise pour cela que les caractéristiques spécifiquement utiles à cette tâche. Par ailleurs, les RCP, c’est-à-dire les réunions de concertation pluridisciplinaires qui adaptent le contenu des référentiels pour les cas où leur application directe n’est pas satisfaisante, regroupent des experts de différentes disciplines. Chaque expert apporte dans ce cadre sa propre part à la solution, selon la discipline qu’il représente, et collabore avec les autres experts, d’autres disciplines, dans la construction d’une solution globale.

Nous nous intéressons ainsi dans cette section à la représentation selon plusieurs points de vue des connaissances utiles à la prise de décision en cancérologie, plus particulièrement de celles contenues dans les référentiels. Nous nous appuyons pour cela sur le langage C-OWL.

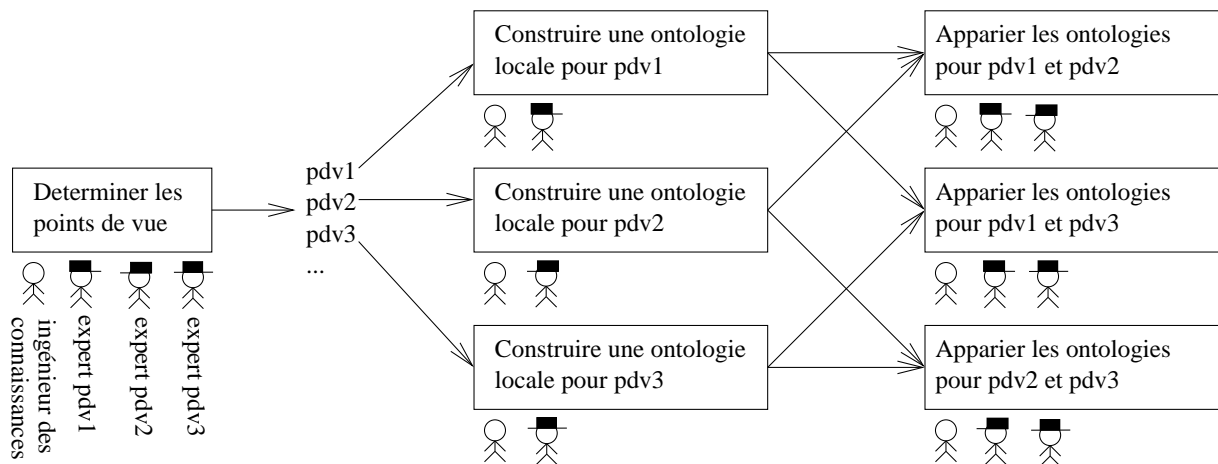


FIG. 4.6 – Démarche pour la construction d'une représentation multi-points de vue des connaissances.

4.3.1 Principes de la représentation

Habituellement, la construction d'une ontologie suppose la mise en œuvre d'un consensus entre les acteurs du domaine, afin d'établir les éléments de connaissances partagés par l'ensemble de ces acteurs et de les représenter indépendamment de tout contexte d'utilisation [Guarino, 1997]. Ici, l'idée est toute autre. Il s'agit de construire une représentation multi-points de vue, un point de vue correspondant dans ce cadre à la représentation des connaissances utiles à une application ou à un groupe de personnes particulier, qui coexiste et collabore avec d'autres. C-OWL nous permet de représenter ces points de vue sous forme d'ontologies locales, ou contextes. Les appariements entre contextes permettent l'échange et la réutilisation de connaissances d'un contexte à l'autre, combinant ainsi les points de vue en une représentation globale.

La démarche pour modéliser et formaliser ces connaissances multi-points de vue sur la base du modèle de C-OWL est ainsi différente de la démarche habituelle pour la construction d'ontologies. On ne cherche pas à établir un consensus, mais au contraire à distinguer les points de vue. Pour cela, trois étapes sont nécessaires (voir figure 4.6). Il s'agit dans un premier temps de *déterminer les points de vue* pertinents dans le domaine, puis de *construire une ontologie locale* pour chacun des points de vue et enfin d'*établir des appariements* entre ces ontologies locales, indiquant ainsi des correspondances entre les connaissances manipulées par les différents points de vue.

Déterminer les points de vue. Il s'agit ici de déterminer quels sont les points de vue à représenter. Lorsqu'un domaine est suffisamment vaste et complexe, il est souvent organisé selon plusieurs services, plusieurs tâches, plusieurs groupes de travail ou encore plusieurs communautés. Cette organisation apporte une division *a priori* du domaine en points de vue, division sur laquelle on pourra s'appuyer pour cette étape. Par ailleurs, déterminer les points de vue est aussi l'occasion de distinguer, parmi les experts du domaine, ceux qui sont le mieux à même de modéliser les connaissances relatives à chacun des points de vue, selon leurs spécialités. Dans ce cadre, les éléments indiqués dans la figure 4.6 sont largement idéalisés : on ne peut connaître tous les experts avant d'avoir déterminé les points de vue et certains experts peuvent être impliqués dans la définition de plusieurs points de vue.

Construire une ontologie locale. Le principe ici est de construire une ontologie locale, un contexte formalisé en OWL, pour chacun des points de vue déterminés. Une telle ontologie doit se focaliser sur les connaissances utiles au point de vue considéré, *indépendamment des autres points de vue*. Il s'agit de ne couvrir que la partie du domaine pertinente relativement au point de vue, en cherchant à obtenir une représentation simple et homogène, au plus proche de la vision des spécialistes et des utilisateurs.

Apparier les ontologies. Dans cette dernière étape, les relations qu'entretiennent implicitement les points de vue sont explicitées au travers d'appariements entre les ontologies locales. Il est important de se rappeler que, selon le modèle de C-OWL, les appariements sont directionnels : une passerelle est un élément de connaissances défini au sein d'un contexte particulier et n'est ainsi utilisable que relativement au point de vue correspondant. La question à se poser pour chacun des points de vue est de savoir quels sont les éléments de connaissances d'autres points de vue que celui-ci est en mesure de « voir » et de réutiliser. Il s'agit ensuite d'établir des correspondances permettant d'indiquer l'interprétation qui doit être faite de ces éléments de connaissances extérieurs au sein de l'ontologie locale.

4.3.2 Représentation du référentiel pour le traitement du cancer du sein en C-OWL

Dans cette section, nous illustrons la représentation multi-points de vue de référentiels avec C-OWL sur l'exemple du référentiel pour le traitement du cancer du sein. Celui-ci est inclus dans un document plus large couvrant aussi le diagnostic de ce cancer et est accessible au travers du site Web du réseau Oncolor [(RSO), 2005e].

Déterminer les points de vue

Déterminer les points de vue peut être une tâche difficile dans certains domaines. Il s'agit de trouver des divisions élémentaires dans les connaissances du domaine, indépendantes des autres, mais collaborant entre elles. Ici, le référentiel est divisé en différentes parties, s'intéressant à différentes phases du traitement et impliquant différentes disciplines :

1. La partie *chimiothérapie néo-adjuvante* : la chimiothérapie néo-adjuvante est un traitement chimiothérapique qui peut être recommandé avant la chirurgie et dont l'objectif principal est de freiner, voire d'endiguer, la progression de la tumeur, afin de permettre un geste chirurgical moins large.
2. La partie *chirurgie* : la chirurgie est en quelque sorte la phase centrale du traitement du cancer du sein. L'objectif de cette partie est de décider quel traitement chirurgical doit être employé selon les caractéristiques du patient et de sa maladie.
3. La partie *radiothérapie* : la radiothérapie a pour objectif de compléter la chirurgie en éliminant les éventuelles cellules envahies encore présentes, en particulier dans les ganglions. L'objectif est de décider quelles parties de l'anatomie du patient il est utile d'irradier.
4. La partie *traitement médical adjuvant* : les traitements adjuvants sont eux aussi des traitements complémentaires. Il peut s'agir de traitements en chimiothérapie ou en hormonothérapie.

Ces parties sont relativement indépendantes, bien qu'elles partagent certaines notions et fassent parfois référence aux décisions prises dans d'autres parties. Cette division du document est par ailleurs utilisée quotidiennement par les médecins et apparaît comme bien adaptée à leur utilisation. Il semble donc pertinent de s'appuyer sur cette organisation *a priori* des connaissances pour déterminer les points de vue. Le choix aurait pu être fait de considérer la partie *traitement médical adjuvant* comme deux points de vue différents, un pour la chimiothérapie et l'autre pour l'hormonothérapie, voire de regrouper la chimiothérapie néo-adjuvante et la chimiothérapie complémentaire. Mais nous avons préféré conserver

la structure initiale du référentiel. Dans la suite nous nous intéresserons donc à quatre points de vue, ceux de la *chimiothérapie néo-adjuvante*, de la *chirurgie*, de la *radiothérapie* et du *traitement médical adjuvant*, représentés dans quatre contextes de C-OWL, nommés respectivement O_{cna} , O_{chir} , O_{radio} et O_{tma} .

Remarque. Pour l'exemple présenté ici, on ne s'intéressera qu'à une sous-partie des connaissances décisionnelles contenues dans les référentiels. La représentation en est de plus simplifiée. Le traitement du cancer du sein étant un domaine particulièrement compliqué, il ne serait pas pertinent d'être complètement exhaustif et précis dans le cadre de cet exemple. Par ailleurs, il s'agit seulement ici d'illustrer la représentation de points de vue. Les éléments de décision mentionnés dans cet exemple n'ont ainsi pas tous été validés par des médecins.

Construire les ontologies locales

La représentation d'un point de vue s'appuie sur des éléments de vocabulaire propre à la partie du référentiel correspondante. Elle repose sur un diagramme (voir chapitre 1) s'intéressant aux caractéristiques de patients utiles selon le point de vue considéré. Chaque contexte est ainsi mis en œuvre selon les principes indiqués dans le chapitre 2 pour la représentation des référentiels en OWL (voir section 2.3.1).

Les points de vue considérés ici ont en commun de porter sur la recommandation de traitements pour le cancer du sein. Les classes de patients (*Patient*) et de traitements (*Traitement*) sont ainsi présentes dans chacun des quatre contextes. Néanmoins, ces classes ne représentent pas nécessairement les mêmes éléments dans l'ensemble des points de vue : elles ne doivent pas être interprétées de la même façon selon le contexte. Par exemple, un patient dans O_{chir} est une personne atteinte de cancer du sein, alors qu'un patient dans O_{tma} est une femme⁷ atteinte de cancer du sein et pour laquelle un traitement chirurgical a déjà été proposé. De la même façon, seuls les traitements radiothérapeutiques sont considérés dans O_{radio} alors qu'un traitement sera nécessairement chirurgical dans O_{chir} . Le lien existant entre les patients et les traitements recommandés par le référentiel est représenté dans tous les points de vue par une propriété nommée *recommandation*.

Dans une optique de résolution de problèmes, les instances de la classe *Patient* correspondent à des descriptions de problèmes et les instances de la classe *Traitement* correspondent aux solutions. Chaque point de vue est alors constitué des connaissances nécessaires à la résolution d'un type de problème particulier, un problème local, répondant à une question particulière (par exemple, « Quelle est la chirurgie recommandée ? ») dans le cadre d'un problème plus global (ici, « Quel est le traitement recommandé ? »).

La *chimiothérapie néo-adjuvante* correspond à un ensemble de cures de chimiothérapie (suivant le protocole FEC 100), dont les résultats sont évalués au bout de la troisième cure [(RSO), 2005a]. La décision à prendre correspond à la poursuite des cures, en fonction de l'évolution de la tumeur. Si la tumeur régresse ou se stabilise, la chimiothérapie doit être poursuivie, sinon, le passage du patient en RCP doit être envisagé. Il s'agit aussi d'évaluer si la réponse à la chimiothérapie est suffisante pour permettre une chirurgie conservatrice. Dans le cadre du traitement du cancer du sein, une chirurgie conservatrice est une ablation partielle qui épargne les parties essentielles du sein.

⁷Certains traitements hormonothérapeutiques proposés dans le référentiel pour le traitement du cancer du sein ne sont applicables qu'aux femmes.

Par exemple, considérons l'extrait suivant du contexte O_{cna} :

$$\begin{aligned} \text{P-RepSR} &\equiv \text{Patient} \sqcap \exists \text{chimioNA.}(\text{ChimioNeoAdjuvante} \sqcap \\ &\quad \exists \text{reponse.}(\text{Regression} \sqcup \text{Stabilisation})) \\ \text{P-RepSR} &\sqsubseteq \exists \text{recommandation.PoursuiteChimio} \end{aligned}$$

Celui-ci indique que le référentiel recommande de poursuivre la chimiothérapie (classe `PoursuiteChimio`) en cas de stabilisation ou de régression de la tumeur ($\exists \text{reponse.}(\text{Regression} \sqcup \text{Stabilisation})$). Les réponses de types `Stabilisation` ou `Régression` sont de plus associées à une propriété nommée *suffisante* permettant d'indiquer par un booléen si la tumeur a suffisamment évolué pour permettre une chirurgie conservatrice.

La chirurgie est le traitement central dans le cadre de la prise en charge d'un patient atteint de cancer du sein. Pour simplifier, nous nous contenterons ici de décrire les connaissances décisionnelles liées à la *chirurgie des tumeurs opérables d'emblée* [(RSO), 2005b], qui est parmi les situations les moins complexes à traiter. Il s'agit dans ce cas de décider entre une *mastectomie partielle* et une *mastectomie de Patey*. Une mastectomie partielle correspond à une ablation conservatrice du sein, se concentrant uniquement sur la tumeur. Elle est nécessairement associée à un *curage axillaire des deux étages inférieurs*. Un curage axillaire correspond au prélèvement de ganglions dans le but de les analyser. L'indication « deux étages inférieurs » concerne l'endroit où les ganglions doivent être prélevés. Une mastectomie de Patey correspond à une ablation totale du sein, associée à un curage axillaire complet, c'est-à-dire des trois étages. Par exemple, un patient ayant une tumeur opérable d'emblée, qui n'est par ailleurs pas multifocale, sera traité par une mastectomie partielle. Cela se traduit par les axiomes suivants dans O_{chir} :

$$\begin{aligned} \text{P-TOE-NMF} &\equiv \text{Patient} \sqcap \exists \text{aTumeur.}(\text{TumeurOpérableDEmblée} \sqcap \exists \text{aFoyer.}(\text{FoyerSimple})) \\ \text{P-TOE-NMF} &\sqsubseteq \exists \text{recommandation.MastectomiePartielle} \end{aligned}$$

La *radiothérapie* a pour objectif de compléter la chirurgie en éliminant les cellules envahies éventuellement restantes, en particulier dans les ganglions. Il s'agira de choisir la ou les zones qu'il sera utile d'irradier, parmi le sein, la paroi de la zone prélevée par une mastectomie partielle, la zone sus-claviculaire, ou la chaîne mammaire interne du sein (CMI), éventuellement élargie (CMIE) [(RSO), 2005c]. Les caractéristiques du patient prises en compte pour cette décision concernent la chirurgie proposée, la localisation de la tumeur dans le sein, sa taille, la présence d'une atteinte cutanée et la présence de cellules envahies dans les ganglions. À titre d'exemple, les axiomes

$$\begin{aligned} \text{P-CC-NP-LII} &\equiv \text{Patient} \sqcap \exists \text{chirurgie.}(\text{AblationConservatrice} \sqcap \\ &\quad \exists \text{envahissementGanglionnaire.}(\text{true} \sqcap \\ &\quad \exists \text{aTumeur.}(\exists \text{localisation.}(\text{Inféro-Interne}))) \\ \text{RadioSeinCMIE} &\equiv \text{Irradiation} \sqcap \exists \text{zone.}(\text{Sein} \sqcap \exists \text{zone.}(\text{CMIE})) \\ \text{R-Sein-CMIE} &\equiv \exists \text{recommandation.}(\text{RadioSeinCMIE}) \\ \text{P-CC-NP-LII} &\sqsubseteq \text{R-Sein-CMIE} \end{aligned}$$

indiquent qu'un patient ayant subi une chirurgie conservatrice, dont les ganglions sont envahis et ayant une tumeur localisée dans le quadrant inféro-interne du sein devra être traité par une radiothérapie du sein et de la chaîne mammaire interne élargie. Il faut remarquer qu'il est possible dans le cadre du

traitement du cancer du sein de recommander *de ne pas faire de radiothérapie*. On représentera ce type de recommandation par un axiome de la forme

$$P \sqsubseteq \neg \exists \text{recommandation.Radiothérapie}$$

qui indique que pour les patients représentés par des instances de la classe P, une radiothérapie ne pourra être recommandée.

La partie la plus importante (en taille) du référentiel pour le traitement du cancer du sein concerne le *traitement médical adjuvant* [(RSO), 2005d]. Elle se concentre sur la recommandation d'une hormonothérapie ou d'une chimiothérapie en complément de la chirurgie. L'hormonothérapie repose sur le contrôle des hormones ayant une influence sur le développement de la tumeur. Il y a deux types d'hormonothérapies utilisées dans le cadre du traitement du cancer du sein. L'hormonothérapie suppressive consiste en la suppression des fonctions des ovaires, productrices d'hormones, de façon chimique ou chirurgicale (ablation des ovaires). L'hormonothérapie additive consiste en une cure de Tamoxifène, un produit anti-œstrogènes. Les chimiothérapies envisageables consistent en 4 à 6 cures d'injections selon les protocoles FEC 50 (chimiothérapie de niveau 1) ou FEC 100 (chimiothérapie de niveau 2). Le choix du traitement médical adjuvant est guidé par des caractéristiques telles que l'état des récepteurs hormonaux, le nombre de ganglions envahis, l'âge du patient, ainsi que la taille de la tumeur et son grade (1, 2 ou 3). Le référentiel indique par exemple qu'un patient n'ayant pas de ganglion envahi et des récepteurs hormonaux positifs, qui serait âgé de moins de 35 ans et avec une tumeur de grade supérieur ou égal à 2 ou de taille supérieure ou égale à 1 cm, sera traité par une chimiothérapie de niveau 1 :

$$\begin{aligned} P\text{-GNI-RHP-AI35-TGS2ouTTS1} &\equiv \text{Patient} \sqcap \exists \text{âge.EntierInfE35} \sqcap \exists \text{ ganglionsEnvahis.0} \sqcap \\ &\quad \exists \text{récepteursHormonaux.RHPositifs} \sqcap \\ &\quad \exists \text{tumeur.}(\exists \text{grade.}(\text{grade2} \sqcup \text{grade3}) \sqcup \exists \text{taille.RéelSupE1}) \\ P\text{-GNI-RHP-AI35-TGS2ouTTS1} &\sqsubseteq \exists \text{recommandation.ChimioNiv1} \end{aligned}$$

où *EntierInfE35* et *RéelSupE1* sont des types de données représentant respectivement les entiers inférieurs ou égaux à 35 et les réels supérieurs ou égaux à 1. De plus, le statut des récepteurs hormonaux est défini en fonction du statut des récepteurs aux œstrogènes et à la progestérone selon les axiomes

$$\begin{aligned} \text{RHPositifs} &\equiv \exists \text{récepteurOestrogène.Positif} \sqcup \exists \text{récepteurProgestérone.Positif} \\ \text{RHNégatifs} &\equiv \exists \text{récepteurOestrogène.Négatif} \sqcap \exists \text{récepteurProgestérone.Négatif} \end{aligned}$$

Appariement

Dans cette dernière étape, on cherche à établir des relations entre les quatre points de vue décrits précédemment, sous forme d'appariements. Chacune des ontologies locales représentées correspond à la vision d'une ou deux disciplines de la cancérologie, impliquées dans une phase du traitement particulière. Ces différentes phases de traitement correspondent à différents moments dans la prise en charge du patient. Ainsi, la chirurgie ayant lieu après la chimiothérapie néo-adjuvante, il semble naturel que les décisions en chirurgie s'appuient sur les décisions prises pour la chimiothérapie néo-adjuvante. De la même façon, les points de vue de la radiothérapie et du traitement médical adjuvant sont intéressés à échanger des informations avec la chirurgie. La figure 4.7 résume les appariements qu'il sera pertinent de mettre en œuvre selon ce principe. Dans cette figure, une flèche d'un contexte O_i vers un contexte O_j indique la mise en œuvre d'un appariement M_{ij} non-vide. Comme précédemment indiqué, nous n'utiliserons, pour décrire ces appariements, que des passerelles de types *dedans* ($\xrightarrow{\sqsubseteq}$) et *dessus* ($\xrightarrow{\sqsupseteq}$) entre les classes des contextes représentés.

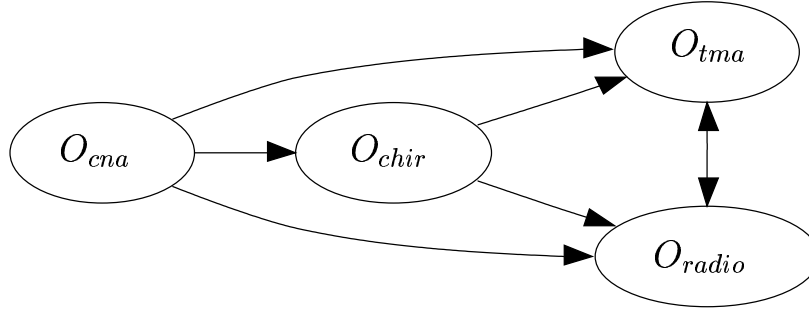


FIG. 4.7 – Appariements mis en œuvre dans la représentation du référentiel pour le traitement du cancer du sein.

Appariement de O_{cna} vers O_{chir} . Tout patient en chimiothérapie néo-adjuvante sera nécessairement, par la suite un patient en chirurgie. Par contre, les patients en chirurgie n'ont pas nécessairement été des patients en chimiothérapie néo-adjuvante. Ainsi, l'appariement de O_{cna} vers O_{chir} contient la passerelle

$$cna:Patient \xrightarrow{\sqsubseteq} chir:Patient$$

Par ailleurs, la chimiothérapie néo-adjuvante a pour objectif de réduire la taille de la tumeur afin de permettre une mastectomie partielle : il s'agit de rendre la tumeur *opérable d'emblée*. Ainsi, des passerelles permettent de déduire si la tumeur devra être considérée comme opérable d'emblée dans O_{chir} , en fonction du résultat de la chimiothérapie néo-ajuvante indiqué dans O_{cna} :

$$\begin{aligned}
 cna:RepSuf &\equiv \exists chimioNA. (ChimioNeoAdjuvante \sqcap \exists reponse. (\ni \text{ suffisante.true})) \\
 cna:RepInSuf &\equiv \exists chimioNA. (ChimioNeoAdjuvante \sqcap \exists reponse. (Progression \sqcup \\
 &\qquad \qquad \qquad \ni \text{ suffisante.false})) \\
 chir:TOE &\equiv \exists aTumeur. TumeurOpérableDEmblée \\
 chir:TNOE &\equiv \exists aTumeur. TumeurNonOpérableDEmblée \\
 cna:RepSuf &\xrightarrow{\sqsubseteq} chir:TOE \\
 cna:RepInSuf &\xrightarrow{\sqsubseteq} chir:TNOE
 \end{aligned}$$

Appariement de O_{chir} vers O_{radio} . La question du traitement radiothérapique ne se pose que pour les patients ayant subi une chirurgie. Ainsi, l'appariement de O_{chir} à O_{radio} contient la passerelle

$$chir:Patient \xrightarrow{\sqsupseteq} radio:Patient$$

Par ailleurs, les décisions en radiothérapie reposent sur les décisions concernant le traitement chirurgical. En effet, une des caractéristiques du patient prises en compte dans ce cadre est la chirurgie effectuée. Un patient pour qui une recommandation de mastectomie partielle est proposée en chirurgie sera considéré, dans le contexte de la radiothérapie, comme un patient ayant subi une chirurgie conservatrice. Ainsi, entre autres exemples, les passerelles

$$\begin{aligned}
 chir:\exists recommandation.MastectomiePartielle &\xrightarrow{\sqsubseteq} radio:\exists chirurgie.AblationConservatrice \\
 chir:\exists recommandation.MastectomieDePatey &\xrightarrow{\sqsubseteq} radio:\exists chirurgie.AblationNonConservatrice
 \end{aligned}$$

sont incluses dans l'appariement entre O_{chir} et O_{radio} .

Appariement de O_{chir} vers O_{tma} . De la même façon que pour la radiothérapie, un patient pour lequel la question du traitement médical adjuvant se pose est un patient pour lequel une chirurgie a été proposée. Ainsi, l'appariement de O_{chir} vers O_{tma} contient la passerelle

$$\text{chir:Patient} \xrightarrow{\exists} \text{tma:Patient}$$

Appariement de O_{cna} vers O_{radio} . Il est établi par le référentiel que tout patient ayant subi une chimiothérapie néo-adjuvante doit être traité en radiothérapie *comme si le statut ganglionnaire était positif*. Pour simplifier, cette règle est représentée dans l'appariement entre O_{cna} et O_{radio} par la passerelle

$$\text{cna:}\exists\text{chimioNA.ChimioNeoAdjuvante} \xrightarrow{\exists} \text{radio:}\exists\text{envahissementGanglionnaire.true}$$

Appariement de O_{cna} vers O_{tma} . La chimiothérapie est un traitement qui ne peut être effectué à trop grandes doses pour un même patient. Ainsi, le référentiel indique qu'un patient ayant été traité par chimiothérapie néo-adjuvante ne devra pas être traité ensuite par chimiothérapie dans le cadre du traitement médical adjuvant. Ceci peut-être représenté, entre autres exemples, par la passerelle :

$$\text{cna:}\exists\text{chimioNA.ChimioNeoAdjuvante} \xrightarrow{\exists} \text{tma:}\neg\exists\text{recommandation.Chimiothérapie}$$

Appariement de O_{radio} vers O_{tma} et de O_{tma} vers O_{radio} . Les décisions prises en radiothérapie n'ont pas d'influence sur celles concernant le traitement médical adjuvant, et vice versa. Néanmoins, certaines caractéristiques des patients, telles que le statut ganglionnaire ou la taille de la tumeur, sont représentées dans les deux contextes O_{radio} et O_{tma} . Ainsi, les passerelles sont utiles ici pour permettre le partage des informations sur les patients d'un contexte à l'autre. À titre d'exemple, dans l'appariement de O_{tma} à O_{radio} , les passerelles

$$\text{tma:}\exists\text{ganglionsEnvahis.0} \xrightarrow{\exists} \text{radio:}\exists\text{envahissementGanglionnaire.false}$$

$$\text{tma:}\exists\text{ganglionsEnvahis.EntierSupE1} \xrightarrow{\exists} \text{radio:}\exists\text{envahissementGanglionnaire.true}$$

permettent de compléter la description du patient concernant le statut ganglionnaire en radiothérapie en fonction de l'information sur le nombre de ganglions envahis telle que représentée selon le point de vue du traitement médical adjuvant. tma:EntierSupE1 est un type de données représentant les entiers supérieurs ou égaux à 1. Les passerelles inverses

$$\text{radio:}\exists\text{envahissementGanglionnaire.false} \xrightarrow{\exists} \text{tma:}\exists\text{ganglionsEnvahis.0}$$

$$\text{radio:}\exists\text{envahissementGanglionnaire.true} \xrightarrow{\exists} \text{tma:}\exists\text{ganglionsEnvahis.EntierSupE1}$$

sont incluses dans l'appariement entre O_{tma} et O_{radio} pour permettre le même type d'échange dans l'autre sens.

4.3.3 Raisonnement multi-points de vue pour l'application des référentiels

Le raisonnement global sur les classes (la subsomption) n'est pas utilisé dans l'exemple précédent. Les passerelles n'ont pas d'influence sur l'organisation des classes en hiérarchie dans ce cas. Le principal mécanisme de raisonnement multi-points de vue utilisé pour l'aide à la décision dans le cadre de l'application des référentiels est l'instanciation globale (voir la section 4.2.3). Pour comprendre de quelle

façon ce mécanisme permet de conclure à une solution reposant sur la combinaison de multiples points de vue, nous allons l'illustrer sur un exemple concernant le traitement du cancer du sein.

Imaginons le cas d'une patiente âgée de 33 ans et atteinte d'un cancer du sein. La tumeur de cette patiente n'est pas multi-focale, est opérable d'emblée, de grade 2, localisée dans le quadrant inféro-interne du sein et mesure 1,8 cm. Les récepteurs hormonaux sont positifs et le curage ne révélera pas de ganglion envahi.

Le médecin cherchant à établir un traitement pour cette patiente pourra la considérer selon un ou plusieurs des quatre points de vue définis, c'est-à-dire se placer dans un ou plusieurs des quatre contextes O_{cna} , O_{chir} , O_{radio} et O_{tma} . Pour cet exemple, nous cherchons à obtenir les recommandations du référentiel concernant la chirurgie, la radiothérapie et le traitement médical adjuvant, la chimiothérapie néo-adjuvante n'étant pas appliquée dans ce cas. La patiente sera donc représentée par trois instances locales, $chir:p$, $radio:p$ et $tma:p$, de la classe des patients. Différentes assertions sont déclarées dans les différents contextes pour traduire les caractéristiques décrites plus haut

O_{chir}	O_{radio}	O_{tma}
Patient(p)	Patient(p)	Patient(p)
TumeurOpérableDEmblée(t)	envahissementGanglionnaire(p, false)	Tumeur(t)
aTumeur(p, t)	Tumeur(t)	tumeur(p, t)
FoyerSimple(f)	aTumeur(p, t)	Grade2(g)
aFoyer(t, f)	Infero-Interne(l)	grade(t, g)
	localisation(t, l)	taille(t, 1.8)
		RHNegatifs(rh)
		récepteursHormonaux(p, rh)
		âge(p, 33)

De plus, afin d'indiquer que ces instances représentent différents points de vue sur la même patiente, des passerelles d'équivalences sont déclarées entre chacune d'elles :

$$\begin{array}{ll}
 chir:p \stackrel{\equiv}{\rightarrow} radio:p & chir:p \stackrel{\equiv}{\rightarrow} tma:p \\
 tma:p \stackrel{\equiv}{\rightarrow} radio:p & tma:p \stackrel{\equiv}{\rightarrow} chir:p \\
 radio:p \stackrel{\equiv}{\rightarrow} tma:p & radio:p \stackrel{\equiv}{\rightarrow} chir:p
 \end{array}$$

Sur la base de assertions décrites dans O_{chir} et des axiomes contenus dans la représentation du référentiel pour ce point de vue, il peut être déduit qu'une mastectomie partielle est recommandée. Dans O_{radio} , la passerelle :

$$chir:\exists recommendation.MastectomiePartielle \stackrel{\sqsubseteq}{\rightarrow} radio:\exists chirurgie.AblationConservatrice$$

permet de compléter les informations connues sur la patiente représentée par $radio:p$. Une radiothérapie du sein et de la chaîne mammaire interne élargie est donc recommandée. Dans O_{tma} , la passerelle

$$radio:\exists envahissementGanglionnaire.false \stackrel{\sqsubseteq}{\rightarrow} tma:\exists ganglionsEnvahis.0$$

permet de réutiliser l'information concernant l'envahissement ganglionnaire depuis O_{radio} . Une chimiothérapie de niveau 1 est alors recommandée pour la patiente représentée par $tma:p$. Finalement, la solution recommandée par le référentiel est représentée par trois instances de classes de traitements ($chir:MastectomiePartielle$, $radio:RadioSeinCMIE$ et $tma:ChimioNiv1$), c'est-à-dire trois points de vue sur la solution, qui se complètent pour former une solution globale.

4.3.4 Apports de la représentation multi-points de vue

Le premier critère que l'on peut chercher à évaluer concernant l'apport de la représentation multi-points de vue avec C-OWL est le gain en expressivité du langage. Mais la pertinence de ce critère reste sujette à discussion, d'autant que, d'un point de vue purement formel, le langage utilisé ici n'est pas plus expressif que OWL. En effet, il est montré dans [Borgida et Serafini, 2002] qu'une base de connaissances en LDD peut être transformée sans perte d'informations en une base de connaissances en LD classique, de type *SHIQ*. Ainsi, les contextes d'un espace de contextes en C-OWL peuvent être regroupés en une ontologie unique exprimée en OWL DL. Néanmoins, [Bouquet *et al.*, 2004] et [Serafini et Taminin, 2005] introduisent pour C-OWL un principe de *non-propagation de l'incohérence* qui n'est pas pris en compte dans cette transformation. L'incohérence d'un contexte en C-OWL ne doit pas avoir d'influence sur la cohérence des autres contextes de l'espace de contextes. Mais, ce principe n'est pas utilisé dans notre application. Nous nous intéressons en effet à construire des points de vue cohérents, localement autant que globalement.

La représentation de points de vue, telle que mise en œuvre grâce à C-OWL, a par contre un intérêt flagrant en terme d'ingénierie et de gestion des connaissances. La démarche proposée consiste en effet à construire un ensemble de représentations locales, correspondant chacune à la vision d'une partie du domaine, destinée à une tâche, une application ou un groupe de personnes particulier. Ces représentations sont ensuite appariées deux à deux. De cette façon, il n'est pas nécessaire de mettre en œuvre un consensus global à l'ensemble du domaine. Chaque point de vue peut se concentrer sur une représentation simple, homogène et la plus exhaustive possible des connaissances qu'il juge utiles selon son propre intérêt. Par ailleurs, un ensemble de contextes correspondant aux points de vue présents dans le domaine est plus simple à maintenir qu'une ontologie unique agrégeant l'ensemble de ces points de vue. En effet, une évolution dans les connaissances d'un point de vue particulier peut dans ce cas être réalisée localement, sans avoir à remettre en cause l'ensemble de la représentation. De la même façon, la représentation multi-points de vue simplifie l'utilisation et l'accès aux connaissances. Les acteurs du domaine, selon le point de vue qu'ils adoptent, ont la possibilité de se concentrer uniquement sur les contextes qu'ils jugent pertinents et obtenir de cette façon une représentation des connaissances adaptée à leurs usages. La collaboration et le partage de connaissances avec d'autres points de vue sont alors réalisés automatiquement, au travers des appariements.

À titre d'exemple, une représentation du référentiel pour le traitement du cancer du sein avait été construite précédemment, dans un langage proche des LD et n'intégrant pas la notion de point de vue [d'Aquin *et al.*, 2004a]. La construction de cette représentation supposait de mettre en commun l'ensemble des expertises du domaine. Il était par ailleurs nécessaire de choisir entre différentes représentations possibles pour certains éléments, le choix mis en œuvre ne correspondant pas nécessairement à l'habitude de l'ensemble des acteurs du domaine. Par exemple, du fait des besoins liés au traitement médical adjuvant, il a été décidé de représenter le statut ganglionnaire par le nombre de ganglions envahis, ce qui est en désaccord avec la représentation habituelle en radiothérapie. De plus, l'apparition d'une nouvelle technique, par exemple en chirurgie, supposait d'ajouter ou de mettre à jour des éléments de connaissances parmi les plus de 1000 classes qui composaient la représentation du référentiel. Une telle évolution pouvait potentiellement remettre en cause une grande partie de cette représentation, y compris des éléments sans rapport avec la chirurgie. Dans la représentation multi-points de vue de ce même référentiel, un point de vue apporte une représentation simple, composée seulement de quelques dizaines de classes et qui évolue indépendamment des autres points de vue.

Les mécanismes de raisonnements comme la subsomption ou l'instanciation sont reconnus comme particulièrement complexes en LD [Baader *et al.*, 2003] et donc en OWL. De ce fait, une des problématiques parmi les plus importantes à l'heure actuelle au sein du Web sémantique concerne le *passage à l'échelle*. Nous nous appuyons sur C-OWL pour construire des ontologies modulaires,

qui représentent les différents points de vue intervenant dans le domaine. Comme montré dans [Serafini et Tamilin, 2005], à propos du système DRAGO, C-OWL permet de mettre en œuvre des mécanismes de raisonnements distribués, où un moteur d'inférences est associé à chacun des contextes de l'espace de contextes. Les inférences globales, combinant les connaissances contenues dans plusieurs contextes, sont ainsi réalisées de façon décentralisée, par la communication « pair-à-pair » entre les moteurs d'inférences [Serafini et Tamilin, 2004]. La complexité du raisonnement et les ressources nécessaires peuvent de cette manière être distribuées entre les moteurs d'inférences, s'exécutant éventuellement sur différentes machines. De plus, les contextes d'un espace de contextes étant relativement indépendants les uns des autres, chaque moteur d'inférences pourra être optimisé pour ne prendre en compte que le niveau d'expressivité nécessaire à la représentation dans le contexte qui lui correspond.

Finalement, nous nous intéresserons dans le chapitre suivant aux apports de la représentation multi-points de vue dans le cadre de l'*adaptation* des référentiels. Un mécanisme de RÀPC décentralisé y est en effet défini sur la base d'une représentation en C-OWL des connaissances du domaine et des connaissances d'adaptation.

4.4 De l'approche ascendante de la construction d'ontologies

Dans ce chapitre, un exemple de représentation multi-points de vue du référentiel pour le traitement du cancer du sein a été présenté. Celui-ci regroupait quatre contextes, correspondant aux points de vue de différentes disciplines de la cancérologie, impliquées dans différentes phases du traitement. Mais, ainsi regroupés dans un espace de contextes, on peut remarquer que l'ensemble de ces quatre points de vue constitue lui même un point de vue particulier, coexistant avec d'autres : celui du *traitement du cancer du sein*. Il serait donc intéressant de relier ce point de vue, *englobant* les quatre déjà définis, avec ceux du diagnostic, de la surveillance ou des essais cliniques, eux même éventuellement constitués d'autres *sous-points de vue*. Il serait de cette façon établi le point de vue, encore plus englobant, de la *prise en charge de patients atteints de cancer du sein en Lorraine*, ce dernier pouvant lui même être mis en relation avec la prise en charge d'autres types de cancer ou avec la représentation de référentiels d'autres régions de France, d'autres pays, etc.

Le processus illustré ci-dessus sur l'exemple de la cancérologie relève de ce que nous pourrions appeler une *approche ascendante de la construction d'ontologies*. En comparaison, la démarche courante pour cela relève plutôt d'une approche descendante. Il s'agit en effet habituellement de construire une représentation la plus générale et partagée possible, couvrant une grande part du domaine, pour ensuite éventuellement la spécialiser pour des applications particulières. Ce que nous envisageons ici consiste à l'inverse à considérer en premier lieu la représentation de connaissances locales, de points de vue élémentaires, et de les combiner à l'aide des appariements pour former des représentations de plus en plus générales. On obtient de cette façon une ontologie couvrant la globalité du domaine et composée d'ontologies locales, directement opérationnelles.

Concernant les formalismes de représentation, deux de ceux mentionnés au début de ce chapitre permettent d'organiser les points de vue en hiérarchie : [Attardi et Simi, 1995] et [McCarthy, 1993]. En particulier, John McCarthy développe l'idée que tout contexte *c* est nécessairement placé dans un *contexte englobant*, qui explicite les éléments laissés implicites dans *c* (les *suppositions*). Dans le contexte de la poste, par exemple, la notion d'adresse correspond implicitement à celle d'adresse postale, supposition qu'il faudra expliciter dans un contexte plus général, s'intéressant aussi aux adresses électroniques. Un autre exemple, en cancérologie, concerne la notion de patient. Un patient est implicitement considéré comme quelqu'un pour lequel on suspecte un cancer du sein dans le contexte du diagnostic. Dans celui du traitement, un patient est une personne pour laquelle un cancer du sein a été diagnostiqué. A l'heure actuelle, les contextes d'un espace de contextes en C-OWL ne sont pas organisés en hiérarchie. Un

contexte englobant serait dans ce formalisme représenté au même niveau que ceux qu'il englobe. Il serait alors intéressant d'inclure une représentation explicite de ce type de hiérarchisation des contextes à C-OWL, afin de permettre la construction ascendante d'ontologies telle que décrite ici.

Chapitre 5

Raisonnement à partir de cas décentralisé selon plusieurs points de vue

Comme nous l'avons montré au chapitre 3, le processus de RÀPC repose sur trois types de connaissances : les connaissances du domaine, les connaissances d'adaptation et les cas. Par ailleurs, nous avons vu dans le chapitre précédent les bénéfices qu'apporte la représentation explicite de multiple points de vue au sein d'ontologies contenant les connaissances du domaine. En particulier dans le cadre du projet KASIMIR, cela permet de prendre en compte les différents points de vue que constituent les disciplines de la cancérologie. Les mécanismes de raisonnements utilisés pour l'application des référentiels peuvent ainsi s'appuyer sur ces points de vue et sur les relations qui les lient. Mais les connaissances spécifiques au RÀPC, en particulier les connaissances d'adaptation, sont elles aussi dépendantes du point de vue adopté. Un chirurgien cherchant à établir un traitement chirurgical ne considérera pas comme étant similaires les mêmes patients qu'un radiothérapeute s'intéressant à établir une radiothérapie. Dans ce chapitre nous proposons un mécanisme de RÀPC *décentralisé* exploitant des connaissances multi-points de vue, telles qu'elles sont représentées en C-OWL. Celui-ci est appliqué comme mécanisme de raisonnement pour l'aide à la décision dans le cadre de l'adaptation des référentiels. Dans cette optique, chaque point de vue contient les connaissances de domaine et les connaissances d'adaptation utiles à la résolution d'un problème, la prise en charge d'un patient, selon une discipline particulière. Mais au-delà de l'application en cancérologie, nous cherchons à répondre ici, au travers du RÀPC décentralisé, au problème du raisonnement adaptatif dans l'environnement multi-ontologies du Web sémantique.

5.1 Pourquoi prendre en compte les points de vue en RÀPC

La première motivation pour la prise en compte des points de vue au sein du processus de RÀPC provient de l'application à l'aide à la décision dans le cadre de l'adaptation des référentiels. En effet, lorsqu'un patient sort du cadre des référentiels, c'est-à-dire que ceux-ci n'apportent pas de solution satisfaisante au problème que pose la prise en charge du patient, la décision à prendre est établie par adaptation des solutions fournies par les référentiels, au cours d'une RCP. Ces réunions regroupent des médecins de différentes disciplines et chaque membre d'une RCP apporte sa propre part à la solution, selon les éléments de connaissances propres à sa discipline d'appartenance et en collaborant avec les autres médecins, d'autres disciplines. C'est cette multiplicité de vues dans le processus de résolution de problèmes que l'on souhaite prendre en compte et utiliser dans le cadre du RÀPC. Mais, indépendamment de l'application, asseoir un processus de RÀPC décentralisé sur une représentation multi-points de vue des connaissances relève de plusieurs approches et de plusieurs problématiques du RÀPC en général.

5.1.1 Raisonner à partir de plusieurs cas

Selon [Gebhardt *et al.*, 1997] un cas est dit « complexe » lorsqu'il possède notamment la particularité d'être utilisable de plusieurs façons et qu'il existe ainsi différentes façons de le considérer. Dans les domaines où les cas sont complexes, un seul cas source n'est pas nécessairement suffisant pour résoudre un problème. Pour cette raison, des travaux se sont intéressés à la mise en œuvre de processus de RÀPC dans lesquels plusieurs cas sources sont extraits de la base de cas au cours de la remémoration, chacun couvrant une part de la question posée. La réutilisation de ces multiples cas sources, au cours de l'adaptation, nécessite alors la mise en œuvre de mécanismes permettant de les combiner pour former une solution globale cohérente. Au sein du système DÉJÀ VU [Smyth, 1996], par exemple, le problème cible est préalablement *décomposé* en différents sous-problèmes, sur la base d'une hiérarchie de problèmes abstraits, c'est-à-dire ne contenant que les caractéristiques utiles à la résolution d'un sous-problème particulier. Un cas source différent est remémoré pour chacun de ces sous-problèmes et la solution de celui-ci est adaptée pour former une solution au sous-problème considéré. Les multiples solutions obtenues sont finalement intégrées comme différentes parties de la solution globale. Dans le système EADOCs [Netten et Vingerhoeds, 1996], les différents cas sources ne sont pas utilisés simultanément et combinés par la suite, comme dans DÉJÀ VU, mais intégrés au fur et à mesure, de façon incrémentale, dans la construction de la solution. Enfin, une autre approche pour la réutilisation de plusieurs cas consiste à considérer les cas sources remémorés non pas comme des éléments à intégrer, mais comme des alternatives. C'est ce que fait le système SAXEX [Arcos et Lopez de Mántaras, 1997] par exemple, qui s'appuie sur différentes représentations du problème cible, appelées *perspectives*, pour retrouver différents cas sources et finalement n'utiliser que celui qu'il considère comme étant le « meilleur », c'est-à-dire le plus similaire au problème cible.

5.1.2 Raisonner avec plusieurs vues sur la similarité

Les multiples vues selon lesquelles on peut considérer les cas complexes amènent à la nécessité de combiner plusieurs « fonctions » de similarité, c'est-à-dire plusieurs façons d'appréhender la similarité entre deux cas [Börner, 1998]. Par exemple, au sein du système FABEL, les cas sont considérés selon différents *aspects* (voir par exemple [Gebhardt *et al.*, 1997] et [Börner, 1998]). Un aspect est composé de deux fonctions, une fonction de représentation et une fonction de « distance ». La *fonction de représentation* transforme la représentation originelle d'un cas pour la restreindre à certaines caractéristiques. La *fonction de distance* permet de mesurer la similarité entre deux cas tels qu'ils sont vus dans l'aspect considéré, c'est-à-dire dont la représentation est restreinte par la fonction de représentation. Finalement, la similarité globale entre deux cas correspond à une combinaison des similarités calculées selon chacun des aspects, en prenant en compte une notion d'*importance d'un aspect*. De plus, disposer de plusieurs façons de mesurer la similarité entre cas permet de mettre en œuvre différents index pour les cas stockés, apportant par ce biais différentes vues sur la base de cas et différentes façons d'y accéder (voir [Risland *et al.*, 1993] pour un exemple de remémoration guidée par plusieurs index).

5.1.3 Raisonnement décentralisé

L'intelligence artificielle décentralisée, telle qu'elle est présentée dans [Demazeau et Müller, 1989], concerne l'activité d'agents autonomes, ayant leurs propres connaissances et leurs propres buts, qui coexistent et collaborent avec d'autres agents du même type. De la même façon, en s'appuyant sur les différents points de vue présents dans le domaine, on cherche à construire un système dans lequel le RÀPC est à la fois local à un point de vue et où chacun des points de vue collabore pour construire une solution globale. Un premier avantage de ce type d'approche concerne le passage à l'échelle. Dans chaque point

de vue le RÀPC se focalise sur un sous-ensemble des connaissances disponibles et fonctionne simultanément à d'autres processus de RÀPC s'exécutant éventuellement sur différentes machines. De plus, l'article [Leake et Sooriamurthi, 2003] montre que distribuer de cette façon les connaissances utiles au RÀPC (dans le cas de [Leake et Sooriamurthi, 2003], la base de cas) permet d'améliorer la qualité des résultats. En effet, l'hypothèse sur laquelle repose le RÀPC est que des problèmes similaires ont des solutions similaires, cette supposition étant plus facilement vérifiable dans des domaines restreints. Distribuer les connaissances utiles au RÀPC selon leur capacité à répondre à un problème particulier, selon un point de vue donné, permet ainsi d'obtenir de meilleures solutions locales et dont la combinaison en une solution globale est de meilleure qualité que si celle-ci avait été construite de façon centralisée. Par ailleurs, mettre en œuvre un système de RÀPC sur la base des points de vue décentralisés permet d'exploiter les liens existant entre les points de vue comme des connaissances utiles à la résolution de problèmes, à la manière de ce qui est réalisé dans les systèmes multi-agents où l'interaction entre agents représente un aspect central (voir [Nagendra Prasad *et al.*, 1996] et [Ontañón et Plaza, 2003] pour des exemples de RÀPC multi-agents).

5.1.4 RÀPC dans le cadre du Web sémantique

Le Web est un environnement naturellement peu structuré et distribué. En particulier, coexistent au sein du Web sémantique de multiples ontologies concernant un même domaine, mais ne couvrant pas nécessairement les mêmes aspects, les mêmes tâches ou le même niveau de granularité. La mise en œuvre d'un mécanisme de RÀPC pour le Web sémantique doit ainsi pouvoir bénéficier de la prise en compte de ces différents points de vue, représentés par autant d'ontologies inter-connectées. Il s'agit de résoudre un problème sur la base de connaissances locales à un point de vue, une ontologie, et en sollicitant les autres points de vue lorsque cela est utile, à la manière de ce qui est réalisé dans les *systèmes de raisonnements pair-à-pair* (*peer-to-peer inference systems*, voir par exemple [Adjiman *et al.*, 2004]). Par exemple, dans le cadre des logiques de descriptions distribuées, le système DRAGO [Serafini et Tamilin, 2005] implémente un mécanisme de raisonnement où différents moteurs d'inférences sont utilisés pour raisonner sur différentes ontologies. Un moteur d'inférences particulier peut faire appel à un autre lorsque des connaissances provenant d'une autre ontologie lui sont utiles pour réaliser des inférences au sein de l'ontologie locale.

5.2 RÀPC décentralisé avec C-OWL

La prise en compte des points de vue au sein du processus de RÀPC a en premier lieu été étudiée dans le cadre de la représentation des connaissances par objets [d'Aquin *et al.*, 2002, d'Aquin *et al.*, 2004c], sur la base d'un langage proche de TROEPS ([Mariño, 1993], voir chapitre 4). Ces travaux ont ensuite été repris sur la base du langage C-OWL [d'Aquin *et al.*, 2005b], celui-ci étant plus général que TROEPS et naturellement intégré aux technologies du Web sémantique. Nous nous concentrerons donc ici sur la présentation du RÀPC décentralisé avec C-OWL, tel qu'il est décrit dans [d'Aquin *et al.*, 2005b].

Rappel sur C-OWL. Les connaissances en C-OWL sont représentées au sein de plusieurs ontologies locales, appelées contextes (voir section 4.2). Les éléments de connaissances, c'est-à-dire les classes, les propriétés et les individus, sont préfixés selon le contexte dans lequel ils sont définis. Par exemple, $i : a$, $i : C \sqcap \exists p.D$ ou $i : C \sqsubseteq D$ sont des expressions provenant du contexte O_i . Enfin, les éléments de deux contextes différents (O_i et O_j) peuvent être liés par des *passerelles* de la forme $i : C \sqsubseteq j : D$, $i : C \sqsupseteq j : D$ ou $i : a \sqsupseteq j : b$. Les inférences réalisées au travers de ces passerelles sont résumées en section 4.2.3.

5.2.1 Principe général

Les contextes de C-OWL sont utilisés ici pour représenter les différents points de vue intervenant dans le raisonnement. Chaque contexte contient les connaissances (du domaine, d'adaptation et les cas) utiles à la résolution d'un problème local, c'est-à-dire relevant d'un certain point de vue. Ainsi, le problème cible à résoudre est représenté par un ensemble $\{i:cible\}_i$ d'individus correspondant aux problèmes locaux, avec autant de problèmes locaux $i:cible$ qu'il y a de points de vue (contextes) O_i à considérer. De plus, afin d'exprimer le fait que ces problèmes locaux correspondent à différentes vues sur un seul et même problème, des passerelles $i:cible \xrightarrow{\equiv} j:cible$ sont déclarées pour chaque couple de contextes (O_i, O_j) présents au sein de cet espace de contextes (avec $i \neq j$).

Le mécanisme de RÀPC permettant de résoudre un problème cible représenté par un ensemble de problèmes locaux est conçu comme un processus *décentralisé* : il est à la fois *local* à chaque point de vue et *collaboratif*, dans le sens où il repose sur le partage de connaissances entre points de vue. Le RÀPC décentralisé est ainsi fondé sur deux opérations principales, appliquées de façon distribuée dans chacun des contextes O_i de l'espace de contextes :

- (a) *Le RÀPC local* qui correspond au processus de RÀPC habituel, tel qu'il a été vu au chapitre 3, appliqué sur les connaissances locales au point de vue considéré (représentées par le contexte O_i) dans le but de construire une solution locale $i:Sol(cible)$ au problème local $i:cible$.
- (b) *La complétion du cas cible* qui correspond à la part collaborative du RÀPC décentralisé. Il s'agit ici de compléter les informations connues sur le cas cible local, concernant le problème $i:cible$ ou sa solution $i:Sol(cible)$, en faisant appel aux connaissances et aux inférences dans d'autres points de vue grâce aux passerelles.

Ces deux opérations sont appliquées dans chaque contexte tant que de nouvelles inférences peuvent être réalisées. La solution correspond finalement à l'ensemble $\{i:Sol(cible)\}_i$ des solutions locales, construites à la fois selon les connaissances locales à chaque point de vue et par la combinaison des points de vue au travers des passerelles.

5.2.2 RÀPC local à un point de vue

Un contexte O_i contient les connaissances nécessaires à la résolution d'un problème local $i:cible$, afin d'établir une solution locale $i:Sol(cible)$. Il est donc considéré ici qu'un problème vu selon un certain point de vue est résolu par ce même point de vue sur la solution. Au sein d'un contexte O_i , les cas sources sont ainsi représentés par des couples $i:(source, Sol(source))$, liant un problème source local $i:source$ à sa solution locale $i:Sol(source)$ ⁸. Selon ce même principe, les reformulations sont elles aussi considérées comme des connaissances locales, décrites sous la forme $i:(r, \mathcal{A}_r)$ et liant des relations $i:r$ entre problèmes à des fonctions d'adaptation $i:\mathcal{A}_r$.

Un contexte O_i en C-OWL correspond à une ontologie et contient ainsi une hiérarchie de classes pouvant être utilisées pour indexer les cas sources de la base de cas. Un index $i:idx(source)$ correspond à une abstraction d'un problème source $i:source$, ne retenant que les informations pertinentes selon le point de vue de O_i , c'est-à-dire celles amenant à la solution $i:Sol(source)$. Ainsi, une solution $i:Sol(source)$ d'un problème source peut être utilisée pour résoudre un problème $i:cible$ si $i:cible$ est reconnu comme étant une instance de la classe $i:idx(source)$, index de $i:source$. Selon ce principe, le mécanisme d'instanciation est utilisé lors de la phase de remémoration du processus de RÀPC local à un contexte O_i afin de retrouver un index de problème source à réutiliser. Plus précisément, il s'agit de construire un chemin de similarité liant un problème $i:pb_0$, instance de la classe

⁸La représentation des connaissances utiles au RÀPC en C-OWL repose directement sur la représentation en OWL détaillée en section 3.2.2. Pour une question de simplicité, nous utiliserons ici les notations propres au RÀPC pour désigner ces connaissances et non leurs correspondances en OWL.

index $i:\text{idx}(\text{source})$ d'un problème $i:\text{source}$, au problème $i:\text{cible}$. Ce chemin de similarité est alors composé de relations $i:r_k$ définies dans O_i :

$$i:\text{source} \xrightarrow{\text{isa}} i:\text{idx}(\text{source}) \xleftarrow{\text{isa}} i:\text{pb}_0 \ i:r_1 \ \text{pb}_1 \ \dots \ \text{pb}_{q-1} \ i:r_q \ i:\text{cible}$$

où $\xrightarrow{\text{isa}}$ signifie « est une instance de ». Sur la base de ce chemin de similarité, la phase d'adaptation du processus de RÀPC local à O_i a pour charge de construire un chemin d'adaptation liant $i:\text{Sol}(\text{source})$ à $i:\text{Sol}(\text{cible})$ et composé de fonctions d'adaptation locales $i:\mathcal{A}_{r_k}$. Celles-ci sont issues des reformulations $i:(r_k, \mathcal{A}_{r_k})$ qui constituent les connaissances d'adaptation propres à O_i . Chaque processus de RÀPC local est ainsi « contenu » dans un point de vue particulier, s'appuyant sur ses propres connaissances pour construire son point de vue sur la solution.

5.2.3 Combinaison des points de vue selon les passerelles

Les passerelles correspondent à des relations sémantiques entre des éléments de connaissances décrits dans différents contextes. Dans le cadre du RÀPC décentralisé, elles permettent de représenter l'influence d'un point de vue sur un autre. Nous montrons ci-dessous, au travers d'un exemple, comment les passerelles sont utilisées conjointement au RÀPC local pour combiner les points de vue. Il s'agit, sur la base du raisonnement distribué en C-OWL (voir section 4.2.3), de compléter la description du problème $i:\text{cible}$ et de sa solution $i:\text{Sol}(\text{cible})$ dans un contexte O_i , en fonction des inférences réalisées dans d'autres contextes.

Pour cet exemple, nous considérons trois contextes appelés O_1 , O_2 et O_3 . Un problème source est représenté dans ce cadre par sa classe index, au travers d'un axiome *problème-solution* (voir section 3.2.3). Par exemple, l'expression $1:I1 \equiv \text{Problème} \sqcap \exists p1.C1$ représente un problème source dans le contexte O_1 et l'axiome problème-solution $1:I1 \sqsubseteq \exists a\text{PourSolution}.S1$ permet d'associer à toute instance de la classe de problèmes $1:I1$ une instance de la classe de solutions $1:S1$. De la même manière, les deux problèmes sources (classes index) $2:I2$ et $3:I3$ sont respectivement définis dans O_2 et O_3 (voir première et deuxième lignes de la figure 5.1). Comme indiqué dans la section 5.2.1, le problème cible est représenté par trois instances locales de la classe *Problème* (cinquième ligne de la figure 5.1) et ces instances sont reliées par des passerelles d'équivalence, afin de préciser qu'il s'agit de trois points de vue différents sur le même problème global (troisième ligne de la figure 5.1). Des passerelles entre certaines classes des différents contextes ont de plus été déclarées (quatrième ligne de la figure 5.1), permettant ainsi le partage de connaissances entre points de vue. Finalement, la description du problème cible est réalisée au travers d'un ensemble d'assertions (cinquième, sixième et septième lignes de la figure 5.1).

Les différentes étapes du processus de RÀPC décentralisé, décrites entre la huitième et la onzième ligne de la figure 5.1, sont détaillées ci-dessous. Comme point de départ des trois processus de RÀPC locaux, les problèmes $1:\text{cible}$, $2:\text{cible}$ et $3:\text{cible}$ sont d'abord instanciés dans leurs contextes respectifs.

- Dz1. Dans le contexte O_2 , $2:\text{cible}$ est reconnu comme une instance de la classe $2:\exists p21.C21$.
- Dz2. Les deux passerelles $2:\exists p21.C21 \stackrel{\sqsubseteq}{\rightarrow} 1:\exists p1.C1$ et $2:\text{cible} \stackrel{\equiv}{\rightarrow} 1:\text{cible}$ permettent de compléter la description de $1:\text{cible}$ en fonction des connaissances contenues dans O_2 concernant $2:\text{cible}$. $1:\text{cible}$ est alors reconnu comme une instance de la classe $1:\exists p1.C1$ et donc de la classe $1:I1$.
- Dz3. Au travers de l'axiome problème-solution $1:I1 \sqsubseteq \exists a\text{PourSolution}.S1$, une solution, instance de la classe $1:S1$, est associée au problème $1:\text{cible}$. Celui-ci est alors considéré comme une instance de la classe $1:\exists a\text{PourSolution}.S1$.
- Dz4. La description du problème $2:\text{cible}$ dans O_2 peut être complétée selon les deux passerelles $1:\exists a\text{PourSolution}.S1 \stackrel{\sqsubseteq}{\rightarrow} 2:\exists p23.C23$ et $1:\text{cible} \stackrel{\equiv}{\rightarrow} 2:\text{cible}$. L'individu $2:\text{cible}$ est ainsi reconnu comme une instance de la classe $2:\exists p23.C23$.

	O_1	O_2	O_3
1	$I1 \equiv \text{Problème} \sqcap \exists p1.C1$	$I2 \equiv \text{Problème} \sqcap \exists p21.C21 \sqcap \exists p22.C22$	$I3 \equiv \text{Problème} \sqcap \exists p3.C3$
2	$I1 \sqsubseteq \exists a\text{PourSolution}.S1$	$I2 \sqsubseteq \exists a\text{PourSolution}.S21$	$I3 \sqsubseteq \exists a\text{PourSolution}.S31$
3	$2:\text{cible} \xrightarrow{\equiv} 1:\text{cible}$	$1:\text{cible} \xrightarrow{\equiv} 2:\text{cible}$	$2:\text{cible} \xrightarrow{\equiv} 3:\text{cible}$
4	$2:\exists p21.C21 \xrightarrow{\sqsubseteq} 1:\exists p1.C1$	$1:\exists a\text{PourSolution}.S1 \xrightarrow{\sqsubseteq} 2:\exists p23.C23$	$2:\exists a\text{PourSolution}.S22 \xrightarrow{\sqsubseteq} 3:\exists a\text{PourSolution}.S32$
5	Problème(cible)	Problème(cible)	Problème(cible)
6		C21(a)	
7		p21(cible, a)	
8		Dz1. $\exists p21.C21(\text{cible})$	
9	Dz2. $\exists p1.C1(\text{cible})$		
10	Dz3. $\exists a\text{PourSolution}.S1(\text{cible})$	Dz4. $\exists p23.C23(\text{cible})$	
11		Dz5. $\exists a\text{PourSolution}.S22(\text{cible})$	Dz6. $\exists a\text{PourSolution}.S32(\text{cible})$

FIG. 5.1 – Un exemple de RÀPC décentralisé. Les première et deuxième lignes introduisent des problèmes sources. Les troisième et quatrième lignes définissent des appariements associés aux contextes. De la cinquième à la septième ligne sont décrites des assertions concernant le problème cible. Les différentes étapes du RÀPC décentralisé sur cet exemple sont finalement décrites entre la huitième et la onzième ligne.

Dz5. Comme expliqué plus bas, le processus de RÀPC local au contexte O_2 est alors en mesure de construire une solution au problème $2:\text{cible}$, sous la forme d'une instance de $2:S22$. $2:\text{cible}$ peut ainsi être reconnu comme une instance de $2:\exists a\text{PourSolution}.S22$.

Dz6. Finalement dans O_3 , il est inféré que $3:\text{cible}$ est une instance de $3:\exists a\text{PourSolution}.S32$ sur la base des passerelles $2:\exists a\text{PourSolution}.S22 \xrightarrow{\sqsubseteq} 3:\exists a\text{PourSolution}.S32$ et $2:\text{cible} \xrightarrow{\equiv} 3:\text{cible}$.

La solution du problème cible représenté par les trois problèmes locaux $1:\text{cible}$, $2:\text{cible}$ et $3:\text{cible}$ correspond ainsi à la combinaison de trois solutions locales, représentées par trois instances, respectivement de $1:S1$, $2:S22$ et $3:S32$.

Parmi l'ensemble des étapes de ce processus, les étapes Dz3. et Dz5. se distinguent comme relevant du RÀPC *local*, s'appuyant sur les connaissances définies dans le contexte considéré (respectivement O_1 et O_2) pour établir une solution au problème local. Les étapes Dz2., Dz4. et Dz6. sont quant à elles des étapes de *complétion du cas cible*, reposant sur les passerelles pour inférer des connaissances dans le contexte local, selon les connaissances contenues dans d'autres contextes.

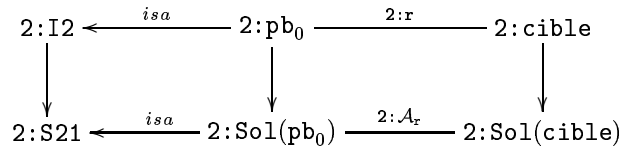


FIG. 5.2 – Le chemin de similarité et le chemin d'adaptation du RÀPC local à O_2 .

Détails du processus de RÀPC local Dz5. Afin d'établir par adaptation une solution pour le problème $2:\text{cible}$ le processus de RÀPC local à O_2 (étape Dz5.) s'appuie sur une reformulation de la forme $2:(r, \mathcal{A}_r)$ (voir figure 5.2). À ce moment du processus de RÀPC décentralisé, $2:\text{cible}$ a été reconnu comme une instance de la classe $2:\text{Problème} \sqcap \exists p21.C21 \sqcap \exists p23.C23$. Durant la phase de remémoration, la relation $2:r$ entre problèmes amène à la création d'un problème intermédiaire $2:pb_0$ à

partir de $1:cible$. La relation $2:r$ est telle que la différence entre $2:pb_0$ et $2:cible$ tient dans le fait que le premier est une instance de $2:\exists p22.C22$ alors que le second est une instance de $2:\exists p23.C23$. $2:pb_0$ peut ainsi être reconnu comme une instance de la classe $2:Problème \sqcap \exists p21.C21 \sqcap \exists p22.C22$, et donc de $2:I2$. Sur la base de l'axiome problème-solution $2:I2 \sqsubseteq \exists aPourSolution.S21$ défini dans O_2 , $2:pb_0$ est alors associé à une solution $2:Sol(pb_0)$, instance de $2:S21$. La fonction d'adaptation $2:A_r$ est utilisée dans le chemin d'adaptation pour construire une solution $2:Sol(cible)$ au problème $2:cible$, à partir de la solution $2:Sol(pb_0)$ de $2:pb_0$. Cette fonction est telle que la différence entre ces deux individus tient dans le fait que $2:Sol(pb_0)$ est une instance de $2:S21$, alors que $2:Sol(cible)$ est une instance de $2:S22$. La solution du problème local $2:cible$ est ainsi une instance $2:Sol(cible)$ de la classe $2:S22$.

5.3 Exemple d'application dans le cadre du traitement du cancer du sein

Dans cette section, nous présentons un exemple d'application du mécanisme de RÀPC décentralisé pour l'aide à la décision dans le cadre de l'adaptation des référentiels. Cet exemple conserve la même structure et s'appuie sur les mêmes inférences que celui détaillé dans la section précédente (voir figure 5.1), les éléments de connaissances utilisés étant simplement remplacés par d'autres, provenant du domaine du traitement du cancer du sein. Il faut remarquer que cet exemple, bien qu'inspiré de cas réels, est fortement modifié et arrangé afin d'illustrer de façon la plus concise possible le mécanisme de RÀPC décentralisé appliqué à l'aide à la décision dans le cadre du traitement du cancer du sein.

La base de cas et le modèle des connaissances du domaine sont ici contenus dans une représentation en C-OWL du référentiel pour le traitement du cancer du sein. Dans l'exemple qui suit, trois contextes, O_{cna} , O_{chir} et O_{radio} , sont considérés, afin de représenter les trois points de vue respectivement de la chimiothérapie néo-adjuvante, de la chirurgie et de la radiothérapie. Ces trois contextes correspondent respectivement aux contextes O_1 , O_2 et O_3 de l'exemple précédent. La représentation du référentiel dans un contexte O_i peut être vue comme un ensemble d'axiomes problème-solution de la forme $i:P \sqsubseteq recommandation.T$, où $i:P$ est une classe de patients, sous-classe de $i:Patient$, $i:T$ est une classe de traitements, sous-classe de $i:Traitement$, et $i:recommandation$ est la propriété correspondant à $i:aPourSolution$. Par exemple, on considérera dans O_{cna} que les patients ayant une tumeur de taille inférieure à 4 cm doivent être traités par des cures de chimiothérapie néo-adjuvante. Ainsi, l'axiome problème-solution $1:I1 \sqsubseteq aPourSolution.S1$ de l'exemple de la figure 5.1 se traduit dans ce cadre par :

$$cna:Patient \sqcap \exists tailleTumeur.inf4 \sqsubseteq \exists recommandation.ChimioNéoAdjuvante$$

avec $inf4$ le type des réels inférieurs à 4. De la même façon, O_{chir} contient l'axiome

$$chir:Patient \sqcap \exists aTumeur.(\exists taille.supe4) \sqcap \exists chimioNéoAdjuvante.true \sqsubseteq \exists recommandation.MastectomieDePatey$$

signifiant que pour les patients ayant une tumeur d'au moins 4 cm ($supe4$ étant le type des réels supérieurs ou égaux à 4) et pour qui une chimiothérapie néo-adjuvante doit être appliquée, une mastectomie de Patey, c'est-à-dire une ablation totale du sein associée à un curage axillaire complet, est recommandée. Dans O_{radio} , l'axiome

$$radio:Patient \sqcap \exists envahissementGanglionnaire.true \sqsubseteq \exists recommandation.RadioCMI$$

indique que pour les patients ayant des ganglions infectés, une irradiation de la chaîne mammaire interne est recommandée.

Les passerelles entre classes indiquées dans le figure 5.1 peuvent ainsi être traduites dans les trois contextes O_{cna} , O_{chir} et O_{radio} de la façon suivante :

[p1] $chir:\exists aTumeur.(\exists taille.inf4) \stackrel{\equiv}{\rightarrow} cna:\exists tailleTumeur.inf4$

[p2] $cna:\exists recommandation.ChimioNéoAdjuvante \stackrel{\equiv}{\rightarrow} chir:\exists chimioNéoAdjuvante.true$

[p3] $chir:\exists recommandation.MastectomieTotale \stackrel{\equiv}{\rightarrow} radio:\exists recommandation.RadioParoi$

La première passerelle ([p1]) permet à O_{chir} de partager les connaissances concernant les tailles de tumeur des patients avec O_{cna} . La description du problème en chirurgie peut être complétée selon la solution trouvée en chimiothérapie néo-adjuvante sur la base de la deuxième passerelle ([p2]). La troisième passerelle ([p3]) exprime le fait que, quand une ablation totale est recommandée en chirurgie, une irradiation de la paroi de la zone extraite doit être effectuée en radiothérapie.

De plus, dans le contexte O_{chir} , des connaissances d'adaptation sont déclarées, sous la forme d'une reformulation $chir:(r, \mathcal{A}_r)$. La relation $chir:r$ entre problèmes s'applique entre une instance de $chir:Patient$ ayant une tumeur de petite taille (inférieure à 4 cm), mais couvrant une part importante du sein (plus de 60%), et une instance de $chir:Patient$ ayant une tumeur de taille importante (4 cm et plus). En d'autres termes, un patient ayant une petite tumeur dans un sein peu volumineux est considéré, pour la chirurgie, comme similaire à un patient ayant une tumeur de grande taille. L'application de la fonction d'adaptation $chir:\mathcal{A}_r$ consiste à supprimer le curage d'une mastectomie de Patey, c'est-à-dire à la remplacer par une mastectomie totale (sans curage axillaire).

Dans cet exemple, nous considérons un problème cible représenté par trois problèmes locaux : $cna:cible$, $chir:cible$ et $radio:cible$. Ceux-ci sont reliés par des passerelles d'équivalence ($\stackrel{\equiv}{\rightarrow}$) comme indiqué à la section 5.2.1 et dans l'exemple de la figure 5.1. Chacun de ces individus locaux est une instance de la classe *Patient* (et donc aussi de la classe *Problème*), c'est-à-dire que les assertions $cna:Patient(cible)$, $chir:Patient(cible)$ et $radio:Patient(cible)$ sont réalisées au sein de O_{cna} , O_{chir} et O_{radio} respectivement. De plus, $chir:cible$ est décrit comme représentant un patient ayant une petite tumeur couvrant une grande partie du sein, c'est-à-dire que O_{chir} contient l'assertion $chir:\exists aTumeur.(\exists taille.inf4 \sqcap \exists couvre.supe60)(cible)$, avec *supe60* le type des réels supérieurs ou égaux à 60.

Suivant l'ensemble des « renommages » indiqués, la résolution de ce problème cible suit les six étapes décrites à la section précédente. L'information concernant la taille de la tumeur est dans un premier temps partagée entre les contextes O_{chir} et O_{cna} , grâce à la passerelle [p1]. Une chimiothérapie néo-adjuvante est alors recommandée pour le patient représenté par $cna:cible$, selon les connaissances contenues dans O_{cna} . La passerelle [p2] permet alors de compléter la description du problème local $chir:cible$, en fonction de la solution établie en chimiothérapie néo-adjuvante. Dans O_{chir} , la reformulation $chir:(r, \mathcal{A}_r)$ est alors appliquée dans un processus de RÀPC local, considérant $chir:cible$ comme similaire à un patient ayant une tumeur de grande taille. Selon l'axiome problème-solution contenu dans O_{chir} , la solution pour ce type de patients correspond à une mastectomie de Patey. Cette solution est ainsi remplacée par une mastectomie totale, selon la fonction d'adaptation $chir:\mathcal{A}_r$, pour devenir une solution de $chir:cible$. Finalement, la solution établie pour la chirurgie, la mastectomie totale, permet d'appliquer la passerelle [p3] afin d'établir une solution en radiothérapie.

Un point intéressant de cet exemple est que le problème cible est considéré de façons différentes, selon trois points de vue différents. En particulier, il est vu comme un patient ayant une petite tumeur pour la chimiothérapie néo-adjuvante (O_{cna}) et comme un patient ayant une tumeur de grande taille pour la chirurgie (O_{chir}).

5.4 Note sur l'implémentation

Au chapitre 3, un système de RÀPC s'appuyant sur des connaissances du domaine, d'adaptation et des cas représentés en OWL a été décrit sous la forme d'un service Web. La représentation des connaissances utiles au RÀPC en OWL ainsi que l'architecture fondée sur les services Web facilite la mise en œuvre et la distribution des multiples processus de RÀPC locaux nécessaires au RÀPC décentralisé. Concernant les mécanismes de raisonnements relevant des passerelles, à la base de la combinaison des points de vue, nous nous appuyons sur le système DRAGO, décrit dans [Serafini et Tamilin, 2005]. DRAGO est un moteur d'inférences sur les LDD, le formalisme à la base de C-OWL, et implémente le mécanisme de *subsumption globale* reposant sur les passerelles (voir section 4.2.3). Néanmoins, ce système est à l'heure actuelle toujours en développement et doit encore être étendu afin de permettre une mise en œuvre complètement opérationnelle du RÀPC décentralisé.

5.5 D'autres approches pour la prise en compte de points de vue au sein du RÀPC

Dans ce chapitre, un mécanisme de RÀPC décentralisé reposant sur une représentation multi-points de vue des connaissances en C-OWL a été présenté. Celui-ci est conçu indépendamment du domaine d'application et est ainsi applicable dans de nombreux autres domaines. Néanmoins, la prise en compte des points de vue, telle qu'elle est réalisée dans ce cadre, est pensée pour répondre aux besoins particulier du projet KASIMIR concernant l'aide à la décision pour l'adaptation des référentiels. Dans des applications différentes, s'appuyant sur une notion différente de point de vue, d'autres choix auraient pu être envisagés. Par exemple, il est parfois considéré que des objets sont similaires *s'il existe un point de vue dans lequel ils sont identiques* (voir par exemple [Sauvagnac, 2000]). Ce principe constitue une définition de la similarité, reposant directement sur la notion de point de vue et qui pourrait être appliquée dans le cadre du RÀPC. Elle suppose néanmoins que l'on soit capable d'*établir un point de vue*, de le construire pour les besoins de la similarité, contrairement à ce qui est fait dans notre approche où les points de vue sont prédéfinis. Il serait ainsi intéressant dans le cadre du RÀPC, d'intégrer à la représentation de points de vue la possibilité de construire dynamiquement des points de vue, par exemple par composition d'autres points de vue.

Par ailleurs, dans l'approche proposée ici, le mécanisme de RÀPC est considéré comme local à un point de vue particulier. Les passerelles sont utilisées pour le partage de connaissances entre les multiples processus de raisonnements locaux et permettent ainsi de les combiner. Par opposition, on pourrait imaginer que le mécanisme permettant de résoudre un problème selon un certain point de vue s'appuie sur la *reformulation* de ce problème dans un autre point de vue. Cela supposerait que la similarité et l'adaptation s'appuient directement sur les appariements entre points de vue, par exemple par la mise en œuvre de reformulations de la forme $(\xrightarrow{x}, \xrightarrow{A_x})$, où \xrightarrow{x} et $\xrightarrow{A_x}$ seraient des relations entre éléments de deux points de vue différents, au même titre que les passerelles.

Chapitre 6

Travaux en cours au sein du projet KASIMIR

Ce chapitre présente deux travaux en cours concernant la représentation des connaissances et le raisonnement au sein du projet KASIMIR : l'intégration du flou dans la représentation en OWL et l'acquisition de connaissances d'adaptation pour le RÀPC.

6.1 Prise en compte de types de données flous en OWL

Comme la plupart des logiques « classiques », OWL et les LD sont bien adaptés pour la représentation d'éléments de connaissances ayant des frontières définies de façon stricte, mais ne sont pas conçus pour exprimer des *notions vagues*. On pourra par exemple aisément concevoir que la classe des femmes soit définie en OWL comme celle des personnes de sexe féminin ($Femme \equiv Personne \sqcap \exists \text{sexe.Féminin}$). Exprimer la classe des adultes dans ce formalisme semble par contre moins évident. Cela suppose de choisir un seuil concernant l'âge au-delà duquel une personne pourra être considérée comme adulte. Par exemple, avec un seuil à 20 ans, celle-ci s'écrirait $Adulte \equiv Personne \sqcap \exists \text{âge.supe20}$, avec supe20 le type des entiers supérieurs ou égaux à 20. Mais, la classe des adultes se rattachant par essence à une notion vague, il serait plus satisfaisant de pouvoir la définir en OWL par la classe des personnes ayant *environ 20 ans et plus*. Cette section résume les éléments présentés dans l'article [d'Aquin *et al.*, 2005c] montrant comment OWL peut être étendu pour la représentation et le raisonnement avec des *types de données flous*, comme celui des entiers d'*environ 20 et plus*.

Concernant le projet KASIMIR, la motivation pour une extension de OWL pour la représentation et le raisonnement avec des types de données flous provient de seuils numériques conditionnant la recommandation de certaines décisions. Par exemple, il peut être établi dans le référentiel pour le traitement du cancer du sein que les patients ayant, en plus d'autres caractéristiques, une taille de tumeur de plus de 4 cm doivent subir une mastectomie totale :

$$C1 \equiv C0 \sqcap \exists aTumeur.\exists \text{taille.sup4}$$
$$C1 \sqsubseteq \exists \text{recommandation.MastectomieTotale}$$

sup4 étant le type des réels supérieurs à 4. Pour les patients ayant les mêmes caractéristiques, mais avec une tumeur de 4 cm et moins, la recommandation du référentiel sera par contre une mastectomie partielle :

$$C2 \equiv C0 \sqcap \exists aTumeur.\exists \text{taille.infe4}$$
$$C2 \sqsubseteq \exists \text{recommandation.MastectomiePartielle}$$

inf e4 étant le type des réels inférieurs ou égaux à 4. Ainsi, un patient représenté par une instance de la classe C0 et ayant une tumeur de 4,1 cm sera classifié comme une instance de C1 et donc associé à une recommandation de mastectomie totale. Mais lorsqu'un cas se trouve comme ici très proche d'un seuil de décision, fixé de façon arbitraire, il serait préférable dans le cadre de l'aide à la décision que les deux possibilités soient proposées au médecin, éventuellement ordonnées selon une « préférence » (ici la mastectomie totale, puis la mastectomie partielle). L'objectif de l'intégration de types de données flous dans la représentation des référentiels est ainsi de réduire l'effet de seuils de décision trop strictes, en les rendant flous.

6.1.1 Logiques de descriptions floues

Plusieurs travaux se sont intéressés à l'extension du formalisme des LD vers des LD floues. Certaines de ces extensions sont présentées et comparées dans [d'Aquin *et al.*, 2004d]. En LD « classique », les classes sont interprétées par des sous-ensembles du domaine d'interprétation $\Delta^{\mathcal{I}}$. Il semble ainsi naturel dans les LD floues d'interpréter les classes comme des sous-ensembles flous de $\Delta^{\mathcal{I}}$.

L'appartenance d'un élément x à un ensemble flou n'est pas, comme pour les ensembles classiques, une indication booléenne (x appartient ou n'appartient pas à l'ensemble), mais peut être graduelle, selon un degré d'appartenance entre 0 et 1. Ainsi, étant donné un ensemble X , un sous-ensemble flou de X est caractérisé par une *fonction d'appartenance* $A : x \in X \mapsto A(x) \in [0; 1]$. Pour simplifier, on ne fera pas de distinction dans la suite entre un sous-ensemble flou et sa fonction d'appartenance (pour plus d'informations sur la théorie des sous-ensembles flous, se référer à [Dubois et Prade, 2000]). Avec A et B deux sous-ensembles flous de X , A est un sous-ensemble de B , noté $A \subseteq B$, si pour tout $x \in X$, $A(x) \leq B(x)$.

Les LD classiques sont interprétées grâce à des notions ensemblistes classiques : ensemble, relation binaire, appartenance, etc. Les extensions floues des LD sont associées à une sémantique exprimée grâce à la théorie des sous-ensembles flous : une classe C s'interprète comme un sous-ensemble flou $C^{\mathcal{I}}$ du domaine d'interprétation $\Delta^{\mathcal{I}}$, une propriété p comme une relation binaire floue $p^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$ et l'appartenance d'un individu a à une classe C , $a^{\mathcal{I}} \in C^{\mathcal{I}}$ dans le cas classique, devient le degré d'appartenance de $a^{\mathcal{I}}$ à $C^{\mathcal{I}}$: $C^{\mathcal{I}}(a^{\mathcal{I}})$. Les différentes LD floues ont par ailleurs en commun l'essentiel de ce qui concerne l'interprétation des constructeurs de classes. Pour les constructeurs de la LD \mathcal{ALC} , on aura par exemple (voir notamment [Straccia, 1998]) :

$$\begin{aligned}
 \top^{\mathcal{I}}(x) &= 1 \\
 \perp^{\mathcal{I}}(x) &= 0 \\
 (C \sqcap D)^{\mathcal{I}}(x) &= \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\} \\
 (C \sqcup D)^{\mathcal{I}}(x) &= \max\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\} \\
 (\neg C)^{\mathcal{I}}(x) &= 1 - C^{\mathcal{I}}(x) \\
 (\exists p.C)^{\mathcal{I}}(x) &= \sup_{y \in \Delta^{\mathcal{I}}} \min\{p^{\mathcal{I}}(x, y), C^{\mathcal{I}}(y)\} \\
 (\forall p.C)^{\mathcal{I}}(x) &= \inf_{y \in \Delta^{\mathcal{I}}} \max\{1 - p^{\mathcal{I}}(x, y), C^{\mathcal{I}}(y)\}
 \end{aligned}$$

pour tout $x \in \Delta^{\mathcal{I}}$. Finalement, les LD floues se distinguent principalement par la façon dont elles étendent les LD classiques pour l'introduction du flou [d'Aquin *et al.*, 2004d]. Cela peut être grâce à l'association de degrés de vérité aux axiomes ou aux assertions, à l'utilisation de modificateurs sur les concepts (« plus ou moins », « très », etc.) ou encore à des prédicats flous dans des domaines concrets. L'extension de OWL que nous présentons dans la suite relève de cette dernière approche.

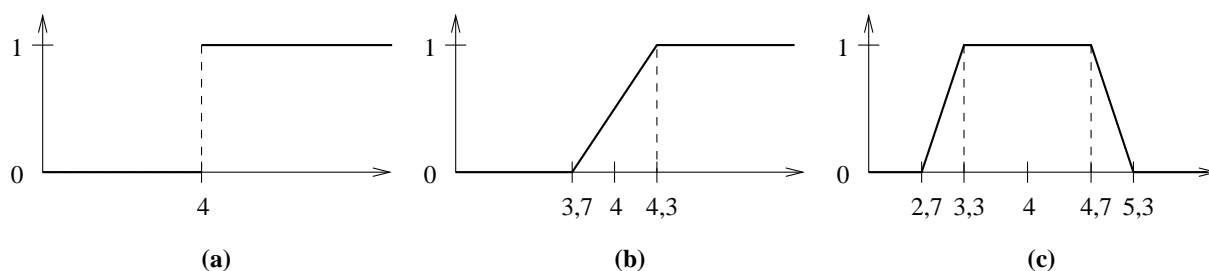


FIG. 6.1 – Fonctions d'appartenance pour l'ensemble classique des réels supérieurs à 4 (a), pour l'ensemble flou des réels supérieurs à environ 4 (b) et pour l'ensemble flou des réels entre environ 3 et environ 5 (c).

6.1.2 Extension de OWL pour le prise en compte de types de données flous

Comme montré dans la section 2.1.3, OWL manipule des données concrètes, comme les entiers, au travers des types de données de XML SCHÉMA. Pour la représentation des référentiels au sein du projet KASIMIR, des types de données dérivés par restriction de types numériques sont utilisés (voir section 2.3.1). Par exemple, le type `sup4` des réels supérieurs à 4 est décrit dans ce cadre par :

```
<xsd:simpleType name="sup4">
  <xsd:restriction base="float">
    <xsd:minExclusive value="4.0"/>
  </xsd:restriction>
</xsd:simpleType>
```

Un tel type de données représente un ensemble de valeurs, en l'occurrence ici l'ensemble des réels supérieurs à 4. Afin d'étendre OWL pour l'introduction du flou, nous nous appuyons sur la notion de *types de données flous* pour représenter des ensembles flous de valeurs. Dans les systèmes manipulant des ensembles flous, il est courant de se limiter à des fonctions d'appartenance ayant la forme de trapèzes (voir figure 6.1). L'ensemble des valeurs pour lesquelles le degré d'appartenance à un ensemble flou A est égal à 1 est appelé le *noyau* de A (en anglais, *core*). Il s'agit du segment supérieur du trapèze. L'ensemble des valeurs pour lesquelles le degré d'appartenance à A est différent de 0 est appelé le *support* de A . Il s'agit du segment inférieur du trapèze. Par exemple, pour l'ensemble flou des réels entre environ 3 et environ 5 de la figure 6.1(c), le noyau correspond à l'intervalle $[3, 3; 4, 7]$ et le support à l'intervalle $[2, 7; 5, 3]$. Ces deux éléments sont suffisants pour caractériser un ensemble flou ayant une fonction d'appartenance trapézoïdale [Dubois *et al.*, 2005]. Il est ainsi possible de décrire selon ce principe une extension de la syntaxe de XML SCHÉMA dans laquelle le type `f sup4` des réels supérieurs à environ 4 est défini par une expression de la forme :

```
<fxsd:fuzzyType name="f sup4">
  <fxsd:core value="sup4.3">
  <fxsd:support value="sup3.7">
</xsd:simpleType>
```

avec `sup4.3` et `sup3.7` deux types de données classiques, représentant respectivement les réels supérieurs à 4,3 et supérieurs à 3,7.

Concernant les inférences associées à OWL flou, nous nous sommes principalement concentrés sur l'extension de la subsomption en OWL classique. Trois types d'inférences de ce genre peuvent être

mis en œuvre sur la base des types de données flous : la subsomption floue, la subsomption relative à un degré α et le degré de subsomption. Le terme *subsomption floue* désigne le test indiquant si une classe C subsume une classe D , noté ici $C \sqsubseteq D$, c'est-à-dire si pour tout modèle \mathcal{I} de l'ontologie considérée, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, au sens de l'inclusion entre ensembles flous. Étant donné deux classes C et D , le *degré de subsomption* de C dans D est la valeur $F_{\sqsubseteq}(C, D) \in [0; 1]$. En nous appuyant sur les travaux décrits dans [Tresp et Molitor, 1998], nous proposons la définition suivante pour F_{\sqsubseteq} :

$$F_{\sqsubseteq}(C, D) = \inf\{F_{\sqsubseteq}(C^{\mathcal{I}}, D^{\mathcal{I}}) \mid \mathcal{I} : \text{modèle de la base de connaissances}\}$$

où F_{\sqsubseteq} est une mesure du degré d'inclusion entre ensembles flous. Par exemple, si on définit F_{\sqsubseteq} par $F_{\sqsubseteq}(A, B) = \inf_x F_{\Rightarrow}(A(x), B(x))$, avec F_{\Rightarrow} une implication floue, il faut choisir F_{\Rightarrow} pour que l'on ait $F_{\sqsubseteq}(A, B) = 1$ si et seulement si $A \subseteq B$, et ainsi $F_{\sqsubseteq}(A, B) = 1$ si et seulement si $A \sqsubseteq B$. Dans [Dubois *et al.*, 1991], plusieurs implications floues vérifiant cette propriété sont présentées. Finalement, la *subsomption relative à un degré α* , ou α -subsomption, entre une classe A et une classe B peut être définie par : $A \sqsubseteq_{\alpha} B$ si $F_{\sqsubseteq}(A, B) \geq \alpha$.

L'article [d'Aquin *et al.*, 2005c] montre comment ces trois mécanismes d'inférences peuvent être implémentés sur la base d'un moteur d'inférences pour LD existant, en ne modifiant de ce moteur que la partie en charge du raisonnement sur les types de données. Cela peut être réalisé de façon simple pour ce qui concerne la subsomption floue et l' α -subsomption. Le calcul du degré de subsomption nécessite par contre la mise en œuvre de techniques plus sophistiquées. D'autres mécanismes d'inférences, tels que l'instanciation floue, pourront par ailleurs être facilement implémentés sur cette base.

6.2 Acquisition de connaissances d'adaptation (ACA)

Comme vu au chapitre 3, le RÀPC selon l'approche suivie au sein du projet KASIMIR repose à la fois sur des cas, des connaissances du domaine et des connaissances d'adaptation. Obtenir des cas n'est généralement pas considéré comme une tâche difficile. L'acquisition de connaissances du domaine, par contre, est reconnue comme une part importante dans la conception d'un système à base de connaissances et de nombreux travaux d'ingénierie des connaissances s'y sont intéressés. Obtenir et modéliser les connaissances d'adaptation nécessaires à la mise en œuvre d'un système de RÀPC s'avère de la même façon une tâche essentielle, nécessitant l'application de méthodes et d'outils dédiés. Pour cette raison, l'*acquisition de connaissances d'adaptation (ACA)* est à l'heure actuelle considérée comme une thématique primordiale au sein du projet KASIMIR. L'article [Lieber *et al.*, 2004] présente et compare quelques travaux en RÀPC relevant de l'ACA. De façon générale, on distinguera deux approches principales de l'ACA, applicables selon les sources de connaissances disponibles : les méthodes d'ACA auprès d'experts et les outils pour l'ACA semi-automatique.

6.2.1 ACA auprès d'experts

L'article [Lieber *et al.*, 2003] présente les conclusions d'une expérience d'ACA auprès d'experts, réalisée sous la forme de sessions de concertation entre médecins, ergonomes et informaticiens. Le matériau de base de ces sessions était des comptes-rendus d'adaptations réalisées au sein de RCP. Il s'agissait, en s'appuyant sur le modèle des reformulations, de modéliser les adaptations décrites, afin d'en extraire des *schémas généraux d'adaptation*. Cette étude a par exemple permis de mettre en avant les problèmes liés à la proximité de seuils numériques lors de la prise de décision, amenant à développer des fonctionnalités de représentation du flou au sein du projet KASIMIR (voir section précédente).

Une deuxième étude pour l'ACA auprès d'experts a été menée par la suite par les ergonomes impliqués dans le projet (voir la thèse de Vanina Mollo [Mollo, 2004]). Celle-ci s'est appuyée sur 15 cas de

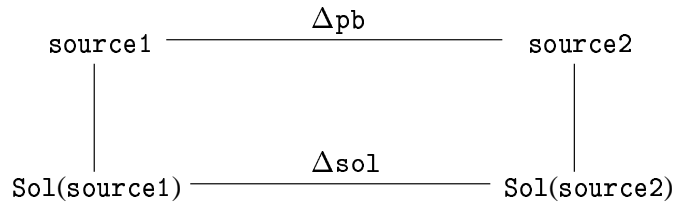


FIG. 6.2 – Un couple de cas sources considéré comme un cas d'adaptation.

cancer du sein, créés artificiellement par un expert cancérologue, dont 14 sortant du cadre du référentiel. Ces cas ont été proposés à une quinzaine de médecins lorrains à qui il a été demandé, dans un premier temps, de donner et d'expliquer leurs décisions les concernant. Dans un deuxième temps, ces mêmes médecins ont été confrontés aux décisions de leurs collègues (selon la technique dite d'*allo-confrontation*). Cette étude s'est par ailleurs elle aussi appuyée sur une modélisation des adaptations selon le modèle des reformulations.

6.2.2 ACA semi-automatique

L'ACA semi-automatique consiste appliquer des techniques d'apprentissage automatique afin d'obtenir des connaissances d'adaptation à partir de données. La question des données disponibles pour l'apprentissage est donc essentielle. On pourrait par exemple imaginer extraire ces connaissances à partir de cas d'adaptation issus de RCP, ce qui supposerait que ceux-ci soient en nombre suffisant et sous une forme exploitable. Une autre possibilité consiste à *construire* des cas d'adaptation à partir de la base de cas. Le système CABAMAKA, actuellement développé au sein du projet KASIMIR, exploite cette deuxième approche (voir [d'Aquin *et al.*, 2004b] pour une description précise des principes de CABAMAKA). Étant donné un couple de cas sources ((source1, Sol(source1)), (source2, Sol(source2))) de la base de cas, on considère que les connaissances d'adaptation que l'on cherche à apprendre ont permis de résoudre source2, c'est-à-dire d'établir Sol(source2), par adaptation de la solution Sol(source1) de source1. En d'autres termes, source1 joue le rôle du problème source dans le cas d'adaptation et source2 celui du problème cible (voir figure 6.2). L'apprentissage consiste à extraire, à partir d'un ensemble de ces couples de cas source, des règles permettant de calculer la variation entre deux problèmes source et cible, noté Δ_{pb} , la variation Δ_{sol} à appliquer à la solution Sol(source) pour obtenir Sol(cible). Le système CABAMAKA s'appuie pour cela sur les principes de l'ECBD (extraction de connaissances à partir de bases de données). Il comprend, comme c'est généralement le cas des systèmes d'ECBD, trois étapes principales : la préparation des données, la fouille de données et l'interprétation des résultats de la fouille.

La préparation des données comporte généralement un formatage et un filtrage. Dans le cas de CABAMAKA le *formatage* consiste à représenter les couples (Δ_{pb} , Δ_{sol}) présents au sein de la base de cas sous une forme exploitable par le processus de fouille, c'est-à-dire par des ensembles de propriétés booléennes. La première étape consiste ainsi à associer à chacun des cas sources (source, Sol(source)) de la base de cas, les ensembles $\Phi(\text{source})$ et $\Phi(\text{Sol}(\text{source}))$ de propriétés booléennes traduisant les caractéristiques respectivement de source et de Sol(source). Cette étape est dépendante du formalisme utilisé pour représenter les problèmes et les solutions. S'ils sont représentés par des individus en OWL, il pourra s'agir par exemple de créer une propriété pour chacune des classes d'appartenance et chacune des relations de l'individu, y compris celles pouvant être déduites de l'ontologie. Par exemple, avec les

axiomes et les assertions suivants :

$$\begin{aligned} \text{Patient} &\sqsubseteq \text{Problème} \\ \text{Patiente} &\equiv \text{Patient} \sqcap \exists \text{sexe.Feminin} \\ \text{Patiente}(p) &\quad \hat{\text{age}}(p, 45) \end{aligned}$$

le résultat de la première étape de formatage sur le problème représenté par l'individu p donnera l'ensemble de propriétés $\Phi(p) = \{\text{Patiente}, \text{Patient}, \text{Problème}, \hat{\text{age}}:45, \text{sexe:Féminin}\}$.

La deuxième étape de formatage s'appuie sur les ensembles de propriétés $\Phi(\text{source1})$, $\Phi(\text{source2})$, $\Phi(\text{Sol}(\text{source1}))$ et $\Phi(\text{Sol}(\text{source2}))$ créés à partir de chacun des couples de cas sources $((\text{source1}, \text{Sol}(\text{source1})), (\text{source2}, \text{Sol}(\text{source2})))$ de la base de cas, pour créer une représentation sous forme de propriétés booléennes des variations Δ_{pb} et Δ_{sol} , respectivement de source1 à source2 et de $\text{Sol}(\text{source1})$ à $\text{Sol}(\text{source2})$. La variation entre deux ensembles de propriétés $E1$ et $E2$ est représentée par un triplet $\Delta E = (E^=, E^-, E^+)$, où $E^=$ est l'ensemble des propriétés communes à $E1$ et $E2$, E^- l'ensemble des propriétés présentes dans $E1$ mais pas dans $E2$ et E^+ l'ensemble des propriétés présentes dans $E2$ mais pas dans $E1$. On peut de cette manière représenter la variation Δ_{pb} entre source1 et source2 par le triplet $(\Phi_{\text{pb}}^=, \Phi_{\text{pb}}^-, \Phi_{\text{pb}}^+)$, ainsi que la variation Δ_{sol} entre $\text{Sol}(\text{source1})$ et $\text{Sol}(\text{source2})$ par le triplet $(\Phi_{\text{sol}}^=, \Phi_{\text{sol}}^-, \Phi_{\text{sol}}^+)$. Les six ensembles de propriétés ainsi créés pour un couple de cas sources sont ensuite agrégés en un seul, en ajoutant à chacune des propriétés des marqueurs permettant d'indiquer duquel des six ensembles elle provient : si une propriété p est présente au sein de l'ensemble Φ_{sol}^+ , elle sera considérée sous la forme p_{sol}^+ dans l'ensemble des propriétés représentant le couple $(\Delta_{\text{pb}}, \Delta_{\text{sol}})$. Par exemple, si les quatre ensembles créés à la première étape sont :

$$\begin{aligned} \Phi(\text{source1}) &= \{a, b, c\} & \Phi(\text{Sol}(\text{source1})) &= \{A, B\} \\ \Phi(\text{source2}) &= \{b, c, d\} & \Phi(\text{Sol}(\text{source2})) &= \{B, C\} \end{aligned}$$

alors le couple $(\Delta_{\text{pb}}, \Delta_{\text{sol}})$ sera représenté par l'ensemble de propriétés booléennes :

$$\{a_{\text{pb}}^-, b_{\text{pb}}^=, c_{\text{pb}}^=, d_{\text{pb}}^+, A_{\text{sol}}^-, B_{\text{sol}}^=, C_{\text{sol}}^+\}$$

Le *filtrage* consiste à sélectionner les propriétés et les ensembles de propriétés considérés comme pertinents et permet ainsi d'orienter une session d'ACA vers un type de recherche particulier. Il peut être appliqué avant la première étape du formatage, entre les deux étapes ou après la deuxième.

La fouille de données effectuée au sein de CABAMAKA repose sur un algorithme de recherche de *motifs fermés fréquents* au sein des ensembles de propriétés correspondant aux couples $(\Delta_{\text{pb}}, \Delta_{\text{sol}})$ créés à l'étape précédente. Un motif m est un ensemble de propriétés et $f(m)$ correspond à l'ensemble des objets de la base de données possédant m . m est considéré comme fréquent si le nombre des éléments le possédant dépasse un seuil f_{min} donné : $|f(m)| \geq f_{\text{min}}$. m est fermé s'il est maximal pour l'inclusion des motifs m' tels que $f(m') = f(m)$. L'idée est qu'un motif fermé fréquent apporte potentiellement un élément de connaissances du fait qu'il généralise un ensemble de données : sont oubliées les propriétés particulières qui n'apparaissent que dans des contextes particuliers. Par exemple, supposons que les propriétés de l'ensemble mentionné plus haut apparaissent fréquemment ensemble, à l'exception de $b_{\text{pb}}^=$. On aura alors le motif fermé fréquent :

$$m = \{a_{\text{pb}}^-, c_{\text{pb}}^=, d_{\text{pb}}^+, A_{\text{sol}}^-, B_{\text{sol}}^=, C_{\text{sol}}^+\}$$

L'algorithme de recherche de motifs fermés fréquents utilisé pour CABAMAKA est une implémentation de l'algorithme CHARM, développée dans le cadre du système CORON [Szathmary et Napoli, 2005].

L'aide à l'interprétation a pour objectif d'assister la construction de nouvelles connaissances à partir des motifs extraits. Par exemple, on peut, à partir du motif m décrit ci-dessus, construire une *règle d'adaptation* de la forme :

- si**
- (1) *source* et *cible* partagent la propriété c
 - (2) a est dans *source* mais pas dans *cible*
 - (3) d est dans *cible* mais pas dans *source*
 - (4) $\text{Sol}(\text{source})$ possède les propriétés A et B
 - (5) $\text{Sol}(\text{source})$ ne possède pas la propriété C

alors $\text{Sol}(\text{cible})$ peut être créée en ajoutant C et en supprimant A de $\text{Sol}(\text{source})$.

Le problème est que les motifs fermés fréquents en sortie du processus de fouille sont très nombreux et qu'il est ainsi difficile de les analyser pour n'en sélectionner que les plus pertinents. Une étude est en cours afin d'établir des *indices statistiques* permettant d'ordonner les motifs pour mettre en avant ceux susceptibles d'être utilisés pour construire des connaissances d'adaptation, à l'image de ce qui est fait dans [Cherfi, 2004] pour la fouille de textes.

6.2.3 Vers une approche mixte de l'ACA

Les schémas d'adaptation issus de l'ACA auprès d'experts ont l'avantage d'être intelligibles, c'est-à-dire qu'ils sont facilement explicables par des experts du domaine, mais sont par contre difficilement opérationnalisables en tant que connaissances d'adaptation pour un service de RAPC tel que celui décrit au chapitre 3. Les règles d'adaptation issues de l'acquisition semi-automatique constituent quant à elles des connaissances d'adaptation directement opérationnalisables, mais ont le défaut d'être difficilement intelligibles. Une approche de l'ACA combinant les deux démarches, auprès d'experts et semi-automatique, est ainsi envisagée au sein du projet KASIMIR. Il pourrait s'agir par exemple de considérer une règle d'adaptation issue de la fouille comme une mise en œuvre particulière d'un des schémas d'adaptation construits avec les experts, le schéma constituant de cette façon une « interprétation » possible de la règle en vue de lui fournir une explication.

Conclusion

Ce mémoire de thèse présente la mise en œuvre d'un portail sémantique dédié à la gestion de connaissances décisionnelles en cancérologie. Dans un premier temps, l'architecture de ce portail a été développée comme une application des principes et des technologies du Web sémantique. Les connaissances contenues dans les référentiels utilisés par les cancérologues pour la prise de décision ont ainsi été modélisées et représentées selon les technologies standard pour la représentation des connaissances sur le Web sémantique, notamment OWL. La diffusion de ces connaissances est réalisée par l'intermédiaire d'un serveur de connaissances, reposant sur les mécanismes de raisonnements associés à OWL et fournissant par ce biais un accès intelligent aux connaissances. Des interfaces graphiques ont ainsi pu être mises en place, s'exécutant à distance et exploitant les capacités de raisonnement du portail pour l'aide à la décision dans le cadre de l'application des référentiels. L'ensemble des éléments qui composent cette architecture est d'ores et déjà opérationnel. Des référentiels sont en cours de formalisation (comme par exemple le référentiel pour la surveillance du cancer du colon) en concertation avec les partenaires du domaine médical participant au projet KASIMIR. Les représentations en OWL de ces référentiels serviront de base à la mise en place d'un portail sémantique reposant sur l'architecture présentée ici dans les mois à venir. L'utilisation concrète des développements réalisés au cours de cette thèse sera ainsi l'occasion d'une évaluation auprès des médecins lorrains. De plus, il est important de remarquer que, hormis les connaissances elles-mêmes, les éléments constituant ce portail sémantique sont indépendants du domaine d'application. Bien que construits pour répondre aux besoins de la gestion des connaissances en cancérologie, ceux-ci sont en effet développés avec l'objectif d'être réutilisés dans d'autres domaines, médicaux ou non. Ils constituent en ce sens un environnement générique pour la mise en œuvre de portails sémantiques.

Cette architecture de portail du Web sémantique fournit par ailleurs la base technologique nécessaire à la mise en œuvre de capacités de représentation et de raisonnement répondant aux besoins du projet KASIMIR pour la gestion des connaissances en cancérologie. Les principes du raisonnement à partir de cas (RÀPC) sont en particulier appliqués pour l'aide à la décision dans le cadre de l'adaptation des référentiels. Il s'agit dans un premier temps de permettre la formalisation des connaissances nécessaires à l'adaptation des référentiels pour les cas auxquels ils ne répondent pas. Pour cela, un modèle pour la représentation des *connaissances d'adaptation* utilisées en RÀPC a été défini sur la base du langage OWL. Ce modèle est conçu pour représenter des notions propres au RÀPC, comme celles de similarité et d'adaptation, en lien avec les connaissances du domaine, contenues dans les ontologies du Web sémantique. Il sert ainsi de cadre à la représentation et à la réutilisation de connaissances d'adaptation, jouant de cette façon un rôle d'intermédiaire entre les ontologies de domaine et un mécanisme de RÀPC indépendant du domaine. Sur la base de ce modèle, un service Web implémentant les opérations de remémoration et d'adaptation du RÀPC a ainsi pu être développé, apportant, en complément des inférences habituellement utilisées dans ce cadre, un nouveau mécanisme de raisonnement pour le Web sémantique. Celui-ci reste néanmoins à l'état de prototype et de nombreux points doivent encore être améliorés pour qu'il puisse être complètement intégré au portail sémantique KASIMIR.

Dans des domaines complexes et multi-disciplinaires, comme c'est le cas de la cancérologie, co-

existent souvent différentes façons de considérer les éléments de connaissances, différentes façons de les représenter et finalement différentes façons de les utiliser. Nous avons pour cette raison exploré dans ce mémoire les apports de la représentation et du raisonnement multi-points de vue, en particulier dans le cadre de la gestion des connaissances en cancérologie. En effet, la prise de décision en cancérologie fait intervenir différentes disciplines médicales, telles que la chimiothérapie, la chirurgie ou la radiothérapie. Chacune d'elles apporte sa propre façon de voir les problèmes et sa propre part aux solutions. Permettre l'utilisation conjointe de plusieurs représentations alternatives du domaine est une problématique étudiée dans de nombreux formalismes de représentation des connaissances. Parmi ceux-ci, nous avons choisi de nous appuyer sur C-OWL, dont les mécanismes de représentation multi-contextes nous permettent de mettre en œuvre et de relier les différents points de vue impliqués dans les décisions en cancérologie. Cette forme de représentation multi-points de vue a montrée, par le biais de l'application au projet KASIMIR, des apports aussi bien en termes de « passage à l'échelle » qu'en ce qui concerne les difficultés de l'ingénierie des connaissances. Il apparaît en effet plus simple, dans des domaines où différents points de vue cohabitent, de construire et de mettre en relation différentes représentations plutôt que de chercher à établir une représentation agrégeant l'ensemble de ces points de vue. La maintenance et l'évolution des connaissances s'en trouvent de la même façon facilitée.

Finally, this multi-points of view representation of knowledge is exploited within a decentralized RÀPC mechanism. This one relies on the representation of knowledge of the domain, of adaptation knowledge and of cases according to several points of view to distribute the RÀPC mechanism within different local processes, each one focusing on a particular point of view concerning the problem to solve. The relations existing between the points of view are then used for the exchange of knowledge and the collaboration between points of view, bringing in this way a mean to combine the inferences realized within the multiple reasoning processes. The apports of this decentralized RÀPC mechanism are measured not only in terms of calculation time, but also in the fact that it constitutes a reasoning mechanism at the same time distributed and collaborative. It responds in this way to the needs of the aid to the decision for the adaptation of referentials within the multi-disciplinary framework of oncology, while being particularly well adapted to the multi-ontologies environment of the Semantic Web. A service of decentralized RÀPC is thus in the process of development in order to be integrated into the semantic portal KASIMIR. This one relies on the RÀPC service developed in chapter 3 and on the inference engine for C-OWL DRAGO [Serafini et Tamilin, 2005]. This latter being itself in the process of development, it remains at the moment still insufficient to allow a completely operational decentralized RÀPC.

Outre les travaux en cours concernant l'intégration du flou dans la représentation en OWL et l'acquisition de connaissances d'adaptation présentés dans le chapitre 6, les perspectives auxquelles amènent cette thèse se regroupent en trois thématiques principales : le déploiement de portails du Web sémantique, le RÀPC pour le Web sémantique et la gestion de connaissances multi-points de vue.

Outils pour le déploiement de portails sémantiques. Comme indiqué plus haut, bien que prévus pour répondre aux besoins du projet KASIMIR concernant la gestion des connaissances décisionnelles en cancérologie, les éléments qui constituent le portail sémantique développé ici sont conçus de façon générique, c'est-à-dire de façon à être réutilisables dans d'autres domaines que la cancérologie. Néanmoins, le déploiement de cette architecture reste une tâche techniquement difficile et plusieurs points doivent encore être développés afin de simplifier la mise en œuvre de nouveaux portails sur cette base. En particulier, les mécanismes de raisonnements utilisés, notamment les inférences associées aux logiques de descriptions, sont particulièrement gourmands en ressources. Pour ce qui concerne le temps de calcul, des mécanismes de caches ont d'ores et déjà été intégrés, permettant de ne pas avoir a ré-

exécuter les inférences déjà réalisées par ailleurs (voir section 2.4). La représentation et le raisonnement multi-points de vue apportent eux aussi une amélioration sensible sur ce point. Mais d'autres techniques devront encore être envisagées dans le sens du « passage à l'échelle », afin de permettre le déploiement de ce type de portail de façon complètement opérationnelle dans un environnement aussi vaste que celui du Web. Parmi les techniques existantes, certaines formes de raisonnements approximatifs (comme par exemple celle développée dans [Groot *et al.*, 2005]) ainsi que des techniques de compilation de connaissances semblent particulièrement prometteuses (voir [Wache *et al.*, 2004] pour un état des recherches sur le « passage à l'échelle » au sein du Web sémantique).

En plus des outils mis en œuvre pour la construction, l'édition et la maintenance des connaissances, des outils permettant d'assister la construction et de gérer le fonctionnement et l'évolution du portail lui-même semblent nécessaires. Il s'agit dans ce cadre de développer un environnement permettant par exemple la mise en place du serveur de connaissances au sein d'un réseau, d'en connaître le niveau d'utilisation, d'en tester les performances, de gérer les redéploiements en cas de panne, etc. La mise en œuvre des fonctionnalités du portail sous formes de services Web facilite d'ores et déjà ce type de tâches. Il serait dans ce cadre particulièrement intéressant d'exploiter des descriptions sémantiques de ces services, sur la base de langages tels que OWL-S [Martin *et al.*, 2004].

La question des interfaces est de plus particulièrement importante pour l'acceptation du portail auprès des utilisateurs. Trois points particuliers concernant celles-ci restent à être explorés dans le cadre du portail KASIMIR. En effet, les interfaces développées ici (ÉDHIBOU et NAVHIBOU, voir section 2.5) sont construites comme des outils génériques, applicables sur des connaissances représentées en OWL, quel que soit le domaine d'application. Elles sont automatiquement générées en fonction de l'ontologie considérée. Mais malgré l'avantage que procure ce type d'interfaces concernant la réutilisabilité, celles-ci sont peu conviviales et nécessitent d'être « personnalisées » en fonction de l'application pour laquelle elles sont employées. Un mécanisme permettant de décrire l'aspect que doit prendre l'interface, tout en conservant son caractère générique et auto-générée, fait partie des développements en cours dans le cadre du portail sémantique KASIMIR. Le deuxième point concernant les interfaces est leur intégration avec les extensions pour la représentation des connaissances et le raisonnement développées durant cette thèse. En effet, elles s'appuient à l'heure actuelle uniquement sur les mécanismes d'inférences standard associés à OWL. Permettre la visualisation des résultats du RÀPC et des explications associées, l'édition de connaissances floues ou encore la navigation au sein d'ontologies multi-points de vue sont parmi les développements à envisager dans ce cadre. Enfin, un des avantages importants d'une architecture fondée sur les principes du Web sémantique et des services Web est que celle-ci est en mesure d'interopérer avec d'autres éléments logiciels, inclus dans d'autres applications. Il serait intéressant d'évaluer ce point en intégrant par exemple le portail sémantique avec le système d'informations d'un centre hospitalier. Il s'agirait de le faire collaborer avec les logiciels existants, comme celui en charge des *dossiers patients informatisés*, et de rendre l'accès aux connaissances mieux adapté à l'activité des médecins (transfert de l'interface sur un *assistant personnel* lors des consultations par exemple).

RÀPC, connaissances et Web sémantique. À court terme, les perspectives pour le développement du mécanisme de RÀPC au sein du portail sémantique KASIMIR concernent principalement l'amélioration de l'implémentation actuelle du service décrit au chapitre 3, notamment en termes de temps de calcul, ainsi qu'une mise en œuvre opérationnelle du service de RÀPC décentralisé décrit au chapitre 5. Se dernier repose en effet sur le moteur d'inférences DRAGO, qui doit encore être étendu par l'implémentation de l'*instanciation globale*, ainsi que pour permettre la mise en œuvre d'appariements contenant des cycles dans les ontologies contextualisées de C-OWL. Le chapitre 5 indique de plus comme perspective la prise en compte d'autres formes de points de vue au sein du processus de RÀPC.

Par ailleurs, le RÀPC est un mécanisme de raisonnement nouveau au sein du Web sémantique : les

inférences réalisées dans ce cadre relèvent habituellement de la déduction. Néanmoins, nous pensons que de nombreuses applications du Web sémantique bénéficieraient de l'utilisation d'un service de raisonnement reposant sur l'analogie. Un argument dans ce sens est que le RÀPC a l'avantage, vis-à-vis de mécanismes reposant sur la déduction, d'être bien adapté dans des applications où les connaissances du domaine sont incomplètes ou incomplètement formalisées. Comme c'est le cas pour la cancérologie, disposer d'une ontologie contenant l'ensemble des connaissances utiles reste un idéal difficile à atteindre dans beaucoup de domaines. La mise en œuvre de connaissances d'adaptation et leur opérationnalisation au sein du RÀPC pourra ainsi s'avérer une aide précieuse pour de nombreuses applications du Web sémantique. L'étude de ces applications s'avère une perspective intéressante vis-à-vis du service de RÀPC conçu pour le portail sémantique KASIMIR. Leur mise en œuvre sur la base de ce service et les extensions qu'il serait nécessaire de lui apporter font alors partie des développements à envisager.

Dans le même ordre d'idée, le chapitre 3 montre comment le modèle des reformulations est utilisé pour la représentation des connaissances d'adaptation. Il y est argumenté que, les reformulations étant un modèle très général, il peut être spécialisé pour mettre en œuvre d'autres approches du RÀPC. Une autre perspective consiste ainsi à étudier et comparer ces approches afin d'étendre le modèle des reformulations et, par ce biais, les capacités du mécanisme de RÀPC implémenté. En particulier, le fait de formaliser les connaissances d'adaptation au sein d'un langage de représentation des connaissances tel que OWL amène à la possibilité de construire des cas d'adaptation, résultats d'épisodes de résolutions de problèmes, eux-même réutilisables au sein du processus de RÀPC, selon l'approche dite d'*adaptation à partir de cas* (voir par exemple [Leake *et al.*, 1997]).

Gestion de connaissances multi-points de vue et gestion multi-points de vue des connaissances.

Dans le chapitre 4, l'idée d'exploiter les points de vue dans un processus de construction ascendante de connaissances est mentionnée. Il est question dans ce cas de construire une hiérarchie de points de vue, des plus spécifiques aux plus généraux, des plus élémentaires aux plus englobants. Dans le même ordre d'idée, le chapitre 5 indique, comme une perspective, les apports que pourraient avoir la construction dynamique de points de vue dans le cadre du RÀPC. Ces deux points amènent à la même constatation : l'utilité de gérer non seulement le contenu des points de vue, mais aussi les points de vue eux même. Cette perspective peut par ailleurs être vue selon les deux aspects théorique et technique. Il s'agirait en particulier d'étudier comment des points de vue existants pourraient être combinés pour former de nouveaux points de vue (par des *constructeurs de points de vue* de types conjonction, union, différence par exemple). La nature de la relation hiérarchique liant les points de vue ainsi que les inférences réalisables sur cette base ferait aussi partie des points théoriques à considérer. On pourrait dans ce cadre s'appuyer sur une formalisation au *niveau méta* des points de vue, à la manière de ce qui est fait dans [Attardi et Simi, 1995], ou encore explorer pour cela la relation liant la représentation de points de vue aux logiques modales. Sur le plan technique, cette perspective concerne le développement d'un environnement permettant d'assister la construction, la maintenance et la manipulation de représentation multi-points de vue. Il s'agirait de disposer d'outils pour la construction ascendante de connaissances multi-points de vue, assistant leur hiérarchisation et leur combinaison, ainsi que de moteurs d'inférences capables de raisonner à la fois *dans* les points de vue et *sur* les points de vue. Finalement, la gestion multi-points de vue de connaissances ne pourrait se faire sans une *description sémantique des points de vue*. Il serait en effet intéressant d'associer à chaque point de vue une expression permettant de décrire le contenu d'un point de vue, en quoi il se distingue des autres, permettant de cette façon à un utilisateur de reconnaître par exemple dans quels points de vue il est supposé se placer selon la situation dans laquelle il se trouve et le problème qu'il cherche à résoudre.

Bibliographie

- [Aamodt et Plaza, 1994] A. Aamodt et E. Plaza. Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AICom - Artificial Intelligence Communications*, 7(1) :39–59, 1994.
- [Aamodt, 1990] A. Aamodt. Knowledge-Intensive Case-Based Reasoning and Sustained Learning. Dans *Proc. of the 9th European Conference on Artificial Intelligence (ECAI'90)*, rédacteur L. C. Aiello, August 1990.
- [Aamodt, 2004] A. Aamodt. Knowledge-Intensive Case-Based Reasoning in CREEK. Dans *Proc. of the European Conference on Case-Based Reasoning, ECCBR'04*, rédacteurs P. Funk et P. A. González-Calero, volume 3155 de *Lecture Notes in Artificial Intelligence*, pages 1–15. Springer, 2004.
- [Adjiman *et al.*, 2004] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, et L. Simon. Distributed Reasoning in a Peer-to-Peer Setting. Dans *Proc. of the 16th European Conference on Artificial Intelligence, ECAI'2004*, rédacteurs R. López de Mántaras et L. Saitta, pages 945–946. IOS Press, 2004.
- [Arcos et Lopez de Mántaras, 1997] J. L. Arcos et R. Lopez de Mántaras. Perspectives : a declarative bias mechanism for case retrieval. Dans *Proc. of the International Conference on Case-Based Reasoning, ICCBR'97*, rédacteurs D. B. Leake et E. Plaza, volume 1266 de *Lecture Notes in Computer Science*, pages 279–290. Springer, 1997.
- [Arpírez *et al.*, 2003] J.C. Arpírez, O. Corcho, M. Fernández-López, et A. Gómez-Pérez. WebODE in a nutshell. *AI Magazine*, 24(3) :37–47, 2003.
- [(ASC), 2005] The Apache SOAP Community (ASC). Apache SOAP. <http://ws.apache.org/soap/docs/index.html>, dernière consultation : septembre 2005, 2005.
- [Attardi et Simi, 1995] G. Attardi et M. Simi. A formalization of viewpoints. *Fundamenta Informaticae*, 1995.
- [Ayala et Nichol, 2005] D. Ayala et S. Nichol. NuSOAP - SOAP Toolkit for PHP. <http://sourceforge.net/projects/nusoap/> dernière consultation : septembre 2005, 2005.
- [Baader *et al.*, 2003] rédacteurs F. Baader, D. Calvanese, D. McGuinness, D. Nardi, et P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, cambridge, UK, 2003.
- [Baader et Hanschke, 1991] F. Baader et P. Hanschke. A Scheme for Integrating Concrete Domains into Concept Languages. Dans *Proc. of the International Joint Conference on Artificial Intelligence, IJCAI 91*, pages 452–457, 1991.
- [Baker *et al.*, 1999] P. Baker, C. Goble, S. Bechhofer, N. Paton, R. Steven, et A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6) :510–520, 1999.
- [Bechhofer *et al.*, 2001] S. Bechhofer, I. Horrocks, C. Goble, et R. Stevens. OilEd : a Reason-able Ontology Editor for the Semantic Web. Dans *Proc. of Joint German/Austrian Conferences on AI, KI 2001*, volume 2174 de *Lecture Notes in Artificial Intelligence*, pages 396–408. Springer, 2001.

- [Bechhofer *et al.*, 2003] S. Bechhofer, R. Volz, , et P. Lord. Cooking the Semantic Web with the OWL API. Dans *Proc. of the Second International Semantic Web Conference, ISWC 2003*, 2003.
- [Bechhofer, 2003] S. Bechhofer. The DIG Description Logic Interface : DIG/1.1. Technical report, DL Implementation Group, 2003.
- [Bello-Tomás *et al.*, 2004] J. J. Bello-Tomás, P. A. González-Calero, et B. Díaz-Agudo. JColibri : An Object-Oriented Framework for Building CBR Systems. Dans *Advances in Case-Based Reasoning, Proc. of the 7th European Conference on Case-Based Reasoning, ECCBR 2004*, pages 32–46, 2004.
- [Benerecetti *et al.*, 2001] M. Benerecetti, P. Bouquet, et C. Ghidini. On the Dimensions of Context Dependence : Partiality, Approximation, and Perspective. Dans *Proc. of the Third International and Interdisciplinary Conference on Modeling and Using Context*, volume 2116 de *Lecture Notes In Computer Science*, pages 59–72. Springer, 2001.
- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, et O. Lassila. The semantic web. *Scientific American*, mai 2001. <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>.
- [Berners-Lee et Fischetti, 2000] T. Berners-Lee et M. Fischetti. *Weaving the Web*. HarperCollins, 2000.
- [Bobrow et Winograd, 1977] D. Bobrow et T. Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1) :3–46, 1977.
- [Borgida et Serafini, 2002] A. Borgida et L. Serafini. Distributed description logics : Directed domain correspondences in federated information sources. Dans *Proc. of the Intenational Conference on Cooperative Information Systems*, 2002.
- [Börner, 1998] K. Börner. CBR for Design. Dans Lenz *et al.* [Lenz *et al.*, 1998], chapter 8.
- [Bouquet *et al.*, 2002] P. Bouquet, A. Donà, L. Serafini, et S. Zanobini. Contextualized local ontologies specification via CTXML. Dans *Working Notes of the AAAI-02 workshop on Meaning Negotiation*, 2002.
- [Bouquet *et al.*, 2003a] P. Bouquet, C. Ghidini, F. Giunchiglia, et E. Blanzieri. Theories and uses of context in knowledge representation and reasoning. *Journal of pragmatics*, 35(3), 2003.
- [Bouquet *et al.*, 2003b] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, et H. Stuckenschmidt. C-OWL : Contextualizing Ontologies. Dans *Proc. of the International Semantic Web Conference, ISWC 2003*, 2003.
- [Bouquet *et al.*, 2004] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, et H. Stuckenschmidt. Contextualizing Ontologies. *Journal of Web Semantics*, 1(4) :1–19, 2004.
- [Bouthier, 2005] C. Bouthier. Hypertree Java Library. <http://hypertree.sourceforge.net/>, dernière consultation : septembre 2005, 2005.
- [Brachman *et al.*, 1991] R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L. Alperin Resnick, et A. Borgida. Living With CLASSIC : When and How to Use a KL-ONE-Like Language. Dans *Principles of Semantic Networks*, rédacteur J.F. Sowa, chapter 14, pages 401–456. Morgan Kaufmann, 1991.
- [Bresson et Lieber, 1999] B. Bresson et J. Lieber. Classification pour l’aide au traitement du cancer du sein. Dans *Actes des Septième journées de la Société Francophone de Classification, SFC’99*, rédacteurs F. Le Ber, J.-F. Mari, A. Napoli, et A. Simon, pages 53–59, Nancy, septembre 1999. Unité de recherche INRIA Lorraine.
- [Bringay *et al.*, 2005] S. Bringay, C. Barry, et J. Charlet. Les annotations pour gérer les connaissances du dossier patient. Dans *Actes des Journées Françaises d’Ingénierie des Connaissances, IC 2005*, 2005.

-
- [Champin, 2002] P.-A. Champin. *Modéliser l'expérience pour en assister la réutilisation : de la Conception Assistée par Ordinateur au Web Sémantique*. Thèse d'université, Université Claude Bernard Lyon 1, 2002.
- [Charlet, 2002] J. Charlet. L'Ingénierie des connaissances. Développements, résultats et perspectives pour la gestion des connaissances médicales. Habilitation à diriger des recherches, Université Pierre et Marie Curie, décembre 2002.
- [Chen et Wu, 2003] H. Chen et Z. Wu. CaseML : a RDF-based Case Markup Language for Case-based Reasoning in Semantic Web. Dans *From structured cases to unstructured problem solving episodes for experience-based assistance. Workshop at ICCBR-2003*, rédacteurs B. Fuchs et A. Mille, 2003.
- [Cherfi, 2004] H. Cherfi. *Étude et réalisation d'un système d'extraction de connaissances à partir de textes*. Thèse d'université, Université Henri Poincaré – Nancy 1, 2004.
- [Chouraqui, 1986] E. Chouraqui. Le raisonnement analogique : sa problématique, ses applications. Dans *Actes des Journées Nationales sur l'Intelligence Artificielle, Aix-les-Bains*, pages 107–117. CEPADUES-Editions, Toulouse, 1986.
- [Coyle *et al.*, 2004] L. Coyle, D. Doyle, et P. Cunningham. Representing Similarity for CBR in XML. Dans *Advances in Case-Based Reasoning (Procs. of the Seventh European Conference), LNAI 3155*, rédacteurs P. Funk et P. González Calero, pages 119–127. Springer, 2004.
- [d'Aquin *et al.*, 2002] M. d'Aquin, J. Lieber, et A. Napoli. Représentation multi-points de vue des connaissances pour l'adaptation. Dans *Actes du dixième atelier de raisonnement à partir de cas*, rédacteur M. C. Jaulent, Mai 2002.
- [d'Aquin *et al.*, 2004a] M. d'Aquin, S. Brachais, J. Lieber, et A. Napoli. Decision Support and Knowledge Management in Oncology using Hierarchical Classification. Dans *Proc. of the Symposium on Computerized Guidelines and Protocols, CGP-2004*, volume 101 de *Studies in Health Technology and Informatics*, pages 16–30. IOS Press, 2004.
- [d'Aquin *et al.*, 2004b] M. d'Aquin, S. Brachais, J. Lieber, et A. Napoli. Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas — une approche fondée sur des techniques d'extraction de connaissances dans des bases de données. Dans *Actes du 12ème Atelier de Raisonnement à Partir de Cas - RàPC'04*, pages 41–52, 2004.
- [d'Aquin *et al.*, 2004c] M. d'Aquin, J. Lieber, et A. Napoli. Représentation de points de vue pour le raisonnement à partir de cas. Dans *Actes de la conférence Langages et Modèles à Objets, LMO'04*, rédacteurs J. Euzenat et B. Carré, volume 10 de *Revue des sciences et technologies de l'information - série L'objet*, pages 245–258. Lavoisier, 2004.
- [d'Aquin *et al.*, 2004d] M. d'Aquin, J. Lieber, et A. Napoli. Étude de quelques logiques de descriptions floues et de formalismes apparentés. Dans *Actes des rencontres francophones sur la logique floue et ses applications - LFA'04*, pages 255–262, 2004.
- [d'Aquin *et al.*, 2005a] M. d'Aquin, S. Brachais, C. Bouthier, J. Lieber, et A. Napoli. Knowledge Editing and Maintenance Tools for a Semantic Portal in Oncology. *International Journal of Human-Computer Studies (IJHCS)*, 62(5) :619–638, 2005.
- [d'Aquin *et al.*, 2005b] M. d'Aquin, J. Lieber, et A. Napoli. Decentralized Case-Based Reasoning for the Semantic Web. Dans *Proc. of International Semantic Web Conference, ISWC 2005*, volume 3729 de *Lecture Notes in Computer Science*, pages 142–155. Springer, 2005.
- [d'Aquin *et al.*, 2005c] M. d'Aquin, J. Lieber, et A. Napoli. Towards a Semantic Portal for Oncology using a Description Logic with Fuzzy Concrete Domains. Dans *Fuzzy Logic and the Semantic Web*, rédacteur E. Sanchez. Elsevier, 2005.

- [d'Aquin, 2003] M. d'Aquin. Un modèle de connaissances en RDF(S) pour raisonner à partir de cas sur le Web sémantique. Dans *Actes de l'atelier raisonnement à partir de cas, RàPC'03, plate-forme AFIA*, rédacteur J. Lieber, 2003.
- [Davis, 1987] H. E. Davis. VIEWS : Multiple Perspectives and Structured Objects in a Knowledge Representation Language. Bachelor and master of science thesis, MIT, 1987.
- [Díaz-Agudo et González-Calero, 2002] B. Díaz-Agudo et P.A. González-Calero. CBRonto : a Task/Method ontology for CBR. Dans *Proc. of the 15th International FLAIRS 2002 Conference (Special Track on Case-Based Reasoning)*, 2002.
- [Decker *et al.*, 1999] S. Decker, M. Erdmann, D. Fensel, et R. Studer. Ontobroker : Ontology Based Access to Distributed and Semi-Structured Information. Dans *Proc. of the IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics- Semantic Issues in Multimedia Systems*, pages 351–369, 1999.
- [Dekker, 1994] L. Dekker. *FROME : Représentation multiple et classification d'objets avec points de vue*. Thèse d'université, Université des sciences et technologies de Lille, Juin 1994.
- [Demazeau et Müller, 1989] Y. Demazeau et J.-P. Müller. Decentralized Artificial Intelligence. Dans *Decentralized A.I. – Proc. of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, rédacteurs Y. Demazeau et J.-P. Müller, pages 3–13. North-Holland, 1989.
- [Dieng *et al.*, 1999] R. Dieng, O. Corby, A. Giboin, et M. Ribière. Methods and tools for corporate knowledge management. *International on Human-Computer Studies*, 51(3) :567–598, 1999.
- [Dieng *et al.*, 2000] R. Dieng, O. Corby, A. Giboin, J. Golebiowska, N. Matta, et M. Ribière. *Méthodes et outils pour la gestion des connaissances*. DUNOD, 2000.
- [Dieng-Kuntz *et al.*, 2004] R. Dieng-Kuntz, D. Minier, F. Corby, M. Ruzicka, O. Corby, L. Alamarguy, et P.-H. Luong. Medical Ontology and Virtual Staff for a Health Network. Dans *Engineering Knowledge in the Age of the Semantic Web, Proc. of the 14th International Conference, EKAW 2004*, volume 3257 de *Lecture Notes in Computer Science*, pages 187–202. Springer, 2004.
- [Dubois *et al.*, 1991] D. Dubois, H. Prade, et J.-P. Rossazza. Vagueness, Typicality and Uncertainty in Class Hierarchies. *International Journal of Intelligent Systems*, 6 :167–183, 1991.
- [Dubois *et al.*, 2005] D. Dubois, J. Mengin, et H. Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. Dans *Fuzzy Logic and the Semantic Web*, rédacteur E. Sanchez. Elsevier, 2005.
- [Dubois et Prade, 2000] rédacteurs D. Dubois et H. Prade. *Fundamentals of Fuzzy Sets*, volume 7 de *The Handbooks of fuzzy sets*. Kluwer, 2000.
- [Ducournau *et al.*, 1998] rédacteurs R. Ducournau, J. Euzenat, G. Masini, et A. Napoli. *Langages et modèles à objets : états des recherches et perspectives*. Collection Didactique. INRIA, Rocquencourt (FR), 1998.
- [(EBMWG), 1992] Evidence-based medicine working-group (EBMWG). Evidence-based medicine. A new approach to teaching the practice of medicine. *Journal of the American Medical Association*, 17 :268, 1992.
- [Euzenat (coordinator), 2004] J. Euzenat (coordinator). State of the art on current alignment techniques. Knowledge Web Deliverable 2.2.3, Knowledge Web network of excellence, 2004.
- [Falquet et Mottaz, 2001] G. Falquet et C.-L. Mottaz. Navigation hypertexte dans une ontologie multi-points de vue. Dans *Actes de la conférence NimesTIC-01*, 2001.
- [Falquet et Mottaz, 2002] G. Falquet et C.-L. Mottaz. A Model for the Collaborative Design of Multi-Point-of-View Terminological Knowledge Bases. Dans *Knowledge Management and Organizational Memories*, rédacteurs R. Dieng et N. Matta. Kluwer, 2002.

-
- [Ferber, 1989] J. Ferber. *Objets et agents : une étude des structures de représentation et de communications en intelligence artificielle*. Thèse de doctorat d'état, Université Pierre et Marie Curie – Paris 6, Juin 1989.
- [Fuchs *et al.*, 1999] B. Fuchs, J. Lieber, A. Mille, et A. Napoli. Towards a Unified Theory of Adaptation in Case-Based Reasoning. Dans *Proc. of the Third International Conference on Case-Based Reasoning (ICCBR-99), Seeon Monastery (Germany)*, rédacteurs K.-D. Althoff, R. Bergmann, et L.K. Branting, Lecture Notes in Artificial Intelligence 1650, pages 104–117. Springer, Berlin, 1999.
- [Fuchs, 1997] B. Fuchs. *Représentation des connaissances pour le raisonnement à partir de cas : le système Rocade*. Thèse d'université, Université Jean Monnet de Saint-Etienne, Octobre 1997.
- [Gebhardt *et al.*, 1997] F. Gebhardt, A. Voß, W. Gräther, et B. Schmidt-Belz. *Reasoning With Complex Cases*. Kluwer academic publishers, 1997.
- [Ghidini et Guinchiglia, 2001] C. Ghidini et F. Guinchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2) :221–259, 2001.
- [Gibaud *et al.*, 2004] B. Gibaud, M. Dojat, H. Benali, O. Dameron, J.P. Matsumoto, M. Pellegrini-Issac, R. Valabregue, et C. Barillot. Towards an Ontology for Sharing Neuroimaging Data and Processing Methods : Experience Learned from the Development of a Demonstrator. Dans *Proc. of the Didamic Workshop, MICCAI 2004 Conference*, rédacteurs M. Dojat et B. Gibaud, pages 15–23, 2004.
- [Golbeck *et al.*, 2003] J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler, et B. Parsia. The National Cancer Institute's Thésaurus and Ontology. *Journal of Web Semantics*, 1(1) :75–80, 2003.
- [Gómez-Albarrán *et al.*, 1999] M. Gómez-Albarrán, P.A. Gonzàles-Calero, B. Díaz-Agudo, et C. Fernández-Conde. Modelling the CBR Life Cycle Using Description Logics. Dans *Proc. of the International Conference on Case-Based Reasoning, ICCBR'99*, rédacteurs K.-D. Althoff, R. Bergmann, et L.K. Branting, volume 1650 de *Lecture Notes in Artificial Intelligence*, pages 147–161. Springer, 1999.
- [Groot *et al.*, 2005] P. Groot, H. Stuckenschmidt, et H. Wache. Approximating Description Logic Classification for Semantic Web Reasoning. Dans *The Semantic Web : Research and Applications, Proc. of the Second European Semantic Web Conference, ESWC 2005*, pages 318–332, 2005.
- [Gruber, 1993] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Dans *Formal Ontology in Conceptual Analysis and Knowledge Representation*, rédacteurs N. Guarino et R. Poli, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.
- [Guarino, 1997] N. Guarino. Understanding, Building, and Using Ontologies. *International Journal of Human-Computer Studies (IJHCS)*, 46(2–3) :293–310, 1997.
- [Guarino, 1998] N. Guarino. *Formal Ontology and Information Systems*, 1998.
- [Guinchiglia et Serafini, 1994] F. Guinchiglia et L. Serafini. Multilanguage hierarchical logics, or : how to do without modal logics. *Artificial Intelligence*, 65(1) :29–70, 1994.
- [Haarslev et Möller, 2001] V. Haarslev et R. Möller. Description of the Racer System and its Applications. Dans *Proc. of the 2001 International Description Logics Workshop*, rédacteurs C. Goble, D. L. McGuinness, R. Möller, et P. F. Patel-Schneider, pages 46–55, 2001.
- [Hahn et Schulz, 2004] U. Hahn et S. Schulz. Building a Very Large Ontology from Medical Thesauri. Dans Staab et Studer [Staab et Studer, 2004].
- [Hammond, 1989] K. J. Hammond. CHEF. [Riesbeck et Schank, 1989], chapter 6, pages 165–212.
- [Hammond, 1990] K. J. Hammond. Explaining and Repairing Plans that Fail. *AI Magazine*, 45(1–2) :173–228, 1990.

- [Haton *et al.*, 1991] J.-P. Haton, N. Bouzid, F. Charpillet, M.-C. Haton, B. Lâasri, H. Lâasri, P. Marquis, T. Mondot, et A. Napoli. *Le raisonnement en intelligence artificielle*. InterEditions, Paris, 1991.
- [Hautamäki, 1986] A. Hautamäki. Points of View and Their Logical Analysis. *Acta Philosophica Fennica*, 41, 1986.
- [Hendler et W3C Communications Team, 2004] J. Hendler et W3C Communications Team. Frequently Asked Questions on W3C's Web Ontology Language (OWL), 2004. <http://www.w3.org/2003/08/owlfaq.html>, dernière consultation : juillet 2005.
- [Horrocks *et al.*, 2003] I. Horrocks, P.F. Patel-Schneider, et F. van Harmelen. From SHIQ and RDF to OWL : the making of a Web Ontology Language. *Journal of Web Semantics*, 1(1) :7–26, decembre 2003.
- [Horrocks et Patel-Schneider, 1998] I. Horrocks et P.F. Patel-Schneider. Optimising Propositional Modal Satisfiability for Description Logic Subsumption. Dans *Artificial Intelligence and Symbolic Computation : International Conference AISC'98*, number 1476 in Lecture Notes in Artificial Intelligence, pages 234–246. Springer, 1998.
- [Horrocks et Sattler, 2001] I. Horrocks et U. Sattler. Ontology Reasoning in the $\mathcal{SHOQ}(\mathcal{D})$ Description Logic. Dans *Proc. of the 17th Int. Joint Conference on Artificial Intelligence (IJCAI 2001)*, rédacteur B. Nebel, pages 199–204. Morgan Kaufmann, 2001.
- [Horrocks, 1998] I. Horrocks. Using an expressive description logic : FaCT or fiction ? Dans *Principles of Knowledge Representation and Reasoning : Proc. of the Sixth International Conference (KR'98)*, rédacteurs A. G. Cohn, L. Schubert, et S. C. Shapiro, pages 636–647. Morgan Kaufmann Publishers, June 1998.
- [Kalfoglou et Schorlemmer, 2003] Y. Kalfoglou et M. Schorlemmer. Ontology Mapping : the State of the Art. *The Knowledge Engineering Review*, 18(1) :1–31, 2003.
- [Kamp, 1996] G. Kamp. Using description logics for knowledge intensive case-based reasoning. Dans *Third European Workshop on Case-Based Reasoning (EWCBR'96), Lausanne, Switzerland*, rédacteurs B. Faltings et I. Smith, pages 204–218. Springer-Verlag, Berlin, 1996.
- [Kayser, 1997] Daniel Kayser. *La représentation des connaissances*. Hermes, 1997.
- [Knublauch *et al.*, 2004] H. Knublauch, R.W. Ferguson, N.F. Noy, et M.A. Musen. The Protégé OWL Plugin : An Open Development Environment for Semantic Web Applications. Dans *Proc. of the Third International Semantic Web Conference, ISWC 2004*, 2004.
- [Koehler, 1994] J. Koehler. An application of terminological logics to case-based reasoning. Dans *KR-94*, pages 351–362, 1994.
- [Koivunen et Miller, 2002] M.-R. Koivunen et E. Miller. W3C Semantic Web Activity. Dans *Semantic Web Kick-off Seminar in Finland*, rédacteur Eero Hyvönen, pages 27–44, 2002. <http://www.w3.org/2001/12/semweb-fin/w3csw>.
- [Kolodner, 1993] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
- [Lassila et McGuinness, 2001] O. Lassila et D. McGuinness. The Role of Frame-Based Representation on the Semantic Web. *Electronic Transactions on Artificial Intelligence*, 6(5), 2001.
- [Leake *et al.*, 1997] D. B. Leake, A. Kinley, et D. C. Wilson. A Case Study of Case-Based CBR. Dans *Proc. of the Second International Conference on Case-Based Reasoning Research and Development, ICCBR 97*, pages 371–382. Springer, 1997.
- [Leake et Sooriamurthi, 2003] D. B. Leake et R. Sooriamurthi. Dispatching Cases Versus Merging Case-Bases : When MCBR Matters. Dans *Proc. of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2003)*, pages 129–133. AAAI Press, 2003.

-
- [Lenz *et al.*, 1998] rédacteurs M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, et S. Wess. *Case-Based Reasoning Technology : From Foundations to Applications, LNAI 1400*, volume 1400 de *Lecture Notes in Artificial Intelligence*. Springer, 1998.
- [Lieber *et al.*, 2002] J. Lieber, M. d'Aquin, P. Bey, B. Bresson, O. Croissant, P. Falzon, A. Lesur, J. Lévesque, V. Mollo, A. Napoli, M. Rios, et C. Sauvagnac. The Kasimir Project : Knowledge Management in Cancerology. Dans *Proc. of the 4th International Workshop on Enterprise Networking and Computing in Health Care Industry, HealthCom 2002*, juin 2002.
- [Lieber *et al.*, 2003] J. Lieber, M. d'Aquin, P. Bey, A. Napoli, M. Rios, et C. Sauvagnac. Acquisition of Adaptation Knowledge for Breast Cancer Treatment Decision Support. Dans *Proc. of the 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003*, 2003.
- [Lieber *et al.*, 2004] J. Lieber, M. d'Aquin, S. Brachais, et A. Napoli. Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas. Dans *Actes du 12ème Atelier de Raisonnement à Partir de Cas - RàPC'04*, pages 53–60, 2004.
- [Lieber et Napoli, 1996] J. Lieber et A. Napoli. Using classification in case-based planning. Dans *Proc. of the 12th European Conference on Artificial Intelligence (ECAI-96)*, rédacteur W. Wahlster, pages 132–136. John Wiley & Sons, Ltd., 1996.
- [Lieber, 1997] J. Lieber. *Raisonnement à partir de cas et classification hiérarchique. Application à la planification de synthèse en chimie organique*. Thèse d'université, Université Henri Poincaré Nancy 1, Octobre 1997.
- [Lieber, 2002] J. Lieber. Strong, Fuzzy and Smooth Hierarchical Classification for Case-Based Problem Solving. Dans *Proc. of the 15th European Conference on Artificial Intelligence (ECAI-02)*, rédacteur F. van Harmelen. IOS Press, Amsterdam, 2002.
- [López *et al.*, 1999] M. Fernández López, A. Gómez-Pérez, J. Pazos Sierra, et A. Pazos Sierra. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(1) :37–46, 1999.
- [Lutz, 2003] C. Lutz. Description Logics with Concrete Domains - A Survey. Dans *Advances in Modal Logics Volume 4*, rédacteurs P. Balbiani, N.-Y. Suzuki, F. Wolter, et M. Zakharyashev. King's College Publications, 2003.
- [Maedche *et al.*, 2002] A. Maedche, S. Staab, N. Stojanovic, R. Studer, et Y. Sure. SEMantic portAL - The SEAL approach. Dans *Creating the Semantic Web*. MIT Press, 2002.
- [Maedche et Staab, 2000] A. Maedche et S. Staab. The Text-To-Onto Ontology Learning Environment (software demonstration). Dans *Proc. of the Eight International Conference on Conceptual Structures, ICCS 2000*, pages 14–18, 2000.
- [Maedche et Staab, 2004] A. Maedche et S. Staab. Ontology Learning. Dans Staab et Studer [Staab et Studer, 2004].
- [Mariño, 1993] O. Mariño. *Raisonnement classificatoire dans une représentation multi-points de vue*. Thèse d'université, Université Joseph Fourier - Grenoble 1, Octobre 1993.
- [Martin *et al.*, 2004] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, et K. Sycara. Bringing Semantics to Web Services : The OWL-S Approach. Dans *Proc. of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004*, 2004.
- [McBride, 2002] B. McBride. Jena : a semantic Web toolkit. *Internet Computing, IEEE*, 6(6) :55–59, 2002.

- [McCarthy, 1993] J. McCarthy. Notes on Formalizing Context. Dans *Proc. of the International Joint Conference on Artificial Intelligence, IJCAI'93*, 1993.
- [Melis *et al.*, 1998] E. Melis, J. Lieber, et A. Napoli. Reformulation in case-based reasoning. Dans *Proc. of the Fourth European Workshop on Case-Based Reasoning, EWCBR-98*, rédacteurs B. Smyth et P. Cunningham, Lecture Notes in Artificial Intelligence 1488, pages 172–183. Springer, 1998.
- [Melis, 1995] E. Melis. A Model of Analogy-Driven Proof-Plan Construction. Dans *Proc. of the International Joint Conference on Artificial Intelligence, IJCAI'95*, pages 182–189, 1995.
- [Metzger, 2002] J.-L. Metzger. Using DL for a case-based explanation system. Dans *Proc. of the International Workshop on Description Logics - DL2002*, 2002.
- [Mille, 1998] A. Mille. Associer expertise et expérience pour assister les tâches de l'utilisateur. Habilitation à diriger des recherches, Université Claude Bernard, Lyon, Novembre 1998.
- [Milne, 1926] A.A. Milne. *Winnie the Pooh*. 1926.
- [Mollo, 2004] V. Mollo. *Usage des ressources, adaptation des savoirs et gestion de l'autonomie dans la décision thérapeutique*. Thèse d'université en ergonomie, Conservatoire national des arts et métiers, 2004.
- [Musen, 1997] M.A. Musen. Modeling for Decision Support. Dans *Handbook of Medical Informatics*, rédacteurs J. van Bommel et M.A. Musen. Springer, 1997.
- [Nagendra Prasad *et al.*, 1996] M.V. Nagendra Prasad, V.R. Lesser, et S.E. Lander. Retrieval and Reasoning in Distributed Case Bases. *Journal of Visual Communication and Image Representation*, 7(1) :74–87, 1996.
- [Napoli *et al.*, 1997] A. Napoli, J. Lieber, et A. Simon. A classification-based approach to case-based reasoning. Dans *Proc. of the International Workshop on Description Logics (DL'97)*, rédacteurs M.-C. Rousset, R. Brachman, F. Donini, E. Franconi, I. Horrocks, et A. Levy, pages 104–108. Université de Paris-Sud, Orsay, 1997.
- [Napoli, 1997] Amedeo Napoli. Une introduction aux logiques de descriptions. Rapport de Recherche RR 3314, INRIA, 1997.
- [Nelson *et al.*, 2002] S.J. Nelson, T. Powell, et B.L. Humphreys. The Unified Medical Language System (UMLS) Project. Dans *Encyclopedia of Library and Information Science*, pages 369–378. Marcel Dekker, 2002.
- [Netten et Vingerhoeds, 1996] B. Netten et R. Vingerhoeds. Incremental Adaptation for Conceptual Design in EADOCS. 1996. Proc. of the ECAI 96 Workshop on Adaptation in Case-Based Reasoning.
- [Noy *et al.*, 2001] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson, et M. A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2) :60–71, 2001.
- [Noy *et al.*, 2004] N.F. Noy, S. Kunnatur, M. Klein, et M.A. Musen. Tracking changes during ontology evolution. Dans *Proc. of the Third International Conference on the Semantic Web, ISWC 2004*, 2004.
- [Noy et Musen, 2003] N.F. Noy et M.A. Musen. The PROMPT Suite : Interactive Tools For Ontology Merging And Mapping. *International Journal of Human-Computer Studies, IJHCS*, 59(6) :983–1024, 2003.
- [(NWG), 2005] Network Working Group (NWG). Uniform Resource Identifier (URI) : Generic Syntax. RFC 3986, Request for Comments, the Internet Society, janvier 2005. <http://www.gbiv.com/protocols/uri/rfc/rfc3986.html>.
- [Ontañón et Plaza, 2003] S. Ontañón et E. Plaza. Cooperative Case Bartering for Case-Based Reasoning Agent. Dans *Proc. of the 5th Catalanian Conference on AI, CCIA 2002*, volume 2504 de *Lecture Notes in Computer Science*, pages 294–308. Springer, 2003.

-
- [Pan et Horrocks, 2003] J. Pan et I. Horrocks. Web Ontology Reasoning with Datatype Groups. Dans *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)*, rédacteurs D. Fensel, K. Sycara, et J. Mylopoulos, volume 2870 de *Lecture Notes in Computer Science*, pages 47–63. Springer, 2003.
- [Pan et Horrocks, 2005] J. Pan et I. Horrocks. OWL-Eu : Adding Customised Datatypes into OWL. Dans *Proc. of The Second European Semantic Web Conference, ESWC 2005*, rédacteurs A. Gómez-Pérez et J. Euzenat, number 3532 in *Lecture Notes in Computer Science*, pages 153–166. Springer, 2005.
- [Parsia et Sirin, 2004] B. Parsia et E. Sirin. Pellet : An OWL DL Reasoner (poster). Dans *Proc. of the third International Semantic Web Conference, ISWC 2004*, 2004.
- [Peleg *et al.*, 2003] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, et M. Stefanelli. Comparing Computer-interpretable Guideline Models : A Case-study Approach. *Journal of the American Medical Informatics Association*, (10) :52–68, 2003.
- [Pirolli *et al.*, 2001] P. Pirolli, S. K. Card, et M. M. Van Der Wege. Visual Information Foraging in a Focus+Context Visualization. Dans *Proc. of the ACM Conference on Human Factor in Computing Systems (CHI-01)*, pages 506–513, Seattle, 2001. ACM Press.
- [Pusic et Ansermino, 2004] M. Pusic et M. Ansermino. Clinical decision support systems. *BC Medical Journal*, 46(5) :236–239, juin 2004.
- [Quinlan, 1986] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1) :81–106, 1986.
- [(RDF-CWG), 2004a] RDF Core Working Group (RDF-CWG). RDF Vocabulary Description Language 1.0 : RDF Schema. W3C Recommendation, février 2004. <http://www.w3.org/TR/rdf-schema/>.
- [(RDF-CWG), 2004b] RDF Core Working Group (RDF-CWG). RDF/XML Syntax Specification (Revised). W3C Recommendation, février 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [(RDF-CWG), 2004c] RDF Core Working Group (RDF-CWG). Resource Description Framework (RDF) : Concepts and Abstract Syntax. W3C Recommendation, février 2004. <http://www.w3.org/TR/rdf-concepts/>.
- [Rector et Nowlan, 1993] A. Rector et W. Nowlan. The GALEN project. *Computer Methods and Programs in Biomedicine*, 45 :75–78, 1993.
- [Rivière et Dieng, 1997] M. Rivière et R. Dieng. Introduction of Viewpoints in Conceptual Graph Formalism. Dans *Proc. of the 5th International Conference on Conceptual Structures, ICCS'97*, 1997.
- [Richter, 1998] M.M. Richter. Introduction. Dans Lenz *et al.* [Lenz *et al.*, 1998], chapter 1.
- [Riesbeck et Schank, 1989] C. K. Riesbeck et R. C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.
- [Rios *et al.*, 2003] M. Rios, E. Desandes, B. Bresson, I. Klein, V. Demange, et P. Bey. Référentiels de bonnes pratiques cancérologiques : étude comparative de trois supports d’aide à la proposition thérapeutique pour les cancers du sein et de la prostate en lorraine. *Bull Cancer*, 90(4) :363–370, 2003.
- [Rissland *et al.*, 1993] E.L. Rissland, D.B. Skalak, et M.T. Friedman. Case Retrieval through Multiple Indexing and Heuristic Search. Dans *Proc. of the International Joint Conference on Artificial Intelligence, IJCAI'93*. 1993.
- [(RSO), 2005a] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie chimiothérapie néo-adjuvante. http://www.oncolor.org/referentiels/sein/sein_ttt_adj_neo.htm, 2005. dernière consultation : mai 2005.

- [(RSO), 2005b] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie chirurgie des tumeurs opérables d’emblée. http://www.oncolor.org/referentiels/sein/sein_ttt_chir_arb.htm, 2005. dernière consultation : mai 2005.
- [(RSO), 2005c] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie radiothérapie. http://www.oncolor.org/referentiels/sein/sein_ttt_rdth_arb.htm, 2005. dernière consultation : mai 2005.
- [(RSO), 2005d] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie traitement médical adjuvant. http://www.oncolor.org/referentiels/sein/sein_ttt_adj.htm, 2005. dernière consultation : mai 2005.
- [(RSO), 2005e] Réseau de santé Oncolor (RSO). Site internet du réseau de santé en cancérologie de Lorraine, Oncolor. <http://www.oncolor.org/>, 2005. dernière consultation : mai 2005.
- [Salotti et Ventos, 1999] S. Salotti et V. Ventos. Une approche formelle du raisonnement à partir de cas dans une logique de descriptions. *Revue d’Intelligence Artificielle*, 13 :37–72, 1999.
- [Sauvagnac *et al.*, 1999] C. Sauvagnac, J. Stines, A. Lesur, P. Falzon, et P. Bey. Application of Therapeutic Protocols : A Tool to Manage Medical Knowledge. Dans *Proc. of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AMIDM’99*, rédacteurs W. Horn, Y. S. Shahar, G. Lindberg, S. Andreassen, et J. Wyatt, volume 1620 de *LNAI*, pages 86–90. Springer-Verlag, 1999.
- [Sauvagnac, 2000] C. Sauvagnac. *La construction de connaissances par l’utilisation et la conception de procédures. Contribution au cadre théorique des activités métafonctionnelles*. Thèse d’université en ergonomie, Conservatoire national des arts et métiers, 2000.
- [Schreiber *et al.*, 1999] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, et B. Wielinga. *Knowledge Engineering and Management – The CommonKADS Methodology*. MIT Press, 1999.
- [Serafini et Taminin, 2004] L. Serafini et A. Taminin. Local Tableaux for Reasoning in Distributed Description Logics. Dans *Proc. of the International Workshop on Description Logics, DL’04*, rédacteurs V. Haarslev et R. Moeller, pages 100–109, 2004.
- [Serafini et Taminin, 2005] L. Serafini et A. Taminin. DRAGO : Distributed reasoning architecture for the semantic web. Dans *Proc. of the Second European Semantic Web Conference (ESWC’05)*, rédacteurs A. Gomez-Perez et J. Euzenat, volume 3532 de *Lecture Notes in Computer Science*, pages 361–376. Springer, May 2005.
- [Shortliffe, 1986] E. H. Shortliffe. Medical Expert Systems Research at Stanford University. Dans *Proc. of the 20th Computer Science Symposium*, 1986.
- [Simon, 2000] A. Simon. *Outils classificatoires par objets pour l’extraction de connaissances dans les bases de données*. Thèse d’université, Université Henri Poincaré – Nancy 1, 2000.
- [Smyth, 1996] B. Smyth. *Case-Based Design*. PhD. thesis, Trinity College, University of Dublin, April 1996.
- [Séroussi *et al.*, 2001] B. Séroussi, J. Bouaud, et E.-C. Antoine. Oncodoc : a successful experiment of computer-supported guideline development and implementation in the treatment of breast cancer. *Artificial Intelligence in Medicine*, 22 :43–64, 2001.
- [Staab *et al.*, 2001] S. Staab, M. Erdmann, et A. Maedche. Ontologies in RDF(S). *Linköping Electronic Articles in Computer and Information Science*, 6(9), 2001.

-
- [Staab et Studer, 2004] rédacteurs S. Staab et R. Studer. *Handbook on Ontologies*. Springer, 2004.
- [Stollberg *et al.*, 2004] M. Stollberg, H. Lausen, R. Lara, Y. Ding, H. Sung-Kook, et D. Fensel. Towards Semantic Web Portals. Dans *Proc. of the WWW04 Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.
- [Stracia, 1998] U. Stracia. A Fuzzy Description Logic. Dans *Proc. of the 15th National Conferences on Artificial Intelligence (AAAI-98) and the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 534–599, 1998.
- [Stuckenschmidt *et al.*, 2004] H. Stuckenschmidt, F. van Harmelen, L. Serafini, et F. Giunchiglia. Using C-OWL for the Alignment and Merging of Medical Ontologies. Dans *Proc. of the KR workshop on Knowledge representation in medicine, KRMED 2004*, 2004.
- [Stumme et Maedche, 2001] G. Stumme et A. Maedche. FCA-MERGE : Bottom-Up Merging of Ontologies. Dans *Proc. of the International Joint Conference on Artificial Intelligence, IJCAI 2001*, pages 225–234, 2001.
- [Sure *et al.*, 2002] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, et D. Wenke. OntoEdit : Collaborative Ontology Development for the Semantic Web. Dans *Proc. of the first International Semantic Web Conference 2002, ISWC 2002*, volume 2342 de *Lecture Notes in Computer Science*. Springer, 2002.
- [Sure *et al.*, 2004] Y. Sure, S. Staab, et R. Studer. On-To-Knowledge Methodology. Dans Staab et Studer [Staab et Studer, 2004].
- [(SWBPDWG), 2005] Semantic Web Best Practices and Deployment Working Group (SWBPDWG). XML Schema Datatypes in RDF and OWL. W3C Working draft, avril 2005.
- [Szathmary et Napoli, 2005] L. Szathmary et A. Napoli. CORON : A Framework for Levelwise Itemset Mining Algorithms. Dans *Proc. of the Third International Conference on Formal Concept Analysis, ICFCA '05*, pages 110–113, 2005.
- [The Unicode consortium, 2003] The Unicode consortium. *The Unicode Standard, version 4.0*. Addison-Wesley, 2003. voir par exemple <http://www.unicode.org/versions/Unicode4.1.0/>, dernière consultation : juillet 2005.
- [Tresp et Molitor, 1998] C.B. Tresp et R. Molitor. A Description Logic for Vague Knowledge. Dans *Proc. of the 13th European Conference on Artificial Intelligence, ECAI'98*, pages 361–365, 1998.
- [Wache *et al.*, 2004] H. Wache, L. Serafini, et R. Garcia-Castro (coordinators). Survey on Scalability Techniques for Reasoning with Ontologies. Knowledge Web Deliverable 2.1.1, Knowledge Web network of excellence, 2004.
- [(WOWG), 2004a] W3C Web Ontology Working Group (WOWG). Owl web ontology language guide. W3C recommendation, février 2004. <http://www.w3.org/TR/owl-guide/>.
- [(WOWG), 2004b] W3C Web Ontology Working Group (WOWG). Owl web ontology language overview. W3C recommendation, février 2004. <http://www.w3.org/TR/owl-features/>.
- [(WOWG), 2004c] W3C Web Ontology Working Group (WOWG). Owl web ontology language reference. W3C recommendation, février 2004. <http://www.w3.org/TR/owl-ref/>.
- [(WOWG), 2004d] W3C Web Ontology Working Group (WOWG). Owl web ontology language semantics and abstract syntax. W3C recommendation, février 2004. <http://www.w3.org/TR/owl-semantics/>.
- [(WQSWG), 2005] Web Services Description Working Group (WQSWG). Wsdl 2.0 spécifications. W3C working drafts, mai 2005. <http://www.w3.org/2002/ws/desc/>.

- [(XML-CWG), 2004] XML Core Working Group (XML-CWG). Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation, février 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [(XML-SWG), 2004] XML Schema Working Group (XML-SWG). XML Schema Part 2 : Datatypes Second Edition. W3C Recommendation, octobre 2004.
- [(XPWG), 2003] XML Protocol Working Group (XPWG). SOAP specifications. W3C recommendations, juin 2003. <http://www.w3.org/TR/soap/>.

Annexe A

Sources des modèles utilisés pour le RÀPC et algorithmes

A.1 Code OWL FULL du modèle des reformulations

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:kabr="http://kasimir.loria.fr/cbr#"
  xmlns="http://kasimir.loria.fr/cbr#"
  xml:base="http://kasimir.loria.fr/cbr#"

  <owl:Ontology rdf:about="KCBROnto">
    <rdfs:comment>Le modèle des reformulations pour le RàPC en OWL.</rdfs:comment>
  </owl:Ontology>

  <!-- Probleme, solution et aPourSolution -->

  <owl:Class rdf:ID="Probleme">
    <rdfs:label>Probleme pour le RaPC</rdfs:label>
  </owl:Class>

  <owl:Class rdf:ID="Solution">
    <rdfs:label>Solution pour le RaPC</rdfs:label>
  </owl:Class>

  <owl:ObjectProperty rdf:ID="aPourSolution">
    <rdfs:domain rdf:resource="#Probleme"/>
    <rdfs:range rdf:resource="#Solution"/>
  </owl:ObjectProperty>

  <!-- reformulation -->

  <owl:ObjectProperty rdf:ID="reformulation">
    <rdfs:domain rdf:resource="#RelationProblemes"/>
    <rdfs:range rdf:resource="#AdaptationSolutions"/>
  </owl:ObjectProperty>

  <!-- RelationProblemes et AdaptationSolutions sont des types de proprietes :
```

```
    des sous-classes de owl:ObjectProperty -->

<owl:Class rdf:ID="RelationProblemes">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:Class>

<owl:Class rdf:ID="AdaptationSolutions">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:Class>

<!-- Chemin de similarite et chemin d'adaptation -->

<owl:Class rdf:ID="CheminSimilarite">
  <rdfs:subClassOf rdf:resource="#RelationProblemes"/>
</owl:Class>

<owl:FunctionalProperty rdf:ID="relationPrecedente">
  <rdfs:domain rdf:resource="#CheminSimilarite"/>
  <rdfs:range rdf:resource="#RelationProblemes"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="relationSuiivante">
  <rdfs:domain rdf:resource="#CheminSimilarite"/>
  <rdfs:range rdf:resource="#RelationProblemes"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="pbi">
  <rdfs:domain rdf:resource="#CheminSimilarite"/>
  <rdfs:range rdf:resource="#Probleme"/>
</owl:FunctionalProperty>

<owl:Class rdf:ID="CheminAdaptation">
  <rdfs:subClassOf rdf:resource="#AdaptationSolutions"/>
</owl:Class>

<owl:FunctionalProperty rdf:ID="adaptationPrecedente">
  <rdfs:domain rdf:resource="#CheminAdaptation"/>
  <rdfs:range rdf:resource="#AdaptationSolutions"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="adaptationSuiivante">
  <rdfs:domain rdf:resource="#CheminAdaptation"/>
  <rdfs:range rdf:resource="#AdaptationSolutions"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="soli">
  <rdfs:domain rdf:resource="#CheminAdaptation"/>
  <rdfs:range rdf:resource="#Solution"/>
</owl:FunctionalProperty>

</rdf:RDF>
```

A.2 Code OWL de l'exemple pour le traitement du cancer du sein

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:ko="http://kasimir.loria.fr/referentiels/ttt-sein#"
xmlns:kcbr="http://kasimir.loria.fr/cbr#"
xmlns="http://kasimir.loria.fr/exemples/cbr#"
xml:base="http://kasimir.loria.fr/exemples/cbr#"

<owl:Ontology rdf:about="exemple">
  <rdfs:comment>
    Exemple de representation de connaissances d'adaptation pour le traitement
    du cancer du sein dans le projet Kasimir.
  </rdfs:comment>
  <!-- le modele des reformulations -->
  <owl:imports rdf:resource="http://www.loria.fr/~daquin/onto/cbr/kcbr.owl"/>
  <!-- le modele des connaissances du domaine -->
  <owl:imports rdf:resource="http://www.loria.fr/~daquin/onto/ref/ttt-sein.owl"/>
</owl:Ontology>

<!-- liens entre elements du RaPC et elements du domaine -->

<owl:Class rdf:about="http://kasimir.loria.fr/referentiels/ttt-sein#Patient">
  <rdfs:subClassOf rdf:resource="http://kasimir.loria.fr/cbr#Probleme"/>
</owl:Class>

<owl:Class rdf:about="http://kasimir.loria.fr/referentiels/ttt-sein#Traitement">
  <rdfs:subClassOf rdf:resource="http://kasimir.loria.fr/cbr#Solution"/>
</owl:Class>

<owl:ObjectProperty
  rdf:about="http://kasimir.loria.fr/referentiels/ttt-sein#recommandation">
  <rdfs:subPropertyOf rdf:resource="http://kasimir.loria.fr/cbr#aPourSolution"/>
</owl:ObjectProperty>

<!-- premiere reformulation -->

<kcbr:RelationProblemes rdf:about="#localisationPessimiste">
  <rdfs:label>Relation entre un patient ayant une tumeur interne et un
    autre dont la localisation est indéterminée</rdfs:label>
  <rdfs:domain rdf:resource="#P-LInt"/>
  <rdfs:range rdf:resource="#P-LInd"/>
  <kcbr:reformulation rdf:resource="#copie"/>
</kcbr:RelationProblemes>

<kcbr:AdaptationSolutions rdf:about="#copie">
  <rdfs:label>Adaptation par copie</rdfs:label>
  <rdfs:domain rdf:resource="#Solution"/>
  <rdfs:range rdf:resource="#Solution"/>
</kcbr:AdaptationSolutions>

<!-- deuxieme reformulation -->

<kcbr:RelationProblemes rdf:about="#changeSexeFM">
  <rdfs:label>Relation entre un patient de sexe féminin et un autre
    de sexe masculin</rdfs:label>
  <rdfs:domain rdf:resource="#Patiente"/>
  <rdfs:range rdf:resource="#Patient-M"/>
  <kcbr:reformulation rdf:resource="#substitutionOvarTam"/>
</kcbr:RelationProblemes>

```

```

<kcbr:AdaptationSolutions rdf:about="#substitution0varTam">
  <rdfs:label>Adaptation par substitution de l'ovariectomie par
    une cure de Tamoxifène</rdfs:label>
  <rdfs:domain rdf:resource="#TTT-0var"/>
  <rdfs:range rdf:resource="#TTT-Tam"/>
</kcbr:AdaptationSolutions>

</rdf:RDF>

```

A.3 Algorithme pour la remémoration

Fonction remémoration (cible)

entrée : *Problème* cible : le problème à résoudre

sortie : Liste(*CheminSimilarité*) CS : les chemins de similarité, liant cible à des classes
index de problèmes sources

Début

Liste(*Problème*) déjàParcouru ← listeVide()

retourner classificationElastique(cible, déjàParcouru)

Fin

Fonction classificationElastique(pb, déjàParcourus)

entrée : *Problème* pb : le problème à classifier

entrée : Liste(*Problème*) déjàParcourus : les problèmes déjà parcourus dans la recherche

sortie : Liste(*CheminSimilarité*) résultat : les chemins de similarité, liant pb à des classes
index de problèmes sources

Début

Liste(*CheminSimilarité*) résultat ← listevide()

Liste(*Couple(Problème, RelationProblèmes)*) plusProchesProblèmes
← génèrePlusProchesProblèmes(pb)

PourTout couple dans plusProchesProblèmes **faire**

Si le problème p de couple n'est pas présent dans déjàParcourus **alors**

ajouter p dans déjàParcourus

/* utilise le test d'instanciation du serveur de connaissances */

Si p est instance d'une classe index d'un problème source **alors**

transformer couple en un *CheminSimilarité* à une étape

ajouter couple dans resultat

FinSi

Liste(*CheminSimilarité*) reste ← classificationElastique(p, déjàParcourus)

PourTout CS dans suite **faire**

RelationProblèmes relation ← la relation dans couple

CheminSimilarité nouveau ← concaténer(reste, relation, p)

ajouter nouveau à resultat

FinPourTout

FinSi

FinPourTout

retourner resultat

Fin

La fonction génèrePlusProchesProblèmes recherche au sein du modèle des connaissances d'adaptation les relations r_i entre problèmes applicables au problème pb passé en paramètre et génère les problèmes intermédiaires pb_i liés à pb par ces relations. La fonction concaténer construit un chemin de similarité à partir d'un chemin de similarité existant (reste), en lui ajoutant une relation entre problèmes

(relation) et un problème (p).

A.4 Algorithme pour l'adaptation

Fonction adaptation (cheminsSimilarité)

entrée : Liste(*CheminSimilarité*) cheminsSimilarité : les chemins de similarité liant cible à des classes index de problèmes sources

sortie : Liste(*Solution*) solutionsCible : Les solutions de cible, construites par adaptation

Début

Liste(*Solution*) solutionsCible ← listeVide()

PourTout cs **dans** cheminsSimilarité **faire**

CheminAdaptation ca ← construireCheminAdaptation(cs)

Solution solution ← récupérer la dernière solution dans ca

 ajouter solution dans solutionsCible

FinPourTout

retourner solutionsCible

Fin

Fonction construireCheminAdaptation(cheminSimilarité)

entrée : *CheminSimilarité* cheminSimilarité : Un chemin de similarité

sortie : *CheminAdaptation* cheminAdaptation : Le chemin d'adaptation correspondant à cheminSimilarité

Début

Probleme pb ← récupérer le premier problème dans cheminSimilarité

 /* récupérer la solution du premier problème du chemin, c'est-à-dire source, correspond soit à récupérer la solution associée directement à l'individu source, soit à récupérer une instance de la classe de solution associée à son index */

Solution solpb ← récupérer la solution de pb

CheminAdaptation cheminAdaptation ← créer un cheminAdaptation débutant par solpb

PourTout *RelationProblèmes* relation **dans** cheminSimilarité **faire**

AdaptationSolutions adaptation ← récupérer la fonction d'adaptation en

 lien avec relation par une reformulation

Solution sol ← appliquerAdaptation(adaptation, solpb)

 cheminAdaptation ← concatener(cheminAdaptation, adaptation, sol)

 solpb ← sol

FinPourTout

retourner cheminAdaptation

Fin

Dans cet algorithme, les indications de type « récupérer » quelque chose, correspondent à une interrogation du serveur de connaissances. La fonction `appliquerAdaptation` renvoie la solution générée par l'application de la fonction d'adaptation passée en paramètre (`adaptation`) sur la solution passée en paramètre (`sol`). La fonction `concatener` construit un chemin d'adaptation à partir d'un chemin de d'adaptation existant (`cheminAdaptation`), en lui ajoutant une fonction d'adaptation (`adaptation`) et une solution (`sol`).

A.5 Code OWL du modèle des opérations de transformation

<rdf:RDF

 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"


```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns="http://kasimir.loria.fr/transformations#"
xml:base="http://kasimir.loria.fr/transformations#"

<owl:Ontology rdf:about="TransformationOnto">
  <rdfs:comment>
    Modèle des opérations de transformation, utilisées notamment pour
    la construction de chemins de similarité et d'adaptation en RàPC.
  </rdfs:comment>
</owl:Ontology>

<!-- Opération -->

<owl:Class rdf:ID="OperationTransformation">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#applicableSur"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#renvoie"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#genere"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="OperationElementaire">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#OperationTransformation"/>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#operateur"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#OperateurTransformation"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="OperationBinaire">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#OperationElementaire"/>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#operateur"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#OperateurBinaire"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

```

    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#operande"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="OperationComposee">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#OperationTransformation"/>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#premiereOperation"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#OperationTransformation"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#deuxiemeOperation"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#OperationTransformation"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<!-- Operateur -->

<owl:Class rdf:ID="OperateurTransformation">
  <owl:Restriction>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:about="#implementePar"/>
    </owl:onProperty>
    <owl:someValuesFrom rdf:resource="#Service"/>
  </owl:Restriction>
</owl:Class>

<owl:Class rdf:ID="OperateurBinaire">
  <rdfs:subClassOf rdf:resource="#OperateurTransformation"/>
</owl:Class>

<!-- Propriétés -->

<owl:ObjectProperty rdf:ID="applicableSur">
  <rdfs:domain rdf:resource="#OperationTransformation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="renvoie">
  <rdfs:domain rdf:resource="#OperationTransformation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="genere">
  <rdfs:domain rdf:resource="#OperationTransformation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:ObjectProperty>

```

```
<owl:FunctionalProperty rdf:ID="opérateur">
  <rdfs:domain rdf:resource="#OperationTransformation"/>
  <rdfs:range rdf:resource="OperationTransformation"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="premiereOperation">
  <rdfs:domain rdf:resource="#OperationComposee"/>
  <rdfs:range rdf:resource="#OperationTransformation"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="deuxiemeOperation">
  <rdfs:domain rdf:resource="#OperationComposee"/>
  <rdfs:range rdf:resource="#OperationTransformation"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="operande">
  <rdfs:domain rdf:resource="#OperationBinaire"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="implementePar">
  <rdfs:domain rdf:resource="#OpérateurTransformation"/>
  <rdfs:range rdf:resource="#Service"/>
</owl:FunctionalProperty>

</rdf:RDF>
```

Résumé

Cette thèse présente une application et des extensions des principes et des technologies relatives au Web sémantique pour la construction d'un portail dédié à la gestion des connaissances en cancérologie. OWL, langage de représentation d'ontologies, est notamment employé pour la représentation des connaissances contenues dans les référentiels, sortes de protocoles de décision médicaux. Les inférences associées à OWL sont utilisées pour l'aide à la décision dans le cadre de l'application des référentiels. Mais les connaissances contenues dans les référentiels ne permettent pas de prendre en compte toutes les situations et nécessitent d'être *adaptées* pour certains cas particuliers. Pour cette raison, nous proposons une extension des technologies du Web sémantique afin de permettre la représentation de *connaissances d'adaptation* et l'implémentation d'un mécanisme de *raisonnement à partir de cas* (RÀPC) intégré à l'infrastructure du Web sémantique. Par ailleurs, dans des domaines complexes tels que la cancérologie, coexistent différents points de vue, c'est-à-dire différentes façons de considérer les connaissances du domaine et différentes façons de les utiliser. La notion d'ontologie ne permet pas de prendre en compte cette multiplicité de points de vue. Pour cette raison, nous avons étudié les formalismes de représentation des connaissances intégrant de façon explicite la notion de point de vue. Nous nous appuyons sur le langage C-OWL afin d'établir une représentation multi-points de vue des connaissances contenues dans les référentiels en accord avec les technologies du Web sémantique. Sur la base de cette représentation multi-points de vue en C-OWL, un mécanisme de RÀPC décentralisé est défini. Celui-ci exploite les multiples points de vue pour distribuer le RÀPC en différents processus. Les relations existant entre les points de vue sont quant à elles utilisées pour l'échange de connaissances et la collaboration entre ces différents processus. Au travers de l'application en cancérologie, la représentation multi-points de vue montre ainsi des apports concernant les difficultés inhérentes à l'ingénierie des connaissances (acquisition, évolution), la mise en œuvre du mécanisme de RÀPC (utilisation de plusieurs cas, combinaison de points de vue) et le « passage à l'échelle » des mécanismes de raisonnements pour le Web sémantique, RÀPC compris.

Mots-clés : Représentation et gestion des connaissances, Web sémantique, représentation multi-points de vue, raisonnement à partir de cas, cancérologie, OWL, logiques de descriptions, C-OWL.

Abstract

This thesis presents an application and some extensions of the semantic Web principles and technologies for the building of a portal dedicated to knowledge management in oncology. OWL, the Web ontology language, is used for representing the knowledge contained in medical decision protocols. The inferences associated with OWL are applied for decision support for the purpose of protocol application. But the knowledge contained in the protocols is not sufficient to take into account all the possible situations and needs to be *adapted* for some particular cases. For this reason, we propose an extension of the semantic Web technologies in order to allow the representation of *adaptation knowledge* and to implement a case-based reasoning (CBR) mechanism integrated to the semantic Web infrastructure. Furthermore, in complex domains like oncology, there exist multiple viewpoints, that are different ways to consider the domain knowledge and different ways to use it. The idea of ontology does not take into account this multiplicity of viewpoints. For this reason, we have studied the knowledge representation formalisms that explicitly integrate viewpoints. We have chosen the C-OWL language in order to establish a multiple viewpoint representation of the knowledge contained in the protocols, in accordance with the semantic Web technologies. On the basis of this multiple viewpoint representation in C-OWL, a decentralized CBR mechanism is defined. It exploits the viewpoints to distribute CBR in several processes. The relations existing between viewpoints are then used to allow knowledge sharing and collaboration between these multiple processes. Following the application in oncology, the multiple viewpoint representation has shown contributions concerning some of the difficulties of knowledge engineering (acquisition, evolution), the building of the CBR mechanism (use of several cases, viewpoint combination) and the scalability of reasoning mechanisms for the semantic Web, including CBR.

Keywords: Knowledge representation and management, semantic Web, multiple viewpoint representation, case-based reasoning, oncology, OWL, description logics, C-OWL.

