

Université de Montréal

OntoMic, une ontologie pour les laboratoires de microbiologie clinique

par Jean-Guy Levreault

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté  
en vue de l'obtention du grade de Maîtrise ès sciences (M. Sc.)  
en informatique  
option intelligence artificielle

novembre 2018

© Jean-Guy Levreault, 2018

## Résumé

Tous peuvent constater dans leur vie quotidienne l'explosion de la quantité d'informations sous forme de textes qui nous sont destinés, ces textes sont partout, livrets d'instructions, règlements, courriels, textos, panneaux publicitaires. Bien sûr ces textes sont indispensables, mais ils sont eux-mêmes source de complexité, en autres parce que les informations qu'ils renferment peuvent être contradictoires.

Au sein du jeu de textes dont une organisation est responsable, les contradictions, manquements et obsolescences ne sont pas acceptables. Pour s'assurer de la qualité de son jeu de texte, l'organisation doit donc attribuer des ressources à sa maintenance et ces ressources augmentent souvent au fil du temps, car la taille du jeu de texte a presque toujours tendance à s'accroître.

Ce mémoire montre comment une ontologie peut s'attaquer à cette complexité dans le domaine des tests en laboratoire de microbiologie médicale. Une ontologie sert à formaliser un langage pour représenter les informations et à donner un cadre à l'organisation de ces informations.

Les ontologies peuvent être adoptées de façon autonome par les organisations et l'échange d'informations entre organisations qui partagent la même ontologie est grandement simplifié. Notre ontologie OntoMic est basée sur Basic Formal Ontology (BFO) du groupe Open Biological and Biomedical Ontology (OBO) Foundry, BFO offre à OntoMic un cadre solidement éprouvé, de plus le groupe OBO offre plusieurs ontologies pour standardiser les entités de base comme les unités de mesure, ce qui favorise l'interopérabilité.

Nous allons présenter une méthode d'importation des informations contenues dans un jeu de documents dans le formalisme d'OntoMic sans avoir à les modifier. Nous allons montrer comment avec l'ontologie on peut faire une gestion manuelle des informations sans ressources excessives.

Mais surtout nous allons présenter comment les capacités d'inférence automatique qu'offrent les ontologies peuvent être utilisées pour détecter automatiquement les contradictions dans les informations encodées par l'ontologie et quels sont les développements nécessaires pour cette automatisation.

## Remerciements

Je tiens à remercier en premier lieu mon directeur M. Guy Lapalme qui n'a pas hésité à prendre sous sa charge l'étudiant fort peu orthodoxe que je suis. De plus, M Lapalme n'a cessé de croire en moi et en mon projet, même quand je n'y croyais plus moi-même.

Je voudrais également remercier le département d'informatique de l'Université de Montréal qui m'a permis de reprendre sur de nouvelles bases une maîtrise que j'avais délaissée il y a déjà trente ans. Je serai, sans l'ombre d'un doute, le candidat qui a pris le plus de temps pour graduer.

Je tiens à remercier mes collègues médecins-microbiologistes qui ont gracieusement accepté d'adapter mon horaire de travail pour que je puisse poursuivre ce projet.

Et finalement je voudrais remercier les membres de ma famille qui n'ont à aucun moment douté de la bonne poursuite de ce projet et qui m'ont toujours encouragé.

# Table des matières

Résumé .....	1
Remerciements .....	2
Table des matières .....	3
Table des figures .....	11
Table des extraits de documents.....	12
Chapitre 1 Introduction .....	14
1.1 Motivation .....	14
1.2 Notre ontologie OntoMic .....	15
1.3 Plan du mémoire.....	16
1.4 Les conventions utilisées dans le mémoire .....	20
Chapitre 2 Le laboratoire de microbiologie.....	21
2.1 Le cycle d'une analyse en microbiologie .....	21
2.1.1 Le travail d'un laboratoire de microbiologie.....	21
2.1.2 Le travail au laboratoire et en amont et aval .....	25
2.2 Les avoirs informationnels des laboratoires .....	26
2.2.1 Requêtes .....	26
2.2.2 Répertoire d'analyse.....	27
2.2.3 Guide de prélèvement.....	28
2.2.4 Protocoles .....	30
2.2.5 Le SIL, un avoir informationnel particulier .....	31
2.3 Conclusion.....	36

Chapitre 3 Analyse du problème et solution .....	37
3.1 Le défi de gérer le « comment faire » .....	37
3.1.1 Le cas particulier de la microbiologie .....	38
3.1.2 Le contexte réglementaire .....	39
3.1.3 L'évolution des activités des laboratoires .....	39
3.1.4 La gestion de l'information dans le défi global des laboratoires .....	40
3.2 Exemples de problèmes avec les objectifs .....	42
Exemple 1 Incohérences sur contenants acceptables pour spécimens .....	43
Exemple 2 Problèmes avec la vérification de la conformité .....	44
Exemple 3 Prise en charge incomplète d'un document normatif .....	46
Exemple 4 Incohérence entre le SIL et un protocole .....	47
Exemple 5 Omission dans le SIL de la possibilité de demander un test .....	47
3.2.1 Conclusion sur ces exemples.....	48
3.3 La gestion de l'information en place à Saint-Jérôme .....	49
3.3.1 Les protocoles .....	49
3.3.2 Les requêtes .....	51
3.3.3 Le répertoire d'analyse et le guide de prélèvement.....	51
3.3.4 Le SIL.....	52
3.4 Analyse des mécanismes qui ont produit les erreurs.....	52
3.4.1 Absence d'un mécanisme de révision systématique périodique. ....	52
3.4.2 Vision trop étroite de l'impact d'un changement.....	53
3.4.3 Manque d'accès aux informations.....	54
3.4.4 Risque de perpétuation des erreurs selon les personnes ciblées par l'information. ....	57

3.4.5	Implantation nécessaire de la notion de domaine des cultures.....	58
3.4.6	Conclusion sur l’analyse des problèmes. ....	58
3.5	Historique des approches tentées à Saint-Jérôme.....	59
3.5.1	Le système de gestion documentaire en place.....	59
3.5.2	Nouvelle version du système.....	60
3.5.3	Expérimentation avec les wikis.....	63
3.5.4	Conclusion sur l’approche par wiki .....	68
3.6	Le choix d’une méthode d’organisation des données .....	69
3.6.1	Le modèle relationnel.....	70
3.6.2	Le modèle par ontologie.....	74
Chapitre 4	Les ontologies.....	76
4.1	Introduction aux ontologies.....	76
4.1.1	La représentation par triplets.....	78
4.2	Revue de littérature .....	79
4.2.1	Les ontologies et classifications médicales .....	79
4.2.2	Codification dans un domaine connexe .....	82
4.2.3	Conclusion sur la revue de littérature.....	85
4.3	L’ontologie de haut niveau BFO .....	85
4.3.1	Le rôle de BFO.....	86
4.3.2	Présentation générale de BFO .....	86
4.3.3	La hiérarchie de classes de BFO .....	89
4.4	Les inférences logiques avec les ontologies.....	96
4.4.1	L’inférence sur les classes : la subsomption .....	97

4.4.2 Le classement des instances .....	97
4.4.3 Les raisonneurs et l'inférence logique.....	98
4.5 Le langage OWL .....	99
4.5.1 Une TAXONOMY en OWL .....	99
4.5.2 Les axiomes.....	99
4.5.3 BFO et OWL, différences dans le schème de représentation.....	102
4.6 Conclusion.....	103
Chapitre 5 La gestion informationnelle.....	105
5.1 Les documents vus comme des regroupements d'informations.....	105
5.2 La notion d'informations et d'ICE .....	106
5.3 Le processus de gestion informationnelle .....	107
5.4 Les étapes de la gestion.....	108
5.4.1 Repérage des informations .....	108
5.4.2 Le classement des informations .....	111
5.4.3 Encodage des informations .....	112
5.4.4 Entreposage .....	113
5.5 Les stratégies d'utilisation des données .....	114
5.5.1 La stratégie maître – réutilisation.....	114
5.5.2 La stratégie dépôt – comparaison.....	115
5.5.3 Amalgame des deux stratégies .....	116
5.6 Sens, sémantique et logique procédurale. ....	116
5.7 Conclusion.....	117
Chapitre 6 Analogie avec des formulaires .....	118

6.1 Un formulaire pour la lecture d'une gélose après incubation .....	119
6.2 La nécessaire création de formulaires patrons .....	121
Le patron de formulaire pour la lecture des géloses.....	121
6.2.1 L'étape de repérage des colonies d'intérêts .....	122
6.2.2 Le travail sur les colonies d'intérêts.....	125
6.3 Le patron de tâche adapté pour la lecture de la gélose-sang .....	125
6.3.1 Omission de la description des colonies suspectes d'être du <i>S. auréus</i> .....	125
6.3.2 Le besoin d'une confirmation de l'identification du <i>S. auréus</i> .....	125
6.4 Les leçons sur l'encodage tirées de l'analogie par formulaire .....	126
6.4.1 Informations sous-entendues et véritables omissions .....	126
6.4.2 Les frontières des tâches .....	126
6.5 Classement sémantique des extraits de textes avec les patrons de formulaire .....	127
6.6 Les recherches et comparaisons avec les formulaires .....	128
6.6.1 Informations encodées et non en étiquettes de textes .....	129
6.6.2 Encodage et vocabulaire standardisé.....	130
6.6.3 Les patrons de formulaires sont insuffisants .....	131
6.7 L'analogie avec formulaire et le formalisme OntoMic .....	133
Chapitre 7 Les ontologies modèles et partenaires d'OntoMic .....	134
7.1 Le rôle des ontologies BFO, OBI et OntoMic .....	134
7.1.1 Le rôle comparé d'OBI et d'OntoMic.....	134
7.2 IAO.....	135
7.2.1 Le passage par un support concret pour les ICE .....	136
7.2.2 Les facettes des informations et IAO .....	137

Chapitre 8 OntoMic.....	140
8.1 L'organisation des éléments d'OntoMic .....	140
8.1.1 Les entités prédéfinies et celle des applications.....	141
8.1.2 Les comparaisons manuelles et automatiques.....	142
8.2 Le classement des informations pour le traitement manuel .....	144
8.2.1 Les textes et leurs extraits. ....	145
8.2.2 Le classement des <code>plan_specification</code> .....	146
8.3 L'encodage pour le traitement automatique.....	147
8.3.1 La classe qui représente les cultures, les <code>planned_process</code> .....	148
8.3.2 Présentation générale des <code>measurement_datum</code> .....	155
8.3.3 Les <code>measurement_datum</code> comme des ICE d'IAO .....	156
8.3.4 La valeur d'un <code>measurement_datum</code> .....	159
8.3.5 Les balises sur les <code>measurement_datum</code> .....	161
8.4 Le schéma complet d'encodage illustré par la culture de selles .....	162
8.4.1 Les constituants d'OntoMic .....	162
8.4.2 Le passage des textes aux <code>planned_process</code> .....	163
8.4.3 L'encodage en langage machine des logiques, les balises .....	166
8.5 L'ordonnancement des tâches en OntoMic .....	179
8.5.1 Approche de programmation fonctionnelle.....	180
8.5.2 Avantages de l'approche par programmation fonctionnelle .....	183
8.6 Difficultés entre le modèle théorique et son application.....	185
8.6.1 La frontière entre sujet et production des <code>measurement_datum</code> .....	185
8.6.2 Plusieurs façons d'encoder un même protocole.....	187

8.7 Inspiration pour notre modèle .....	188
8.8 Conclusion sur le modèle OntoMic.....	189
Chapitre 9 L'utilisation des données encodées .....	191
9.1 Faire des recherches avec OntoMic.....	191
9.2 L'utilisation des axiomes et de la subsomption .....	194
9.2.1 Les clauses.....	194
9.2.2 Particularité du raisonnement automatique sur les axiomes .....	197
9.2.3 Sévérité des clauses et subsomption.....	200
9.2.4 Découpage de la logique, plusieurs avenues de modélisation.....	207
9.2.5 La présence de contraintes qui interfère pendant la subsomption.....	210
9.2.6 Difficultés d'obtenir des généralisations avec la subsomption. ....	211
9.3 Inférence logique et flexibilité dans l'encodage.....	213
9.4 Stratégie granulaire .....	213
9.4.1 L'inférence globale .....	213
9.4.2 L'inférence granulaire .....	214
9.5 Conclusion sur l'utilisation des données encodées .....	216
Chapitre 10 Conclusion.....	218
10.1 Réflexion sur notre travail.....	221
10.2 Le développement futur d'OntoMic.....	222
10.2.1 Concentration sur le pôle « problème ».....	223
10.2.2 Concentration sur le pôle « modèle » .....	223
10.3 Plan de développement.....	224
Glossaire.....	225

Bibliographie.....227

Annexe 1 Exemple de requête.....229

## Table des figures

Figure 4-1, organisation des dependant continuant dans BFO .....	91
Figure 4-2, organisation des independant continuant dans BFO .....	93
Figure 4-3, organisation des occurrent dans BFO .....	96
Figure 6-1, exemple de formulaire pour la lecture de la gélose-sang .....	120
Figure 6-2, exemple de patron de formulaire adaptable pour toute lecture de gélose .....	123
Figure 6-3, formulaire sans encodage des actions à prendre avec le <i>S.auréus</i> .....	130
Figure 6-4, formulaire avec encodage des actions à prendre avec le <i>S.auréus</i> .....	130
Figure 8-1, schéma de l'organisation générale d'OntoMic.....	141
Figure 8-2, schéma des sous-classes de <code>plan_specification</code> dans OntoMic .....	144
Figure 8-3, le schéma d'encodage des textes et les <code>planned process</code> .....	163
Figure 8-4, exemple d'une balise sur un élément matériel.....	167
Figure 8-5, un exemple de balise sur une mesure catégorielle.....	172
Figure 8-6, illustration du mécanisme de <code>planned_process</code> invité .....	182
Figure 8-7, diagramme de l'ontologie SMART .....	188

## Table des extraits de documents

Extrait de document 2-1, illustration d'un choix de tests de microbiologie .....	27
Extrait de document 2-2, extrait du répertoire d'analyses pour la culture de selles.....	28
Extrait de document 2-3, guide de prélèvements du labo SJ pour les selles.....	29
Extrait de document 2-4, message affiché lors de la saisie d'une culture de selles .....	34
Extrait de document 3-1, traitement du spécimen de selles selon le labo SJ .....	42
Extrait de document 3-2, dispositifs de culture recommandés.....	43
Extrait de document 3-3, dispositifs de culture selon le labo SJ.....	44
Extrait de document 3-4, ajout d'une méthode TAAN pour la recherche des STEC .....	45
Extrait de document 3-5, remplacement de la culture de selles par la recherche de <i>C. difficile</i> ....	47
Extrait de document 3-7, test de Grem sur les selles selon le labo SJ .....	48
Extrait de document 3-7, menu de saisie pour la culture de selles selon le labo SJ.....	48
Extrait de document 3-8, liste des protocoles associés la culture de selles selon le labo SJ.....	49
Extrait de document 3-9, lecture du dispositif gélose-sang selon le labo SJ .....	50
Extrait de document 3-11, écran SoftLab pour message sur recherche de <i>C difficile</i> .....	56
Extrait de document 3-12, règle SoftLab pour remplacement par une recherche de <i>C difficile</i> ....	56
Extrait de document 3-13, liste des approbations pour le protocole de selles du labo SJ.....	61
Extrait de document 3-14, liste de changements au protocole de selles du labo SJ.....	62
Extrait de document 3-15, norme sur le transport de selles selon le labo SJ .....	66
Extrait de document 4-1, hiérarchie de classes pour la culture de selles selon SNOMED-CT.....	80
Extrait de document 5-1, listes des géloses avec leur incubation selon le labo SJ .....	108
Extrait de document 5-2, liste des liquides biologiques selon le protocole du labo SJ.....	110

Extrait de document 5-3, liste des liquides biologiques selon répertoire d'analyses du labo SJ .	110
Extrait de document 6-1, lecture de la gélose-sang selon le labo SJ.....	119
Extrait de document 6-2, texte sur la lecture de la gélose-sang transformé en formulaire.....	127
Extrait de document 6-3, lecture de la gélose-sang selon le labo CSL .....	132
Extrait de document 8-2, lecture de la gélose CIN selon le labo SJ.....	183
Extrait de document 8-3, traitement des isolats candidats selon le labo SJ .....	190
Extrait de document 9-1, encodage en triplets RDF d'une axiome.....	193
Extrait de document 9-2, choix des géloses selon le labo CSL.....	195
Extrait de document 9-3, recherche de Rotavirus l'hivers selon le labo SJ .....	208
Extrait de document 9-4, choix des géloses selon le labo SJ .....	211
Extrait de document 9-5, choix des gélose selon le labo CSL .....	211

# Chapitre 1 Introduction

## 1.1 Motivation

J'œuvre comme médecin-microbiologiste depuis vingt ans au laboratoire de Saint-Jérôme, mes collègues et moi assurons la supervision scientifique du laboratoire de microbiologie. Notre domaine se nomme la microbiologie clinique, car elle s'adresse aux agents qui sont infectieux pour les humains.

Dès le début de ma pratique, je me suis intéressé à l'amélioration de la productivité et de la qualité que l'informatisation pouvait apporter aux laboratoires. Au départ, je me suis surtout attardé au système informatique de gestion des analyses, le Système d'Information de Laboratoire (SIL), afin qu'il soit utilisé au maximum de ses capacités.

Il est cependant apparu évident que la seule optimisation de notre usage du SIL était insuffisante pour assurer efficacité et qualité. En effet, les SIL ne gèrent qu'un des grands groupes d'informations qui circulent dans le laboratoire, celui qui concerne les demandes et résultats d'analyses. Les informations sur le « comment faire » ne font pas partie du champ d'application des SIL, des informations comme la manière de prélever, de transporter, de repérer des résultats aberrants, etc.

Pour des raisons que nous explorerons dans le mémoire, il y a une augmentation constante du volume des informations sur ce « comment faire » et bien gérer ces informations est un véritable défi. En effet, ces informations s'adressent à des gens qui ont des formations et des responsabilités très différentes. Ces informations sont de sources disparates et elles sont regroupées dans divers documents, souvent avec des répétitions.

Il faut assurer que ces informations soient à jour, qu'elles ne se contredisent pas entre copies, qu'elles soient conformes aux normes et finalement qu'elles soient complètes. C'est ce que nous nommerons nos trois objectifs.

1. Cohérence
2. Conformité
3. Exhaustivité

Je me suis particulièrement intéressé à ce défi et j'ai participé à des travaux d'informatisation pour le relever, travaux que nous allons présenter dans ce mémoire. Avec ces expériences, j'ai pu constater qu'il fallait organiser les informations avec un schéma. Une fois organisées, les informations pouvaient être manipulées beaucoup plus facilement pour vérifier l'atteinte des trois objectifs.

Pour organiser nos informations sur le « comment faire », nous avons choisi une approche par ontologie parce que non seulement les ontologies sont consacrées à la représentation et à l'organisation des informations, mais aussi parce qu'elles se prêtent au raisonnement automatique. Avec le raisonnement automatique, nous pouvons automatiser la manipulation des informations pour, par exemple, en vérifier la cohérence.

Comme mon champ d'expertise est la microbiologie, ce mémoire s'attardera aux analyses microbiologiques. Mais la réflexion sous-jacente et l'ontologie qui en découle pourront, je crois, être généralisables à d'autres secteurs de la pratique de laboratoire.

## **1.2 Notre ontologie OntoMic**

L'élément central de notre travail est la construction d'une ontologie. Cette ontologie est appelée OntoMic et elle vise à transformer les textes, les bases de données, les diagrammes ou toutes autres entités qui décrivent le « comment faire » en de nouvelles formulations traitables par machine. Le « comment faire » est nommé logique procédurale dans le reste du document.

La prémisse fondamentale d'OntoMic est qu'un protocole peut être encodé par l'imposition de contraintes sur le déroulement d'une analyse, plutôt qu'en spécifiant une liste de gestes. Ces

contraintes portent sur les entités matérielles et les informations utilisées, produites ou transformées pendant l'exécution de l'analyse de microbiologie. Cette prémisse est bien adaptée au domaine des laboratoires, en effet, les techniciennes savent bien comment faire les actions demandées par les protocoles, les informations les plus utiles les guident dans le choix de ces actions.

Le traitement automatique de la formulation en langage machine sera à même de faire les forages de données essentiels à la vérification manuelle de l'atteinte des trois objectifs. La vérification sera aussi automatisée en ayant recours aux capacités d'inférence logique automatique par des programmes que l'on appelle raisonneur.

### **1.3 Plan du mémoire**

À la section suivante, nous ferons la liste des conventions utilisées dans ce mémoire. Vous trouverez également un Glossaire à la fin du mémoire.

Le **Chapitre 2** est consacré à décrire les laboratoires de microbiologie clinique

Nous prenons comme exemple de celui du CISSS des Laurentides installation Saint-Jérôme, il sera désigné par « SJ ». Le travail accompli dans un laboratoire de microbiologie est décrit avec son vocabulaire de métier. Également, nous identifions quels sont les besoins en informations du laboratoire pour qu'il puisse accomplir sa mission. Par la suite, nous présentons les documents de base que l'on retrouve dans un laboratoire avec des exemples provenant du laboratoire SJ.

Le **Chapitre 3** est consacré aux problèmes avec les trois objectifs.

Nous débutons en expliquant comment le contexte de la pratique de la microbiologie rend ce domaine vulnérable à des problèmes avec les objectifs de cohérence, d'exhaustivité et de conformité.

Puis nous passons à l'étude de problèmes réels survenus avec les avoirs informationnels du laboratoire SJ. À la lumière de ces exemples, nous décrivons comment se fait la maintenance des documents et nous étudions les mécanismes par lesquels sont survenus les problèmes. Puis nous présentons nos conclusions sur l'étude de ces erreurs.

Nous allons voir comment notre étude des erreurs nous a guidé dans la recherche de solutions ainsi que dans des expérimentations sur le terrain. Nous montrons les deux éléments que nous croyons essentiels à toutes prises en charge des avoires informationnels : un vocabulaire standard et un schéma d'organisation des informations.

Le chapitre se termine par une exploration de l'organisation des informations par le biais d'une base de données classique, dit modèle relationnel. Nous montrons les limites de ce modèle et pourquoi nous avons opté pour un autre, celui des ontologies.

Le **Chapitre 4** est consacré aux ontologies.

Les notions sont présentées dans ce chapitre en langue naturelle, afin que les particularités du formalisme d'un langage ne nuisent pas à la compréhension. La dernière section du chapitre est consacrée au langage OWL, qui est le langage le plus fréquemment utilisé pour écrire les ontologies.

Nous allons voir la définition d'une ontologie, puis la recherche de modèles pour OntoMic. Les travaux pertinents sont présentés, avec une attention particulière aux ontologies du groupe Open Biological and Biomedical Ontology (OBO). Nous expliquons le choix des ontologies Ontology for Biomedical Investigations (OBI) et Basic Formal Ontology (BFO) du groupe OBO comme ontologies de base pour notre travail sur OntoMic. L'ontologie BFO est centrale dans notre approche et est présentée en profondeur.

Par la suite nous présentons la notion d'inférence automatique par raisonneur.

Le **Chapitre 5** est consacré aux informations contenues dans les documents

Nous définissons le concept d'une unité d'informations, pour ensuite introduire notre vision d'une gestion de ces unités qui vise à les extraire d'un document et à les rendre disponibles. Nous allons explorer les différentes méthodes d'entreposage et d'utilisation des informations extraites des documents.

Le chapitre se termine par la présentation de la modélisation des documents selon OntoMic. Cette modélisation s'appuie sur une dualité : les documents sont vus comme des entités lexicales, des textes et comme des porteurs de sens. Nous introduisons la notion de sémantique pour représenter le sens d'un texte. Les logiques procédurales sont des cas particuliers de sémantique qui représentent le sens d'un protocole.

Le **Chapitre 6** est consacré à l'analogie des formulaires

À partir de ce chapitre, pour alléger le texte, nous restreignons la présentation de notre encodage aux tests de microbiologie faits par culture.

L'encodage par OntoMic est difficile à comprendre, c'est pourquoi nous introduisons dans ce chapitre une analogie à OntoMic. Avec cette analogie, les protocoles sont encodés comme des formulaires à remplir lors de l'exécution d'une culture. À l'aide d'un exemple tiré d'un protocole, nous allons montrer les capacités d'encodage de cette simple analogie. Puis nous allons étudier les capacités de classement des informations de l'analogie.

Cette analogie sera reprise lors de la présentation d'OntoMic, lorsqu'elle peut aider à maîtriser des concepts clés de l'encodage.

Le **Chapitre 7** est consacré aux ontologies OBI et IAO

La première section de ce chapitre est consacrée à l'ontologie OBI, qui est le modèle pour OntoMic. Nous expliquons les rôles pour lesquels OBI et OntoMic ont été conçus et comment ils diffèrent. Nous montrons, malgré ces rôles différents, qu'OBI et OntoMic partagent le même besoin de description des processus de laboratoire.

La seconde section du chapitre est consacrée à Information Artifact Ontology (IAO). IAO est aussi une ontologie du groupe OBO et son rôle est expliqué dans cette section.

Le **Chapitre 8** est consacré à la présentation d'OntoMic

L'encodage avec OntoMic, hérité d'OBI, est puissant, mais difficile à comprendre, dans ce chapitre il est expliqué. L'explication se fait d'abord en langue naturelle pour être ensuite revue avec l'encodage en OWL.

Le chapitre débute par une présentation générale des grandes classes d'OntoMic et de leur organisation. Puis nous expliquons le cheminement allant d'un extrait de texte jusqu'à l'encodage de la logique procédurale du texte dans des structures d'OntoMic. Nous expliquons comment se fait le classement des textes et de leurs extraits selon la nature des tâches auxquelles ils s'adressent.

Puis nous voyons comment la logique procédurale de ces textes est transformée, encodée, en des conditions que les cultures doivent respecter, conditions qu'OntoMic appelle : balises.

Finalement, pour clore la partie démonstration du chapitre, nous présentons la manière avec laquelle OntoMic encode les flots du travail dictés par les protocoles.

Le chapitre se poursuit par une discussion sur les difficultés avec les multiples possibilités de faire l'encodage. Et finalement il se termine sur une courte présentation de l'ontologie SMART, une inspiration pour OntoMic et par une discussion sur les innovations qui ont été faites pour le modèle OntoMic.

Le **Chapitre 9** est consacré à l'inférence automatique

Ce chapitre illustre les nombreuses difficultés qui sont rencontrées lorsque nous tentons une utilisation pratique des données encodées par OntoMic. Dans ce chapitre, nous verrons que de simples tâches, qui n'impliquent aucun raisonnement automatique, sont difficiles à faire en raison du formalisme particulier d'OntoMic.

Par la suite, nous voyons comment, sans stratégie particulière, l'utilisation de l'inférence automatique est vouée à l'échec. Des stratégies pour tenter de pallier l'échec de l'inférence sur les encodages sont envisagées.

## 1.4 Les conventions utilisées dans le mémoire

Genre dans les désignations de personnes	Utilisation de genre masculin ou féminin selon la proportion dans le groupe désigné.
Langue	Les dénominations provenant de références sont laissées dans leur langue d'origine. Les dénominations issues de notre travail sont en français.
Noms d'agent microbiens	Majuscule au nom du genre, minuscule pour l'espèce, le tout en italique <sup>1</sup> . L'utilisation de l'italique ne s'applique pas dans les désignations de l'ontologie.
Désignations de l'ontologie	Restent invariable Le choix de la police différente ne se fera que lorsque nous voulons souligner que le vocable est une désignation dans les ontologies. Lorsque nous faisons référence à l'entité générale hors du contexte de l'ontologie la police reste celle standard et les termes s'accorderont en genre et en nombre.
Désignations de l'ontologie	Police : Arial, les espaces sont remplacés par le caractère tiret bas
Désignations des propriétés	Police : <i>Arial en italique</i>
Désignations des instances	Police : Arial et entre apostrophes "
Hiérarchie de classes	Notation en escalier
Désignation des documents	Les noms de documents tels qu'ils sont établis localement. Nous ajoutons en préfixe un indicateur du laboratoire d'origine : SJ pour Saint-Jérôme, CSL pour la Cité de la Santé de Laval. En suffixe un indicateur de version : « V » suivi du numéro de version.

Tableau 1-1. Les conventions utilisées dans le mémoire

---

<sup>1</sup> Selon l'article (« Scientific Nomenclature - Emerging Infectious Disease journal - CDC », s. d.)

# Chapitre 2 Le laboratoire de microbiologie clinique

Dans ce chapitre nous allons expliquer le travail qui se fait dans un laboratoire de microbiologie. Il faut connaître ce travail pour comprendre comment OntoMic fonctionne, mais aussi pour comprendre les problèmes avec la gestion des informations.

Le but d'OntoMic est de transformer en langage machine le « comment faire » pour les tâches du laboratoire. Mais nous verrons que ce « comment faire » a des interactions importantes avec des tâches externes au laboratoire comme la sélection d'un test lors de la prescription ou l'interprétation des résultats.

## 2.1 Le cycle d'une analyse en microbiologie

### 2.1.1 Le travail d'un laboratoire de microbiologie

Le but d'une analyse en microbiologie est de démontrer la présence ou l'absence d'un agent infectieux et, si un agent est retrouvé, d'en décrire les caractéristiques. Dans ce mémoire nous avons choisi comme exemple la culture de selles, car elle est assez complexe pour bien illustrer toutes les étapes du travail, tout en étant assez routinière pour être faites dans tous les laboratoires.

#### 2.1.1.1 Les informations sur le spécimen et test choisi, les déterminants du travail

Bien sûr on ne recherche pas les agents infectieux chez le patient au complet, le laboratoire recevra ce que l'on appelle un spécimen. Un spécimen est le résultat de la cueillette chez un individu d'une substance biologique que l'on met dans un contenant<sup>2</sup>. Comme entité physique le spécimen est composé de la substance et de son contenant. Le choix de la substance à étudier est fait par le prescripteur selon la situation clinique du patient. Pour notre exemple de culture de selles, la situation clinique sera le plus souvent une diarrhée. Inversement, ce sont parfois les organismes

---

<sup>2</sup> Strictement parlant notre définition n'est pas assez large. En effet, il y a des spécimens qui ne sont pas composés de substances biologiques comme une prothèse articulaire explantée. Il existe aussi des spécimens qui n'ont aucun lien avec un corps humain comme des tests de stérilité déposés dans des plateaux d'instruments passés à l'autoclave.

pathogènes suspectés qui dictent le spécimen nécessaire, par exemple, pour les tests de dépistage des porteurs de bactéries multirésistantes on cueille le spécimen à partir des régions du corps les plus souvent colonisées par la bactérie.

Le jugement clinique guide également le prescripteur dans la sélection des éventuels organismes pathogènes à prendre en compte, car il est impossible de rechercher sur un spécimen tous les agents infectieux qui existent. Le test prescrit<sup>3</sup> détermine la gamme des organismes pathogènes envisagés. Dans une selle par exemple, on pourra rechercher des pathogènes bactériens comme la *Salmonelle*, c'est la culture de selles classique. Mais on pourra aussi rechercher des virus comme le *Norovirus* (qui cause des éclosions de gastro-entérite) ou encore des parasites, ce seront tous des tests différents.

D'autres éléments que le choix de la substance biologique à recueillir et des pathogènes à rechercher ont aussi une influence sur le travail au laboratoire. Ces éléments sont la **localisation\_ - anatomique** d'où provient la substance, la **méthode\_de\_cueillette** et le choix de contenant. On conçoit bien que l'on ne s'attend pas aux mêmes organismes pathogènes pour un pus provenant du visage que d'un autre provenant de la région anale. Pour la méthode de cueillette, il y aura une différence à faire entre une urine recueillie stérilement par ponction vésicale et une autre recueillie par miction, cette dernière étant exposée à la contamination par les bactéries normalement présentes dans l'urètre. Finalement pour le choix du contenant, on traitera différemment une selle qui a été mise dans un contenant sans ajout et une autre qu'on a mise dans un contenant avec un liquide spécial pour en prolonger la durée de conservation.

Donc si le spécimen lui-même est composé d'une portion de substance biologique et d'un contenant, les **information\_spécimen** comprennent les quatre volets : **substance\_bioologique**, **localisation\_anatomique**, **méthode\_cueillette**, **contenant**.

Le laboratoire détermine comment remplir l'**ordonnance** une fois qu'il sait quels pathogènes sont recherchés par le choix du **test** et les spécifications des **information\_spécimen**.

---

<sup>3</sup> L'ordonnance est le document qui résulte de la prescription.

### **2.1.1.2 La détection des agents infectieux**

Pour certains agents infectieux, on peut faire la recherche directement sur le spécimen. Par exemple la recherche de parasites dans les selles peut se faire directement au microscope sur un spécimen de selles concentré par centrifugation.

La plupart du temps cependant, la taille de la population d'un agent infectieux présente dans le spécimen est trop faible pour être mise en évidence par une méthode directe, on a alors recours à des techniques d'amplification. La culture est la technique d'amplification la plus ancienne et encore la plus utilisée<sup>4</sup>; elle consiste à mettre la portion de substance biologique dans des conditions in vitro qui permettent la multiplication de l'agent infectieux. Si l'agent est bien présent, sa population pourra alors atteindre une taille suffisante pour que l'on soit en mesure de le détecter.

### **2.1.1.3 L'identification d'un agent infectieux et sa caractérisation**

Une fois que l'on a repéré un potentiel agent pathogène, ce que l'on appelle un isolat\_d'intérêt, on le caractérise en lui trouvant une place dans la taxonomie, une identification. La taxonomie utilisée en microbiologie clinique est basée sur la taxonomie internationale du vivant (Federhen, 2012), cette taxonomie est hiérarchique et chaque niveau se nomme taxon. La désignation se termine toujours par un genre et puis par une espèce. Par exemple, la lignée (simplifiée) pour l'être humain est Eukaryota ; Metazoa ; Chordata ; Craniata ; Vertebrata ; Euteleostomi ; Mammalia; Eutheria; Euarchontoglires; Primates ; Haplorrhini ; Catarrhini ; Hominidae ; Homo ; Homo sapiens. Notre genre est donc : Homo et notre espèce est : Homo sapiens.

La place que l'on voudra trouver pour un isolat candidat peut être une identification complète au genre et espèce ou un taxon plus général. Parfois également la désignation que l'on donne à un

---

<sup>4</sup> D'autres méthodes sont : les Tests d'Amplification des Acides nucléiques (TAAN) et la détection d'antigène. Les TAAN fonctionnent en multipliant, si elle est présente, une séquence cible d'ARN ou d'ADN. Les séquences cibles d'acide nucléique deviennent détectables une fois qu'elles ont été multipliées. La détection d'antigène se fait avec des anticorps spécifiques pour une molécule cible (antigène), on marque ces anticorps d'un révélateur pour détecter leur liaison.

isolat candidat dans la taxonomie de la microbiologie clinique ne correspond pas directement à un taxon officiel, mais fait l'objet d'un consensus pour les microbiologistes. Un exemple de ceci est la désignation « levures », bien que les levures soient courantes en microbiologie cette désignation n'existe pas dans la taxonomie internationale. Les « levures » sont en réalité une union de plusieurs taxons parmi les Fungi.

Une fois qu'une désignation taxonomique adéquate pour l'isolat candidat a été trouvée par un processus que l'on nomme **identification**. On doit alors statuer si l'isolat est bel et bien pertinent dans le cadre du spécimen et du type de culture en cours. Par exemple, la présence d'une levure est le plus souvent sans conséquence dans un spécimen respiratoire, mais cette même levure a un pouvoir pathogène redoutable si on la retrouve dans le sang.

Même une fois placé adéquatement dans la taxonomie, un agent infectieux n'est pas une simple copie à l'identique à tous les autres de sa catégorie, on voudra parfois le caractériser plus précisément. La caractérisation la plus fréquente est l'**antibiogramme**, qui est l'une étude de l'activité d'agents antimicrobiens in vitro pour en prédire l'efficacité in vivo.

Une fois que l'on a effectué ce travail d'identification et de caractérisation et que la présence de l'isolat est jugée pertinente, on est passé d'un **isolat\_candidat** à un **isolat\_rapportable**, c'est-à-dire à un isolat prêt à apparaître sur un rapport allant au soignant.

Plusieurs des étapes du travail à faire pour passer d'un isolat candidat à un isolat rapportable sont faites manuellement, mais il existe des automates qui font des tâches d'identification et d'antibiogramme<sup>5</sup>.

#### **2.1.1.4 Émission de rapport et compte-rendu du travail**

Pour toute demande d'analyse, le laboratoire doit conserver une trace du travail qui a été effectué dans un **compte-rendu**. Les comptes-rendus de microbiologie sont en partie saisis manuellement, mais s'y ajoutent des informations générées automatiquement par le SIL, comme l'horodatage lors

---

<sup>5</sup> Étude in vitro de l'activité d'une gamme d'antibiotiques contre un agent infectieux afin de prédire l'activité in vivo

du balayage de l'étiquette du spécimen à la réception, ou des informations en provenance des automates.

Toute demande d'analyse doit faire l'objet d'un rapport au prescripteur, même si le spécimen a été reçu détérioré et que l'on a dû le jeter sans le travailler. Le rapport est créé à partir du compte-rendu du travail, mais est différent de ce dernier. La création du rapport est une étape à part entière du travail de laboratoire, une étape où l'on doit décider quelles sont les informations pertinentes à mettre dans le rapport. L'émission d'un rapport peut devoir être faite à plusieurs reprises sur un même test pour produire un ou des rapports préliminaires urgents.

### **2.1.2 Le travail au laboratoire et en amont et aval**

Il est utile de regrouper les tâches à faire pour une analyse de microbiologie selon qu'elles sont faites au laboratoire ou en amont et en aval de celui-ci. Ce regroupement permet de bien identifier les rôles de chacun dans le cheminement.

Les tâches en amont du laboratoire sont : le choix du test et du spécimen (la décision), la prescription, le prélèvement du spécimen et le transport.

Au laboratoire lui-même : la recherche de l'agent infectieux, la caractérisation d'un éventuel agent infectieux et l'élaboration du rapport.

En aval du laboratoire : l'interprétation du résultat

Les acteurs majeurs dans le cycle sont : le patient, le prescripteur, le préleveur, le transporteur, les techniciennes du laboratoire et le destinataire du rapport.

Dans la section suivante, nous verrons les principaux avoirs informationnels du laboratoire ainsi que les acteurs du processus auxquels ils sont destinés.

## **2.2 Les avoirs informationnels des laboratoires**

La plupart des laboratoires ont des avoirs informationnels de base tels que des requêtes, un répertoire d'analyse, un guide de prélèvements et des protocoles.

En plus de ces avoirs traditionnels, les laboratoires du Québec ont tous un Système d'Information de Laboratoire (SIL). Les SIL consignent toutes les données sur les demandes et résultats d'analyses ainsi que les comptes-rendus. Les SIL sont complexes et assez différents des autres avoirs, nous devons les considérer comme des avoirs informationnels stratégiques. En effet, comme ils sont l'instrument de travail de tous les instants, ils orientent le travail, ne serait-ce qu'en proposant des raccourcis claviers pour saisir les comptes-rendus.

Les éléments de ces avoirs informationnels seront appelés **documents**. Nous avons choisi une définition large du mot document comprenant les documents textes classiques, mais aussi les chiffriers, programmes et les tables de bases de données.

### **2.2.1 Requêtes**

Les requêtes sont destinées à l'étape **prescription**, mais elles contiennent aussi des informations sur le **prélèvement**, la saisie dans le SIL.

Les **requêtes** sont souvent encore sous forme papier. Sur ces requêtes, nous retrouvons un choix de tests par des cases à cocher. Un exemple d'une requête de Saint-Jérôme se trouve à l'Annexe 1 et nous en avons un Extrait de document 2-1 qui montre des choix pour quelques tests de microbiologie.

Il y a beaucoup plus de tests disponibles (de l'ordre des milliers) qu'il est possible d'en inscrire sur une seule requête. Les laboratoires utilisent plusieurs requêtes où le jeu de tests listés est adapté selon les spécialités ou selon la section du laboratoire concerné, il y a des requêtes pour l'oncologie,

**MICROBIOLOGIE**

Date et heure prélèvement : \_\_\_\_\_

Antibiotiques reçus: NON  OUI

**201 Culture d'urine** 

**203 Culture de gorge**

**212 Culture de selles # 1** 

**800 Culture de selles # 2** 

**829 Clostridium difficile** 

Pot culture : stérile  milieu Carry-Blair ou entérique  milieu SAF  stérile bâtonnet adhésif 

Extrait de document 2-1, illustration d'un choix de tests de microbiologie  
Source SJ\_LAB-FOR-139

Le symbole  indique quel contenant est à utiliser

l'obstétrique ou encore dédiées à une sous-section comme l'hématologie, la coagulation... La requête montrée en exemple à l'Annexe 1 se veut une requête générale pour les tests les plus demandés en cabinet.

### 2.2.2 Répertoire d'analyse

Le répertoire contient des informations sur toutes les étapes qui se font en amont du laboratoire. Plusieurs personnes peuvent avoir besoin de consulter le répertoire, même les techniciennes pour retrouver les tests rarement demandés.

Un répertoire des analyses offertes est disponible dans tous les laboratoires, mais il s'agit encore souvent d'un document à consulter à l'écran ou à télécharger. Le passage vers une version dynamique est inégal d'un laboratoire à l'autre. Le répertoire est essentiel pour connaître le `code_test` à saisir dans le SIL ou pour connaître des particularités du test comme le contenant à utiliser. À l'Extrait de document 2-2 nous avons un extrait du répertoire du laboratoire SJ pour la culture de selles, ce répertoire est une page web élaborée à partir d'une base de données.

NOM_ANALYSE ▲	CODE ▲	CODE2 ▲	REQ. ▲	SERVICE ▲	JEUNE ▲	SPÉCIMEN ▲
SELLES NO 1 CULTURE	SECUL	212		MICRO		POT STÉRILE
SELLES NO 2 CULTURE	SECU2	800		MICRO		POT STÉRILE
SELLES NO 3 CULTURE (SI NÉCESSAIRE)	SECU3	801		MICRO		POT STÉRILE

NOM_ANALYSE	COD	COD	RE	SERVICE	JEUN	SPÉCIMEN
-------------	-----	-----	----	---------	------	----------

Extrait de document 2-2, extrait du répertoire d'analyses pour la culture de selles  
Source <http://www.cdsjlabo.org/codes-analyses>

Dans le répertoire de SJ, la colonne CODE correspond au `code_test` qui doit être entré dans le SIL lors de la saisie d'une demande d'analyse, la seconde colonne CODE2 correspond à des alias de ces `code_test` composés de chiffres uniquement. Notez que cette colonne CODE2 correspond aux chiffres qui suivent la case à cocher sur la requête comme nous le voyons à l'Extrait de document 2-1.

### 2.2.3 Guide de prélèvement

Le guide de prélèvement est bien sûr en rapport avec l'étape **prélèvement**, mais il contient aussi d'autres informations comme la préparation nécessaire avant de passer le test. Le guide de prélèvement explique comment la portion de substance biologique doit être recueillie et le type de contenant à utiliser. Il peut contenir des renseignements sur la conservation du spécimen avant le transport, les conditions nécessaires pour le transport et les temps d'attente maximaux.

# Selles: cultures , recherche de parasites et de sang occulte

Extrait de CPR-PON-27 SELLES: Cultures , Recherche de Parasites, d'Oxyures, Recherche de sang occulte

Fomulaire – Prescription et Instructions analyses de selles

## PROCÉDURE

### 1 Procédure générale pour recueillir les selles:

- Utiliser un contenant à grande ouverture propre ou un papier plastique pour recueillir les selles, ou directement de la cuvette
- Éviter la contamination par l'urine (sauf dans la couche ou la culotte d'incontinence)
- Ne pas utiliser de laxatif ou de suppositoire pour expulser les selles
- Ne pas recueillir les selles 7 jours suivant un examen baryté
- Utiliser la cuillère soudée dans le couvercle pour déposer la selle dans le contenant approprié
- Prévenir les parents de jeunes enfants que les milieux de conservation sont des **produits toxiques**. Ce

### 2 Particularités à respecter selon l'agent recherché:

#### 2.1 Culture routine 1 ou 2 (E.Coli, Salmonelle, Shigelle, Campylobacter)

- Prélever chaque spécimen à des **jours différents**, intervalle de 24 heures entre deux sp
- Utiliser un milieu de transport **Cary-Blair** (liquide ROSE)
- Respecter la quantité indiquée par la ligne « plein », bien mélanger
- Conserver le spécimen au **réfrigérateur** pour un **maximum de 24 heures**

Extrait de document 2-3, guide de prélèvements du labo SJ pour les selles

Source <http://www.cdsjlabo.org/procedures-prelevement/selles-cultures-recherche-parasites-doxyures-recherche-sang-occulte>

Au laboratoire SJ le guide de prélèvement est présenté sous forme web, mais provient de documents PDF, le site web est donc statique. Pour la culture de selles, nous avons un exemple du guide à l'Extrait de document 2-3 provenant du site web.

Dans ce cas, la procédure sur le web reprend les informations d'une procédure officielle : SJ\_CPR-PON-027\_V10. Notons que le guide de prélèvement web n'a pas de liens avec le répertoire d'analyse. Un utilisateur qui a retrouvé une analyse dans le répertoire ne peut pas cliquer sur un lien et retrouver la méthode de prélèvement.

## 2.2.4 Protocoles

Au Québec, les protocoles de microbiologie sont traditionnellement des documents officiels élaborés et révisés par les médecins-microbiologistes. Ils sont destinés aux techniciennes, mais comprennent des informations sur les étapes en amont et en aval du laboratoire. Le cas échéant les informations destinées à d'autres personnes que les techniciennes sont reprises dans d'autres documents.

Chaque laboratoire a son propre système de classification des documents inspiré d'une norme ISO. Aux laboratoires SJ et CSL, les trois premières lettres du titre de document indiquent le secteur du laboratoire auquel la procédure s'adresse, les trois autres lettres qui suivent décrivent la nature du document, le reste de la désignation est variable selon les laboratoires. Pour notre mémoire, nous ajoutons les préfixes SJ et CSL qui indiquent le laboratoire d'origine du document, nous ajoutons également un suffixe qui indique la version.

Dans ce mémoire, le mot protocole a un sens plus large que le sens traditionnel expliqué ci-haut. Un protocole est tout document scientifique à caractère officiel dont le contenu est sous la responsabilité de professionnels ou d'organismes normatifs et qui dictent comment faire le travail.

Un protocole n'a pas besoin de s'adresser à la totalité de la procédure d'analyse. Par exemple, un document qui n'énonce qu'une simple norme sur le temps d'incubation sera considéré comme un protocole. Un protocole peut aussi décrire autre chose qu'une procédure, ainsi nous considérons comme un protocole la liste des organismes à déclaration obligatoire aux autorités de santé publique.

Notre définition exclut des protocoles les requêtes, les répertoires d'analyses et les guides de prélèvement, car dans la majorité des laboratoires, ils sont élaborés par le personnel technique sous la gouverne des protocoles. Nous nommons documents secondaires ces documents qui ont un certain caractère officiel, mais dont les informations prennent leurs sources dans les protocoles.

Également, nous introduisons la notion d'organisations responsables des protocoles, par exemple le protocole : Liste des maladies, infections et intoxications à déclaration obligatoire (MADO), l'organisme responsable sera l'INSPQ (Institut National de Santé Publique du Québec).

### **2.2.5 Le SIL, un avoir informationnel particulier**

Les SIL sont d'abord conçus pour une utilisation au sein du laboratoire, mais selon les choix locaux, un SIL peut être impliqué dans toutes les étapes du cheminement d'une analyse.

Comme nous l'avons vu dans la section 2.1.1.4 sur la description du travail au laboratoire, chaque demande d'analyse doit avoir son compte-rendu du travail. De nos jours, ces comptes-rendus sont presque toujours entièrement informatisés dans un SIL, les traces du travail ne sont pas conservées ailleurs sauf pour quelques rares exceptions.

Les SIL assignent à chaque demande de test un identificateur unique. Sur le contenant du spécimen, on colle une étiquette avec cet identificateur qui permet à tout moment de retracer les informations sur le patient, le test demandé et les informations sur le spécimen.

Les SIL en place à Saint-Jérôme et Laval sont des instances de SoftLab avec sa sous-section pour la microbiologie SoftMic. Dans la suite du texte, nous utiliserons SoftLab pour désigner ce SIL. Ces instances de SoftLab sont paramétrées localement et cette paramétrisation est fort élaborée. La paramétrisation est en étroite correspondance avec les pratiques et protocoles des laboratoires. Par exemple les codes tests étant définis localement, le code de la même culture d'urine pourra différer d'un laboratoire à l'autre.

Cette paramétrisation passe par des tables que SoftLab désigne comme « set-up », par exemple :

- Liste de tests avec leur `code_test`
- Liste des contenant

- Liste des dispositif\_de\_culture<sup>6</sup>
- Liste des organisme
- Règles de choix des dispositif\_de\_culture
- Table pour les menus d'aide à la saisie des comptes-rendus
- Table pour les fenêtres d'aide surgissantes

### **2.2.5.1 Le SIL, une implantation des protocoles**

Les SIL peuvent être vus comme des implantations des protocoles, car ils gouvernent plusieurs tâches de la microbiologie clinique comme le choix des dispositifs de culture ou la saisie des comptes-rendus. Également les SIL indiquent les actions à prendre par des messages. Ce sont les tables de « set-up » qui permettent au SIL de jouer ce rôle.

#### 2.2.5.1.1 Le choix des contenants pour le spécimen

Nous avons fait mention à la section 2.1.1.1 que le choix du contenant est une des informations spécimens nécessaires. Ceci est vrai pour tous les secteurs du laboratoire ; en biochimie par exemple les tubes de prélèvements sanguins sont codifiés par la couleur de leur bouchon, certaines analyses ne sont compatibles qu'avec un seul type de tube, qu'une seule couleur de bouchon pour d'autres analyses des choix sont possibles.

C'est le SIL qui indique à la personne qui fait le prélèvement le contenant nécessaire, le SIL SoftLab ne permet qu'un seul choix ce qui est une limitation importante comme nous le verrons à l'Exemple 1

#### 2.2.5.1.2 Le choix des dispositifs de culture

SoftLab gouverne le choix des dispositifs en indiquant à la technicienne une sélection de dispositifs lors de l'ensemencement. Lorsque la technicienne fait une lecture du code barre de l'étiquette du

---

<sup>6</sup> En microbiologie clinique il y a un double emploi du vocable « milieu de culture » pour désigner le mélange de substance destiné à nourrir la croissance des organismes et l'ensemble composé d'un contenant et d'un milieu de culture. Nous avons choisi de désigner par « dispositif de culture » l'ensemble du milieu et de son contenant.

spécimen, le SIL accède alors aux déterminants du travail : le test choisi et les informations spécimen. Avec ces déterminants et ses tables, le SIL génère des étiquettes à apposer sur les dispositifs de culture, ces étiquettes indiquent quels dispositifs choisir.

#### 2.2.5.1.3 Les aides à la saisie des comptes-rendus, l'exemple de la lecture d'une gélose

Le SIL renferme aussi des aides à la saisie des comptes-rendus, ces aides sont aussi une implantation des protocoles. Dans ce qui suit, nous verrons un exemple d'aide à la saisie pour la lecture des géloses de la culture de selles.

Nous avons vu à la section 2.1.1.2 que la culture consistait à mettre une portion de substance dans des conditions in vitro nécessaires à la croissance éventuelle d'agents pathogènes. Pour une culture de selles, on dépose un inoculum de selles provenant du spécimen sur des géloses<sup>7</sup>. Les géloses sont ensuite incubées pour des durées standardisées dans des conditions précises de température et de composition en gaz de l'atmosphère. Après incubation, les géloses sont retirées de l'incubateur et examinées à la recherche de caractéristiques signalant la présence d'organismes pathogènes, on appelle ceci la lecture des géloses et les constatations faites lors de la lecture sont consignées dans le compte-rendu.

Pour la culture de selles, nous avons une gélose nommée McConkey Sorbitol, elle est dédiée à la détection d'un des germes responsables de la maladie du hamburger. Lorsque la technicienne veut saisir le résultat de sa lecture de la gélose Mc Conkey sorbitol dans le compte-rendu elle se rend

---

<sup>7</sup> Une gélose est un dispositif de culture que l'on dira « solide » qui est composé d'une couche d'un milieu de culture gélifié contenue dans une boîte de Pétri.



Extrait de document 2-4, message affiché lors de la saisie d'une culture de selles  
Capture d'écran de SoftLab SJ

dans SoftMic à un champ de texte. Elle a alors accès à un menu contextuel qui est associé à cette gélose, ce menu peut en appeler d'autres en cascade. Le texte du compte-rendu de lecture est généré en grande partie par les choix dans ces menus. Ces menus sont donc un reflet direct des protocoles.

#### 2.2.5.1.4 Les messages

Le SIL envoie des messages aux utilisateurs par le biais de fenêtres surgissantes. L'affichage de ces fenêtres est en réponse à des règles qui sont déclenchées par des éléments dans le compte-rendu ou par des conditions générales comme la date. Ces règles sont implantées par des tables de « set-up ». À la Extrait de document 2-4 nous avons un exemple d'une telle fenêtre surgissante, cette fenêtre apparaît lorsqu'un utilisateur tente de faire la saisie d'une culture de selles pour un patient hospitalisé. La fenêtre indique que l'on doit remplacer la culture standard<sup>8</sup> par la recherche du *C difficile*. Le *C difficile* est une bactérie pathogène redoutable, elle peut se répandre parmi les patients hospitalisés.

#### 2.2.5.1.5 La nomenclature des agents infectieux

SoftMic ne permet pas de mettre sur un rapport un organisme qui n'est pas présent dans sa table d'organisme. Le fait que ces tables soient définies localement est un frein majeur dans l'interopérabilité des rapports de microbiologie. Les incohérences entre laboratoires dans les

---

<sup>8</sup> On fait ce remplacement, car les recherches ont montré que les entéropathogènes classiques comme la *Samonelle* sont très rarement retrouvés au-delà de quelques jours après une admission à l'hôpital.

désignations des organismes sont une des raisons pour laquelle le dépôt centralisé de résultats, Dossier Santé Québec (DSQ), n'a implanté aucune normalisation des résultats de microbiologie.

La standardisation de ces tables est facile pour les organismes qui ont la même désignation dans la taxonomie de la microbiologie clinique que dans la taxonomie internationale. Mais elle est plus difficile pour les organismes exclusifs à la taxonomie de la microbiologie clinique, que nous nommons `organismes_cliniques`. Les `organismes_cliniques` sont souvent regroupés selon des caractéristiques observables en laboratoire et il peut y avoir des différences entre les définitions d'un laboratoire à l'autre. Pour illustrer ceci, prenons le cas de l'organisme clinique Entérocoque Résistant à la Vancomycine (ERV), une bactérie multirésistante maintenant endémique dans nos hôpitaux.

La résistance à la vancomycine peut être le résultat d'assemblages génétiques différents, que l'on dénote par `vanA`, `vanB`, `vanC`... Cependant, comme les tests génétiques sont dispendieux, souvent on aura évalué la résistance à la vancomycine sans savoir quel est l'assemblage génétique en cause. Donc le même vocable ERV peut signifier une définition basée sur la génétique et une autre sur le phénotype de résistance.

Avec une ontologie comme OntoMic, nous serons en mesure d'accompagner un rapport qui mentionne la présence d'ERV du code en langage machine de la définition utilisée pour l'ERV.

### **2.2.5.2 Statut du SIL dans les avoirs informationnels**

Nous pourrions être tentés de concevoir le SIL comme un simple document découlant des protocoles, mais dans la pratique il est différent. En effet, n'importe quels professionnels ou techniciennes sont à même de vérifier si une information du guide de prélèvement est présente et si elle est juste. Il en est tout autrement du SIL, car les informations qu'il contient sont dans un format opaque pour tous sauf ceux qui ont eu une formation poussée.

## **2.3 Conclusion**

Ce chapitre a introduit les tâches et le vocabulaire de la microbiologie clinique. Nous avons montré que nous ne pouvions séparer les informations sur le « comment faire », des informations utiles aux autres intervenants dans le cheminement d'une analyse de laboratoire.

Ce chapitre a également montré le rôle primordial du SIL et comment, selon l'organisation locale, il est utilisé par des personnes ayant des rôles fort variés dans le cycle de l'analyse.

## **Chapitre 3 Analyse du problème et solution**

Dans ce qui suit, nous allons donner des exemples de problèmes retrouvés dans les avoirs en place au laboratoire SJ, puis nous allons analyser ces problèmes pour en dégager les causes. Nous verrons comment les méthodes de gestion ont évolué et pourquoi elles se trouvent maintenant sous forte pression par le contexte actuel des laboratoires.

Nous allons montrer comment une solution informatique de gestion, non pas des documents, mais de l'information contenue dans les documents, est à même de s'adapter au contexte actuel et de limiter les risques.

### **3.1 Le défi de gérer le « comment faire »**

Il y a plusieurs raisons pour lesquelles la gestion des informations sur le « comment faire » est un défi, une de celles-ci est la présence de copies des mêmes informations dans des documents différents.

En effet, le laboratoire doit créer des regroupements d'informations, des documents, pour répondre aux besoins des multiples intervenants dans le processus d'une analyse. Ces documents contiennent nécessairement des redondances, car la même information peut être pertinente pour des personnes qui n'accomplissent pas du tout la même tâche. Par exemple, lors d'un changement de type de contenant pour un prélèvement, le médecin doit être informé, mais aussi l'adjointe administrative qui est responsable de faire les commandes de matériel.

Différentes formes de documents sont nécessaires, car il faut s'adresser aux personnes impliquées dans le processus d'analyse d'une manière adaptée. On n'informe pas les techniciennes des particularités d'un nouvel agent infectieux de la même manière que l'on informe le public sur les heures d'ouverture du laboratoire ou les médecins de l'abandon d'une analyse.

Le laboratoire est donc tenu de maintenir ce que nous appellerons une collection de documents de formats variés au sein desquels il y a des redondances. Le propos de ce mémoire n'est pas sur les choix de ces formats, mais bien sur une méthode pour s'assurer que les documents de la collection

satisfont aux trois objectifs de cohérence, de conformité et d'exhaustivité avec un investissement de ressources raisonnable.

### **3.1.1 Le cas particulier de la microbiologie**

En plus de la difficulté liée aux informations redondantes dans des documents de format divers, une autre difficulté provient de la nature même du travail de laboratoire. En effet, le monde des laboratoires cliniques en est un de détails ; une multitude de micro-informations existent sur chacune des analyses et des tâches. Parfois, des détails varient d'un test à l'autre d'une façon qui est fort difficile à retenir, car il n'y a pas de règles générales. Par exemple, pour un test on devra ne pas tenir compte d'une réaction de faible amplitude sous peine d'un faux positif, tandis que pour un autre test semblable, on devra absolument tenir compte d'une réaction de faible amplitude sous peine d'un faux négatif.

Ce phénomène est particulièrement important en microbiologie, car le travail y est encore en grande partie manuel et le jugement de la technicienne doit intervenir à presque toutes les étapes. Souvent, leurs décisions se basent sur des micro-informations qui changent souvent. Par exemple pour les *Salmonella* et l'antibiotique ciprofloxacine, l'ancienne méthode in vitro ne parvenait pas à détecter toutes les résistances observées in vivo. Une autre méthode a été introduite, puis retirée, puis réintroduite, puis finalement totalement remplacée en 2012.

Les décisions des techniciennes s'appuient sur un ensemble de protocoles élaborés par les microbiologistes. Aucune tâche, sauf celles qui sont administratives, n'est faite sans le soutien d'un protocole dans un laboratoire de microbiologie.

Finalement, il y a des changements majeurs à venir dans la pratique de la microbiologie clinique qui auront pour conséquence le besoin encore plus grand de gestion des collections de documents. Ces changements sont la robotisation du travail manuel de la microbiologie et les fusions à grande échelle des laboratoires du Québec. Les futurs robots des laboratoires devront être programmés et cette programmation devra être fidèle aux protocoles. La fusion exigera la comparaison des protocoles de plusieurs laboratoires pour en extraire une pratique commune.

### **3.1.2 Le contexte réglementaire**

L'encadrement réglementaire des hôpitaux exige l'obtention d'un agrément. Cet agrément porte sur de nombreux secteurs d'activité, par exemple sur la pharmacie, les soins infirmiers, mais aussi sur les services diagnostiques comme les laboratoires. Depuis le tournant du siècle, ce processus d'agrément est devenu plus quantitatif, l'agrément a exigé pour les laboratoires l'application graduelle de la norme ISO 15189. Mettre en vigueur cette norme a demandé un changement radical du mode de formation du personnel et de diffusion de l'information.

Auparavant dans les laboratoires, comme dans la majorité des champs d'activité de la médecine, la formation se faisait par un processus de compagnonnage. La nouvelle technicienne était simplement intégrée au travail quotidien en tant que novice avec peu d'autonomie au départ. Graduellement, au fil des mois, la technicienne devenait de plus en plus autonome et sa performance était jugée par ses formatrices de façon informelle. Il y avait certes des documents de référence, mais beaucoup de savoirs n'étaient transmis que de personne à personne.

Avec les méthodes modernes comme la norme ISO, le bagage de connaissances et d'habiletés nécessaires pour accomplir chaque tâche doit être défini de façon explicite. Par la suite, il faut s'assurer par des critères objectifs que chaque membre du personnel maîtrise les connaissances et les habiletés jugées nécessaires à son travail. Un tel système s'appuie donc sur une documentation détaillée et des processus pour mesurer la maîtrise de cette documentation.

### **3.1.3 L'évolution des activités des laboratoires**

En plus des changements réglementaires, les laboratoires doivent s'adapter à une évolution constante du volume des analyses demandées, des temps de réponse nécessaires et finalement dans la nature même des analyses. Ceci a eu pour conséquence que, malgré l'intégration continue de nouvelles technologies augmentant la productivité, le nombre d'employés du laboratoire SJ est passé d'environ 80 personnes en 2000 à environ 160 en 2018.

L'augmentation des volumes provient en partie de la démographie, mais il y a aussi des changements quant à la nature des analyses demandées. Un exemple de ces changements est la

création d'un nouveau pan complet d'activités en microbiologie pour s'attaquer aux bactéries multirésistantes ou encore hypervirulentes. Avant les années 80, ces bactéries étaient responsables de cas sporadiques ou d'éclotions limitées. Depuis, certaines sont devenues endémiques. Pour en contrôler la transmission dans les milieux de soins, on a recours à des dépistages individuels et de masse. Ce nouveau champ d'activité du laboratoire est très actif, de nouvelles méthodes et de nouveaux agents à dépister s'ajoutent régulièrement.

Un autre changement dans la nature des analyses demandées est l'arrivée des techniques rapides qui permettent d'accélérer la prise de décisions cliniques; ces processus accélérés sont indispensables pour faire face au nombre grandissant de patients à traiter. Des exemples de ces techniques nouvelles se retrouvent avec la détection des résistances aux antibiotiques. Les méthodes classiques *in vitro* pour prédire l'activité d'un antibiotique nécessitent l'étude de la croissance de la bactérie en présence de l'antibiotique pendant au moins seize heures. Il y a maintenant des méthodes plus rapides de détection de certaines résistances par recherche antigénique ou encore par TAAN. Ces techniques ne peuvent remplacer les méthodes classiques, en effet parfois les techniques rapides et traditionnelles donnent des résultats en contradiction les unes avec les autres. Les techniciennes doivent donc non seulement savoir comment effectuer ces nouvelles détections rapides, mais aussi en connaître les limites et comment résoudre les cas discordants.

Les secteurs d'activités autres que le laboratoire ont eux aussi évolué vers des normes plus modernes et ils ont aussi leurs besoins en avoirs informationnels. Dans le processus global d'une analyse des informations comme le domaine d'un test, le contenant nécessaire, les conditions de transport sont surtout utiles aux personnes travaillant en amont et en aval du laboratoire. Les laboratoires doivent donc produire de la documentation pour d'autres personnes que leur propre personnel et s'assurer qu'il y a des mécanismes de diffusion et de mesure de la maîtrise de cette documentation.

### **3.1.4 La gestion de l'information dans le défi global des laboratoires**

Comme le reste du système de santé, les laboratoires sont soumis à des contraintes d'effectifs et financières importantes, ces contraintes sont centrales aux problèmes de gestion de l'information.

En effet, même avec une informatisation minimale, on pourrait parvenir à une gestion adéquate avec des ressources humaines suffisamment nombreuses et stables.

La gestion des avoirs informationnels doit cependant compétitionner avec les autres besoins du laboratoire, besoins qui sont souvent d'une nécessité plus immédiate comme assurer suffisamment de personnel pour traiter les volumes d'analyses et entraîner le nouveau personnel. Ces contraintes ont empêché le laboratoire de Saint-Jérôme d'implanter tous les processus nécessaires et donc entraîné des déficiences.

#### Prélèvement :

Les selles doivent être recueillies sans contamination par l'urine ou par l'eau de cuvette. Elles ne doivent pas être émises suite à l'utilisation de laxatifs ou suppositoires ou les jours suivant un lavement baryté.

Un écouvillonnage rectal peut être fait, mais ce spécimen est inadéquat pour la recherche de toxine de *C. difficile*.

#### Transport :

Si le délai entre le prélèvement et l'arrivée au laboratoire est < 2 heures, on peut envoyer la selle dans un pot stérile. Si le délai est > 2 heures, il faut utiliser un milieu de transport Cary-Blair modifié et un pot stérile. Un délai maximum de 24 heures est accepté.

Pour la recherche de toxine de *Clostridium difficile*, envoyer la selle dans un pot stérile sans milieu de transport.

#### Réception :

Un maximum de 2 spécimens par patient à des jours différents est accepté. Les spécimens doivent parvenir avant le délai de transport prévu. Ne pas ensemer les selles dures ou contenant du baryum et les écouvillons si non dans un milieu de transport. Tout spécimen ne répondant pas à ces critères sera rejeté avec message explicatif au rapport.

Extrait de document 3-1, traitement du spécimen de selles selon le labo SJ

Ssource : SJ\_MIC-PON-072\_V7

Le passage surligné en vert explique le prélèvement, tandis que les passages surlignés en jaune indiquent les spécimens acceptables pour la culture.

## 3.2 Exemples de problèmes avec les objectifs

Rappelons que nos trois objectifs sont:

- Cohérence
- Exhaustivité
- Conformité aux normes

Les informations contenues dans les avoirs sont cohérentes s'il n'y a pas de contradiction entre elles. Elles sont exhaustives si toutes les informations nécessaires sont non seulement présentes,

- A. Media for routine stool culture** (see Table 3.8.1-1 for descriptions and abbreviations)
1. BAP
  2. MAC
  3. Choose one or more (to increase yield) of the following to select for *Salmonella* and *Shigella*.
    - a. HEK
    - b. XLD
    - c. CHROM agar *Salmonella*
    - d. SS (do not use as only selective medium)

Extrait de document 3-2, dispositifs de culture recommandés

Source : CMPH 4<sup>th</sup> edition (Leber, 2016)

Dans ce texte BAP (Blood Agar Plate) équivaut à la gélose-sang, MAC (MacConkey Agar) à Mc Conkey, HEK est Hektoen, les autres dispositifs XLD, CHROM agar et SS ne sont pas utilisés lors de l'ensemencement à Saint-Jérôme.

mais retrouvées là où on s'attend. Finalement, les informations sont conformes, si elles répondent aux normes en vigueur.

### **Exemple 1 Incohérences sur contenants acceptables pour spécimens**

Avec cet exemple, nous illustrons un problème de cohérence entre un protocole, une requête et le guide de prélèvement.

Les passages qui sont soulignés en jaune dans l'Extrait de document 3-1 nous montrent qu'il y a trois types de spécimens qui sont acceptables pour la culture: selles dans un pot avec milieu de transport, selles sans milieu de transport et finalement écouvillonnage rectal avec milieu de transport.

Maintenant, examinons l'Extrait de document 2-1 à la page 27, nous voyons par les pictogrammes qu'un pot avec milieu de transport est le seul choix de contenant mentionné. Finalement, l'Extrait de document 2-2 montre que le répertoire d'analyse ne fait mention que d'un pot stérile sans mention de milieu de transport. Finalement, ni la requête ni le répertoire ne font mention de l'écouvillon.

Milieus :

- Gélose-sang
- Mc Conkey
- Hektoen
- Gélose Campy CSM
- McConkey-Sorbitol avec cefixime et tellurite
- CIN

Extrait de document 3-3, dispositifs de culture selon le labo SJ

Source : SJ\_MIC-PON-072\_V7

Le SIL lui n'offre, lors de la saisie d'une demande de culture de selle, que le contenant avec milieu de transport Cary-Blair. Ceci n'est pas vraiment une omission, car le SIL n'admet tout simplement pas plus d'un choix de contenant.

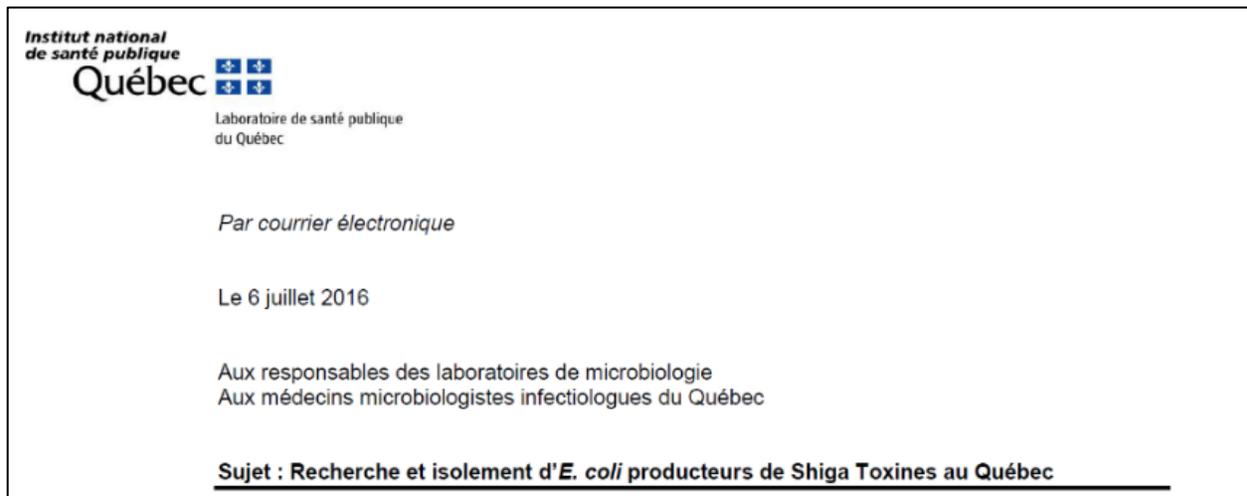
## **Exemple 2 Problèmes avec la vérification de la conformité**

Dans cet exemple, nous allons illustrer des difficultés avec la conformité. Dans les laboratoires, une source d'information très importante est le Clinical Microbiology Procedures Handbook (CMPH) (Leber, 2016) par l'American Society for Microbiology (ASM). Les protocoles de cette source sont reconnus comme des standards et les microbiologistes doivent justifier l'éventuel choix de ne pas respecter ce standard.

L'Extrait de document 3-2, nous donne la liste des géloses jugées essentielles par le CMPH. L'Extrait de document 3-3, nous donne la liste selon le protocole de Saint-Jérôme. Ce dernier répond à l'exigence du CMPH avec gélose-sang, Mc Conkey et Hektoen. Mais il exige trois géloses qui ne sont pas présentes dans la liste du CMPH, soit CIN, Campy CSM et Mc Conkey sorbitol.

La gélose CIN fait partie des géloses jugées facultatives dans le CMPH et son ajout a été choisi par les microbiologistes.

La gélose Campy CSM est dédiée à la recherche des *Campylobacter*, qui nécessite des conditions de culture bien particulières. Le protocole de référence du CMPH ne liste pas de dispositif dédié au *Campylobacter*, car sa recherche est décrite dans un protocole du CMPH distinct. Avant de vérifier la conformité, il est donc essentiel de s'assurer que les organismes pathogènes recherchés sont les



Extrait de document 3-4, ajout d'une méthode TAAN pour la recherche des STEC

Source : Lettre du LSPQ sur le STEC

mêmes. Le domaine d'un test est l'entité qui décrit la gamme de pathogènes qu'un test est à même de détecter.

Finalement, notons, bien que ce ne soit pas une erreur, que le dispositif MC Conkey-Sorbitol avec cefixime et tellurite n'est pas dans la liste du CMPH. Ce dispositif est utilisé pour la détection du *E coli* O157:H7<sup>9</sup> **un des types** d'*E. coli* associé à la "maladie du hamburger". L'ensemble des *E. coli* associé à cette maladie s'appellent Shiga Toxin producing *E. Coli* (STEC).

Le CMPH recommande de faire au minimum la détection des *E coli* O157:H7 dans certaines situations cliniques, le protocole de Saint-Jérôme qui demande de la faire sur toutes les selles est donc conforme. Cependant, nous verrons dans l'exemple suivant qu'à la suite d'une autre recommandation, la recherche a maintenant été étendue à **tous** les STEC dans certaines situations cliniques.

---

<sup>9</sup> Dans *E coli* O157:H7, O et H désigne des antigènes (des molécules capables d'induire la production d'anticorps), le chiffre 157 et 7 indique les sous-types d'antigènes O et H.

### **Exemple 3 Prise en charge incomplète d'un document normatif**

Nous avons vu dans l'exemple précédent que le protocole de Saint-Jérôme faisait la recherche du *E coli* O157:H7 sur toutes les selles, mais non de tous STEC. La situation était similaire à la Cité de la Santé. Devant la gravité des infections à STEC chez les enfants et la nécessité pour la santé publique d'intervenir en cas d'éclosion, le Laboratoire de Santé Publique du Québec (LSPQ)<sup>10</sup> a émis une directive sur la recherche des STEC dont nous voyons l'en-tête à l'Extrait de document 3-4.

Cet avis stipule que les laboratoires doivent modifier leur protocole pour inclure la détection de tous les STEC dans certains cas visés. Les cas visés sont : les cultures de selles pour les patients âgés cinq ans ou moins, les cas où un spécimen sanguinolent est reçu et finalement les cas pour qui le soignant suspecte une "maladie du hamburger".

Le LSPQ a désigné des laboratoires pour offrir une méthode de détection des gènes de production de la toxine par TAAN, le LSPQ demande que pour les cas visés on fasse parvenir à un des centres désignés une partie du spécimen de selles après un prétraitement.

Cette modification a fait l'objet d'un nouveau protocole de traitement de spécimens à Saint-Jérôme soit le SJ\_MIC-PON-197 Recherche E. coli STEC, mais aucune mention de ceci ne se retrouve dans le protocole SJ\_MIC-PON-072\_V7.

De plus dans le répertoire d'analyse sur le web, il n'est fait mention que des O157:H7. La possibilité de demander une recherche de tous les STEC, s'il y a suspicion clinique, n'est pas mentionnée, ni que la recherche sera automatiquement faite chez les 5 ans et moins.

La requête elle ne fait mention ni des O157:H7 ni des STEC, elle ne donne pas d'indications spéciales sur les selles sanguinolentes.

---

<sup>10</sup> Le LSPQ est une division de l'Institut National de Santé Publique du Québec, il s'agit d'un laboratoire de référence faisant certaines analyses et aussi d'un organisme normatif.

Pour tout spécimen provenant de patients hospitalisés depuis > 4 jours, ne faire que la recherche de Clostridium difficile. Pour toute demande spéciale, consulter le microbiologiste du laboratoire.

Extrait de document 3-5, remplacement de la culture de selles par la recherche de *C. difficile*  
Source : SJ\_MIC-PON-072\_V7

Le guide de prélèvement ne fait aucune mention d'une conduite spéciale en cas de selles sanguinolentes.

Finalement pour le SIL une règle a été implantée qui ajoute automatiquement une demande du test génétique pour les enfants. Si le SIL permet l'ajout du nouveau test pour tout patient, aucune mention d'une conduite spéciale en cas de selles sanguinolentes n'est faite.

Nous pouvons conclure de tout ceci que l'implantation de la modification demandée par le LSPQ a été incomplète.

#### **Exemple 4 Incohérence entre le SIL et un protocole**

Selon l'Extrait de document 3-5 provenant du protocole de selles, une demande de culture de selles standard ne devrait plus être faite à partir de la 5<sup>e</sup> journée d'hospitalisation. Cependant, à l'Extrait de document 2-4, nous avons montré le message affiché par le SIL si on tente de saisir une demande de culture de selles standard après la troisième journée d'hospitalisation, donc à partir de la 4<sup>e</sup> journée.

#### **Exemple 5 Omission dans le SIL de la possibilité de demander un test**

À l'Extrait de document 3-7 nous voyons que, selon le protocole Saint-Jérôme, un test Gram<sup>11</sup> peut être demandé sur les selles reçues sans milieu de transport.

Les requêtes ne font pas mention de la possibilité de demander un Gram sur les selles. Les requêtes étant un assortiment de tests choisis, il pourrait s'agir d'une décision plutôt que d'une erreur.

---

<sup>11</sup> Strictement parlant le Gram est une coloration mise au point par Hans Christian Gram, par extension on désigne du nom de la coloration l'ensemble composé d'une lame de microscope et d'un étalement de substance qui a été coloré. Le type de coloration est choisi en fonction des caractéristiques que l'on veut mettre en évidence.

Ensemencement :

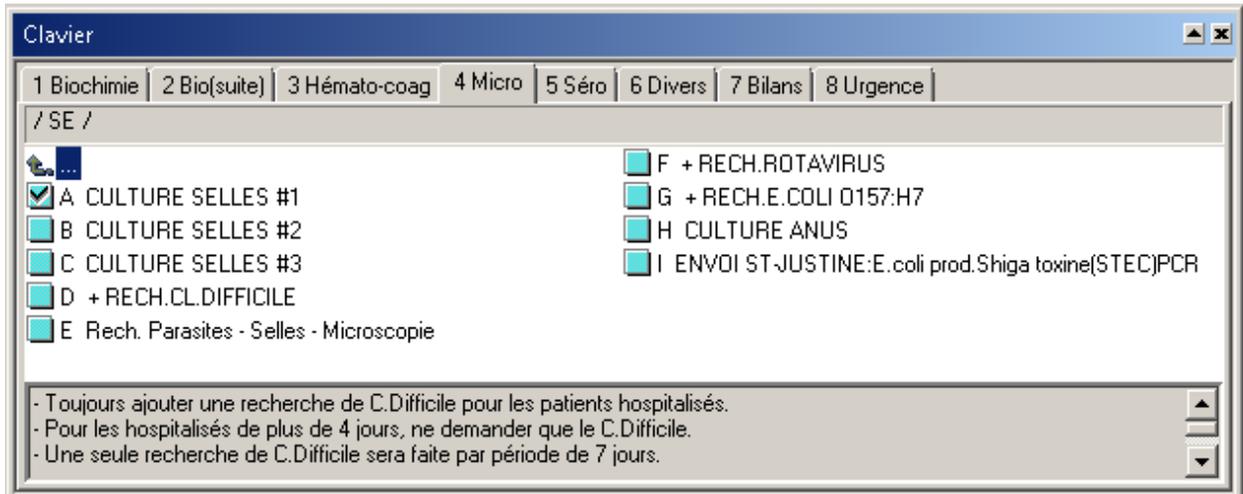
a) Routine :

Examen direct: à faire sur demande sur selles fraîches (< 2 heures) et diarrhéiques seulement.

Faire une coloration de Gram. Rechercher la présence de polynucléaires. Rappporter “ présence ou absence de polynucléaires”.

Extrait de document 3-7, test de Grem sur les selles selon le labo SJ

Source : SJ\_MIC-PON-072\_V7



Extrait de document 3-7, menu de saisie pour la culture de selles selon le labo SJ

Source :SoftLab, laboratoire SJ

Cependant, le répertoire d’analyse, lui, doit monter tous les tests disponibles. Pourtant le répertoire ne fait aucune mention du test de Gram.

Si l’utilisateur utilise les menus du SIL pour faire la saisie d’une culture de selles, alors après avoir choisi l’option « selles » dans un premier menu, le SIL affichera à l’utilisateur le menu illustré à l’Extrait de document 3-7. Nous pouvons voir que dans ce menu il n’est pas fait mention d’un test Gram.

### 3.2.1 Conclusion sur ces exemples

Nous avons facilement retrouvé ces erreurs en partant de notre exemple de culture selles et en vérifiant les avois informationnels de base pour cette culture. Il ne s’agit que d’illustrations pour montrer que des problèmes avec les trois objectifs surviennent dans la vie quotidienne au laboratoire.

### **3. POLITIQUES/PROCÉDURES ASSOCIÉES**

MIC-PON-109 Identification des Campylobacter  
MIC-PON-165 Milieu Hecktoen  
MIC-PON-029 Milieu XLD  
MIC-PON-051 TSI  
MIC-PON-189 C.Diff Quik Chek Complete

Extrait de document 3-8, liste des protocoles associés la culture de selles selon le labo SJ  
Source :SJ MIC-PON-072 V7

Notre but est de comprendre les causes des problèmes et erreurs et de montrer comment notre solution entend y remédier. Dans la section suivante, nous allons voir comment se fait la gestion des avoirs informationnels au laboratoire SJ, puis nous analyserons les mécanismes d'apparition de ces erreurs et problèmes.

## **3.3 La gestion de l'information en place à Saint-Jérôme**

### **3.3.1 Les protocoles**

#### **3.3.1.1 Classement, versionnage, approbation**

Les protocoles produits à l'interne sont développés, maintenus à jour et diffusés exclusivement sous forme électronique avec l'aide d'un logiciel de gestion documentaire maison. Tous les documents y figurent avec leur historique de changements et leurs approbations par signature électronique.

Les protocoles sous responsabilité d'organismes externes, comme le LSPQ, ne sont pas pris en charge par le logiciel maison. Ces documents sont conservés sous forme électronique, mais dans une arborescence de fichiers propre à chaque gestionnaire de terrain sans l'assistance du logiciel de gestion documentaire maison.

Les documents répertoriés par le logiciel maison sont classés selon la sous-section du laboratoire responsable, comme l'indiquent les trois premières lettres du nom de document. Chaque affectation de tâches dans le laboratoire s'est vu associer une liste de documents. Par exemple, on a regroupé les documents qu'une technicienne doit connaître pour effectuer les **ensemencements** : la tâche

Identification :

1) Gélose-sang :

Lire la gélose à 24 heures.

Noter sur la présence ou l'absence de la flore fécale.

Pour le Staph.aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure.

Extrait de document 3-9, lecture du dispositif gélose-sang selon le labo SJ  
Source SJ MIC-PON-072 V7

d'inoculer les dispositifs de culture avec une portion (un inoculum) de la substance biologique. Chaque technicienne a donc une liste de documents à maîtriser selon ses affectations de tâches et elle doit attester de ses lectures par signature électronique.

Cette classification par groupe de tâches est déjà une forme rudimentaire de classement de documents selon les informations qu'ils contiennent. La liste des documents que doit maîtriser une technicienne est également une ébauche de la notion de public cible pour une information.

### **3.3.1.2 Lien de dépendance entre documents**

Une section « POLITIQUES / PROCÉDURES ASSOCIÉES » est présente dans tous les protocoles internes autant du laboratoire SJ que CSL. Cette section liste les autres protocoles dont la maîtrise est nécessaire pour bien effectuer la tâche. Dans cette liste, se trouvent souvent des protocoles pour des procédures outils qui sont utilisées par plusieurs protocoles. Cette liste est bâtie sur la base d'une modalité « a besoin de ». Dans l'Extrait de document 3-8, nous avons cette liste pour le protocole de selles du laboratoire SJ.

Il n'y a pas de directives claires quant aux critères pour qu'une dépendance ait à être listée. Certaines dépendances sont absentes de la liste sans que nécessairement leur importance soit moindre. Par exemple, à l'Extrait de document 3-9, il est spécifié que si la bactérie *Staphylococcus aureus* (*S. aureus*) est retrouvée sur la gélose-sang alors il faut vérifier si la bactérie est résistante à l'antibiotique oxacilline. Il y a une procédure pour faire cette détection de résistance, elle est décrite par le protocole SJ\_MIC-PON-099\_V2. Pourtant nous ne retrouvons pas SJ\_MIC-PON-099\_V2 dans la liste des dépendances de l'Extrait de document 3-8.

Comme les listes de dépendances sont sous forme texte, nous ne pouvons pas renverser le sens de la modalité et pour un document faire une liste des autres documents qui dépendent de lui. Par exemple faire la liste de tous les protocoles qui utilisent SJ\_MIC-PON-099\_V2 Dépistage du SAMR oxascreen.

Cette possibilité est très importante pour maintenir la cohérence. Prenons l'exemple de l'épreuve TSI, qui est dans la liste de dépendances de la culture de selles. Cette épreuve est décrite par le protocole SJ\_MIC-PON-05\_TSI\_V1. Ce test TSI est une épreuve d'identification dont le temps d'incubation est crucial, une modification a été faite au protocole de TSI qui mentionne dans quelles situations l'incubation doit être prolongée. Il serait fort utile de pouvoir recenser les procédures qui utilisent cette épreuve de TSI afin de vérifier si elles ne sont pas en contradiction avec ce besoin d'incubation prolongée.

### **3.3.2 Les requêtes**

Les requêtes sont des documents Word qui sont modifiés au fil du temps. Elles ne sont pas créées à partir d'une base de données et donc, lors d'une mise à jour, chaque requête doit être modifiée individuellement. Elles sont sous la responsabilité d'une équipe du laboratoire qui se consacre au soutien des utilisateurs externes.

### **3.3.3 Le répertoire d'analyse et le guide de prélèvement**

Le répertoire d'analyse et le guide de prélèvement ont évolué à partir de documents qui étaient distribués sous forme papier. Des sites web ont été construits lorsqu'il était devenu trop laborieux de s'assurer que tous les utilisateurs du laboratoire avaient en main les bonnes versions des volumineux documents papier.

La maintenance de ces sites web n'est pas faite par le personnel du soutien aux utilisateurs externes, mais par les personnes du secteur informatique du laboratoire (en autres responsables du SIL). Ils sont basés sur des logiciels en code ouvert comme WordPress et MySQL. Il n'y a pas d'extraction automatique des informations des protocoles, le cas échéant les informations sont reproduites par copier-coller.

### **3.3.4 Le SIL**

Une équipe dédiée à la maintenance du SIL effectue les changements ou autorise un membre du personnel à les faire sous sa supervision. Ce contrôle des modifications apportées au SIL est très serré, car une modification intempestive pourrait avoir des conséquences graves. Une erreur de paramétrisation d'un test peut entraîner par exemple l'absence de l'émission des résultats.

Cette équipe est responsable de la maintenance du SIL, mais aussi de celle de toutes les interfaces avec le SIL comme celle avec le Dossier Santé Québec (DSQ).

## **3.4 Analyse des mécanismes qui ont produit les erreurs**

Avant de proposer notre solution aux problèmes de gestion avec nos trois objectifs, nous allons voir l'origine de ces erreurs et ce qui en a empêché la détection, et qui les a donc maintenues dans le temps.

### **3.4.1 Absence d'un mécanisme de révision systématique périodique.**

À l'Exemple 1 nous avons vu que la requête, le répertoire des analyses et le guide de prélèvements ne sont pas en accord quant aux spécimens et contenants nécessaires pour la culture de selles. La genèse de cette discordance est difficile à retracer, mais nous savons qu'elle s'est maintenue lors des mises à jour de la base de données pour le répertoire et des révisions successives des documents pour les requêtes et pour le guide de prélèvement.

Nous suggérons que cela est la conséquence de l'absence d'un mécanisme de révision périodique de l'exactitude des informations des requêtes, du répertoire et du guide avec les protocoles. Un tel mécanisme serait très lourd avec les instruments en place.

En effet supposons que l'on veuille faire une révision en profondeur des requêtes, on devrait alors examiner chaque requête, aller voir les protocoles pour chacun des tests listés sur la requête, retrouver les informations dans les protocoles et les comparer. Ceci sera laborieux parce qu'il y a plusieurs tests sur les requêtes. Pour notre exemple de requête, il y en a une centaine, donc autant de protocoles à vérifier. De plus, l'information peut être difficile à retrouver dans les protocoles,

comme nous l'avons vu avec l'Exemple 1 où a retrouvé l'exigence d'avoir un milieu de transport pour l'écouvillon dans la section ensemencement. Finalement, le rendement d'une telle vérification serait faible, car les changements de procédure de laboratoire qui affectent le prélèvement sont relativement rares.

En pratique, on a plutôt établi le processus inverse où la révision des requêtes n'est enclenchée que lorsqu'une modification affectant les requêtes est faite à un protocole. Les omissions que nous avons vues montrent que ce processus est faillible. La raison principale est que lors des révisions successives du protocole des selles, il n'y a pas eu changement de la méthode de prélèvement. Donc la modification des autres documents, comme la requête et le répertoire d'analyse n'ont jamais été enclenchés et ainsi les erreurs initiales n'ont jamais été corrigées.

Donc, au minimum à chaque révision d'un protocole, il faut que les documents qui en dépendent soient révisés, cela nécessite l'implication d'équipes de différents secteurs du laboratoire et l'usage d'une table de dépendance fiable. Dans une situation idéale, au moment de la révision, **toutes** les informations de **tous** les documents seraient disponibles au réviseur sans qu'il ait à faire de recherches.

### **3.4.2 Vision trop étroite de l'impact d'un changement**

La modification de protocole de selles dictée par le LSPQ (voir l'Exemple 3 ) soulève des questions. L'implantation de la recherche des STEC n'a pas été complète bien qu'on ait fait des modifications du SIL. Nous pouvons nous demander pourquoi seulement une partie des changements nécessaires a été faite.

Cela provient du fait que la procédure qui doit être appliquée lors de la modification d'un test n'est pas exhaustive. Ces changements ne sont pas faits selon une procédure formellement établie, la marche à suivre n'est transmise que de personne à personne. Les changements qui ont été faits étaient ceux de nécessité immédiate. Par exemple sans la modification du SIL il aurait fallu pour chaque demande de culture de selles faire une vérification manuelle du critère d'âge. Ceci est très lourd et donc la nécessité de modifier le SIL pour faire la vérification automatique de l'âge s'est imposée d'elle-même. Également, il fallait un protocole pour la nouvelle procédure de recherche

de STEC parce que, sans protocole, on ne peut pas former les techniciennes. Les autres changements, comme la modification du guide de prélèvement, n'empêchent pas de débiter les envois au centre de référence.

Nous avons également noté l'absence d'un mécanisme pour nous assurer que la recherche de STEC est faite sur les selles sanglantes. Des décisions n'ont pas été prises quant à savoir quelles personnes sont désignées responsables de l'examen de la selle et de la décision d'ajouter le test STEC. Est-ce que la responsabilité incombe au prescripteur ? Si le prescripteur n'est pas au courant de la nature hémorragique de la diarrhée et que la recherche de STEC n'est pas prescrite, doit-il y avoir un mécanisme de rattrapage accompli par les personnes qui manipuleront le spécimen ?

Pour gérer ce risque d'omission, il faut voir le processus d'une analyse dans son ensemble, allant de l'identification par le soignant du besoin d'une analyse de laboratoire, jusqu'à l'interprétation du résultat par le soignant comme nous l'avons vu à la section 2.1. Notons que notre processus commence par une étape de décision qui a lieu **avant** la prescription et se termine avec une autre étape d'interprétation qui a lieu **après** que le clinicien a été mis au courant du rapport. Une fois que le processus est bien décrit dans son ensemble, on peut assigner des responsabilités aux personnes qui sont impliquées à chacune des étapes.

Chaque changement à un protocole doit être revu pour son impact sur chacune des étapes de ce processus d'analyse étendu et en fonction des besoins d'information de chacun des acteurs.

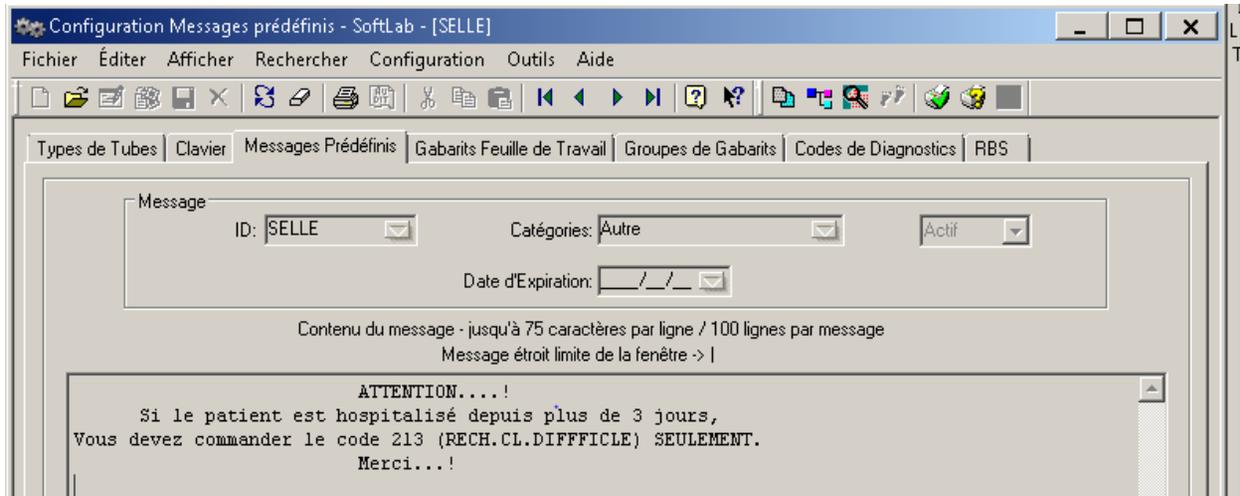
### **3.4.3 Manque d'accès aux informations**

À l'Exemple 4 nous avons noté une discordance entre une fenêtre surgissante du SIL et le protocole pour la recherche de *C difficile*, comme avec les discordances vues ci-haut, c'est la lourdeur du processus de révision qui est l'enjeu principal.

Avec le SIL, retrouver et corriger les discordances est non seulement laborieux et répétitif, mais c'est aussi complexe. Les changements dans la paramétrisation du SIL sont sous la responsabilité d'une équipe dédiée, mais la consultation des tables de paramètres est disponible à tous. En principe donc, n'importe quelle technicienne pourrait retrouver dans les tables les règles qui gouvernent les

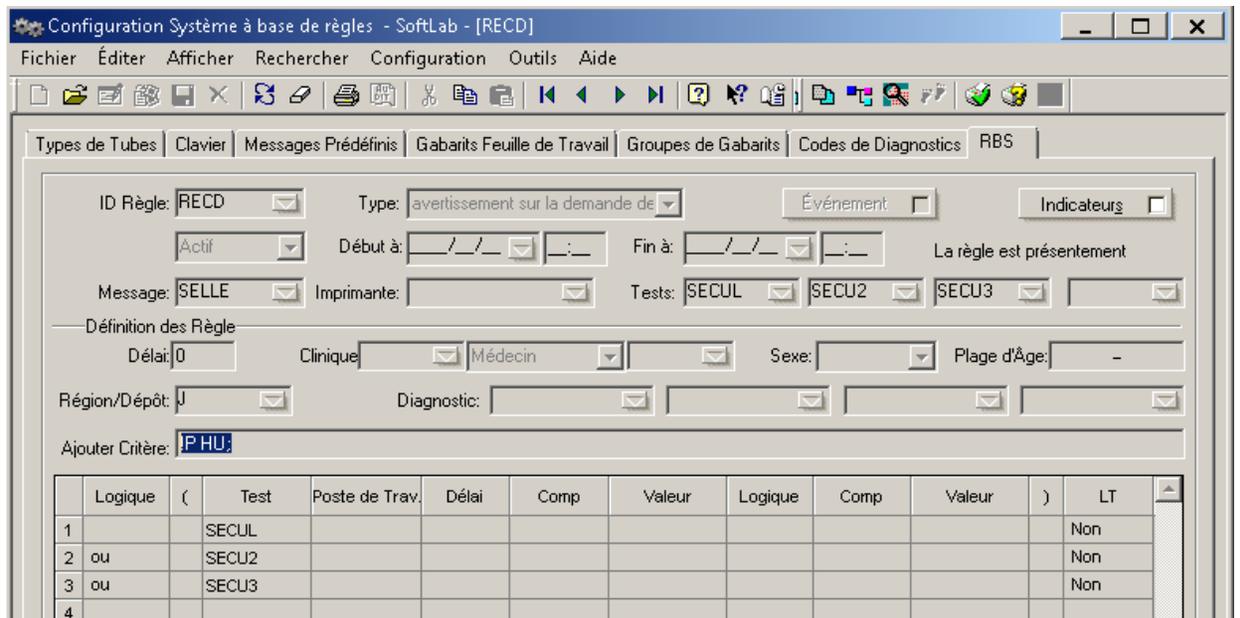
fenêtres surgissantes et vérifier leur exactitude. Cependant, retrouver ces règles et ensuite les comprendre est une tâche qui est loin d'être à la portée d'un utilisateur, même avec une formation. En effet, certaines recherches ne peuvent être faites efficacement qu'en laissant de côté l'interface du SIL, pour utiliser du code SQL directement sur les tables de paramètres. Même une fois la règle retrouvée, en comprendre le libellé n'est pas simple.

Illustrons ceci avec la règle qui gouverne le message sur le *C difficile*. Avec SoftLab, il y a deux éléments pour une fenêtre surgissante, le libellé du message et la règle qui déclenche l'affichage du message. La recherche de la règle n'est pas directement possible avec l'interface, il faut trouver en premier le message et avec l'identificateur du message on retrouve la règle.



Extrait de document 3-11, écran SoftLab pour message sur recherche de *C difficile*

Source : apture d'écran de SoftLab Saint-Jérôme



Extrait de document 3-12, règle SoftLab pour remplacement par une recherche de *C difficile*

Source : capture d'écran de SoftLab Saint-Jérôme

L'interface SoftLab n'offre pas d'instrument de recherche parmi les messages, nous avons retrouvé le message sur le C difficile par une requête SQL parmi les 3886 messages, son identificateur est simplement « SELLE ».

Ensuite, il faut rechercher la règle qui déclenche l'affichage du message. À cette étape, l'interface du SIL est plus conviviale et permet la recherche de la règle en fonction du message qu'elle

déclenche. Une fois la règle retrouvée, il faut en comprendre le fonctionnement ; dans ce cas-ci, il faut savoir que les caractères « !P HU », qui sont surlignés en bleu dans la case « Ajouter Critères » sont des codes pour un critère de déclenchement. « !P » signale un critère basé sur les patients et « HU » stipule quels sont les patients ciblés, « HU » veut dire Hospitalisé ou Urgence.

Avec les documents autres que le SIL, il est possible, bien que ce soit très lourd, de gérer les objectifs par des méthodes manuelles avec des listes de dépendances. Avec le SIL cependant ce n'est pas envisageable. Au minimum, il faut un catalogue pour classer les informations du SIL. Par exemple, il serait relativement facile de classer les textes et les règles des fenêtres surgissantes en fonction des analyses auxquelles elles s'adressent. Dans notre exemple, nous pourrions relier le texte de la fenêtre avec des entrées de catalogue « culture de selles » et « recherche *C. difficile* ».

### **3.4.4 Risque de perpétuation des erreurs selon les personnes ciblées par l'information.**

Nous avons vu que la vérification des documents qui découlent des protocoles est difficile, mais ce n'est pas seul facteur qui fait que les erreurs perdurent. Les erreurs peuvent se perpétuer parce que la personne à qui sont destinées les informations n'est pas à même de les détecter, c'est le cas d'ailleurs de toutes les erreurs que nous avons données en exemple.

Supposons qu'une erreur de paramétrisation du SIL fasse que le SIL exige une gélose XLD au lieu de CIN, n'importe quelle technicienne qui a lu le protocole sera à même de la déceler. Toutefois pour l'erreur sur le délai après hospitalisation et le *C difficile* ; les personnes qui saisissent les demandes d'analyse dans le SIL et qui verront la fenêtre font souvent partie du personnel administratif, qui ne sont pas des employés du laboratoire. Ces personnes ne connaissent pas les protocoles et ne sont donc pas à même de savoir si le délai prévu est de 72 ou 96 heures.

Ce risque de perpétuation lié à la personne cible de l'information est intrinsèque à l'activité du laboratoire et doit être géré par des processus de gestion appropriés. Ces processus de gestion devront se baser sur la description du cheminement complet d'une analyse et sur les rôles des personnes dans ce cheminement.

### 3.4.5 Implantation nécessaire de la notion de domaine des cultures

Nous avons vu deux situations où l'absence d'une définition précise du domaine d'une culture est un problème. Dans l'Exemple 2 nous avons vu que la comparaison avec les protocoles de référence impliquait une prise en compte du domaine. À l'Exemple 3 nous avons vu que le médecin n'avait pas d'information sur la prise en charge ou non des STEC lors de la culture de selles.

### 3.4.6 Conclusion sur l'analyse des problèmes.

S'attaquer aux mécanismes à la base des erreurs que nous avons vues nécessite de produire des contenus, d'implanter une vision de la gestion des informations et finalement de développer des outils informatiques.

La production de nouveau contenu par les microbiologistes sera nécessaire afin de s'assurer que toutes les informations pertinentes au choix du test et à l'interprétation de son résultat soient disponibles. Nous avons vu avec nos exemples que la notion de **domaine** d'une analyse doit être développée.

En second lieu, il faut implanter un paradigme de gestion des avoirs basé sur le cheminement complet des analyses et sur les besoins des personnes le long de ce cheminement. Ce paradigme de gestion permettra d'identifier quelles sont les informations qui sont le plus susceptibles d'erreur et de non-détection.

Finalement, les outils informatiques doivent être capables de présenter pour un sujet tous les documents pertinents et les extractions pertinentes des tables de "set-up" du SIL. Pour assurer la qualité des informations qui sont contenues dans le SIL, il faut que celui-ci ne soit plus une boîte noire pour les professionnels. Les informations devraient être organisées et accessibles selon un schéma établi **par le laboratoire** et non **imposé** par l'application.

Avant d'explorer des approches nouvelles, résumons les éléments et tâches nécessaires à une gestion traditionnelle des documents

- Inventaire des avoirs

Pour s'attaquer aux objectifs dans des documents, il est essentiel d'en avoir un inventaire et de les classer de façon systématique, incluant les documents normatifs externes.

- Table de dépendances

Une table basée sur le principe “a besoin de”, dans cette table pour un protocole on retrouve la liste des autres protocoles dont la compréhension est nécessaire pour bien faire la tâche.

Une autre table sur le principe “informations reprises dans”, qui liste tous les documents qui reprennent les informations du protocole.

- Une permutation des tables de dépendances pour en faire une sur “nécessaire à” et une autre “reprend les informations de”
- Un format fixe pour les protocoles pour retrouver les informations rapidement

Mais il reste une zone grise avec la gestion traditionnelle. Comment gérer les documents comme le SIL ou les requêtes, qui ont de si nombreux liens avec les protocoles? En effet, comme toutes les analyses ou presque sont décrites dans le SIL, alors le document SIL sera dans la table “informations reprises dans” de tous les protocoles sans pointeurs sur les informations pertinentes.

Cette approche est insuffisante et nous en avons développé une autre que nous verrons dans ce qui suit. Il reste que cette approche traditionnelle est un point de départ nécessaire.

## **3.5 Historique des approches tentées à Saint-Jérôme**

### **3.5.1 Le système de gestion documentaire en place**

Le logiciel maison de gestion documentaire en place depuis 2009 a été créé par l'auteur de ce mémoire afin de satisfaire à une des exigences de la norme ISO. En effet, cette norme exige que le laboratoire maintienne un registre des lectures de documents. Lorsqu'une employée signe le registre pour un document, elle affirme non seulement l'avoir lu, mais elle affirme également en maîtriser le contenu. Chaque employée a une liste de documents à maîtriser, liste établie en fonction des tâches qu'elle est appelée à accomplir, on dira que chaque employé a sa liste de lecture. Dès

les premiers moments de l'introduction de la norme ISO, il est apparu que gérer ce registre sans support informatique était impensable d'où la création du système maison.

Le système gère le calendrier de mise à jour, il permet d'assigner à des personnes la charge de créer ou de mettre à jour un document ; il permet également la saisie des jalons du processus d'approbation par signature électronique. Finalement, le système avise les employés de la disponibilité d'une nouvelle version si le document est dans leur liste de lecture.

### **3.5.2 Nouvelle version du système**

Une version améliorée du système maison était planifiée pour satisfaire à la liste de charges pour une gestion documentaire classique. Cette version aurait eu une liste de dépendances informatisées plutôt que simplement dans le texte. Cette nouvelle version aurait également implémenté une fusion entre la base de données du répertoire d'analyse et les protocoles. On aurait ajouté à chaque protocole traitant d'une analyse une section où apparaîtraient les lignes pertinentes extraites du répertoire d'analyse. Cette section, faite d'extraits, se voulait semblable aux infobox qui se retrouvent dans le coin supérieur droit de certaines pages de Wikipédia<sup>12</sup>. Les protocoles devenant ainsi une sorte d'assemblage.

Mais les textes sont déjà un assemblage avec la version actuelle du système. En effet, la première page du texte qui indique les jalons d'approbation est entièrement produite par les entrées de la base de données du système maison. Cette première page pour le protocole de selles se trouve à l'Extrait de document 3-13.

---

<sup>12</sup>Voir <https://fr.wikipedia.org/wiki/Aide:Infobox>

## MIC-PON-72-Culture de selles

Version	Créé/Révisé/Approuvé par	Date d'entrée en vigueur
1	LUC BAILEY/BENOIT FILIATRAULT/	2006/10/05
2	BENOIT FILIATRAULT//	2007/10/09
3	BENOIT FILIATRAULT/BENOIT FILIATRAULT/LUC BAILEY	2008/02/09
	BENOIT FILIATRAULT/BENOIT FILIATRAULT/LUC BAILEY	2010/01/21
4	LUC BAILEY/BENOIT FILIATRAULT/JEAN-GUY LEVREULT	2010/11/30
	LUC BAILEY/BENOIT FILIATRAULT/JEAN-GUY LEVREULT	2013/02/05
5	JEAN-GUY LEVREULT/BENOIT FILIATRAULT/JEAN-GUY LEVREULT	2015/04/29
6	SEBASTIEN GOURD/BENOIT FILIATRAULT/LUC BAILEY	2015/11/13
7	BENOIT FILIATRAULT/BENOIT FILIATRAULT/LUC BAILEY	2016/05/02

Extrait de document 3-13, liste des approbations pour le protocole de selles du labo SJ

Source : SJ\_MIC-PON-072\_V7

De même, la dernière page du protocole qui donne un résumé des changements par numéro de version est également produite à partir de la base de données. Nous en avons un exemple à l'Extrait de document 3-14.

Lorsqu'un changement est fait à un protocole, le système exige de la personne qui a fait le changement une signature électronique et la saisie d'un texte qui résume le changement. Si le changement est assez important, il y a création d'une nouvelle version, cette nouvelle version est ajoutée à la liste de lectures des techniciennes concernées. Les signatures et les résumés de changements sont conservés dans une base de données. Une fois le processus de changement terminé, un nouvel assemblage est effectué et le document est produit avec une première page comportant une nouvelle entrée pour les dernières signatures et également une nouvelle entrée dans la table sur la dernière page avec le résumé des changements.

## Historique des modifications

Version	Date	Modification
1		
2		
3	2010/01/21	SEBASTIEN GOURD - Correction de la mise en forme.
	2010/01/21	SEBASTIEN GOURD - Reconduit sans modification.
4	2010/11/30	JEAN-GUY LEVREAU - Ajout du milieu CIN
	2010/11/30	JEAN-GUY LEVREAU - Modification de la gélose pour E Coli O157 pour un milieu sélectif
	2010/11/30	JEAN-GUY LEVREAU - Précision sur le mode de lecture des géloses
	2010/11/30	JEAN-GUY LEVREAU - Précision sur les modalités de recherche des Aeromonas, Plesiomonas et Vibrio
	2010/11/30	JEAN-GUY LEVREAU - Précision sur les tests "en tube" d'élimination avant de faire l'identification complète
	2010/11/30	JEAN-GUY LEVREAU - Précisions sur délais de transport et condition d'acceptation
	2013/02/05	BENOIT FILIATRAULT - Reconduit sans modification par FILB0.
5	2014/12/12	JEAN-GUY LEVREAU - Commentaire sur la non sortie des Yersinia kristensenii
	2015/02/04	LUC BAILEY - Retrait de recherche Staph. aureus, Pseudo et levures
	2015/03/05	BENOIT FILIATRAULT - Correction de la mise en forme.
6	2015/10/21	SEBASTIEN GOURD - Ajout d'un algorithme décisionnel pour dépister le C.difficile.
	2015/10/21	SEBASTIEN GOURD - Modification au texte sur le C.difficile
	2015/11/12	BENOIT FILIATRAULT - Ajout des heures et de l'algorithme des C.difficile
	2016/01/19	BENOIT FILIATRAULT - Correction de fautes de frappe.
7	2016/04/13	BENOIT FILIATRAULT - Ajout faire antibiogramme sur les campylobacter.
	2016/05/02	LUC BAILEY - Correction de fautes de frappe.

Extrait de document 3-14, liste de changements au protocole de selles du labo SJ

Source : SJ\_MIC-PON-072\_V7

L'ajout d'une section avec les informations pertinentes en provenance du répertoire d'analyse ne pourrait qu'être une nouvelle application de cette construction de documents par assemblage. Mais une complication apparaît rapidement. En effet, cette section avec les informations en provenance du répertoire devrait-elle être dynamique et refléter les derniers changements au répertoire ? Si oui, il est alors possible que la section du protocole avec les importations ne présente pas les mêmes informations que celles qui étaient présentes au moment de l'approbation.

Donc il y a une certaine opposition entre les documents vus comme des entités unitaires et statiques gérées par des approches de gestion documentaire classiques et les informations contenues dans ces documents qui pourraient être dynamiques et provenir d'une base de données.

Des contraintes financières ont retardé le développement d'une seconde version du système maison. Puis le regroupement sous une même direction de plusieurs laboratoires avec la création des CISSS en 2015 a mis un terme au projet. La version initiale du système demeure à ce jour encore en production.

### **3.5.3 Expérimentation avec les wikis**

Cette seconde version du logiciel maison aurait pu pallier les difficultés à maintenir cohérents les protocoles et le répertoire d'analyse par l'intégration dans les protocoles d'éléments provenant d'une base de données sur les analyses. Mais cela ne réglait pas le problème pour les autres documents secondaires comme les requêtes et le guide de prélèvement.

Le fait que ce problème de cohérence n'était pas entièrement résolu avec la seconde version planifiée du système maison nous a poussés à explorer d'autres solutions. Nous pourrions aller plus loin avec le principe de création de documents par assemblage et voir les protocoles comme un assemblage de sous-sections, sous-sections qui pourraient être réutilisées dans les documents secondaires. Bien sûr, ceci sous-entend un format uniforme et très précis pour les protocoles, à la section 3.4.6, nous avons déjà mis à jour ce besoin pour une gestion traditionnelle efficace.

Nous avons alors le choix de tenter d'ajouter cette fonctionnalité à notre logiciel maison ou d'explorer si des logiciels existants pouvaient être utilisés. Mais avant de voir le logiciel qui a été retenu pour nos travaux exploratoires, examinons un second problème qui est apparu au cours de l'utilisation du logiciel maison.

Ce problème est qu'il est impossible pour le personnel d'ajouter des informations aux protocoles sans passer par le processus d'approbation. En conséquence de ce problème, la pratique d'imprimer les protocoles et d'y ajouter des informations au stylo s'est répandue. Nous voulions que les protocoles deviennent les instruments du travail au quotidien et pour cela il fallait qu'on puisse y

faire des ajouts ; par exemple, le nom et le numéro de téléphone d'une personne-ressource dans un laboratoire de référence.

Nous cherchions un modèle qui permettrait ces ajouts sans pour autant compromettre le caractère officiel des protocoles. Le modèle par wiki permet exactement ce genre de modifications « démocratisées » sans pour autant perdre les jalons officiels, de plus certains moteurs de wiki permettent la création de documents composites. Cette approche par wiki a donc été explorée.

### 3.5.3.1 Transclusion avec MediaWiki

MediaWiki, le logiciel de Wikipédia est robuste, disponible gratuitement et fort bien documenté. En plus de ses capacités très approfondies pour encadrer la démocratisation des changements aux documents, MediaWiki prend en charge la transclusion. La transclusion est l'inclusion dynamique dans un document de portions de textes qui proviennent d'un autre document, ce qui est exactement ce dont nous avons besoin pour notre système.

L'approche explorée consistait à rebâtir les protocoles dans une instance de MediaWiki avec un format modulaire préétabli, puis à extraire par transclusion des portions de textes pour bâtir d'autres documents. Par exemple, on pourrait bâtir le guide de prélèvements avec l'extraction de chaque section prélèvements des protocoles.

Il faut une prise en charge des versions des informations ce que MediaWiki permet de gérer. En effet supposons que notre guide de prélèvements ait besoin d'une approbation, ce guide est un document composite, car son contenu est en partie composé d'extractions, mais contient aussi du texte original. Alors, si le guide de prélèvement est approuvé avec sa section sur la culture de selles faites par une extraction de la section pertinente du protocole de selles **version 7**. Advenant une modification du protocole de selles en **version 8**. MediaWiki permet de choisir de présenter le document composite avec l'inclusion de la version du texte en vigueur au moment de l'approbation, soit en version 7 ou encore présenter le document composite modifié avec la nouvelle version 8 du texte inclus. MediaWiki permet même de comparer côte à côte les versions pour voir les changements apportés.

Mais cette approche restait basée sur le texte et non sur les informations, elle était mal adaptée à des tâches comme la maintenance de requête ou encore l'extraction des données pour le répertoire d'analyse, car les informations sources devaient être sous forme de données non pas de texte.

### **3.5.3.2 Semantic MediaWiki**

Semantic MediaWiki est une extension de MediaWiki qui permet d'étendre la notion de transclusion en la rendant possible pour des données. Pour ce faire, il faut bien sûr un mécanisme pour représenter les données. Le formalisme que Semantic MediaWiki implante est celui des triplets : sujet – relation – objet.

Pour Semantic MediaWiki, le sujet de tous les triplets que l'on ajoute à un document est le document lui-même. Les documents se nomment « page » selon la nomenclature MediaWiki. Donc l'élément de gauche de tous les triplets définis sur une page est la page elle-même.

L'élément de droite, la cible de la propriété peut être plusieurs choses. Les cibles peuvent être une autre page, une chaîne de caractères, un nombre, un lien URL. Le choix de la cible peut être laissé libre ou encore encadré par une liste de choix.

L'élément du milieu dans le triplet, la propriété, définit la nature du lien entre le sujet et l'objet. Les propriétés sont prédéfinies par une paramétrisation du wiki.

Finalement, Semantic MediaWiki permet de faire des recherches ciblées parmi les triplets et il permet aussi la réutilisation de triplets dans d'autres pages.

### Transport :

Si le délai entre le prélèvement et l'arrivée au laboratoire est < 2 heures, on peut envoyer la selle dans un pot stérile. Si le délai est > 2 heures, il faut utiliser un milieu de transport Cary-Blair modifié et un pot stérile. Un délai maximum de 24 heures est accepté.

Extrait de document 3-15, norme sur le transport de selles selon le labo SJ

Source : SJ\_MIC-PON-072\_V7

#### 3.5.3.2.1 Exemple de notation avec Semantic MediaWiki

Supposons que nous voulions attacher des triplets sur le choix des contenants à l'extrait de document 3-15. Premièrement, il faut transformer le document SJ\_MIC-PON-072\_V7 en un page de Semantic MediaWiki. Pour ce faire, le document doit être réécrit avec le langage de balisage Wikitext qui est commun à MediaWiki et Semantic MediaWiki.

Par la suite sur la page on créera, toujours en Wikitext, les triplets et on les ancrera sur la page. La mécanique du balisage en Wikitext des triplets de leur ancrage est assez difficile à comprendre et inutile à présenter ici.

Voici les triplets, notons que la couleur du triplet souligne à quel texte d'ancrage le triplet de rattache.

SJ\_MIC-PON-072-V7                      *a\_contenant*                      pot vissé stérile

SJ\_MIC-PON-072-V7                      *a\_contenant*                      pot vissé Cary-Blair

Voici dans le texte original les points d'ancrage choisis

### Transport :

Si le délai entre le prélèvement et l'arrivée au laboratoire est < 2 heures, on peut envoyer la selle dans un pot stérile. Si le délai est > 2 heures, il faut utiliser un milieu de transport Cary-Blair modifié et un pot stérile. Un délai maximum de 24 heures est accepté.

Notons que les cibles des propriétés ne correspondent pas exactement aux textes d'ancrages, cela

permet de ne pas changer le texte et d'utiliser quand même des désignations standardisées. Cette fonctionnalité de Semantic MediaWiki qui permet d'ancrer un triplet sur un élément de textes sera très importante dans notre nouvelle approche de gestion de l'information.

Ceci est une avancée par rapport à la transclusion seule, d'autant plus que Semantic MediaWiki gère aussi la notion de version des triplets. Par exemple, si l'on modifie le texte SJ\_MIC-PON-072-V7 pour changer le type de contenant, Semantic MediaWiki permet que les informations sur le contenant extraites de SJ\_MIC-PON-072\_V7 et reproduites sur d'autres pages restent les mêmes. Semantic MediaWiki fixe par paramétrisation à quel moment une modification d'une information sur sa page d'origine se trouve reflétée sur les pages de destination.

#### 3.5.3.2.2 Difficultés avec les données complexes

Semantic MediaWiki impose toujours le même sujet dans les relations, le même élément de gauche. Il est ainsi plus difficile de représenter certaines associations de données. Par exemple si l'on veut assigner à « pot vissé stérile » le délai de transport de 2 heures et celui de 24 heures à « pot vissé Cary-Blair », il faut recourir au type de données vecteur, car le côté gauche de l'affectation de propriété est toujours SJ\_MIC-PON-72\_V7, il faut également changer la propriété pour *a contenant délai*

MIC-PON-72-Culture de selles	<i>a contenant délai</i>	pot vissé Cary-Blair ; 24 hres
MIC-PON-72-Culture de selles	<i>a contenant délai</i>	pot vissé stérile ; 2 hres

On pourrait avoir besoin de vecteurs à plusieurs dimensions pour ajouter d'autres données comme la température nécessaire pendant le transport ou la nécessité de protéger de la lumière. Semantic MediaWiki permet de gérer cette complexité sans passer par des vecteurs, mais avec une modélisation difficile à maîtriser.

#### 3.5.3.2.3 Organisation des propriétés

Avec MediaWiki les documents sont organisés par catégories et le contenu des documents par entêtes de sections standardisés. Semantic MediaWiki permet ceci, mais avec en plus une arborescence pour les propriétés. Cette arborescence de propriétés est particulièrement nécessaire

avec Semantic MediaWiki, car la sémantique des triplets est uniquement portée par la propriété. Pour illustrer ce concept de hiérarchie de propriétés supposons l'existence d'une page dédiée à l'incubation pour la culture de selles : `incubation culture de selles`

<code>incubation_culture_de_selles.</code>	<code>a_durée_minimale</code>	18 heures
<code>incubation_culture_de_selles</code>	<code>a_durée_maximale</code>	24 heures
<code>incubation_culture_de_selles</code>	<code>a_durée_norme_scalaire</code>	24 heures

Il sera beaucoup plus facile de faire des interrogations si les trois propriétés : `a_durée_minimale`, `a_durée_maximale` et `a_durée_norme_scalaire` sont des sous-propriétés d'une propriété mère commune comme `a_norme_durée`.

### 3.5.3.3 Le recours au Wikitext : un problème avec l'approche par wiki

Nous avons vu que Semantic MediaWiki autant que MediaWiki utilisent le Wikitext pour la mise en page des textes. Le Wikitext est un dérivé de HTML, il se veut plus simple et plus robuste au piratage.

Bien que le Wikitext puisse sembler assez simple d'utilisation pour un utilisateur habile avec l'informatique, il en est tout autrement pour le personnel technique. Il est apparu rapidement irréaliste de faire une retranscription manuelle des protocoles en Wikitext. Nous avons alors testé des transformations automatiques, mais les codes produits se sont avérés fort rébarbatifs à lire et très difficiles à modifier.

À ce jour, bien que la fondation Wikipédia ait travaillé sur le sujet, il n'y a pas encore de remplacement simple pour le Wikitext.

### 3.5.4 Conclusion sur l'approche par wiki

L'exploration des solutions par wiki a eu le mérite de démontrer que, pour s'attaquer à nos trois objectifs, il nous fallait un vocabulaire standardisé et un système d'organisation des données.

NOM_ANALYSE ▲	CODE ▲	CODE2 ▲	REQ. ▲	SERVICE ▲	JEUNE ▲	SPÉCIMEN ▲	REMARQUES ▲
PARASITES, RECHERCHE (SELLES)(INCLUS RECHERCHE DE CRYPTOSPORIDIUM ET GIARDIA)	PARAS	282		MICRO		SELLES X 2	2 POTS DE SELLES, DANS MILIEU SAF. 2 SELLES SUR DEUX JOURS DIFFÉRENTS, DE PRÉFÉRENCE. ACHEMINER LES DEUX SELLES EN MÊME TEMPS AU LABORATOIRE.
SELLES : E. COLI O157:H7 (SI SELLES SANGUINOLENTES)	SE157	214		MICRO		SELLES	
SELLES EXAMEN DIRECT (RECHERCHE LEUCOCYTES SELLES)	EXDIR	234		MICRO		SELLES	SELLES LIQUIDES SEULEMENT
SELLES NO 1 CULTURE	SECUL	212		MICRO		POT STÉRILE	
SELLES NO 2 CULTURE	SECU2	800		MICRO		POT STÉRILE	
SELLES NO 3 CULTURE (SI NÉCESSAIRE)	SECU3	801		MICRO		POT STÉRILE	
SELLES-BK	SEBK	342		MICRO		CONTENANT SELLES	

Extrait de document 3-16, contenant nécessaires pour recueillir les selles selon le labo SJ  
Source : <http://www.cdsjlabo.org/codes-analyses>

Même pour la simple création de textes composites par transclusion, il faut que les protocoles sources soient séparés en sections préétablies : prélèvement, conservation avant transport, transport, réception, etc. Ce besoin d'organisation des données nous a amenés à décrire le concept de gestion informationnelle pour le contenu des documents, concept qui sera introduit dans la prochaine section.

L'exploration de la solution par wiki nous a également montré que la réécriture des protocoles nécessiterait des efforts considérables qui, dans un contexte de ressources limitées, est irréaliste.

Notre solution devra conserver les textes dans leur format actuel ou les transformer automatiquement par lots. Des réécritures pourront être envisagées, mais seulement si l'on peut procéder par étapes ; les processus de révision en place devront également être conservés. Il en découle que toutes les solutions envisageables passeront par une phase d'extraction de données dans des documents déjà en place.

### 3.6 Le choix d'une méthode d'organisation des données

Notre expérimentation avec Semantic MediaWiki nous a montré qu'il nous fallait un vocabulaire standard et un schème d'organisation des données.

Le modèle d'organisation le plus classique est modèle relationnel par base de données. C'est ce que nous avons envisagé pour la seconde version du logiciel maison que nous avons présenté à la section 3.5.2. Dans ce qui suit, nous allons expliquer pourquoi le modèle relationnel n'est pas une bonne solution. Ensuite, nous allons montrer pourquoi les ontologies sont un meilleur système d'organisation des données pour notre problème.

### **3.6.1 Le modèle relationnel**

Pour montrer les limites du modèle relationnel, prenons la base de données qui sous-tend le répertoire d'analyse et abordons sa transformation pour en faire le dépôt universel d'informations sur les analyses. Il y a, entre autres, des ajouts à faire pour décrire le choix du contenant et les modalités du transport.

Examinons ces ajouts pour la culture de selles. Nous pouvons voir à l'Extrait de document 3-16 que dans la colonne SPÉCIMEN, nous retrouvons «SELLES», «POT STÉRILE» et «CONTENANT SELLES». Donc en ce moment se confondent les notions de substances biologiques à prélever et de contenant pour la transporter.

Il faudra donc séparer cette colonne en deux champs CONTENANT et SUBSTANCE, ces champs devront être des choix dans des listes. En effet si nous laissons les champs en entrées libres alors nous pourrions retrouver le même contenant sur deux appellations comme dans l'extrait du répertoire avec «POT STÉRILE» et «CONTENANT SELLES». Il devra y avoir également la possibilité d'afficher plus d'une combinaison CONTENANT et SUBSTANCE par analyse.

Il faudra d'autres colonnes pour décrire les modalités de transport comme la durée maximale et la température à maintenir durant le transport.

### 3.6.1.1 Limitations du modèle de base de données relationnelle

Le modèle relationnel devra être assez élaboré, comme nous le montre notre exemple qui ne se limite qu'aux étapes du choix du contenant et du transport. Cependant, comme chaque méthode d'organisation devra être assez complexe, il ne s'agit pas d'une limitation spécifique au modèle relationnel.

Le véritable problème avec le modèle relationnel est qu'il n'offre pas assez de souplesse. Le contenu et les liens entre les tables doivent être soigneusement planifiés, car, une fois établis, ils sont difficiles à modifier. De même, les relations entre les champs d'une table doivent s'appliquer à chaque ligne intégralement. Dans ce qui suit, nous verrons un exemple de chacun de ces problèmes.

#### 3.6.1.1.1 Rigidité du contenu des tables et des liens entre elles

Avec la culture de selles, nous avons une illustration d'une difficulté du modèle relationnel qui impose l'utilisation d'une table de jointure.

Test	Cueillette	Substance	Contenant	Transport
Culture selles	Défécation	Selles	Pot simple	2 heures
Culture selles	Stomie	Selles	Pot simple	2 heures
Culture selles	Défécation	Selles	Pot avec milieu	2 heures
Culture selles	Stomie	Selles	Pot avec milieu	24 heures
Culture selles	Écouvillonnage rectal	Écouvillonnage rectal	Dispositif écouvillon	24 heures
Parasites selles	Défécation	Selles	Pot avec fixatif	Non spécifié
Parasites selles	Stomie	Selles	Pot avec fixatif	Non spécifié

Tableau 3-1, exemple de table de jointure pour les tests avec les spécimens et transports acceptés  
Trois cueillettes différentes (défécation, stomie, écouvillonnage anal), deux substances (selles, écouvillonnage anal), trois contenants de transport (contenant avec ou sans préservatif, dispositif écouvillon-tube) et finalement deux normes de transport différentes (en dedans de deux ou vingt-quatre heures). Certaines combinaisons ne sont pas compatibles comme défécation – selles – sans préservatif – vingt-quatre heures.

La méthode classique pour gérer ceci dans le modèle relationnel est de définir une table de jointure avec une ligne par combinaison acceptable et de joindre ces lignes à la table principale. Le Tableau 3-1 donne un exemple de table de jointure.

Mais cela suppose que, pour les autres éléments de la culture de selles, le traitement est équivalent, peu importe laquelle des cinq possibilités de combinaisons cueillette – substance – contenant a été choisie, ce qui n'est pas le cas. En effet, pour ensemer une selle, on trempe un écouvillon dans le spécimen puis on dépose un inoculum sur la gélose ; si nécessaire, l'écouvillon peut être rechargé en le trempant de nouveau dans le spécimen. Bien sûr, recharger est impossible si le spécimen est reçu sur un écouvillon. Pour tenir compte de cette possibilité, il faudra donc ajouter d'autres colonnes à notre table de jointure.

La rigidité du modèle relationnel fait que celui-ci doit être entièrement prédéterminé pour pouvoir prendre en charge tous les cas possibles. Dans notre exemple, il faut préciser d'avance toutes les colonnes qui seront dans la table de jointure.

Nous avons un exemple d'une solution inélégante à une situation non prévue au départ avec le choix des déterminants du travail avec le SIL SoftLab. Nous avons vu à la section 2.1.1 que le déterminant de la microbiologie est la combinaison d'un test et du descripteur du spécimen. Ce descripteur est composé de quatre facettes : la substance biologique, la localisation anatomique, la méthode cueillette et le contenant. Le SIL ne prévoit qu'une seule possibilité de contenant, déjà une limitation importante, il n'offre qu'une seule autre modalité nommée SOURCE pour désigner la localisation anatomique, la substance biologique et la méthode de cueillette.

Dans la liste des SOURCE de l'instance SoftLab de Saint-Jérôme, se retrouve des localisations anatomiques : OREILLE DROITE, des substances et méthodes de cueillette : URINE PONCTION VESICALE, des méthodes de cueillette seules : ASPIRATION À L'AIGUILLE (liquide). SOURCE est donc un amalgame hétéroclite des trois facettes du descripteur de spécimen, cet amalgame n'est pas une erreur dans la paramétrisation du SIL, mais un moyen de contourner les nombreuses limitations qui découlent du modèle déficient et rigide.

#### 3.6.1.1.2 Rigidité des relations entre les colonnes des tables

Avec le modèle relationnel, chaque ligne d'une table désigne des éléments qui sont en relation. Dans le répertoire d'analyse le fait que le NOM ANALYSE : SELLES NO 1 CULTURE soit sur la même ligne que CODE : SECUL veut dire qu'ils sont «en relation», mais sans plus de

qualification sur la nature du lien. Nous pouvons déduire le sens à donner à ce lien en fonction des noms de champs.

Dans notre cas nous pouvons utiliser la convention qu'un élément de la colonne NOM ANALYSE est en relation *a code SIL* avec l'élément correspondant sur la même ligne de la colonne CODE. Ce sens, cette relation *a code SIL*, devra être le **même** pour toutes les lignes de la table.

Mais il serait parfois utile que le sens donné à la relation ne soit pas exactement le même pour toutes les lignes. Par exemple, certaines analyses n'ont pas de code SIL, car elles sont gérées par un autre système : le SIIATH (Système d'Information Intégré sur les Activités Transfusionnelles et d'Hémovigilance) qui a ses propres codes. Nous voudrions pouvoir mettre le code SIIATH dans la même colonne, mais alors la sémantique de la relation *a code SIL* ne sera plus valable pour toute la table.

#### 3.6.1.1.3 Difficultés dans la représentation des règles

Une dernière difficulté avec le modèle relationnel est avec la représentation des règles qui sont très fréquentes en microbiologie clinique: elles peuvent s'adresser à chaque étape du processus d'une analyse et porter sur chacun des éléments de ce processus. Par exemple dans le protocole sur les selles de SJ nous avons les règles indiquées par le Tableau 3-2 :

Élément du processus sur lequel la règle est basée	Caractéristique de l'élément	Règle
Patient	Âge	Faire automatiquement la recherche de STEC chez les patients de 5 ans et moins
	Localisation	Ne pas faire la culture de selles ordinaires chez les patients hospitalisés depuis plusieurs jours
Spécimen	Date du prélèvement	Faire la recherche du Rotavirus sur les selles pédiatriques de janvier à mai
	Date du prélèvement	Rejeter un spécimen de selles de plus de 24 heures d'âge
	Apparence	Ne pas ensemer les selles dures ou contenant du Baryum
Milieus de culture	Incubation	Durée minimale et maximale

Tableau 3-2, exemple de règles pour la culture de selles selon le labo SJ  
Source SJ\_MIC-PON-072\_V7

La représentation de ces règles par des entrées de base de données n'est pas simple. Par exemple dans le SIL, chaque règle est représentée par une ligne d'une table à 114 colonnes, les colonnes de cette table sont une énumération de tous les éléments sur lesquels on pourrait éventuellement mettre un critère avec un indicateur sur la valeur du critère le cas échéant.

Nous aimerions une représentation plus claire de ces règles et pouvoir comparer automatiquement certaines règles. Nous aimerions pouvoir vérifier automatiquement qu'une procédure interne qui stipule de 18 à 24 heures d'incubation est conforme, bien que plus sévère, à la recommandation d'un document normatif qui mentionne une incubation de 16 à 24 heures. On peut encoder et comparer des règles avec une base de données, mais pas de façon intrinsèque au modèle.

### 3.6.2 Le modèle par ontologie

Le modèle par ontologie, que nous verrons en détail au chapitre suivant, est plus souple que le modèle relationnel, car il est évolutif. Avec une ontologie, nous pouvons modifier des liens entre les éléments ou en ajouter, sans toucher à l'organisation déjà en place.

Une ontologie permet de déclarer entre les éléments qui la composent des relations bien plus complexes que celles qui sont possibles avec le modèle relationnel. Ces relations complexes font qu'une ontologie est à même d'encoder des règles comme celles que nous avons vues au Tableau 3-2 sans recours à une programmation hors de l'ontologie.

Les ontologies se prêtent aussi à l'inférence logique automatique, ce qui veut dire révéler des éléments de logiques nouveaux par déduction automatique sur les éléments déjà présents dans l'ontologie. C'est cette capacité de raisonnement automatique des ontologies qui a été la motivation première de ce mémoire.

Un dernier avantage des ontologies sur le modèle relationnel provient plus de l'usage que l'on fait en général des ontologies que de ses capacités comme telles. En effet, contrairement aux bases de données, les ontologies ont souvent comme objectif une certaine universalité, au moins au sein d'un domaine d'activité. Cet objectif était présent même dans les versions plus anciennes du concept d'ontologie comme les taxonomies. Les codifications modernes comme OWL sont d'emblée conçues pour la diffusion, entre autres, avec le recours à des identificateurs sous la forme d'Internationalized Resource Identifier (IRI).

Les éléments de base de la microbiologie clinique sont les mêmes dans le monde entier, les notions de milieux de transport et de culture, de conditions et durée d'incubation sont communes à tout le domaine. Les efforts de standardisation dans le domaine ont été et sont encore très nombreux, mais cette standardisation ne s'appuie pas encore sur des instruments modernes de codification et de diffusion comme les ontologies.

## Chapitre 4 Les ontologies

Nous avons vu à la fin du chapitre précédent que nous avons choisi le modèle par ontologie parce qu'il est souple et évolutif, parce que ses capacités d'expression sont impressionnantes et parce qu'il se prête à l'inférence automatique. Dans ce chapitre, nous allons approfondir nos connaissances sur les ontologies.

Nous allons commencer par une définition des ontologies. Par la suite, nous allons discuter de notre recherche d'ontologies modèles pour notre problème et décrire celles que nous avons trouvées, en particulier Basic Formal Ontology (BFO). Le formalisme que BFO propose est crucial pour notre ontologie OntoMic, il sera développé en profondeur. Nous allons également expliquer comment se font les inférences logiques.

Notre présentation des ontologies, de BFO et de l'inférence logique se fera en langue naturelle, sans le recours à un formalisme autre que celui de sujet – relation - objet. Nous avons fait ce choix, car il permet de s'attarder uniquement à la logique sans confusion avec la syntaxe d'un langage formel.

Finalement, nous allons présenter les grandes lignes d'OWL, le langage formel avec lequel sont écrites la majeure partie des ontologies, dont OntoMic. Nous allons montrer comment les concepts en langue naturelle que nous avons vus sont traduits en énoncés OWL.

### 4.1 Introduction aux ontologies

La définition d'une ontologie peut être difficile à cerner, nous avons adopté celle de Barry Smith, Waclaw Kusnierczyk, Daniel Schober et Werner Ceusters :

A TAXONOMY is a tree-form graph-theoretic representational artifact with nodes representing universals or classes and edges representing is\_a or subset relations. An ONTOLOGY is a representational artifact, comprising a taxonomy as proper part, whose representational units are intended to designate some combination of universals, defined classes, and certain relations between them. (Arp, Smith, & Spear, 2015)

Cette définition est fortement influencée par l'ontologie Basic Formal Ontology (BFO) des mêmes auteurs. BFO est à la base de notre travail sera présenté en profondeur à la section 4.3. Cette définition introduit des concepts qui sont eux-mêmes à définir, nous allons présenter ici le concept de classes. D'autres concepts comme `universal` et `defined_class` seront définis à la section 4.3 sur BFO.

Une classe dans une ontologie peut être vue comme une niche de rangement, il y a plusieurs façons de ranger les choses, il y a donc plusieurs de façons de définir ce qui doit se retrouver dans une niche, dans une classe. La tâche d'une TAXONOMY est de classer les « choses », c'est-à-dire les individual. Chaque classe d'une TAXONOMY a sa clé de rangement et la hiérarchie de classe est une suite de généralisations des clés de rangement.

Il est souvent plus facile de comprendre une généralisation en s'attardant au contenu des classes qu'à leurs clés de rangement. Par exemple, nous savons tous que la classe « mammifère » généralise la classe « chat », mais bien peu d'entre nous comprennent la généralisation de clés sous-jacente. Il y a donc une nuance logique entre inclusion et généralisation pour les classes, mais dans la pratique ces termes sont utilisés de façon interchangeable.

L'ontologie décrit également des relations canoniques entre les classes, c'est-à-dire les relations qui doivent être présentes pour toutes les instances ; c'est la partie ONTOLOGY. Un exemple de relation canonique est celle qui unit un spécimen et le patient duquel il provient. Aucun spécimen ne peut exister sans avoir été prélevé chez un patient, l'ontologie exprime donc cette relation entre la classe `spécimen` et la classe `patient`, ainsi que son caractère obligatoire.

Cette définition en langue naturelle ne stipule pas avec quel formalisme les relations entre les éléments de l'ontologie sont définies. Le formalisme le plus fréquent est celui de sujet – relation – objet que nous avons vu avec Semantic MediaWiki à la section 3.5.3.2 et dont nous allons illustrer les capacités.

### 4.1.1 La représentation par triplets

Pour montrer comment on peut représenter des données avec des triplets et comment ceux-ci peuvent exprimer des logiques complexes nous allons écrire avec des triplets le contenu présenté dans le Tableau 3-1. Ce tableau nous donne les combinaisons acceptables entre méthode de cueillette – substance biologique – contenant – transport pour la culture de selles.

Nous pouvons représenter les informations de ce tableau avec les triplets suivant :

Selles	<i>substance_acceptée_par</i>	Culture_selles
Écouvillon_rectal	<i>substance_acceptée_par</i>	Culture_selles
Écouvillon_rectal	<i>se_cueille_par</i>	Écouvillonnage_rectal
Selles	<i>se_cueille_par</i>	Défécation
Selles	<i>se_cueille_par</i>	Stomie

(Pot\_simple *choix\_contenant\_pour*  
(selles **et** *substance\_acceptée\_par* Culture\_selles))  
*compatible\_avec* transport\_2\_heures

Si les cinq premiers énoncés sont limpides, il n'en est pas de même du dernier qui est assez complexe. Cet énoncé se traduit en langue naturelle comme :

Les Pots simples qui sont choisis pour contenir des selles, selles qui sont dédiées à une Culture selles. Ces Pots sont compatibles avec un transport de 2 heures.

Cet énoncé n'impose pas le transport en 2 heures pour un Pot simple contenant une autre substance que Selles. Il n'impose pas non plus le critère de 2 heures à un Pot simple contenant une Selle qui est prévue pour un autre test qu'une Culture de selles.

Ceci illustre bien les impressionnantes capacités d'expression du modèle par ontologie, mais qui sont au prix d'une lourdeur et d'une certaine opacité des énoncés. De plus, plusieurs énoncés différents peuvent représenter la même logique. Donc, par rapport au modèle relationnel nous gagnons en expressivité et aussi en souplesse, mais nous devons porter une attention constante à contrôler la complexité du modèle.

## **4.2 Revue de littérature**

Nous avons souligné que notre définition d'une ontologie est fortement liée à l'ontologie BFO. Nous avons découvert BFO lors de notre recherche d'une ontologie modèle pour notre problème. Avant d'aller plus loin, nous allons donc dès maintenant présenter les résultats de notre recherche dans la littérature.

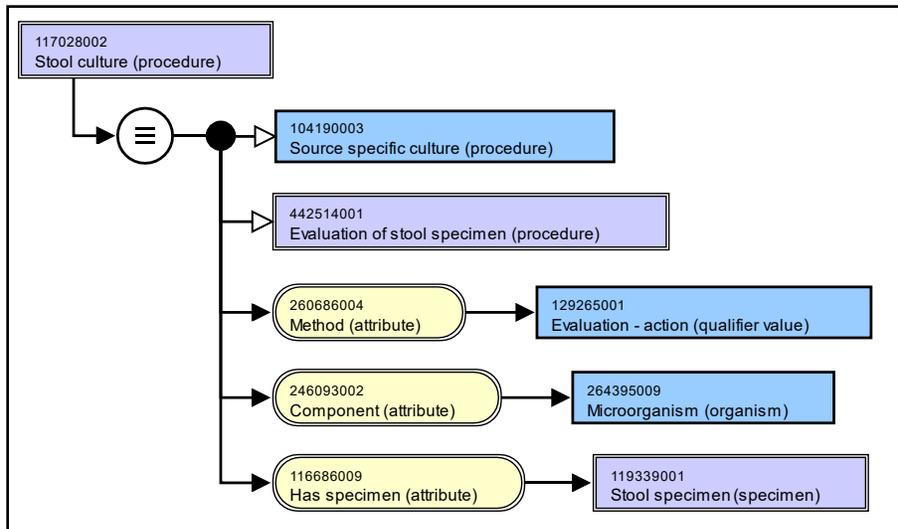
Le domaine médical est un champ d'application majeur pour les ontologies. Bien que les ontologies médicales soient abondantes, nous n'en avons pas retrouvé qui s'intéressent aux logiques procédurales des laboratoires cliniques.

Notre recherche a été étendue aux ontologies et classifications médicales qui ont un rapport indirect avec le domaine des laboratoires et nous avons également exploré les travaux pour le domaine connexe de l'expérimentation scientifique en laboratoire.

### **4.2.1 Les ontologies et classifications médicales**

Nous avons vu avec l'Exemple 3 et avec l'analyse de ses causes à la section 3.4 que même si le propos d'un protocole est centré sur la technique de laboratoire, nous devons toujours analyser un protocole à la lumière de son impact sur l'ensemble du cheminement de l'analyse. Cheminement, qui commence dès la prise de décision par le soignant de prescrire et qui se termine par l'interprétation des résultats.

En conséquence, nous avons considéré une large gamme de travaux à la recherche de référence pour notre ontologie. Entre autres, nous nous sommes attardés aux travaux en cours qui visent l'encodage des demandes d'analyses et de leurs résultats. Ces travaux sont d'envergure, très actifs et chapeautés par de grandes organisations et au minimum ils nous donnent des références pour notre vocabulaire standard.



Extrait de document 4-1, hiérarchie de classes pour la culture de selles selon SNOMED-CT  
 Source : <http://browser.ihtsdotools.org>

#### 4.2.1.1 LOINC

Le travail le plus avancé dans le domaine des laboratoires est Logical Observation Identifiers Names and Codes (LOINC) (« What LOINC is — LOINC », s. d.). LOINC vise à donner des désignations standardisées aux résultats des tests de laboratoire, donc un moyen d'échanger des résultats. LOINC n'est pas présentée sous la forme d'une ontologie classique, elle ne décrit pas de classes donc pas de relations canoniques entre les classes. LOINC est donc une taxonomie présentée sous forme de tableau.

Sur le site web de LOINC on explique que ce qui est encodé est la « question posée par le test », cette question est : qu'est-ce qui a été caractérisé (par exemple le potassium), dans quoi (dans le sérum) et avec quelle échelle de mesure ? Le même descripteur de résultat LOINC pour le potassium sérique peut être utilisé pour commander le test.

Pour la microbiologie, la relation entre le résultat et la demande de test n'est pas aussi directe, une culture de selles peut avoir comme résultat l'absence d'entéropathogènes ou la présence d'une des cinq espèces d'entéropathogènes standards. Le résultat pourrait aussi être la présence de tous autres organismes qui auront été jugés pertinents selon le cas du patient.

Ceci fait que pour la culture d'urine LOINC a neuf codes de résultats, ces codes sont à choisir selon le nombre de bactéries rapportées et selon la durée de la culture; de toute évidence ces codes ne peuvent servir à définir le test commandé.

Finalement, LOINC ne gère pas le descripteur complet du spécimen, LOINC ne tient pas compte de la méthode de prélèvement (urine par miction ou par ponction vésicale), ne précise pas non plus la localisation anatomique, par exemple « pleural fluid » est par définition un liquide qui provient de la cavité pleurale, mais est-ce la cavité pleurale gauche ou droite ?

#### **4.2.1.2 SNOMED**

Une autre codification Systematized Nomenclature of Medicine -- Clinical Terms (« SNOMED Home page », s. d.) se veut une systématisation de tout le vocabulaire médical. SNOMED-CT est une ontologie complète avec une hiérarchie de classes et des relations canoniques comme nous pouvons le voir à l'Extrait de document 4-1 pour la culture de selles.

SNOMED\_CT est de très grande taille avec plus de trois cent mille classes (« SNOMED CT - Summary | NCBO BioPortal », s. d.). La nomenclature de SNOMED-CT comprend des descripteurs pour les spécimens, ce qui manque à LOINC. Les méthodes de prélèvements y sont répertoriées ainsi que les localisations anatomiques. Cependant, la classification de SNOMED-CT pour les tests de microbiologie est assez confuse, les ascendances sont multiples et n'ont pas un modèle clairement établi comme LOINC.

Les organisations responsables de SNOMED-CT et LOINC ont maintenant une entente de partenariat pour harmoniser leurs travaux, mais ceci n'est qu'à ses débuts.

#### **4.2.1.3 DSQ**

Au Québec, le DSQ (Dossier Santé Québec) regroupe les résultats de toutes les analyses de laboratoires. À la base, le DSQ est un dossier numérisé où les résultats sont présentés en PDF, avec le format de rapport du laboratoire qui a effectué l'analyse. Certains résultats clés ont été normalisés afin de permettre de suivre leur évolution et de les comparer même s'ils ont été faits dans des laboratoires différents.

Aucun test de microbiologie n'a été normalisé dans le DSQ et ceci autant en ce qui concerne la désignation des cultures que des résultats.

#### **4.2.1.4 Codifications du domaine des laboratoires en langue naturelle**

Il existe de nombreux documents normatifs. Certains de ceux-ci renferment des éléments de taxonomies qui sont exploitables, ces documents sont souvent non publics comme ceux de l'organisme normatif CLSI (Clinical Laboratory Standards Institute). Cet éditeur diffuse maintenant des documents HTML bâtis avec des transformations XML des documents originaux.

Pour le moment leur système permet de comparer une section du document avec cette même section dans les versions antérieures. Ce système repose donc sur une architecture de document avec des identificateurs de sections stables, comme l'approche que nous avons envisagée avec MediaWiki.

Leur système permet également les annotations des sections, encore une fois un besoin que nous avons identifié avec notre approche par wiki. Finalement, le système ne permet pas d'accéder aux informations granulaires contenues dans les sections, on ne peut pas par exemple rechercher et extraire du document le chiffre qui accompagne une norme sur une température.

#### **4.2.2 Codification dans un domaine connexe**

Dans le domaine connexe de l'expérimentation scientifique biomédicale, nous avons trouvé trois ontologies : OBI, EXACT et SMART Protocols. Toutes les trois visent à décrire les protocoles expérimentaux et à annoter les données produites pour permettre à d'autres équipes de reprendre les expériences pour en valider les résultats.

L'Ontology for Biomedical Investigations (OBI) (Bandrowski, Brinkman, Brochhausen, et al., 2016) fait partie du groupe d'ontologies gérées par l'OBO Foundry (Open Biomedical Ontologies Foundry) (Arp et al., 2015). Les ontologies de ce groupe s'adressent principalement aux sciences biomédicales et visent l'interopérabilité. La plus connue étant GO (Gene Ontology) qui relie les gènes à leur fonction. Les ontologies du groupe OBO sont toutes basées sur BFO (Basic Formal Ontology), une ontologie de haut niveau initiée en 2002 par deux philosophes Barry Smith et Pierre Grenon.

#### 4.2.2.1 EXACT

Experiment ACTions (EXACT) (Soldatova, Nadis, King, et al., 2014) n'est pas une ontologie du groupe OBO même si, dans sa version actuelle, EXACT 2 est basée sur BFO. EXACT se veut une description précise des actions effectuées au cours d'une expérience. Pour ce faire, EXACT définit des actions expérimentales canoniques comme *move*, *wait*, *heat*. OBI quant à elle vise plus à décrire les entrants et sortants d'une étape de travail que les actions elles-mêmes.

Par exemple pour codifier l'ensemencement de la gélose sang dans le protocole de selles, EXACT va décrire comme un algorithme

Bouger

quoi: écouvillon

position\_départ: non\_spécifiée

position\_arrivée: « endroit le plus sanglant ou le plus purulent de la selle »

Bouger

quoi : écouvillon

position\_départ: contenant\_selles

position\_arrivée: gélose-sang

OBI va plutôt décrire

Processus: charger\_inoculum

entrant: selles\_dans\_contenant

entrant: écouvillon

sortant: écouvillon\_chargé\_selles

élément\_info: « choisir l'endroit le plus sanglant ou le plus purulent de la selle »

processus: ensemercer\_gélose\_écouvillon

entrant: écouvillon\_chargé\_selles

entrant: gélose-sang\_non\_ensemencée

sortant: gélose-sang\_ensemencée

Avec OBI, c'est l'utilisateur qui définit lui-même une extension d'OBI pour son domaine à partir des structures de base. Dans notre exemple l'action « bouger » est présente dans EXACT, tandis que « charger inoculum » sera défini par l'utilisateur.

EXACT n'est pas un formalisme aussi systématique que celui d'OBI. Par exemple les actions prédéfinies dans EXACT vont de simples comme *move* à des actions composites comme *neutralize*. Ces différents niveaux de complexité des tâches font que l'on pourra coder la même action de manières différentes ; par exemple, ajouter une base pour neutraliser un liquide gastrique peut se coder par « neutraliser » ou encore par « ajouter » solution basique à liquide gastrique.

EXACT a voulu limiter son nombre d'actions, il n'y a l'action « tremper » « dip » dans EXACT, alors qu'elle serait plus appropriée que « bouger » pour notre exemple. Il aurait donc fallu faire des additions à EXACT pour l'adapter à nos besoins.

#### **4.2.2.2 SMART Protocols**

Selon ses auteurs le but de SeMAnTic RepresenTation for Protocols (SMART Protocols) (Giraldo, García, & Corcho, s. d.) est « SP aims to formalize the description of experimental protocols, which we understand as domain-specific workflows embedded within documents. ». Le but de SMART est assez semblable à OntoMic : représenter les logiques des protocoles qui sont contenus dans des documents. Le modèle qu'OntoMic applique aux documents est inspiré de celui de SMART. Pour les deux ontologies, les documents peuvent être vus comme des « textes » avec des métadonnées classiques comme auteur, titre, etc., mais ils sont également vus comme des recueils d'informations que les ontologies doivent modéliser.

SMART réutilise des classes définies par OBI et par EXACT. Pour l'aspect « workflow » SMART fait appel à une autre ontologie : P-Plan, elle-même un dérivé d'un standard PROV-O du W3C. Ces importations diverses sont parfois inutiles, comme l'utilisation d'une classe « Step » de p-plan alors que nous pouvons facilement utiliser la classe de BFO :*process*. Également, nous allons voir dans la section suivante que l'aspect workflow n'est pas central pour notre modélisation et que donc nous n'avons pas besoin des classes supplémentaires d'EXACT.

#### **4.2.2.3 OBI: le standard de facto et notre choix de modèle d'ontologie**

Dans le domaine de l'expérimentation, OBI est le standard de facto, d'ailleurs EXACT et SMART font appel aux classes d'OBI. OBI est aussi l'ontologie qui représente le mieux la dualité protocoles et expériences réalisées avec les protocoles. OBI y parvient en suivant rigoureusement le

formalisme de BFO et en introduisant une ontologie partenaire pour les informations: IAO (Information Artifact Ontology).

Le développement d'OBI est actif avec plusieurs contributeurs, des conférences téléphoniques au moins hebdomadaires et des listes de messagerie. Notre choix d'OBI a ainsi moins de risque de devenir obsolète.

Pour toutes ces raisons, nous avons choisi OBI pour notre ontologie modèle. Même si notre ontologie OntoMic suivra le modèle OBI, elle ne sera pas basée sur OBI, car OBI contient des classes qui ne sont pas utiles à notre champ d'application. Par contre, OntoMic sera basée sur BFO, toutes les classes de BFO se retrouveront dans OntoMic.

### **4.2.3 Conclusion sur la revue de littérature**

LOINC et SNOMED-CT ont été peu utiles dans la construction de la structure d'OntoMic. Elles seront essentielles lorsqu'il faudra créer dans OntoMic toutes les classes nécessaires à la représentation du domaine. Ce qui nous a été le plus utile a été la lecture attentive des articles sur OBI et surtout la manipulation de la formulation OWL d'OBI qui est bien documentée.

La décision de prendre OBI comme modèle pour OntoMic a eu beaucoup plus d'impacts que nous l'avions cru. En effet OBI, formalise les logiques procédurales d'une manière très particulière, a priori difficile à comprendre, mais ce formalisme est très puissant. Une bonne partie du travail sur ce mémoire a été consacré à rendre intelligible ce formalisme.

Pour comprendre le formalisme d'OBI et d'OntoMic, il faut maîtriser l'ontologie de haut niveau BFO. BFO sera donc présenté à la section suivante.

## **4.3 L'ontologie de haut niveau BFO**

BFO est ce que l'on appelle une ontologie de haut niveau. Il est beaucoup plus facile d'expliquer ce qu'est une ontologie de haut niveau par un exemple que d'en donner une définition formelle. Nous allons donc illustrer le rôle d'une ontologie de haut niveau en expliquant le rôle de BFO.

Soulignons seulement que, comme elles suivent le même formalisme de base, les ontologies bâties avec la même ontologie de haut niveau sont plus facilement interoperables.

### **4.3.1 Le rôle de BFO**

BFO se charge d'organiser des concepts de bases, comme la relation fondamentale qu'il y a entre une caractéristique d'une chose et la chose elle-même. Par exemple, c'est BFO qui donnera le cadre pour représenter la résistance à un antibiotique d'une bactérie, car il s'agit bien d'une caractéristique (la résistance), qui est attachée à une chose (une bactérie).

BFO fournit aussi un cadre pour représenter des notions plus subtiles, comme la différence entre le décompte des colonies et cette quantité dans l'absolu. La quantité de colonies est unique et existe même si personne ne l'a mesurée, par contre le décompte de cette quantité peut se faire plus d'une fois et ne pas donner la même valeur à chaque répétition.

### **4.3.2 Présentation générale de BFO**

BFO est décrite en langue naturelle et c'est ainsi qu'elle sera présentée ici, cependant pour être utilisée par des machines BFO a été traduite dans un langage formel : l'OWL, nous verrons ce langage à la section 4.5.

BFO est voulue comme étant de petite taille, avec un formalisme précis et solidement ancré dans des principes logiques et philosophiques. BFO a été développé d'abord et avant tout pour représenter des choses concrètes (Arp et al., 2015). BFO est un formalisme élégant et polyvalent, ce formalisme a été appliqué aux sciences biomédicales, mais aussi au domaine militaire (Morosoff, Rudnicki, Bryant, et al., 2015).

BFO s'intéresse à la classification des entités plus qu'aux relations entre elles. Une fois que deux entités qui doivent être en relation sont classées, il n'y a souvent qu'un seul choix de propriétés possible pour la relation. Notons au passage que ceci est tout à fait à l'opposé de la modélisation de Semantic MediaWiki où la charge sémantique est d'abord portée par les relations.

Également, BFO vise à décrire les choses de la façon la plus concrète possible: « x » est marié à « y » ne nous renseigne pas sur pourquoi nous pouvons affirmer que « x » est marié à « y ». BFO tentera de rester plus près de la réalité en stipulant que « x » et « y » sont tous deux acteurs dans un processus de « vie maritale », nous pourrions définir, si nous le voulons, le processus de vie maritale, son début, son officialisation, sa fin.

Nous pouvons illustrer ce recours à des constructions plus concrètes avec un autre exemple: la codification du contenant pour le spécimen. Avec BFO, on ne devrait pas décrire directement une classe « contenant à spécimen » ; en effet, l'usage que l'on fait de ce matériel n'est pas intrinsèque au contenant, on pourrait par exemple prendre le même contenant pour contenir un réactif plutôt que pour transporter un spécimen.

Ce qui est intrinsèque au contenant c'est sa nature de servir à contenir quelque chose, on pourra donc définir directement une classe `contenant`. Mais la classe `contenant_à_spécimen` sera définie par une règle, elle sera composée des `contenants` auxquels on a attribué le rôle de servir aux spécimens.

#### **4.3.2.1 Les `universal` et `defined_class`**

Les classes `contenant` et `contenant_à_spécimen` décrites à la section précédente sont des exemples d'`universal` et de `defined_class` respectivement. Ces notions sont présentes dans la définition même de ce qu'est une ontologie, définition que nous avons vue à la section 4.1. Avec cette définition, la `TAXONOMY` s'adresse aux `universal` et l'`ONTOLOGY` s'adresse aux `defined_class`.

Notons l'absence des `individual` dans cette définition. En effet pour BFO les individus ne font pas partie de l'ontologie, mais de son application. L'ontologie ne s'attarde pas aux particularités d'une instance « x ».

Par exemple, selon BFO, OntoMic ne doit pas s'intéresser au fait que **la** gélose-sang avec le numéro de série « x » est contaminée<sup>13</sup>. Par contre, OntoMic **doit** décrire les règles qui permettent de déclarer une gélose contaminée et ainsi faire une `defined_class` : `gélose_contaminée`.

#### 4.3.2.1.1 Les universal

Pour BFO chaque `universal` se généralise par un seul autre `universal` selon sa nature même. C'est le principe de l'ascendance unique. Comme nous pouvons le voir dans l'exemple suivant avec les contenants.

Hiérarchie des `universal` pour contenant (`container`) selon OBI et BFO, allant de la classe la plus générale sur la première ligne à la plus spécifique sur la dernière ligne. L'`universal` le plus général, `continuant`, est une des classes fondamentales de BFO que nous verrons à la section 4.3.3.2.

```
continuant
  material_entity
    processed_material
      container
```

Les `container` se généralisent dans la classe `processed_material` qui sont des `material_entity` qui ont été transformées ou créées par un processus délibéré. Même si nous sortons la classe contenant du contexte d'OntoMic, il restera toujours que le contenant a été fabriqué et qu'il peut contenir, ce qui traduit le caractère universel de sa classe.

#### 4.3.2.1.2 Les defined class

Les `defined_class` sont construites à partir des `universal` sur la base de règles, comme nous l'avons vu avec `contenant_transport_spécimen`. Mais la présence d'une règle sur une classe ne veut pas nécessairement dire qu'il s'agit d'une `defined_class`. Il faut faire la différence entre une classe **décrite** par une règle et une véritable `defined_class` qui sera **peuplée** selon une règle.

---

<sup>13</sup> Un dispositif de culture est contaminé lorsqu'il contient la croissance d'un microbe qui provient de l'air ou de toutes autres sources que d'un spécimen.

La classe `processed_material` est une `universal` décrite par une règle, mais elle n'est pas peuplée selon cette règle, on s'attend à ce que l'ontologie déclare quels individus font partie de `processed_material`. Un autre exemple sera `spécimen`, un `spécimen` est toujours composé d'une `substance_biolgique` et d'un contenant. Il y a une règle qui décrit `spécimen`, mais l'appartenance à `spécimen` doit être déclarée et non déduite par raisonnement.

Par contre, l'appartenance à une `defined_class` comme `contenant_spécimen` devrait être déduite et non déclarée par l'ontologie, un individu est automatiquement membre de la classe s'il satisfait à la règle.

#### 4.3.2.1.3 L'inférence logique

Déduire l'appartenance d'un `individual` à une `defined_class` selon ses caractéristiques est ce que l'on appelle une inférence logique. Un autre type d'inférence que l'on peut faire est de déduire la relation entre deux `defined_class`, ou entre une `defined_class` et un `universal`. Les deux classes peuvent être mutuellement exclusives, incluses l'une dans l'autre ou simplement sans lien. Il est à noter qu'il n'y a pas d'inférence à faire entre deux `universal` car leur relation est déclarée par l'ontologie.

Ce sont les inférences logiques qui permettront à OntoMic de vérifier la cohérence, l'exhaustivité et la conformité. Les inférences logiques sont donc essentielles à notre travail et la section 4.4 leur est consacrée.

### 4.3.3 La hiérarchie de classes de BFO

#### 4.3.3.1 Les continuant et occurrent

BFO divise les entités en deux grandes classes `continuant` et `occurrent`.

Un `continuant` peut se transformer: apparaître (naître), disparaître (mourir), mais à tout moment de son existence un `continuant` est une même entité complète. Il y a cependant des points charnières où le `continuant` cesse d'exister pour être remplacé par un autre. Il est facile de concevoir qu'une personne amputée demeure la même entité avant et après son amputation. Mais

la dépouille de cette personne n'est plus la personne, car son essence même est perdue. En effet, l'essence d'une personne n'est pas dans son assemblage d'organes, mais dans le fait qu'elle est vivante. De même, ce qui reste d'un spécimen après en avoir utilisé une partie pour ensemencement est le même spécimen ; cependant une fois que nous avons transféré ce reste dans un nouveau contenant, nous avons une autre entité: une aliquote du spécimen. En effet, comme nous l'avons vu, un spécimen est la combinaison d'une substance biologique et de son contenant.

Un **occurrent**, par opposition, n'existe au complet que dans son étendue dans le temps ; un **occurrent** n'est pas une chose tangible qui se modifie dans le temps, mais le processus de changement lui-même. Aucun instantané ne peut montrer l'**occurrent** au complet. Le processus de transport de spécimens est un exemple d'**occurrent**.

#### **4.3.3.2 Les classes filles de continuant**

BFO divise les continuant en **independent\_continuant** qui existent de façon autonome et en **dependent\_continuant** qui ne peuvent pas exister sans être rattaché à un **independent\_continuant**. Un exemple d'**independent\_continuant** est un milieu de culture liquide, la turbidité de ce milieu sera un **dependent\_continuant**, en effet aucune turbidité ne peut exister de façon autonome, elle doit toujours être la turbidité de quelque chose.

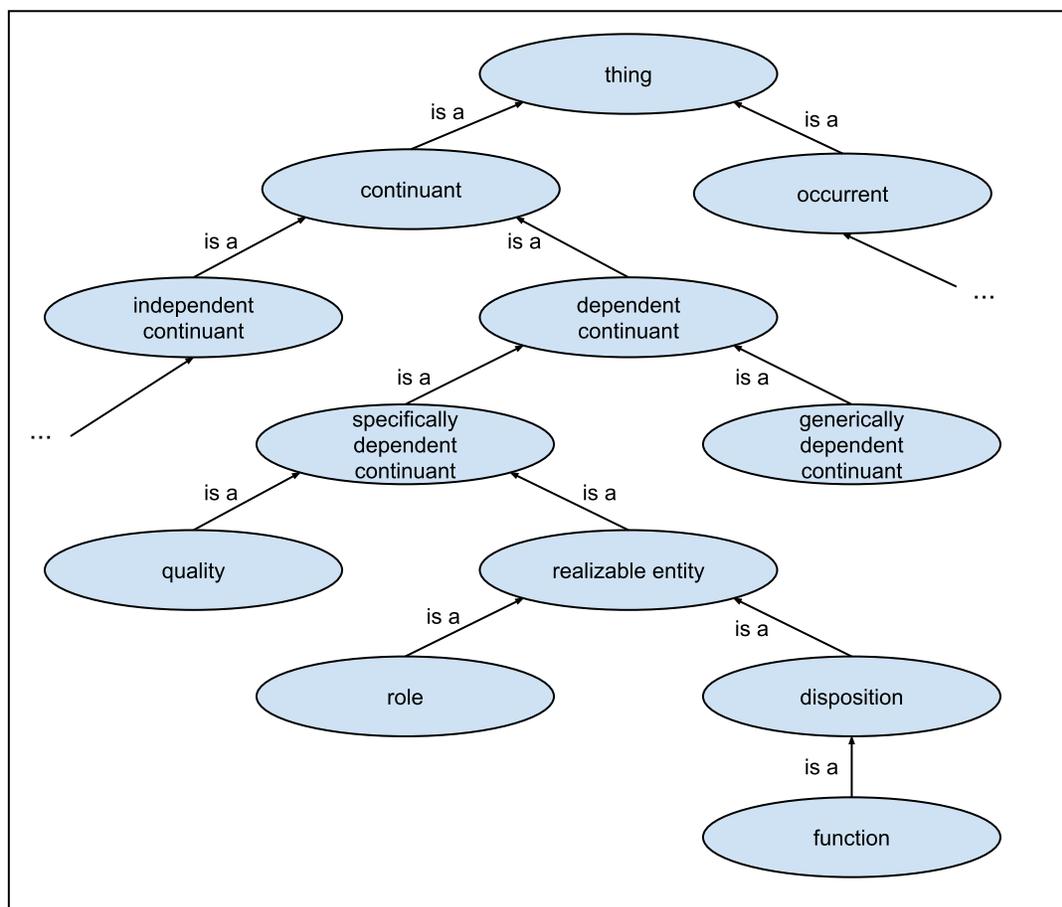


Figure 4-1, organisation des dependant continuant dans BFO

#### 4.3.3.2.1 Les dependent continuant

La Figure 4-1 illustre le développement de la classe `specifically dependent_continuant`<sup>14</sup>, qui se divise en deux groupes: les `quality` et les `realizable_entity`. Une `quality` se rattache à un `independent_continuant` et est toujours présente. La turbidité d'un liquide peut diminuer dans le temps par sédimentation, mais il y aura toujours une turbidité observable. Une `realizable_entity` ne sera pas toujours manifeste chez son `independent_continuant` elle devient manifeste à la suite d'un événement, d'un `occurrent`, c'est une potentialité. Un exemple de `realizable_entity` est l'activité antibiotique d'une molécule, cette potentialité de la molécule n'est pas manifeste dans la

<sup>14</sup> Nous étudions ici qu'un seul des deux sous-types des `dependent_continuant` soit les `specifically dependent_continuant`, l'autre sous-type des `generically dependant_continuant` serons vus à la section 7.2.1.

fiolle qui la contient, elle ne devient manifeste que lorsque la molécule est mise en présence de bactéries *in vitro* ou *in vivo*.

Les *realizable\_entity* sont subdivisées à leur tour en *role*, *disposition* et *function*, un *role* est attribué à un *independent\_continuant* par un agent extérieur et sous-entend une certaine sélection pour des candidats capables de remplir le *role* tandis que pour les *disposition* ou *function* la caractéristique est intrinsèque à un *independent\_continuant*.

Lorsque nous avons défini les contenants à spécimen, nous avons donné un exemple de *role*, ce ne sont pas tous les contenants qui peuvent être des contenants à spécimen, ils doivent être étanches, stériles, résistants aux impacts. Il y a donc un agent extérieur qui juge qu'un contenant est apte à remplir le *role* de contenant à spécimen. Un autre exemple est le *role* de technicienne de laboratoire que l'on attribue à un individu, ce *role* n'est évidemment pas attribuable à tous les individus de la classe *homo\_sapiens*. Finalement, la perte d'un *role* ne change pas l'essence de l'*independent\_continuant* auquel il est rattaché, une technicienne de laboratoire demeure la même personne même si elle réoriente sa carrière. Mais la perte d'une *disposition* ou d'une *function* sous-entend que l'*independent\_continuant* a été altéré, de façon à ce que la potentialité que constitue la *disposition* ou la *function* disparaissent.

Un contenant peut perdre son rôle de contenant à spécimen sans être physiquement altéré si par exemple une nouvelle réglementation exige une plus grande résistance aux impacts, mais pour perdre sa *function* de contenir il doit alors être brisé.

Une *function* est une *disposition* avec la caractéristique supplémentaire que l'*independent\_continuant* qui en est porteur a été conçu ou a évolué par sélection pour accomplir la *function*. Un exemple de *disposition* est la capacité de l'*E. coli* de transformer le lactose en acide<sup>15</sup>. Cependant, la capacité de l'enzyme *bêta-galactosidase* de l'*E. coli* à dégrader le lactose est une *function*, car elle est le résultat d'une évolution pour accomplir cette tâche. Cette distinction est un peu délicate

---

<sup>15</sup> Il y a des *E. coli* qui ne fermentent pas le lactose, mais ce sont de rares exceptions. On trouvera une discussion sur comment tenir compte des exceptions à la page 15 du chapitre 7 de (Bandrowski, Brinkman, Brochhausen, et al., 2016)

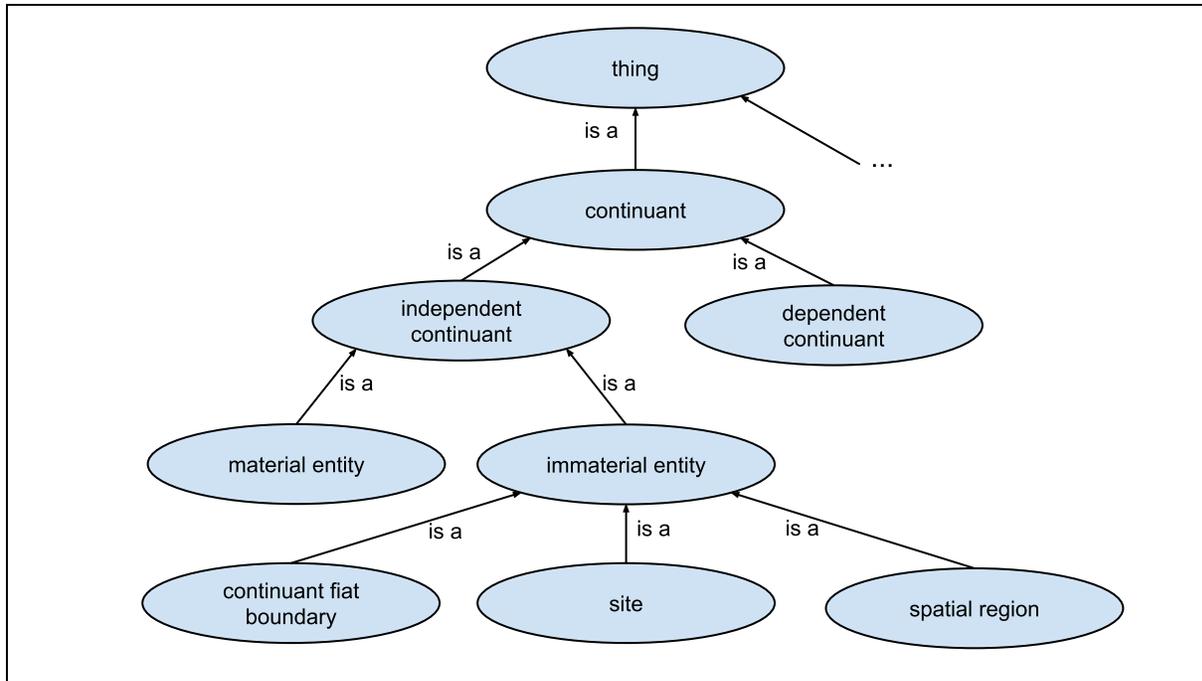


Figure 4-2, organisation des independant continuant dans BFO

à manipuler ; par exemple, on sait que la bactérie *Salmonella typhi* peut donner la fièvre typhoïde chez l'être humain, mais en tenant compte du fait que plusieurs données montrent que *S. typhi* a évolué spécifiquement pour être pathogène, pouvons-nous dire *S. typhi* a la fonction d'infecter l'humain ?

Pour aider à distinguer si nous devons utiliser une **disposition** ou une **fonction**, nous ajoutons à la définition de BFO que pour un `independent_continuant` la perte de la ou des fonction qui lui sont associées veut dire que non seulement il a été changé, mais qu'en plus que son existence même n'est plus assurée. Par exemple, une centrifugeuse a comme fonction de centrifuger ; si elle ne peut plus l'accomplir, alors cela signifie que sa nature a été changée (elle est brisée), mais qu'en plus elle est inutile et risque d'être envoyée à la casse. Pour un clone de *Salmonella typhi*, si sa fonction de donner la typhoïde est perdue alors sa survie est en jeu, nous pouvons voir ceci avec les formes atténuées de *S typhi* produites en laboratoire pour servir de vaccin, bien que ces formes atténuées survivent à un passage chez l'humain en laissant une certaine immunité, elles ne parviennent pas à se répandre en masse dans la nature.

#### 4.3.3.2 Les independent continuant

La Figure 4-2 montre que BFO sépare les `independent_continuant` en deux classes `material_entity` pour ceux qui ont un caractère tangible comme une personne et les `immaterial_entity` qui bien qu'intangibles sont néanmoins bien réel comme l'espace réservé pour l'accueil des géloses dans un incubateur.

Un `continuant_fiat_boundary` décrit la frontière d'une entité, ce n'est pas une entité physique comme la plèvre qui recouvre le poumon, mais une construction abstraite qui englobe l'entité et est de taille minimale. Les `continuant_fiat_boundary` sont surtout utiles lorsque nous avons à décrire l'interface entre deux entités matérielles, par exemple la zone de contact entre un milieu de culture gélifié et un amas de bactéries qui poussent à sa surface.

Un `site` est une zone de l'espace incluse dans un `material_entity`, un `site` se déplace avec le `material_entity` qui le contient. La surface d'une gélose est un `site`, la position de cette surface est fixe relativement à la gélose.

Une `spatial_region` est, comme un `site`, une région de l'espace, mais contrairement au `site` sa localisation n'est pas fixée par rapport à un `material_entity`, mais par rapport à un point de référence géographique. Un `site` peut bouger dans l'espace et donc changer de `spatial_region`, mais une `spatial_region` est immobile par rapport à sa référence. L'adresse civique du laboratoire est une `spatial_region`.

#### 4.3.3.3 Les occurrents

La Figure 4-3 décrit la hiérarchie de classes pour les `occurrent`. La classe de base est `process` ; un `process` a toujours une relation avec un `independent_continuant`, soit que ce dernier est l'acteur du `process` soit qu'il le subit. Ce qui arrive toujours à quelque chose ou par quelque chose. La lecture des dispositifs, le repérage des colonies d'intérêt, le travail sur les `colonie_d'intérêt` sont tous des exemples de `process`.

Rappelons que les `process` sont également le passage obligé pour que les `realizable_entity` s'expriment, ce passage obligé peut être un processus spontané comme dans le cas de la disposition

du E coli à dégrader le lactose ou un processus délibéré comme l'exécution du rôle de technicienne de laboratoire pour une analyse.

On classe dans les *occurents* en plus des *process* les *process\_boundary* et les *temporal\_region*. Un *process\_boundary* est un événement qui marque un point précis dans le déroulement d'un *process* comme son début ou sa fin. Ces événements charnières sont considérés comme étant indivisibles et survenant instantanément, les *process\_boundary* indiquent donc le niveau de granularité du *process*, granularité dans le temps et dans le niveau de détails. Par exemple pour *ensemencement\_selles*, le *process\_boundary* qui marque le début sera le chargement de l'inoculum en trempant un écouvillon dans la selle. Nous ne décrirons pas qu'il faut débarrasser et saisir un écouvillon stérile, saisir le pot de selles, agripper le couvercle, le dévisser...

Les *temporal\_region* sont à zéro ou une dimension ; celles à zéro dimension sont des localisations dans le temps jugées instantanées et celles à une dimension des intervalles de temps. Les *process\_boundary* surviennent dans une *temporal\_region* à zéro dimension et les *process* dans celles à une dimension.

#### 4.3.3.4 Les relations dans BFO

Comme nous l'avons mentionné plus haut, la modélisation en BFO porte plus sur la définition des choses que sur le type de lien entre les choses. BFO ne définit que quelques relations (avec leur réciproque, si elle existe, entre parenthèses).

1. *part\_of\_(has\_part)*: une relation classique qui s'applique autant au continuant qu'au *occurent*
2. *located\_in\_(location\_of)*: relie un continuant avec sa localisation dans l'espace
3. *adjacent\_to*: relie deux régions de l'espace qui sont adjacentes
4. *occurs\_in*: relie un *process* avec la région de l'espace où il se produit
5. *derives\_from\_(derives\_into)*: relie un continuant qui se transforme en un autre

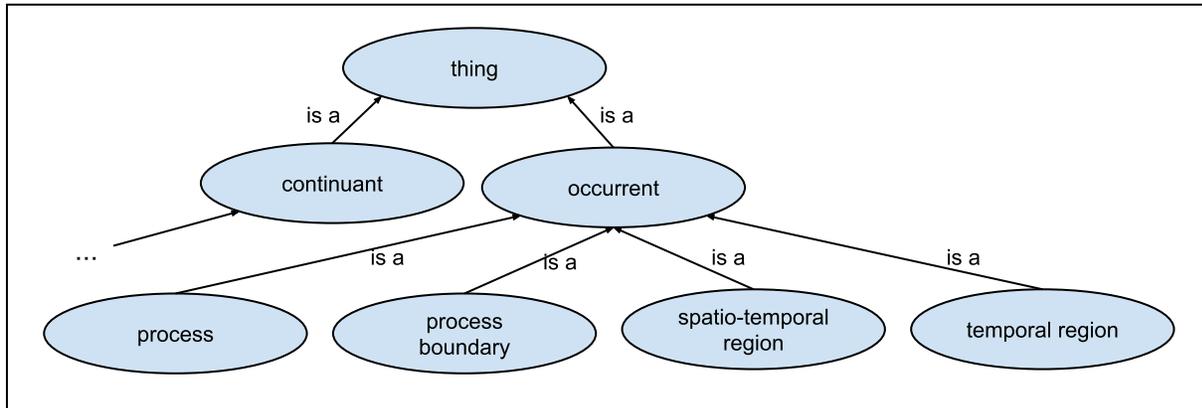


Figure 4-3, organisation des *occurrent* dans BFO

6. *has\_participant* (*participate\_in*): relie un processus avec un continuant par qui le processus arrive ou à qui le processus arrive.
7. *inheres\_in* (*bearer\_of*): relie un *dependent continuant* avec son *independent continuant*
8. *realizes* (*realized\_in*): relie un processus avec un *realizable entity* qui devient manifeste dans le processus.

## 4.4 Les inférences logiques avec les ontologies

À la section 4.3.2.1.3 nous avons introduit le concept d'inférence logique, nous allons l'explicitier ici. Faire des inférences logiques sur une ontologie veut dire ajouter des éléments de logique nouveaux à l'ontologie par déduction sur les éléments déjà présents.

Les inférences peuvent se faire sur les classes, les individus et les relations. *OntoMic* n'utilise pas les inférences sur les relations, car ce n'est pas nécessaire. En effet, *OntoMic* ne définit pas de nouvelles relations par rapport à celles fournies par BFO et OBI.

Il nous reste les inférences sur les classes et les individus, pour *OntoMic* les inférences sur les classes sont de loin les plus importantes comme nous le verrons plus loin.

Dans ce qui suit, nous allons présenter l'inférence logique en langue naturelle comme nous avons présenté BFO, par la suite nous verrons comment le concept formel d'inférence logique est implanté par des raisonneurs sur les ontologies écrites en OWL.

### 4.4.1 L'inférence sur les classes : la subsomption

Dans le domaine des ontologies, la subsomption pour une classe est la recherche d'autres classes de l'ontologie qui la généralisent. La subsomption pour une classe peut aussi être vue comme la recherche d'autres classes qui la contiennent, sous réserve de la nuance logique entre généralisation et inclusion que nous avons vue à la section 4.1.

Pour illustrer l'utilité de la subsomption pour OntoMic, prenons notre exemple du STEC, une cause de colites hémorragiques et supposons que nous avons les définitions suivantes:

La colite hémorragique est un syndrome entérique

Les STEC sont responsables de colite hémorragique

Les entéropathogènes comprennent tous les agents capables de causer un syndrome entérique

Alors par déduction nous pouvons affirmer que les STEC sont une sous-classe des entéropathogènes. Cet exemple est une utilisation triviale de la subsomption. Les inférences de subsomption que nous aurons à faire pour vérifier l'atteinte de nos trois objectifs sont beaucoup plus complexes.

On peut démontrer que toutes les inférences logiques que l'on peut faire sur les classes se ramènent à la subsomption. Par exemple, si nous voulons vérifier s'il est possible qu'une `defined_class` « A » contiennent des individus, il suffit de vérifier qu'il n'y a pas de relation de subsomption entre A et `Nothing`, `Nothing` étant la classe vide canonique. Vérifier la satisfaisabilité de la règle à la base d'une `defined_class` peut sembler trivial, mais comme une règle peut se baser sur des éléments qui eux-mêmes peuvent aussi soumis à des règles, nous pouvons avoir de longues chaînes d'implication avec des conséquences souvent inattendues.

### 4.4.2 Le classement des instances

Toutes les ontologies doivent avoir une classe maîtresse dont toutes les autres sont sous-classes. Cette surclasse générale se nomme le plus souvent `Thing`, toutes les instances, les `individual` sont, au minimum, des instances de `Thing`.

Nous pouvons faire des inférences sur l'appartenance d'un **individual** à des classes. Ce concept de recherche d'appartenance à des classes s'applique autant aux **universal** qu'aux **defined\_class**, mais comme l'appartenance à un **universal** est déclarative seule l'appartenance aux **defined\_class** est intéressante.

Dans OntoMic très peu d'entités sont décrites comme des individus, la plupart des entités sont des classes. Par exemple pour les bactéries qu'est-ce qui constitue un individu ? Chez un malade atteint d'« une » bactérie, il y a en fait une population de milliards de cellules bactériennes qui l'infectent. Notre modélisation se veut le plus près de la réalité possible, donc les instances d'une classe de bactéries comme STEC sont des cellules bactériennes uniques, peu importe si leur existence est très courte avant d'éclater ou de se diviser.

Pour OntoMic « la » bactérie qui infecte un patient n'est pas une instance, mais ce que BFO appelle un **object\_aggregate**. Les unités qui composent un **object\_aggregate** sont des **member** qui n'ont pas d'importance individuelle. La règle qui décrit l'**universal** : **object\_aggregate**, exige seulement que les **member** soient des instances d'une classe commune. Les membres de l'**object\_aggregate** qui représentent « la » bactérie qui infecte le patient sont tous dans la même classe et c'est cette classe qui donne les caractéristiques de « la » bactérie.

Il y aura donc très peu d'individus dans OntoMic et, par ce fait, les inférences sur les individus seront peu utilisées.

### **4.4.3 Les raisonneurs et l'inférence logique**

Dans ce qui précède, nous avons introduit l'inférence comme un concept formel, un peu comme si c'était un humain qui démontrait la véracité d'une inférence comme on démontre un théorème.

Si la logique d'une ontologie suit le formalisme du langage OWL alors l'inférence peut être faite automatiquement. Nous verrons le langage OWL à la section suivante, nous verrons également les particularités du raisonnement automatique en OWL, particularités qui génèrent parfois des résultats pour le moins surprenants.

## 4.5 Le langage OWL

Pour être utilisables par des machines les ontologies ne peuvent être simplement décrites en langue naturelle, elles doivent être formalisées par un langage. Le Web Ontology Language (OWL) développé par le World Wide Web Consortium (W3C) est le standard de facto dans le domaine. Toutes les ontologies du groupe OBO sont maintenant distribuées dans le langage OWL.

Les énoncés en OWL peuvent tous être transformés en triplets : sujet – relation – objet, selon le modèle Resource Description Framework (RDF), également du W3C. Ces triplets peuvent servir à déclarer les instances des classes, les classes elles-mêmes, la hiérarchie de classes, les relations et les règles qui définissent un `universal` ou qui peuplent une `defined_class`.

### 4.5.1 Une TAXONOMY en OWL

OWL permet de déclarer les classes et les instances avec la relation spéciale *type* et les mots clés « `class` » et « `individual` »<sup>16</sup>. OWL déclare une classe comme sous-classe d'une autre avec l'aide de la relation *subClassOf*. Simplement avec ces deux relations OWL peut définir une taxonomie et classer des individus dans la taxonomie.

Mais bien sûr, OWL est bien plus riche que ce que les deux relations ci-haut offrent. La description de toutes les possibilités d'OWL dépasse de loin le cadre de ce mémoire. Par contre, dans la section suivante nous décrirons ce que l'on appelle des axiomes, car sont des constructions d'OWL absolument cruciales pour OntoMic. C'est avec des axiomes que les logiques procédurales sont encodées.

### 4.5.2 Les axiomes

On peut définir des constructions complexes seulement avec des triplets sujet – propriété – objet, parce qu'avec certaines relations le membre de droite peut être une combinaison de plusieurs

---

<sup>16</sup> OWL a plusieurs syntaxes, ceci est une simplification, mais qui ne change pas le formalisme fondamental.

éléments. C'est le cas pour deux relations prédéfinies de OWL : *subClassOf* et *equivalentClass* le terme de gauche, l'objet, peut être une classe de l'ontologie, mais **aussi** une construction logique.

Pour illustrer ceci, prenons la classe **spécimen** d'OntoMic. Nous avons vu à la section 2.1.1.1, que les **spécimen** sont composés de deux entités physiques : une portion de **substance\_biolologique** et un **contenant**.

La définition en langue naturelle de **spécimen** peut être traduite par l'axiome suivant:

**spécimen** ( $\equiv$  ou  $\sqsubseteq$ ) **material\_entity**  
**and has\_part some** **portion\_substance\_biolologique**  
**and has\_part some** **contenant**

Prenons les éléments de l'énoncé de gauche à droite. Le premier mot « **spécimen** » est le nom de la classe à laquelle l'axiome s'applique.

Ensuite viennent les symboles  $\equiv$  ou  $\sqsubseteq$ , ces symboles remplacent les relations *equivalentClass* et *subClassOf* respectivement. Le choix de  $\equiv$  veut dire que **spécimen** est déclaré équivalente à la combinaison logique de droite, tandis que le symbole  $\sqsubseteq$  veut dire que la classe **spécimen** est incluse dans la combinaison logique. Nous verrons à la section 4.5.2.2 l'implication logique du choix de l'un ou l'autre.

À droite des symboles  $\equiv$  ou  $\sqsubseteq$  se trouvent des éléments de la combinaison logique qui définissent une classe virtuelle. Les éléments de la combinaison logique peuvent être des classes ou encore énoncés spéciaux appelés restrictions. Les restrictions génèrent elles aussi une classe virtuelle selon un mécanisme que nous verrons à la section suivante.

La classe virtuelle globale est faite en combinant les éléments de droite par des opérateurs **and** ou **or** qui correspondent respectivement à l'intersection et l'union.

Dans notre exemple, le premier élément de droite est simplement une désignation de classe. Les deux autres sont des exemples de restrictions.

#### 4.5.2.1 Les restrictions

L'élément *has\_part some* *portion\_substance\_biolgique* génère une classe virtuelle. Cette classe est composée de **toutes** les entités de l'ontologie qui sont en relation *has\_part* avec **un ou des individus quelconques** de la classe *portion\_substance\_biolgique*.

La « quantité » d'individus nécessaires pour satisfaire la restriction est « **un ou des individus quelconques** », ce qui est déclaré par l'usage du mot clé **some**, que l'on nomme quantificateur. Il y a d'autres quantificateurs que **some** et nous les expliquerons lors de leur utilisation.

L'élément *has\_part some* *contenant* génère la classe virtuelle composée des entités de l'ontologie qui sont en relation *has\_part* avec un ou des individus quelconques de la classe *contenant*.

Une restriction est donc la combinaison d'une relation, d'un quantificateur et d'une classe. Cette classe peut être une simple désignation comme dans les exemples précédents, mais elle peut être aussi une classe virtuelle résultant d'une combinaison logique. Nous voyons que comme les restrictions peuvent être imbriquées, et ce sur plusieurs niveaux, les éléments peuvent devenir complexes.

Donc la classe virtuelle de notre axiome pour spécimen est celles de toutes les *material\_entity* qui ont sont en relation *has\_part* avec au moins une instance de *portion substance biologique* et en relation *has\_part* avec au moins une instance de *contenant*

#### 4.5.2.2 Les conséquences du choix de $\equiv$ ou $\sqsubseteq$

Le choix de  $\equiv$  ou  $\sqsubseteq$  donne une interprétation différente. Le  $\equiv$  veut dire que dès qu'une entité matérielle a une *portion substance biologique* et un *contenant\_spécimen* attachés par *has\_part* alors c'est un spécimen. Tandis que l'usage de  $\sqsubseteq$  veut dire que tous les spécimens sont des entités matérielles avec les deux parties, mais qu'il peut exister des entités matérielles avec ces parties qui ne sont pas des spécimens.

L'utilisation de l'équivalence de classe  $\equiv$  peut parfois générer des résultats surprenants lors du raisonnement automatique. Par exemple si nous utilisions l'équivalence de classe nous aurions presque le fait qu'un urinoir rempli est un spécimen ! En effet, un urinoir rempli a *has part* une substance biologique et un urinoir est un contenant. C'est seulement parce selon BFO un tout n'est pas partie de lui-même que l'équivalence est bloquée. En effet si contenant avait *has part* contenant alors comme urinoir est un contenant, un urinoir rempli serait un spécimen...

### 4.5.3 BFO et OWL, différences dans le schème de représentation

OWL se veut un langage pour toutes les ontologies, donc OWL tente d'imposer le moins de restrictions possible à la modélisation et assume le moins possible de choses. Par exemple, OWL ne présume pas que les classes sont disjointes, un individu peut être dans deux classes en même temps, ce qui n'est pas possible pour les universal de BFO.

Dans ce qui suit, nous verrons les principales différences entre BFO et OWL.

#### 4.5.3.1 Absence du concept de relations entre classes dans OWL

Pour BFO, les relations peuvent être entre deux individus, entre un individu et une classe et finalement entre deux classes. Pour OWL, les relations sont entre individus seulement<sup>17</sup>, pour représenter les autres types de relations de BFO, il faut passer par des axiomes.

#### 4.5.3.2 Différences entre axiome d'OWL et `defined_class` de BFO

À la section 4.3.2.1.2 nous avons introduit dans notre présentation de BFO la nuance entre classe décrite par une règle et classe peuplée selon une règle. Nous avons alors vu que seules les classes peuplées selon une règle sont de `defined_class`. Il n'y a pas d'équivalents en OWL de ces concepts, une classe a ou n'a pas d'axiomes. De même, il n'y a pas de moyen autre que par convention de conception d'imposer pour les universal l'ascendance unique.

---

<sup>17</sup> La restriction des relations d'individu à individu est vraie sauf pour quelques relations prédéfinies par OWL, par exemple la relation *subClassOf* est entre deux classes.

### 4.5.3.3 Les relations de hiérarchie de classe

Pour BFO, il n'existe qu'un seul marqueur de hiérarchie de classe : la relation *is\_a* et cette relation ne s'applique que pour la hiérarchie des *universal*. Pour BFO, définir une hiérarchie de classes pour les *defined\_class* n'a pas de sens, si hiérarchie il y a, elle doit être déduite par raisonnement.

Pour OWL, n'importe quelle classe peut être déclarée sous-classe d'une autre avec la relation *subClassOf*. Avec OWL une classe gouvernée par un axiome impose cet axiome à toutes ses sous-classes et ceci est exploité dans la conception des ontologies.

### 4.5.3.4 Les individus

Pour BFO, tous les individus sont des *particular* par rapport à des *universal*, on dira qu'un *particular* *instantiates* un *universal* et pour BFO les individus ne font pas partie de l'ontologie, mais de son application. BFO n'a donc que le minimum de relations pour les *particular*, entre autres BFO n'a pas de relations pour déclarer qu'un individu est membre d'un *defined\_class*. En effet pour BFO l'appartenance d'un *particular* à une *defined\_class* doit être déduite et non déclarée.

Pour OWL, il n'y a pas de notion de *particular*, OWL définit des *individual*. Pour OWL, on peut déclarer un *individual* membre de n'importe quelle classe et de plus d'une classe à la fois. Pour OWL déclarer un *individual* comme instance d'une classe avec un axiome impose à l'*individual* les règles de l'axiome.

## 4.6 Conclusion

Nous avons donc une bonne compréhension de la modélisation par une ontologie et nous avons désigné notre ontologie modèle OBI. La découverte d'OBI a amené celle de l'ontologie de haut niveau BFO. Le formalisme de BFO est le fondement de notre travail, il sert de base à la construction de toutes nos structures de modélisation.

Nous avons vu les bases de l'écriture en OWL, en particulier nous avons vu les axiomes. C'est sur ces axiomes que se fera l'inférence logique, concept qui est très important pour OntoMic, car c'est avec l'inférence logique que nous comparons les protocoles.

Maintenant, il nous reste à voir le lien entre les documents et OntoMic, ce sera l'objet du prochain chapitre.

## **Chapitre 5 La gestion informationnelle**

Dans ce chapitre, nous allons présenter une nouvelle manière de voir les documents comme des assemblages d'éléments d'information. Puis nous allons formaliser cette notion d'éléments d'information, pour ensuite expliquer comment on les extrait des documents.

Finalement, nous allons spécifier les étapes du passage d'éléments d'information dans des documents à leur représentation par des éléments d'OntoMic et nous verrons comment utiliser les informations mises dans OntoMic pour traiter nos trois objectifs.

### **5.1 Les documents vus comme des regroupements d'informations**

De nos expériences avec le logiciel maison et les wikis a émergé une nouvelle conception des documents. Un document est vu comme un assemblage d'éléments d'information. Le fait qu'un ensemble d'informations se trouvent regroupées dans un document signifie qu'elles ont au moins une caractéristique commune.

Dans le cas du protocole de culture de selles, les informations contenues dans ce document sont à cet endroit, parce que justement elles partagent toutes la caractéristique de traiter de la culture de selles. Nous pouvons imaginer des protocoles créés selon une autre modalité de regroupement, par exemple, selon l'étape du travail à laquelle elles s'adressent. Nous pourrions ainsi avoir des protocoles de prélèvement, de transport, d'ensemencement qui s'adresse à toutes les cultures.

Notre méthode de gestion des informations a pour but de faire abstraction de la modalité de regroupement que le document représente. Nous délimitons et marquons les informations et nous leur donnons un ensemble de caractéristiques, par la suite nous pouvons créer des documents par regroupement d'informations selon un sous-groupe de caractéristiques qui nous intéresse.

Par exemple, pour faire un guide de prélèvement, nous commençons par marquer les informations qui traitent du prélèvement dans les protocoles d'urines, des gorges, etc.. Puis, nous regroupons les informations sur le prélèvement de tous les protocoles. Nous nous trouvons alors à faire un classement de l'information en premier selon l'étape du travail (le prélèvement) et en second lieu

sur le test (la culture de selles, la culture d'urine...). Le protocole n'est qu'un classement avec comme premier critère le test et puis en second les étapes du travail (le prélèvement, le transport, la réception...)

Le but de la gestion informationnelle est de laisser la gestion des contenants d'informations que sont les documents aux processus de gestion documentaire classique et de travailler avec le contenu des documents selon une procédure en cinq étapes que nous présentons dans ce chapitre.

## 5.2 La notion d'informations et d'ICE

Les éléments de base de notre gestion informationnelle sont les Information Content Entity (ICE), des entités construites pour représenter des informations et qui ont un identificateur unique. Pour qu'une entité soit considérée comme une ICE, il faut qu'elle ait été construite dans le but délibéré de contenir et de faire circuler des informations.

Les ICE ont toutes les caractéristiques suivantes, caractéristiques que nous appellerons facettes:

**Étendue :** caractéristique d'une ICE qui n'a de pertinence que dans une comparaison avec une autre ICE. Un protocole est une ICE de plus grande étendue que l'ICE constituée par la section prélèvement du protocole.

**Nature :** exprime ce qu'est l'ICE : une mesure, une description de tâches.

**Sujet :** décrit l'entité qui est le propos de l'élément d'information. Le poids d'un spécimen est de nature « mesure du poids » et a comme sujet « spécimen ». Un texte qui décrit la tâche d'ensemencement a comme nature « description de tâches » et comme sujet « ensemencement ».

**Production :** pour exister, une ICE doit avoir été produite. L'ICE est qualifiée selon la tâche qui l'a générée. Notons que nous traitons de la tâche de production et non de la notion d'auteur.

**Utilisation :** une ICE pourrait être produite, mais jamais utilisée. Cependant comme l'ICE est délibérément produite alors elle l'est nécessairement dans le but d'être utilisée. Donc, au minimum,

une ICE a une utilisation planifiée. Comme pour la facette production, l'utilisation désigne la tâche qui utilise et non pas la personne ou machine qui utilise.

Il y a toujours un niveau d'abstraction entre une ICE et les entités tangibles, comme entre l'ICE qui représente un texte et le texte imprimé que l'on peut tenir dans les mains. Pour certains types d'ICE, ce niveau d'abstraction est encore plus élevé, par exemple pour les ICE qui servent à encoder les logiques procédurales des protocoles. Pour ce faire, OntoMic utilise des ICE qui n'ont pas d'élément concret correspondant. Ces ICE, bien que pleinement caractérisées avec leurs cinq facettes, n'existent qu'à l'intérieur du monde virtuel créé par l'ontologie.

Ces relations entre les ICE et les entités réelles, tangibles sont assez difficiles à maîtriser. Nous verrons à la section 7.2 comment IAO, l'ontologie partenaire d'OBI, conçoit les ICE et leur relation avec des éléments concrets.

### **5.3 Le processus de gestion informationnelle**

Notre schéma de gestion informationnelle s'ajoute à la gestion documentaire classique ; les notions de versionnage, de processus d'approbation et autres tâches de la gestion documentaire demeurent les mêmes pour les contenants d'informations que sont les documents.

La gestion informationnelle s'intéresse au contenu des documents et s'appuie sur un processus avec les étapes suivantes:

**Repérage :** les informations pertinentes sont retrouvées dans les documents, on les délimite et on leur assigne un identificateur. Ceci nous donne une ICE brute.

**Classement :** on classe les ICE brutes dans notre ontologie selon le « sens » qu'elles représentent. Il n'y a pas de transformation des informations à cette étape. La notion de « sens » d'une ICE brute fera l'objet de la section 5.6.

**Encodage :** les logiques procédurales contenues dans les ICE brutes sont transformées en des logiques interprétables par machine selon le formalisme de notre ontologie. Ces logiques encodées

**Milieus :**

- Gélose-sang
- Mc Conkey
- Hektoen
- Gélose Campy CSM
- McConkey-Sorbitol avec cefixime et tellurite
- CIN

Incuber à 35°C à air ambiant pour 24 heures.

Incuber la gélose Campy CSM dans une jarre en microaérophilie à 37°C pour 72 heures.

Incuber la gélose CIN pour 48 heures à air ambiant à 25°C.

Extrait de document 5-1, listes des géloses avec leur incubation selon le labo SJ

Source SJ\_MIC-PON-072\_V7

ont une représentation bien tangible en des énoncés OWL. Ces énoncés OWL sont nos ICE encodées.

**Entreposage :** les ICE encodées sont mises dans un dépôt

**Utilisation :** on accède aux ICE traduites du dépôt pour les comparer ou les réutiliser

À notre connaissance, cette approche est nouvelle dans le domaine des laboratoires, en particulier il n'y a pas de système de gestion informationnelle en place dans les laboratoires de Québec.

## **5.4 Les étapes de la gestion**

### **5.4.1 Repérage des informations**

À cette première étape, il s'agit de repérer les informations pertinentes dans les documents sources pour créer une ICE qui passera ensuite à l'étape de classement, dès cette étape on donne un identificateur à l'ICE.

Certaines publications (Soldatova et al., 2014) explorent l'usage de méthodes automatisées de repérage et d'extraction des informations du texte d'un protocole, ce serait l'idéal. Pour faire une extraction automatique, il faut une description et une classification des informations, une ontologie

comme OntoMic peut remplir ce rôle. Cependant, programmer une extraction automatique est un travail important, qui viendra seulement dans une étape ultérieure du développement d'OntoMic.

Pour les besoins d'expérimentations avec OntoMic, nous avons procédé à des repérages manuels.

#### **5.4.1.1 Les documents sous forme de texte**

Le repérage se fait au fil de la lecture, pour assurer de la pertinence le lecteur doit bien sûr connaître les objectifs de la codification. Une fois qu'il a repéré une information, le lecteur doit en préciser le début et la fin. À l'Extrait de document 5-1, nous avons un texte qui liste des géloses et précise les conditions d'incubations pour ces géloses. Pour OntoMic, chaque incubation a ses géloses et des conditions, donc dans ce texte nous aurons donc trois incubations, une à 35<sup>0</sup> pour 24 heures, une en microaérophilie (atmosphère à teneur réduite en oxygène) et une à la température de la pièce.

NOM_ANALYSE ▲	CODE ▲	CODE NUM. ▲	REQ. ▲	SERVICE ▼	JEUNE ▲	SPÉCIMEN ▲	REMARQUES ▲	DATE_MODIF ▲
LIQUIDE ABDOMINAL	LBio	916		MICRO		1 ROUGE # 2		2015-02-25
LIQUIDE AMNIOTIQUE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE D'ASCITE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE DE DIALYSE PÉRITONÉALE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE GASTRIQUE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE LYMPHATIQUE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE PÉRICARDIQUE	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE PÉRITONÉAL	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE PLEURAL	LBIO	916		MICRO		1 ROUGE #2		2015-02-25
LIQUIDE SYNOVIAL (GENOU, COUDE, CHEVILLE)	LBIO	916		MICRO		1 ROUGE		2015-02-25

NOM\_ANALYSE    COI    COI    RE    micro    JEUI    SPECIMEN    REMARQUES    DATE\_MOI

Affichage de l'élément 1 à 10 sur 10 éléments (filtré de 1.105 éléments au total)    << < 1 > >>

Extrait de document 5-3, liste des liquides biologiques selon répertoire d'analyses du labo SJ

Source : <http://www.cdsjlabo.org/codes-analyses>

Cette procédure s'adresse aux liquides suivants :

- Liquide synovial
- Liquide pleural
- Liquide péricardique
- Liquide d'ascite ou péritonéal
- Liquide amniotique

Extrait de document 5-2, liste des liquides biologiques selon le protocole du labo SJ

Source MIC-PON-065\_Liquides biologiques

Bien que nous présentions le processus de gestion informationnelle en des étapes séparées, nous pouvons voir avec l'exemple de l'incubation que le lecteur doit connaître la classification pour faire un repérage efficace.

#### 5.4.1.2 Les documents sous forme de données organisées

Idéalement, la tâche de repérage se ferait automatiquement dans un jeu de données organisées, comme la base de données du répertoire d'analyses. Mais il y a une difficulté importante qui survient lorsqu'il faut faire concorder le schéma d'organisation de la source de données avec le schéma de notre ontologie.

Pour donner un exemple de différences dans l'organisation des données, éloignons-nous de la culture de selles, qui a une définition stable, pour nous attarder à la culture de liquide biologique.

À l'Extrait de document 5-3 nous avons les entrées du répertoire d'analyses pour les liquides biologiques, notons que toutes ces entrées ont le même code SIL, donc pour le SIL il s'agit de la même analyse. Du côté des protocoles nous avons le MIC-PON-065\_Liquides biologiques que nous voyons à l'Extrait de document 5-2, dans cet extrait se trouve la liste des substances biologiques auxquelles le protocole s'adresse.

Cinq entrées qui se retrouvent dans le répertoire ne sont pas dans le protocole. Pour quatre d'entre elles, il s'agit d'une différence de nomenclature, mais pour l'entrée : *liquide de dialyse péritonéale*, le protocole SJ\_ MIC-PON-064 lui est spécifique.

OntoMic a adopté comme déterminant le choix de test et les informations spécimens. Pour la culture des liquides biologiques, l'organisation dans OntoMic est différente de l'organisation dans le SIL, dans le répertoire d'analyse et dans les protocoles. Il est difficile d'envisager de faire une correspondance autrement qu'à la main.

Dans d'autres cas, l'organisation de la source est tellement opaque que faire une correspondance automatique entre la source et OntoMic est difficilement envisageable. C'est le cas des tables de paramétrages du SIL. En effet, nous y trouvons des éléments cryptiques comme « !P HU; » comme nous l'avons vu à la section 3.4.3.

## **5.4.2 Le classement des informations**

À cette étape le but n'est pas la transformation des ICE bruts, mais leur classement selon une organisation fournie par OntoMic. Le classement se fait toujours selon la tâche décrite par l'ICE, le classement obéit au paradigme présenté à la section 3.4.6 qui veut que les informations soient toujours rattachées à leur place dans le cheminement de l'analyse.

Pour illustrer ce classement, reprenons les ICE que nous avons rencontrés lors de notre examen des erreurs à la section 3.2.

À l'**Exemple 1** nous avons vu des incohérences selon les sources sur le choix du contenant pour le spécimen. Le choix du contenant est associé à l'étape **prélèvement** du cheminement de

l'analyse, même si l'ICE global qui contient ce choix est une requête associée à l'étape prescription.

En effet, OntoMic est une ontologie qui décrit des gestes réels, comme OBI et BFO. Le geste de choisir le contenant se fait lors du prélèvement, même si les informations sur ce geste peuvent être dans des documents associés à d'autres étapes. Le choix du contenant est donc rattaché à l'étape prélèvement.

À l'**Exemple 2** nous avons vu une ICE qui décrit les géloses qui doivent être ensemencées. Cette ICE est attachée à une sous-étape d'ensemencement qui se nomme `sélection_dispositif_culture`.

À l'**Exemple 3** nous avons vu la lettre du LSPQ qui demande que l'on fasse un ajout à la culture de selles pour les STEC. Il n'est pas évident de décider à quelle étape du processus de culture de selles l'ICE est attachée. OntoMic, s'inspirant d'OBI, définit pour chaque étape un objectif, donc la lettre s'attache à l'étape globale de la culture de selles et qui en décrit l'objectif.

À l'**Exemple 4** nous avons un exemple de fenêtre surgissante qui demande le remplacement d'un test par un autre. Bien que cette ICE apparaisse à l'utilisateur lors de la saisie de l'analyse, cette ICE est rattachée à la `sélection_test` qui est une sous-étape de la tâche prescription.

À l'**Exemple 5** nous avons vu que le protocole donnait la possibilité de faire le test optionnel de coloration de Gram pendant la culture de selles. L'ICE qui stipule cette possibilité est aussi attachée à l'étape `sélection_test`.

### **5.4.3 Encodage des informations**

L'encodage comprend deux étapes, la traduction dans un vocabulaire standard de l'élément et la représentation de sa logique. Nous avons vu un exemple de traduction avec un vocabulaire standard à la section 3.5.3.2 où le fragment de texte « Cary-Blair modifié et un pot stérile » est remplacé par la désignation standard « pot vissé Cary-Blair »

Si la traduction est une étape relativement facile, il en est autrement avec la représentation de la logique. Des nuances subtiles dans la modélisation peuvent empêcher la comparaison automatique.

#### 5.4.3.1 L'encodage et la comparaison automatique

Reprenons l' Exemple 4 sur la fenêtre surgissante sur le remplacement de la recherche de la culture de selles par une recherche de *C. difficile*.

Pour le SIL la règle d'utilisation veut que le message soit déclenché lorsqu'un code test pour une culture de selles est saisi et que le patient est hospitalisé depuis plus de trois jours.

Ce qui est dicté par le protocole est présenté à l'Extrait de document 3-5 où nous pouvons lire la phrase :

« Pour tout spécimen provenant de patients hospitalisés depuis > 4 jours, ne faire que la recherche de Clostridium difficile. Pour toute demande spéciale, consulter le microbiologiste du laboratoire ».

Ceci veut simplement dire de faire le remplacement **après** la réception du spécimen.

Le remplacement dicté par le SIL se fait à la saisie, tandis que selon le protocole le remplacement se fait à la réception et finalement OntoMic attache la règle à `sélection_test`. Les tâches ne sont pas faites aux mêmes moments et ainsi ne sont pas comparables ; la discordance sur les durées d'hospitalisation risque de ne pas être détectée.

La détection automatique des discordances est donc très sensible aux nuances d'encodage, ceci est un indice que cette détection ne sera pas aussi simple que nous l'avions envisagé.

#### 5.4.4 Entreposage

Une fois les éléments d'information repérés, délimités, identifiés, classés et encodés il faut entreposer toutes ces données et pouvoir les rappeler par la suite. Certains modèles, comme le modèle relationnel, comprennent d'emblée leur méthode d'entreposage et leur stratégie de rappel.

Pour le modèle par ontologie, au départ notre approche était bien simple: mettre l'ensemble de nos données dans le fichier texte qui représente l'ontologie, on appelle ce type de fichier sérialisation. Mais tout entreposer dans le fichier de sérialisation est en conflit avec la comparaison automatique à l'aide de l'inférence logique comme nous le verrons à la section 9.4.

## 5.5 Les stratégies d'utilisation des données

Il y a deux stratégies pour s'attaquer à l'objectif de cohérence pour les avoirs informationnels. La stratégie maître - réutilisation consiste à n'avoir qu'une seule source pour chaque information, de ne jamais recopier, mais de réutiliser. C'était l'inspiration derrière notre Expérimentation avec les qui est décrite à la section 3.5.3. Avec cette stratégie, les documents sont des composites créés par des requêtes sur un répertoire de données maître, les documents doivent donc être réécrits.

L'autre stratégie, le dépôt – comparaison, est également basée sur une réserve centrale d'informations. La cohérence est maintenue par la comparaison d'informations de sources diverses, les documents n'ont pas à être changés.

### 5.5.1 La stratégie maître – réutilisation

#### 5.5.1.1 Enjeux avec la stratégie maître – réutilisation

**Impossible à appliquer intégralement :** Cette stratégie est impossible à appliquer intégralement. En effet, certains documents comme le SIL ne sont accessibles qu'en lecture pour des raisons évidentes de stabilité et de sécurité. Nous pourrions allouer au SIL le statut de protocole et considérer qu'il est ainsi une composante du répertoire maître, mais alors il faudra extraire des informations des tables du SIL ce qui n'est pas une simple tâche.

**Documents doivent être réécrits :** De profondes modifications des documents sont à faire. Ils devront être réécrits pour être basés sur des requêtes sur le répertoire.

**Le versionnage est difficile à gérer :** Nous devons décider si le processus de maintenance des avoirs informationnels se fera par des mises à jour directement sur le dépôt ou bien si nous conservons les protocoles comme source maître d'informations. La mise à jour directe du dépôt est

un abandon complet du système en place, ce qui est difficilement envisageable. Si les protocoles demeurent la source maître d'informations alors on se heurtera au problème de conflit de versions que nous avons illustré dans notre approche par wiki à la section 3.5.3.1

**Des comparaisons restent à faire :** La vérification de la conformité implique nécessairement des comparaisons entre les informations du répertoire maître et avec celles de la référence. Il faut donc bâtir un moteur de comparaison.

## **5.5.2 La stratégie dépôt – comparaison**

### **5.5.2.1 Avantages par rapport à maître - réutilisation**

**S'applique à tous les documents :** Nous pouvons implanter la stratégie progressivement, par exemple, ne commencer que par le choix des contenants à spécimens.

**Documents peuvent rester les mêmes :** Aucune modification des documents n'est nécessaire. Le repérage peut être contenu dans un fichier indépendant du document.

**Plus de version à gérer :** Le moteur de comparaison signalera les discordances qui devront être résolues à la main.

**Les documents internes et externes se traitent de la même façon :** Les tâches de repérer, classifier et encoder les informations sont les mêmes peu importe si le document fait partie des avoirs locaux où s'il s'agit d'un document d'un autre laboratoire ou un document normatif.

### **5.5.2.2 Enjeux avec dépôt – comparaison**

**Sensible aux différences de classification et d'encodage :** Avec la stratégie maître – réutilisation, il faut bâtir des requêtes sur le dépôt. Si l'information n'est pas classée ou encore si elle est encodée d'une façon inattendue, la requête n'aura et l'erreur dans le dépôt deviendra évidente.

Mais avec la stratégie dépôt – comparaison, si deux informations ne sont pas encodées de la même manière elles risquent de ne pas être retracées et de ne pas être comparées sans que cela ne soit facilement détectable.

**Plus de ressources humaines à long terme :** Repérer, classifier et encoder une information est très semblable au travail qu'il faut faire pour bâtir une requête. Avec la stratégie dépôt – comparaison, lorsqu'une information est modifiée, le moteur de comparaison signalera la discordance, mais ne pourra pas la résoudre, tandis qu'avec la stratégie maître – réutilisation nous pouvons automatiser les mises à jour.

### 5.5.3 Amalgame des deux stratégies

Nous avons présenté les deux stratégies comme des opposés. Mais en réalité, rien n'empêche que pour un document l'information sur le contenant à spécimen soit peuplée par une requête sur le dépôt, tandis que dans un autre, elle soit laissée native au document, mais repérée et placée dans le dépôt.

## 5.6 Sens, sémantique et logique procédurale.

Nous avons vu à la section 5.2 que les ICE avaient un degré d'abstraction par rapport à une matérialisation d'une information. Ce degré d'abstraction permet, par exemple, de décrire le roman « Vingt mille lieues sous les mers » comme un individu unique même si ce roman de Jules Verne existe en de multiples exemplaires.

Mais pour OntoMic ce degré d'abstraction entre une ICE et un support physique a été insuffisant pour gérer la notion de « sens » d'une ICE. Nous avons nommé sémantique le « sens » d'une ICE.

Pour OntoMic les sémantiques sont des absolus théoriques auxquels on accède que par des ICE comme des textes, des diagrammes, des vidéos, etc. Ces ICE **décrivent** les sémantiques, mais conceptuellement aucune ICE n'est assez « riche » pour **déterminer** une sémantique, contrairement à une édition texte intégral de « Vingt mille lieues sous les mers » qui détermine le roman. Les ICE sont toujours des approximations de la sémantique qu'ils décrivent.

OntoMic s'intéresse particulièrement à une sous-classe particulière de sémantique que nous nommons logique procédurale et qui est définie, informellement, comme le « comment faire ».

OntoMic a le but de transformer les logiques procédurales contenues dans les documents en un langage machine pour rendre ces logiques procédurales traitables automatiquement.

Ce recours à des structures abstraites pour les sémantiques et logiques procédurales peut paraître inutilement complexe, mais en réalité il simplifie la modélisation. En effet sans les sémantiques, il est difficile de représenter la dualité d'un texte comme à la fois une entité lexicale et une entité porteuse de sens. Il est aussi beaucoup plus difficile de représenter le fait que deux textes peuvent avoir le même sens ou comment un seul texte peut être interprété de plus d'une façon et donc avoir plus d'une sémantique.

Nous avons un exemple de deux textes qui ont le même sens avec l'Extrait de document 6-1 où nous pouvons lire la phrase « Lire la gélose à 24 heures ». Une phrase plus complète serait « Lire la gélose après une incubation de 24 heures dans des conditions standards ». Si nous considérons l'omission dans la première phrase comme un simple sous-entendu, sans ambiguïté, alors le texte avec la phrase modifiée désignera la même sémantique, la même logique procédurale.

## **5.7 Conclusion**

Ce chapitre nous a donné les définitions et les descriptions des concepts clés qui permettent de bien encadrer le passage d'un document en une série d'axiomes écrits en OWL dans OntoMic et l'utilisation de ces informations encodées en axiomes pour atteindre nos trois objectifs.

Mais le processus de transformation des logiques procédurales en des énoncées OWL est difficile à maîtriser, comme plusieurs concepts clés d'OntoMic. Pour bien le comprendre, nous allons consacrer le chapitre suivant à une analogie avec des formulaires pour expliquer comment OntoMic fonctionne.

## Chapitre 6 Analogie avec des formulaires

Pour cette analogie, nous créons un laboratoire fictif, dans ce qui suit vous verrez que ce laboratoire est bien étrange... Néanmoins, cette analogie reflète le fonctionnement de la gestion informationnelle et d'OntoMic. En fait, lorsqu'un élément du flot de la gestion informationnelle ou de la modélisation en OntoMic est difficile à cerner, il nous a souvent été utile de revenir à cette analogie et de voir l'élément dans le contexte de notre laboratoire fictif.

Ce laboratoire n'a pas de SIL, il fonctionne uniquement en mode papier. Ce laboratoire a accès à des protocoles, mais ceux-ci proviennent d'un organisme extérieur et sont dans une langue étrangère. Comme il n'est pas à la portée de tous de comprendre ces protocoles, des formulaires papier sont créés à partir des protocoles. Les formulaires sont faits à l'aide d'un traitement de texte à partir d'une banque de patrons de formulaires.

La création des formulaires est bien ardue, avant de consacrer des ressources à la création de formulaires pour un protocole, la direction a au moins besoin d'en connaître le propos. Un employé donne ces informations à la direction simplement en annotant des passages du protocole avec des étiquettes collantes, ces étiquettes sont faites avec des éléments des patrons de formulaires.

La direction de ce laboratoire affirme que comprendre ces formulaires est aussi complet que comprendre les protocoles. De plus, comme elle est responsable de la qualité, la direction a besoin d'une traçabilité du processus de création des formulaires. C'est pourquoi, sur chaque élément du formulaire, on appose une étiquette qui indique le passage du protocole qui est la source de l'élément.

Finalement, la direction tient à ce que ses formulaires soient en conformité avec les protocoles, mais aussi avec d'autres normes, provenant d'autres sources. Le moyen qu'elle a trouvé pour y arriver est de créer des formulaires pour chacune des sources et de les comparer. Comme les formulaires sont tous créés à partir de la même banque de patrons, la tâche est simplifiée.

1) Gélose-sang :

Lire la gélose à 24 heures.

Noter sur la présence ou l'absence de la flore fécale.

Pour le Staph.aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure.

Extrait de document 6-1, lecture de la gélose-sang selon le labo SJ

Source : de SJ\_MIC-PON-065-V7

Dans ce qui suit, nous allons voir la création d'un de ces formulaires pour notre laboratoire fictif, la source du formulaire est toujours notre protocole de culture de selles et celui-ci est en français, un petit accroc à notre analogie. Par la suite, nous allons montrer les patrons de formulaire et indiquer pourquoi ils sont essentiels. Pour finalement discuter de la comparaison manuelle des protocoles et voir quels sont les éléments essentiels au passage à une comparaison automatique.

## 6.1 Un formulaire pour la lecture d'une gélose après incubation

Un formulaire a son champ d'application, c'est-à-dire une description des situations dans lesquelles le formulaire doit être utilisé. Un formulaire comporte plusieurs éléments; des cases pour la saisie; des bulles d'informations; des règles expliquées par des textes; des cases pour des calculs, des cases pour recopier des données provenant d'autres formulaires, des renvois à d'autres formulaires et finalement il y a des indications sur quelles informations doivent être obligatoirement présentes pour que le formulaire soit considéré comme complété.

Illustrons ceci avec l'Extrait de document 6-1 de notre protocole de cultures de selles qui explique comment faire la lecture de la gélose-sang après incubation. Le formulaire illustré à la Figure 6-1 encode la logique procédurale de l'extrait.

Les descriptions des éléments du formulaire en partant de sa partie supérieure gauche en progressant vers le coin inférieur droit.

Le titre de notre formulaire indique son champ d'application: « lecture de gélose incubée », tâche qui s'appliquant au dispositif « gélose-sang ».

Lecture des géloses incubée: gélose-sang

Gélose-sang incubée 24 hres  Oui  Non

Flore fécale  Présence  Absence

Staph auréus  forte prédominance - culture pure  non prédominant - absence

Compte rendu

Dépistage résistance oxacilline par gélose SJ

Noter sur la gélose présence ou absence flore fécale

Pour le Staph auréus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou si culture pure.

1) Gélose-sang : Lire la gélose à 24 heures. Noter sur la présence ou l'absence de la flore fécale. Pour le Staph.aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure.

Figure 6-1, exemple de formulaire pour la lecture de la gélose-sang

La phrase « Lire la gélose à 24 heures. » est encodée par un identifiant sur le formulaire « Gélose-sang incubée 24 h » qui est attaché à une case à cocher oui/non. Pour encoder le fait que l'incubation doit être reprise si elle est de moins de 24 heures le formulaire indique par une flèche et un symbole d'arrêt que la lecture de la gélose-sang doit être interrompue.

Par la suite, nous trouvons une case à cocher avec comme identifiant « Flore fécale » et avec les choix absence/présence pour représenter la phrase du protocole « Noter sur la gélose la présence ou absence de flore fécale ». Pour faire le lien entre le formulaire et le protocole, la phrase est mise dans une bulle attachée à l'élément du formulaire qui l'encode.

La case à cocher suivante encode la phrase « Pour le *Staphylocoque auréus*, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure », si la case avec le libellé « forte prédominance – culture pure » est choisie alors la flèche indique que l'on doit remplir

un autre formulaire « Dépistage résistance oxacilline par gélose ». Également, la phrase est mise dans une bulle attachée à son élément d'encodage.

Notre formulaire comprend une case de texte pour le compte-rendu, car, sans que ceci soit explicitement mentionné dans le l'extrait, nous savons que chaque action faite pendant une culture doit se voir reflétée par une saisie dans le compte-rendu.

La phrase « Pour le *Staphylocoque auréus*, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure » a été encodée avec le niveau de complexité le plus élevé. Avec un niveau de complexité plus bas, nous aurions pu simplement faire un formulaire avec bulle d'information qui contiendrait la phrase, sans case pour une saisie.

Ce choix de niveau de complexité est très important, tous les aspects du protocole qui seront représentés seulement par des bulles échapperont d'emblée à un traitement automatique, contrairement à ceux traités par case à cocher ou par appel à un autre formulaire.

## **6.2 La nécessaire création de formulaires patrons**

Le formulaire à la Figure 6-1 décrit bien le texte de l'Extrait de document 6-1, mais il a été fait sans l'aide d'un patron. Il sera difficilement comparable à d'autres formulaires faits eux aussi sans patron, car les éléments à comparer, les étiquettes d'information, les cases à cocher, etc. ... pourraient ne pas se retrouver au même endroit.

Il nous faut donc faire un patron de formulaire, pour ce faire il faut bien cerner la tâche. Dans notre analyse de la tâche il nous faut déjà faire une précision sur le formulaire de la Figure 6-1. En effet, ce formulaire se nomme « lecture de gélose incubée », alors que notre analyse de la tâche a montré que celle-ci doit être séparée en une sous-tâche repérage\_colonie\_d'intérêt et une ou plusieurs autres sous-tâches travail\_isolat\_d'intérêt.

### **Le patron de formulaire pour la lecture des géloses**

Pour comprendre comment faire le patron de formulaire, il nous faut revenir à notre description du travail fait en microbiologie. À la section 2.1.1.2, nous avons vu que le but de la culture était de

faire proliférer les organismes susceptibles d'être pathogènes pour les rendre détectables ; ces pathogènes potentiels sont nommés organismes d'intérêts. Sur les milieux solides, on repère les organismes d'intérêts selon la morphologie macroscopique des amas d'organismes qui découlent de leur prolifération. On nomme ces amas colonies et nous avons donc des colonies d'intérêts.

Voyons comment faire un patron de formulaire pour la sous-étape de repérage.

## **6.2.1 L'étape de repérage des colonies d'intérêts**

### **6.2.1.1 Analyse de la tâche de repérage**

À cette étape, la technicienne recherche les organismes d'intérêts sur la gélose, elle a donc besoin des descriptions des colonies d'intérêts produites par ces organismes. Puis les différents types de colonies d'intérêts sont regroupés selon les similitudes de leurs apparences<sup>18</sup>. Ces regroupements morphologiques s'appelle isolats d'intérêts.

---

<sup>18</sup> En microbiologie il y a un principe fondamental qui stipule que les colonies qui partagent la même apparence sont des clones du même organisme.

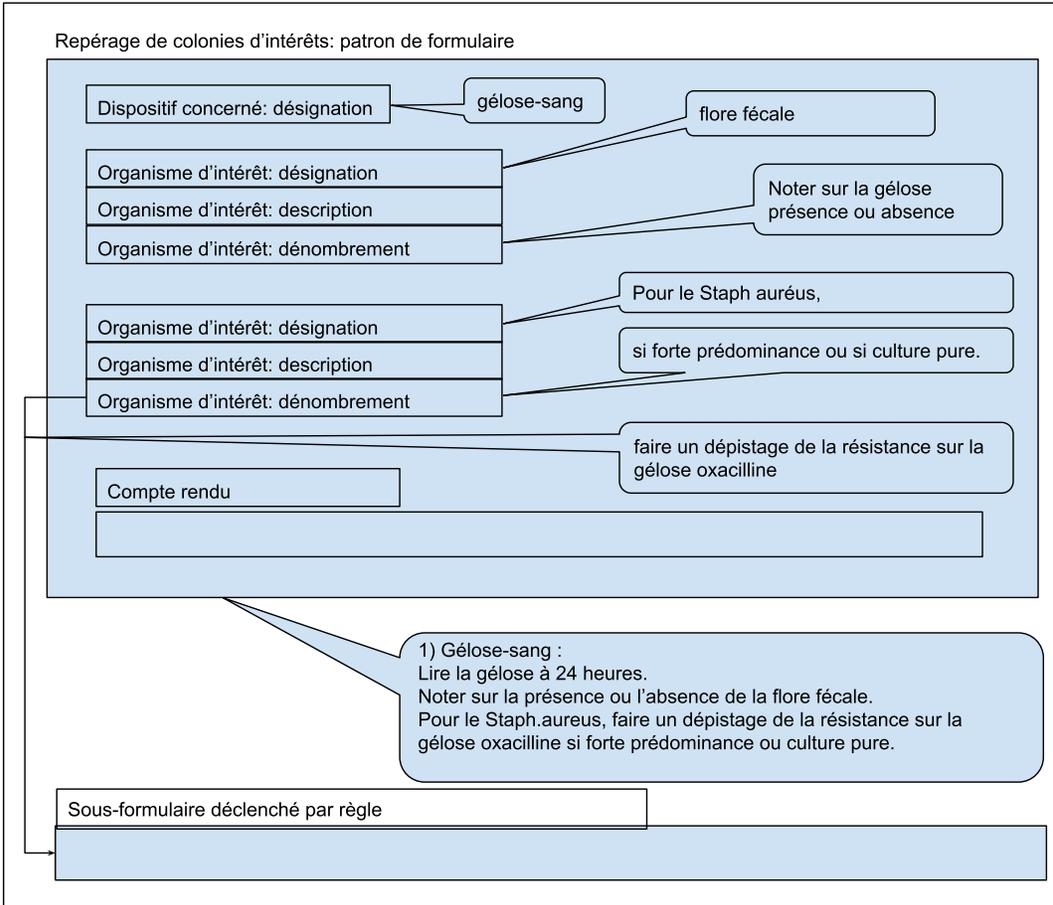


Figure 6-2, exemple de patron de formulaire adaptable pour toute lecture de gélose

Notons que, dans la même culture, il peut y avoir plus d'un isolat d'intérêt pour un même type de colonie d'intérêt. Par exemple, bien que les *S. aureus* partagent tous la caractéristique de faire des colonies jaunâtres, l'intensité de cette coloration varie d'un clone de *S. aureus* à l'autre.

Pour chacun des isolats d'intérêts, il y a un décompte, qui peut être très sommaire, comme dans notre cas où la flore fécale est quantifiée par « absence » ou « présence ».

Par la suite, il y aura des règles de décision sur ces isolats d'intérêts afin de savoir si l'on doit aller plus loin et envoyer l'isolat à une étape ultérieure d'identification et de caractérisation.

Notre protocole de selles applique une règle sur le décompte des colonies de *S. aureus* pour décider si un isolat d'intérêt de *S. aureus* doit passer à l'étape suivante. Les règles basées sur le nombre de colonies sont fréquentes, d'autres règles se basent sur des tests rapides à faire sur les colonies.

Comme les règles basées sur décompte ne demandent qu'un examen de la gélose, elles sont appliquées à l'étape de repérage. L'exécution des tests rapides se retrouve à l'étape travail\_isolat\_-d'intérêt.

D'autres types de règles peuvent s'appliquer à l'étape de repérage. Ces règles sont celles qui dépendent du contexte extérieur à l'examen de la gélose, comme l'âge du patient ou le résultat d'autres cultures pour le même patient. Nous verrons des exemples de règles basées sur le contexte à la section 6.6.3.

### **6.2.1.2 L'encodage du repérage dans un patron de formulaire**

La Figure 6-2 illustre le patron de formulaire pour la sous-tâche repérage\_colonie\_d'intérêt. Ce patron est conçu pour être applicable à la lecture de toute gélose, il comprend un champ pour désigner la gélose visée lors de son adaptation.

Il comprend des blocs d'informations pour chacune des colonies\_d'intérêt, dans le diagramme nous avons illustré trois de ces blocs. Un bloc doit comprendre une désignation de la colonies\_d'intérêt, une description de cette dernière et finalement un dénombrement.

Notons que notre diagramme de patron ne comprend pas de case à cocher pour le résultat du dénombrement. Pour ces cases, il faudra d'autres patrons de formulaire, car il y a plusieurs façons de faire un dénombrement. Le dénombrement de notre exemple est basé sur l'abondance relative de la colonies\_d'intérêt par rapport aux autres colonies. D'autres échelles ne sont pas relatives comme « rare », « abondant ». Chacune des échelles se traduira par un patron de case à cocher.

Nous n'avons pas inscrit au diagramme la possibilité de règles autres que sur le décompte et ceci pour la même raison que pour les cases à cocher des décomptes. En effet, les règles peuvent porter sur chacun des éléments d'une culture et nous n'écrivons pas une règle basée sur l'âge du patient de la même manière qu'une règle sur la saison en cours lorsque le spécimen est recueilli.

Il faudra donc des patrons de formulaires pour chaque type de décompte et chaque type de règle. Pour faire un formulaire spécifique pour un protocole, il faut adapter le formulaire patron et aussi faire un assemblage de patrons de sous-formulaires pour compléter certains éléments.

## **6.2.2 Le travail sur les colonies d'intérêts**

Une fois qu'un regroupement de colonies (un isolat\_d'intérêt) a été repéré, il faut des épreuves supplémentaires pour s'assurer qu'il s'agit d'un organisme véritablement pertinent. Ces épreuves supplémentaires sont séparées en deux groupes, celles qui se font en temps réel pendant l'examen du dispositif de culture et les autres qui donnent un résultat en différé. Le travail\_colonie\_d'intérêt comprend toutes les épreuves qui se font en temps réel.

Notre exemple de lecture de la gélose-sang ne comprend pas de travail\_colonie\_d'intérêt. Le protocole demande en effet une détection de la résistance à l'oxacilline, qui nécessite une incubation jusqu'au lendemain.

## **6.3 Le patron de tâche adapté pour la lecture de la gélose-sang**

Au lieu de faire directement un formulaire à partir de l'Extrait de document 6-1, comme nous l'avons fait à la Figure 6-1, nous allons maintenant prendre notre patron de formulaire de la Figure 6-2 et le transformer pour notre extrait de protocole. Dans cette application du patron de formulaire, nous rencontrons des difficultés que nous allons voir dans ce qui suit.

### **6.3.1 Omission de la description des colonies suspectes d'être du *S. auréus***

La première difficulté est que dans le texte de l'Extrait de document 6-1, la description d'une colonie susceptible d'être un *S. auréus* est manquante alors que le patron en exige une. Le *S. auréus* est si fréquent en microbiologie qu'il est attendu que toutes les techniciennes sachent comment le repérer. Des recours à des savoirs assumés connus sont très fréquents dans les protocoles. Nous pouvons traiter le sous-entendu en laissant vide l'étiquette consacrée à description de la colonie dans le formulaire.

### **6.3.2 Le besoin d'une confirmation de l'identification du *S. auréus***

Il y a une autre omission dans le protocole : le besoin de recourir à une épreuve d'identification formelle du *S. auréus*. En effet, la suspicion sur la base de l'apparence des colonies est insuffisante.

Un test simple, en temps réel, est utilisé, il est fort probable que, dans notre cas, il y ait une omission dans le protocole.

Nous pouvons modéliser notre protocole de différentes façons. Une méthode suppose un repérage parfait, dans ce cas on considère que la confirmation n'est pas nécessaire et que l'apparence de colonie seule est suffisante pour l'identification. Une autre méthode suppose que l'identification visuelle peut être faillible et demande un processus de confirmation sans le définir. Cette seconde méthode nous semble plus adéquate, car elle met en évidence l'omission, par contre elle nécessite un encodeur expérimenté qui saura détecter l'omission.

La Figure 6-2 montre le patron de formulaire pour la sous-tâche repérage des colonies\_d'intérêt de la lecture de la gélose-sang. Pour cet exemple, nous avons supposé un repérage parfait.

## **6.4 Les leçons sur l'encodage tirées de l'analogie par formulaire**

### **6.4.1 Informations sous-entendues et véritables omissions**

Les deux omissions dans notre protocole exemple n'ont pas la même importance. Pour la première, il s'agit simplement de laisser un élément du patron de formulaire vide, cela n'a pas de conséquences sur la modélisation. La seconde omission pourrait entraîner une fausse modélisation sans la vigilance de l'encodeur.

### **6.4.2 Les frontières des tâches**

Nous avons défini travail\_colonies\_d'intérêt comme toutes tâches se faisant en temps réel sur un isolat d'intérêt pendant la lecture de la gélose. Dans les faits, la lecture des géloses associées à un spécimen est un processus non linéaire. Il y a souvent des retours sur une gélose déjà examinée à la lumière d'informations obtenues par l'examen d'une autre gélose ou par la disponibilité du résultat d'un test, certes rapide, mais non instantané.

Il est important que, durant le déroulement d'une culture, on puisse se retrouver dans les formulaires et séparer ceux qui sont complétés et de ceux qui sont en suspens. Pour ce faire, il nous

1) Gélose-sang :

Organisme d'intérêt : désignation

Lire la gélose à 24 heures.  
 Noter sur la présence ou l'absence de la flore fécale.  
 Pour le Staph.aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure.

Organisme d'intérêt : désignation

Extrait de document 6-2, texte sur la lecture de la gélose-sang transformé en formulaire

Source : SJ\_MIC-PON-072\_V7

faut définir les frontières des tâches. Nous avons choisi d'imposer le principe qu'une tâche ne peut se terminer avant que toutes les **sous-tâches** qu'elle a engendrées ne soient elles-mêmes terminées.

Dans notre cas, il faudra définir une tâche englobante pour les géloses qui sont lues en même temps<sup>19</sup>, cette tâche se terminera lorsque les lectures des géloses individuelles seront terminées.

## 6.5 Classement sémantique des extraits de textes avec les patrons de formulaire

Nous nous rappelons le besoin de la direction de notre laboratoire fictif de connaître les grandes lignes des protocoles, qui sont écrits dans une langue étrangère. Pour ce faire, un employé prépare une version annotée avec des étiquettes provenant des patrons de formulaire.

---

<sup>19</sup>Certains organismes fragiles comme le Campylobacter ne peuvent supporter l'atmosphère ambiante plus que quelques dizaines de minutes. Les géloses dédiées au Campylobacter sont lues en lots, à part des autres, pour minimiser la durée d'exposition.

Nous avons un exemple d'annotation avec l'Extrait de document 6-2. Nous avons attaché à l'élément de texte « flore fécale » une étiquette « Organisme d'intérêt : désignation » qui provient du patron de formulaire. Maintenant, nous savons que « flore fécale » désigne un groupe de bactéries que le protocole a jugé pertinent de repérer. Nous avons appliqué le même étiquetage à la mention « *Staph. auréus* ».

Supposons que nous voulons une liste des protocoles de culture pour lesquels la présence de *S. auréus* est jugée pertinente<sup>20</sup>. Il est possible, avec un bon classement, de rechercher manuellement les protocoles qui ont un élément de texte marqué par l'étiquette « Organisme d'intérêt : désignation ». Puis, parmi ceux-ci, sélectionner ceux pour qui l'étiquette est attachée à du *S. auréus*.

Mais en admettant que la structure des formulaires soit accessible électroniquement, il serait beaucoup plus rapide de faire la recherche dans les formulaires.

## 6.6 Les recherches et comparaisons avec les formulaires

Nous avons vu comment nous pouvions rechercher des informations en passant par des annotations ajoutées aux protocoles. Mais la direction de notre laboratoire fictif a exigé que dans chaque formulaire on garde les liens entre les extraits de textes sources et les éléments de formulaire. Les traces de ces liens sont conservées dans des étiquettes provenant du texte source qui sont attachées aux éléments du formulaire.

Nous pourrions prendre ce chemin pour faire des recherches. Et en plus comme toutes les informations analogues des protocoles sont encodées de la même façon, ne pourrions-nous pas automatiser les comparaisons ?

Bien sûr, pour automatiser les comparaisons la direction de notre laboratoire devra investir dans un logiciel plus élaboré qu'un simple traitement de texte pour les formulaires, mais aussi des pratiques d'encodage devront être mises en place comme nous allons le voir dans ce qui suit :

---

<sup>20</sup> Pour certaines cultures, comme celle de la gorge, on ne tient pas compte du *S. auréus*.

### 6.6.1 Informations encodées et non en étiquettes de textes

Pour que les informations du formulaire électronique puissent servir à des comparaisons automatiques, il faut qu'elles soient encodées, et non laissées dans des étiquettes de textes. Prenons l'exemple du passage sur le *S. aureus* du protocole simplement mis dans une étiquette de texte dans le formulaire comme à la Figure 6-3 par opposition à un encodage comme à la Figure 6-4.

Dans le premier cas, le seul moyen d'avoir accès électroniquement aux informations serait d'utiliser des techniques de traitement des langues naturelles pour examiner le texte dans l'étiquette. Dans le second cas, nous pouvons accéder électroniquement au fait que la règle est en rapport avec le *S. aureus*, car c'est le vocable qui est attaché à la boîte de cases à cocher. Nous pouvons accéder électroniquement à la description de la flèche qui pointe vers le formulaire dédié à la recherche de résistance et savoir que son origine est la case à cocher marquée « forte prédominance – culture pure ».

### Lecture gélose incubée : gélose-sang

Gélose-sang incubée 24 hres

Oui  
 Non

Flore fécale

Présence  
 Absence

Noter sur la gélose présence ou absence flore fécale

Pour le Staph auréus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou si culture pure.

A red hexagon symbol is present to the right of the 'Oui/Non' checkboxes, with a red arrow pointing from the 'Non' checkbox to it.

Figure 6-3, formulaire sans encodage des actions à prendre avec le *S.aureus*

### Lecture gélose incubée : gélose-sang

Gélose-sang incubée 24 hres

Oui  
 Non

Flore fécale

Présence  
 Absence

Noter sur la gélose présence ou absence flore fécale

Staph auréus

forte prédominance - culture pure  
 non prédominant - absence

Pour le Staph auréus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou si culture pure.

Dépistage résistance oxacilline par gélose SJ

A red hexagon symbol is present to the right of the 'Oui/Non' checkboxes, with a red arrow pointing from the 'Non' checkbox to it. A red arrow also points from the 'forte prédominance - culture pure' checkbox to the 'Dépistage résistance oxacilline par gélose SJ' section.

Figure 6-4, formulaire avec encodage des actions à prendre avec le *S.aureus*

## 6.6.2 Encodage et vocabulaire standardisé

Nous avons vu que pour rendre le protocole intelligible pour la direction, un employé avait annoté les mots « flore fécale » dans le protocole avec l'étiquette « Organisme d'intérêt : désignation ».

Quand le formulaire a été fait à partir du protocole, on a fait l'inverse, on a donné au champ « Organisme d'intérêt : désignation » du protocole l'étiquette « flore fécale » qui provient du texte du protocole. Dans notre cas aucune étiquette n'a pu être mise sur « Organisme d'intérêt : description », car cette description est manquante dans le protocole.

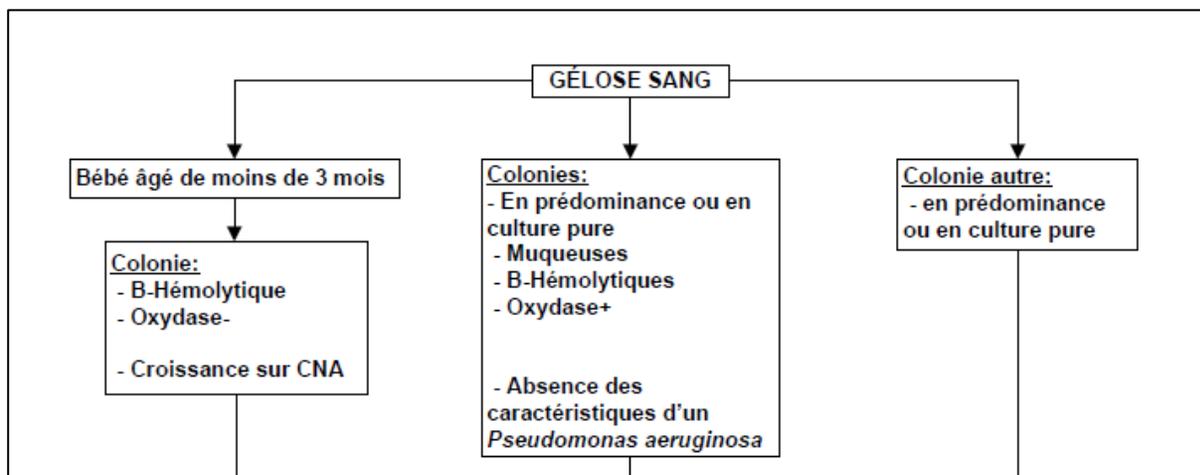
Supposons que nous voulons savoir comment les différents protocoles traitent la flore fécale. Il faut faire une recherche non seulement sur les mots « flore fécale », mais aussi sur les synonymes « flore entérique », « flore stercorale », « flore digestive basse ».

La recherche sera grandement simplifiée si, au moment de la création du formulaire, on avait attaché à l'élément « organisme d'intérêt : désignation » une étiquette avec la désignation standardisée « colonie typique de la flore fécale ». À ce moment, les recherches peuvent se faire sans avoir à tenir compte des synonymes.

### **6.6.3 Les patrons de formulaires sont insuffisants**

Malheureusement, des protocoles similaires, correctement encodés avec les patrons de formulaires, peuvent être différents dans leur implantation de la logique de flot. Prenons un exemple issu du protocole de la Cité de la Santé, ce protocole demande une recherche du Streptocoque du groupe B dans les selles de nourrisson de trois mois ou moins. Cette recherche n'est pas exigée par les organismes normatifs et n'est pas en place à Saint-Jérôme.

À l'Extrait de document 6-3, nous avons le diagramme qui indique comment faire la lecture de la gélose-sang selon le protocole de la Cité de la Santé. Notons au passage un autre exemple d'information manquante ; en effet, il n'y a pas de critère sur la quantification de l'isolat suspect. L'information sous-entendue est que la présence d'une seule colonie est pertinente.



Extrait de document 6-3, lecture de la gélose-sang selon le labo CSL

Source : CSL\_PON-MIC-TSP\_SELLE\_V7

Par contre, dans ce cas, le protocole fournit une description des colonies à suspecter. Soit :  $\beta$ -hémolytique, oxydase négative et croissance sur la gélose Columbia Nalidixic Acid Agar (CNA), cette gélose inhibe la croissance des bactéries autres que le *Streptocoque* du gr B.

Ce protocole de lecture de la gélose-sang nous donne deux exemples de règles pour décider si un isolat\_d'intérêt est pertinent. Deux règles qui sont basées sur le contexte extérieur, soit une règle sur l'âge du patient et l'autre sur le résultat de la lecture de la gélose CNA.

Nous pourrions créer un formulaire en ajoutant comme organisme d'intérêt le *Streptocoque* gr B chez les nourrissons avec une case à cocher pour la présence ou non de l'organisme et une autre case pour l'âge de 3 mois ou moins. Une comparaison, qui pourrait être automatique, de la structure des formulaires permettrait de déceler l'absence de mention du *Streptocoque* gr B dans le formulaire de Saint-Jérôme donc dans le protocole qui en est à l'origine.

Mais imaginons qu'au lieu de ceci nous ayons décidé de faire deux formulaires de lecture de gélose-sang différents selon l'âge du patient. La recherche de *Streptocoque* du groupe B serait encodée seulement dans le formulaire pour les 3 mois ou moins.

Le critère d'âge dans le premier cas s'applique sur une organisme d'intérêt, tandis que le second cas c'est au formulaire entier qu'il s'applique. Il n'est pas simple de déduire automatiquement que les deux constructions sous-tendent la même logique.

## 6.7 L'analogie avec formulaire et le formalisme OntoMic

Nous avons montré par un exemple qu'une approche par formulaires peut représenter la logique procédurale d'un protocole. Autant la création de formulaires peut être souple, autant elle doit être rigide si nous voulons utiliser les formulaires pour les recherches et comparaisons, d'où la nécessité de patrons de formulaires. Si nous voulons automatiser les comparaisons, il faut aller plus loin et utiliser les patrons de formulaires de façon uniforme,

OntoMic peut être vue comme une extension de l'approche par formulaire, en effet OntoMic peut encoder tout ce que l'on peut mettre dans un formulaire et faire des patrons.

En plus, OntoMic peut faire de l'inférence logique. Nous pouvons espérer que l'inférence logique permettra de satisfaire nos objectifs sans passer par une représentation aussi rigide que celle qui est nécessaire avec des formulaires. Par exemple, nous aimerions qu'OntoMic puisse considérer comme équivalentes les deux représentations de l'exigence sur le *Streptocoque* gr B vues à la section 6.6.3.

# Chapitre 7 Les ontologies modèles et partenaires d'OntoMic

## 7.1 Le rôle des ontologies BFO, OBI et OntoMic

### 7.1.1 Le rôle comparé d'OBI et d'OntoMic

OBI, l'ontologie modèle d'OntoMic est surtout utilisée pour annoter des données expérimentales. OBI comprend des classes pour décrire quelle caractéristique de quelle entité a été mesurée, par quelle méthode la mesure a été faite et avec quelle unité de mesure. Si nous exprimons ceci dans le langage de notre gestion informationnelle introduit à la section 5.2, nous dirons qu'OBI décrit la nature d'une mesure (ce qui est mesuré), le sujet (l'entité sur laquelle la caractéristique est mesurée) et sa production (méthode de mesure). Cependant pour décrire complètement la donnée, il faut une description plus précise de sa production que la seule méthode de mesure.

Prenons par exemple la classe d'OBI « *assay detecting IFN-gamma production* » qui décrit les dosages d'interféron. Avec OBI, on caractérise une valeur numérique dans une base de données en lui attachant une étiquette qui mentionne qu'il s'agit d'une valeur en sortie d'un « *assay detecting IFN-gamma production* » et une autre étiquette qui indique les unités de mesure.

Mais supposons que nous ayons une expérience qui mesure l'augmentation de la concentration d'IFN après une stimulation antigénique, il y aura dans la base de données des mesures avant et après stimulation. Dans cette base, on trouvera vraisemblablement des noms de champs comme « valeur de dosage d'interféron préstimulation » et « valeur de dosage d'interféron post-stimulation ». OBI marquera ces champs comme étant en sortie d'un processus « *assay detecting IFN-gamma production* », ce qui indique bien la nature du champ, mais ne précise pas si le dosage est avant ou après la stimulation.

Pour préciser plus loin les champs l'utilisateur d'OBI va créer deux spécialisations de « *assay detecting IFN-gamma production* », ces spécialisations pourraient être appelées : « *assay préstimulation* » et « *assay post-stimulation* ». OBI ajoutera à « *assay préstimulation* » l'exigence de survenir **avant** un processus de stimulation immunitaire « *epitope specific IFN-g release by T cells* » et au « *assay post-stimulation* » de survenir **après** ce même processus. En marquant les

champs selon qu'ils sont en sortie d'un processus pré ou post stimulation, on peut alors faire une interprétation machine des champs.

Donc, pour donner un sens aux données produites lors d'une expérience, OBI assigne des étiquettes aux dosages et en plus une représentation du cadre dans lequel les dosages ont été faits. OBI décrit donc les étapes des expériences, avec leur enchaînement, ce qui est tout à fait adapté aux besoins d'OntoMic.

OntoMic utilise ces structures descriptives analogues à celle d'OBI, mais pas pour annoter des résultats comme OBI, mais pour rendre compte des textes qui sont la source de ces structures descriptives.

Mais même si OntoMic et OBI partagent le besoin de décrire les processus, OntoMic a besoin de structures plus élaborées que celles offertes par OBI. Des structures pour décrire des enchaînements de tâches par exemple. Nous verrons ceci en détail au chapitre suivant.

## 7.2 IAO

L'ontologie IAO est utilisée autant par OBI que par OntoMic pour toutes les informations qui ont été intentionnellement créées, ces informations sont nommées ICE (Information Content Entity) par IAO. La création de IAO a été jugée nécessaire par les concepteurs d'OBI pour décrire des entités qui n'étaient pas dans le champ d'application d'OBI, par exemple les réglementations. Notre utilisation d'IAO pour OntoMic sera conforme à sa définition et concernera des informations intentionnellement produites.

Pour notre étude d'IAO nous nous attarderons particulièrement à deux types d'ICE qui sont très importants pour OntoMic, les `plan_specification` et les `measurement_datum`. Dans ce qui suit, nous allons introduire IAO par le biais des `plan_specification` qui sont utilisés pour les étapes repérage et classements des informations de la gestion informationnelle. Les `measurement_datum` sont utilisés dans l'étape d'encodage et seront abordés plus loin.

## 7.2.1 Le passage par un support concret pour les ICE

Dans cette section, nous allons expliquer comment IAO implante la notion d'abstraction des ICE par rapport aux entités matérielles que nous avons abordée à la section 5.6.

Pour IAO, les ICE sont des `generically_dependent_continuant`, une classe issue de BFO que nous n'avons pas encore vue, mais qui stipule que l'information doit avoir un support pour être manipulée.

Pour expliquer ce qu'est un `generically_dependent_continuant`, il nous faut revenir aux sections 4.3.3.1 et 4.3.3.2 de la présentation de BFO où nous avons introduit la distinction entre `occurrent` et `continuant`. Nous avons alors vu que `continuant` se subdivise à son tour en `independent_continuant` et `dependent_continuant`, mais `dependent_continuant` lui aussi se subdivise en `specifically_dependent` et en `generically_dependent`.

Jusqu'à maintenant, tous les `dependent_continuant` que nous avons vus étaient des `specifically_dependent`, par exemple des `quality` comme la turbidité d'un liquide, la turbidité est spécifique à un seul liquide, deux liquides n'auront jamais exactement la même turbidité. La relation d'un `specifically_dependent` avec son `independent_continuant` est donc fixe sur un seul individu.

Tout comme un `specifically_dependent_continuant`, un `generically_dependent_continuant` ne peut pas exister seul, cependant il peut transférer sa dépendance d'un `independent_continuant` à un autre. Le texte d'un protocole est un exemple de `generically_dependent_continuant` ; bien qu'il ne peut exister sans être écrit sur du papier ou inscrit sur un disque dur, il n'a pas de relation fixe avec aucun support: le papier peut être déchiré, le disque effacé, tant que le protocole est présent sur au moins un support il existe encore.

Le processus d'inscription d'un `generically_dependent_continuant` sur un support s'appelle la concrétisation en BFO, un `generically_dependent_continuant` *is\_concretized\_as* un `specifically_dependent_continuant` par un `process`.

L'impression sur papier est un exemple d'un processus de concrétisation où le `generically_dependent_continuant` :texte est transformé par le processus: impression en un `specifically_dependent_continuant` :`pattern_marque_d'encre` sur un `independant_continuant` :papier. Le processus de concrétisation étant souvent de peu d'intérêt pour la modélisation et on n'en tient donc pas compte.

Les classes de IAO sont toutes des `generically_dependent_continuant`, les informations ont besoin d'un support pour être vues, comprises, échangées, mais aucun de ces supports n'a à être fixe.

## **7.2.2 Les facettes des informations et IAO**

Lors de l'introduction du concept de gestion informationnelle au Chapitre 5, nous avons introduit les cinq facettes des éléments d'information soit l'étendue, la nature, le sujet, la production et l'utilisation. Cette conception provient de nos lectures sur IAO.

### **7.2.2.1 L'étendue d'une ICE**

La notion d'étendue est utilisée pour l'étape de repérage de la gestion informationnelle. Elle n'a pas d'encodage en OntoMic, car le modèle BFO ne comprend pas de relation de comparaison.

### **7.2.2.2 La nature d'un ICE**

Le concept de nature d'une information, de nature d'une ICE, est représenté en IAO par une hiérarchie de classes. Il y a dans IAO des entités comme `documents`, `symbol` qui sont des chaînes de caractères qui représentent d'autres entités, des `directive_information_entity` qui servent à encadrer des processus.

Pour OntoMic, les `directive_information_entity` sont la représentation des entités abstraites logiques procédurales que nous avons vues à la section 5.6. Nous pouvons dire qu'OntoMic représente le « sens » d'un texte, qui dans notre cas est un « comment faire », par des `directive_information_entity`. Cette classe comprend les `plan_specification` et les `objective_specification`. La plupart des « comment faire » seront des `plan_specification`, mais OntoMic

utilise les *objective\_specification*, pour par exemple, décrire la liste des organismes pathogènes qu'une culture est à même de détecter. Dans ce qui suit, nous utiliserons seulement les *plan\_specification* pour faciliter la lecture du texte.

Tous les *plan\_specification* partagent la même nature, même s'ils se retrouvent dans une hiérarchie de sous-classes comme *plan\_global* et *plan\_étape*. Ces sous-classes ne sont pas des *universal* mais des *defined\_class*, la nature est donnée par l'*universal*. Ce qui change entre un *plan\_global* et un *plan\_étape* est son sujet et non sa nature d'être la représentation d'une logique procédurale.

### 7.2.2.3 Le sujet d'une ICE

Tous les ICE doivent être reliés à d'autres entités par la relation *is\_about*. Cette relation donne le sujet de l'élément ICE, la cible choisie pour *is\_about* est très importante pour donner un sens à l'ICE. Par exemple, un symbol 'URCUL' sera *is\_about* *culture\_d'urine*.

IAO est très souple quant aux cibles de la relation *is\_about* qui peuvent être n'importe quels éléments de l'ontologie. OntoMic quant à elle restreint la plupart du temps les cibles d'*is\_about* aux éléments propres à OntoMic et non pas aux éléments créés avec OntoMic pour encoder des protocoles.

Le degré de précision sur le sujet de l'ICE que doit apporter *is\_about* est une convention. Pour OntoMic la nature et le sujet de l'ICE doivent ensemble fournir assez d'information pour que l'ICE prise de façon isolée ait un sens.

Le *plan\_specification* : *plan\_selon\_SJ\_MIC-PON-072\_V7* issu du protocole *SJ\_MIC-PON-072-Culture\_de\_selles\_V7* est *is\_about* l'entité *culture\_de\_selles\_générale*, la cible du *is\_about* pour un *plan\_specification* est donc plus large que les seules cultures réalisées sous sa gouverne.

### 7.2.2.4 Leur production

Bien que toutes les ICE aient une production, dans les cas des textes des protocoles la description de leur production n'est pas dans le champ d'application d'OntoMic.

Par contre, c'est la gestion informationnelle qui crée les `plan_specification` utilisées pour représenter les logiques procédurales. Faire un lien entre un texte et un `plan_specification` est un processus non déterministe, en effet nous avons vu des exemples où les tâches peuvent être subdiviser de différentes manières. Nous conservons une trace du lien entre le texte source et le `plan_specification` avec une relation spéciale: *denotes*.

#### **7.2.2.5 Leur utilisation**

Pour représenter l'utilisation du `plan_specification` par un processus, IAO passe par une concrétisation du `plan_specification` comme nous avons vu à la section 7.2.1. Le `plan_specification` est concrétisé en une instance de `plan` et, par la suite, par la réalisation de cette instance de `plan` dans une instance de `planned_process`.

Cette construction, qui est bien abstraite, n'a pas d'application avec OntoMic. En effet, OntoMic ne s'intéresse pas à l'exécution des `planned_process`, mais à leur encadrement. Dans les faits, cette construction ne sert qu'à stipuler quel plan dirige quel processus.

Donc dans ce chapitre nous avons vu qu'OntoMic et OBI partagent le besoin de représenter le déroulement des processus. Ces mécanismes de représentations d'OBI sont repris par OntoMic pour encoder les logiques procédurales. Nous avons également expliqué qu'IAO sert à représenter les textes comme entités lexicales, le sens des textes (dans notre cas les logiques procédurales) et les données produites et consommées pendant l'exécution des processus.

## Chapitre 8 OntoMic

Nous avons présenté à la section 4.3 les relations fondamentales de BFO et au Chapitre 7 nous avons vu le rôle de l'ontologie modèle OBI et celui de l'ontologie partenaire IAO. Nous avons tout ce qu'il faut en main pour expliquer l'organisation d'OntoMic, avec l'aide de notre analogie de formulaires. Pour simplifier, nous allons d'abord présenter OntoMic en langue naturelle et ensuite l'implantation en OWL.

Nous allons d'abord expliquer l'organisation de haut niveau d'OntoMic, comment les éléments principaux interagissent entre eux. Puis nous allons montrer comment sont classées les logiques procédurales tirées des textes sources. Ce classement se fait de façon classique pour une ontologie et est facile à comprendre.

L'encodage, c'est-à-dire la transformation des logiques procédurales en langage machine, est par contre plus complexe. Cet encodage se base sur une modélisation fort peu intuitive au départ, son explication requiert notre analogie des formulaires. Il fait appel également à des notions supplémentaires sur IAO.

Finalement, nous verrons la méthode avec laquelle OntoMic encode le flot de l'exécution des processus. Cette méthode est elle aussi inhabituelle, avec OntoMic la logique de flot ne s'encode pas comme un algorithme avec OntoMic.

### 8.1 L'organisation des éléments d'OntoMic

Le diagramme à la Figure 8-1 illustre l'organisation et les interactions entre les principaux éléments. Débutons la présentation des entités par la division verticale du diagramme entre GLOBAL et APPLICATION.

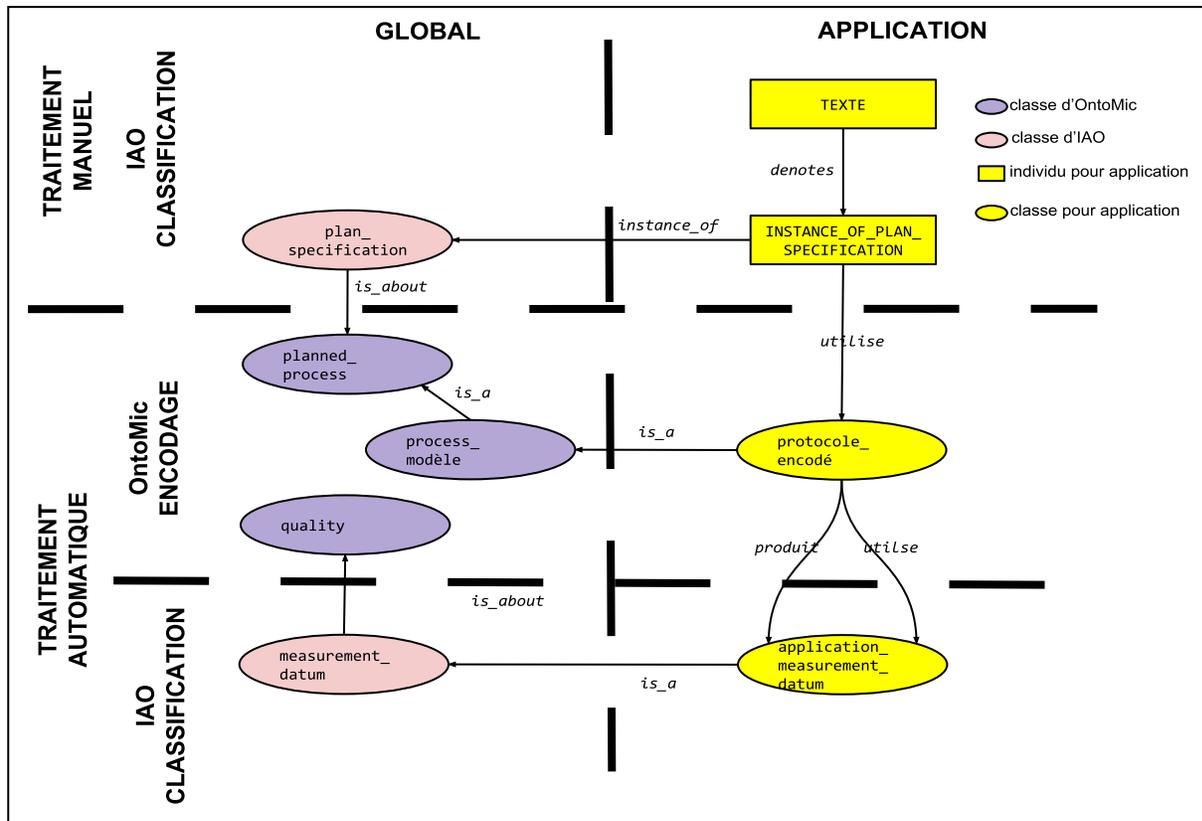


Figure 8-1, schéma de l'organisation générale d'OntoMic.

- Les entités de couleur  sont des éléments provenant d'OBI.
- Celles en  sont ceux qu'un utilisateur définit pour encoder.
- Celles en  proviennent d'IAO.

### 8.1.1 Les entités prédéfinies et celle des applications

OntoMic comprend les classes générales, le côté GLOBAL, que tous les protocoles peuvent utiliser pour leur encodage. Ces classes sont pour la plupart des classes filles de classes importées d'OBI ou d'IAO.

Nous classifions et encodons un texte en créant de nouvelles entités du côté APPLICATION qui sont des combinaisons et/ou des adaptations des entités provenant du côté GLOBAL.

## **8.1.2 Les comparaisons manuelles et automatiques**

En séparant les entités par la ligne horizontale supérieure, le diagramme de la Figure 8-1 fait avec une distinction entre le traitement automatique et manuel des informations. C'est-à-dire entre la comparaison manuelle de protocoles assistée par OntoMic et la comparaison automatisée.

### **8.1.2.1 Traitement manuel**

Pour comparer manuellement des informations contenues dans les textes, il faut que celles-ci soient retraçables et classifiées. Les entités qui se trouvent au-dessus de la ligne horizontale supérieure servent au classement des textes et de leurs extraits. Ces tâches d'organisation des informations et de classement sont considérées dans notre modélisation comme faisant partie d'IAO.

Du côté APPLICATION, nous transformons les textes en des assortiments d'extraits par le processus de repérage des informations décrit à la section 5.4.1. Les informations contenues dans les textes sont alors retraçables. Pour les classifier, nous organisons les extraits en les reliant à des éléments prédéfinis, ces éléments prédéfinis sont du côté GLOBAL et ils sont hérités d'IAO. Ils sont classés dans une hiérarchie ayant à sa tête `plan_specification`, comme nous le voyons à la Figure 8-2.

### **8.1.2.1 Traitement automatique**

Le traitement automatique est représenté en deux couches séparées par la ligne horizontale inférieure. Les éléments de la première couche sont dédiés à l'encodage comme tel, tandis que ceux de la deuxième couche servent à classer les données produites et consommées durant l'exécution des cultures.

#### **8.1.2.1.1 L'encodage par des éléments d'OntoMic**

Le traitement automatique se fait en créant une représentation machine du « sens » des extraits de textes, c'est-à-dire une représentation machine de la logique procédurale décrite par le texte.

Il y a des représentations machines du côté GLOBAL et du côté APPLICATION. Les représentations machines du côté GLOBAL servent de modèle et de modules à assembler pour construire les représentations machines du côté APPLICATION.

Pour faire les représentations machines, nous avons également recours aux données produites et consommées qui se trouvent dans la seconde couche.

#### 8.1.2.1.2 Classement des données produites et consommées

La seconde couche du traitement automatique contient les données produites et consommées pendant l'exécution des cultures, les principales étant les `measurement_datum`, qui sont les mesures faites pendant l'exécution d'un processus. Les protocoles imposent souvent des règles sur des données, par exemple sur un décompte de colonies, des éléments de cette seconde couche du traitement automatique serviront à bâtir ces règles. Tous ces éléments sont organisés par IAO.

Les `measurement_datum` du côté GLOBAL servent de modèle à celles du côté APPLICATION. Nous aurons également des critères du côté GLOBAL qui servent, en combinaison avec les éléments d'encodage d'OntoMic, à bâtir des règles.

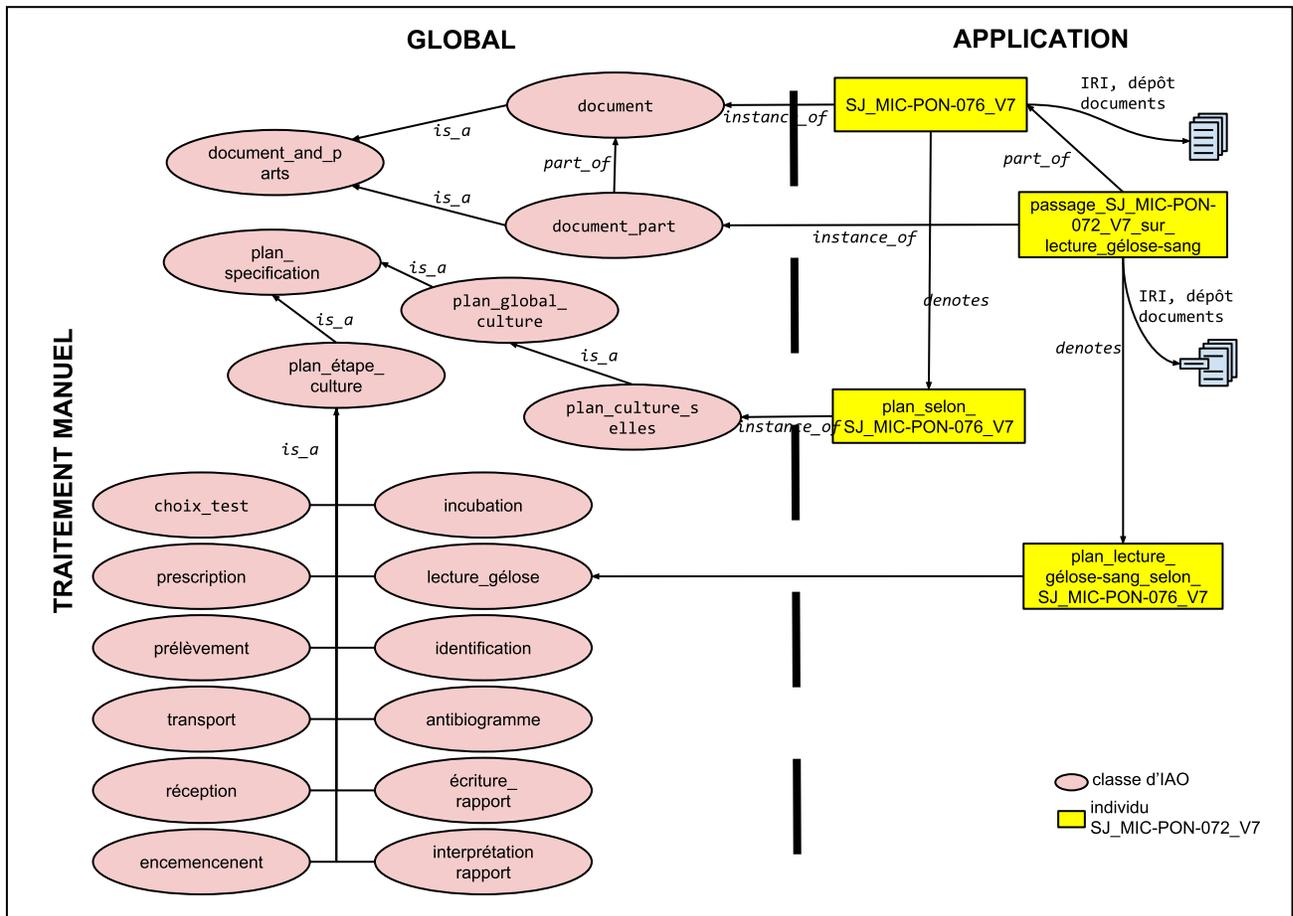


Figure 8-2, schéma des sous-classes de plan\_specification dans OntoMic

## 8.2 Le classement des informations pour le traitement manuel

Comme nous l'avons vu ce classement correspond à l'étage supérieur de diagramme de la Figure 8-1. Nous avons une illustration plus détaillée de ce classement à la Figure 8-2

## 8.2.1 Les textes et leurs extraits.<sup>21</sup>

Nous avons vu que nous transformions les textes en des assortiments d'extraits par le processus de repérage des informations. Pour classer les textes et leurs extraits, nous débutons par les séparer en entités lexicales et en sémantiques.

### 8.2.1.1 Les textes et extraits comme entités lexicales

Les entités lexicales textes et extraits de textes sont des **instances** des classes d'IAO : `document` et `document_part`. Ceci est une des rares utilisations des individus dans OntoMic. En effet, comme nous le verrons plus loin, l'encodage des logiques procédurales se fait essentiellement avec des classes et non des instances.

Ces instances d'IAO, `document` ou `document_part`, sont représentés par des IRI comme tous les éléments d'une ontologie écrite avec OWL. Ces IRI ont la particularité d'être choisis pour être des URL référant aux textes ou aux extraits dans un dépôt. Les URL permettent de les retracer facilement.

L'organisation des protocoles et des autres documents est variable selon les laboratoires. Cette organisation est indépendante d'OntoMic qui impose seulement le classement des textes selon les éditeurs.

Les extraits de textes ont, quant à eux, les contraintes d'être continus et d'être reliés par la relation *part\_of* à leur texte complet source. Le fractionnement du texte est géré par l'encodeur à l'étape de repérage, comme nous l'avons vu à la section 5.4.1.1.

### 8.2.1.2 Les textes et extraits modélisés comme des logiques procédurales

Une fois que nous avons créé un `document` ou un `document_part`, nous créons la ou les instances de `plan_specification` qui en représentent la logique procédurale. Le lien entre une instance de

---

<sup>21</sup> OntoMic vise d'autres documents que les textes, comme les bases de données, mais comme la logique avec les autres types de documents est la même, nous n'en mentionnons pas l'existence ici.

document ou document\_part et sa ou ses instances de plan\_specification est fait avec la relation *denotes*, qui est une sous-relation<sup>22</sup> de *is\_about*. L'exigence d'IAO d'avoir d'une relation *is\_about* pour les document ou document\_part, comme pour tous ses autres éléments, est donc satisfaite.

OntoMic n'impose pas une relation un à un entre les plan\_specification et les document ou document\_part. En effet, les informations sur une même logique procédurale peuvent ne pas être contiguës dans le texte, elles peuvent même se retrouver dans des textes différents. Il peut donc y avoir plus d'une instance de document ou document\_part pour une même logique procédurale. Également, les informations de plusieurs sous-tâches peuvent être imbriquées dans un texte d'une façon inextricable<sup>23</sup>. Il peut donc y avoir plus d'une logique procédurale pour un seul document ou document\_part.

### 8.2.2 Le classement des plan\_specification

Le classement des textes et des extraits se fait par une hiérarchie de sous-classes de plan\_specification. OntoMic définit deux grandes sous-classes de plan\_specification : plan\_global\_culture et plan\_étape\_culture. La classe plan\_global\_culture comprend les logiques procédurales qui correspondent à un protocole complet et la classe plan\_étape\_culture comprend les logiques pour une étape du cheminement d'une culture.

Les sous-classes de plan\_étape\_culture sont basées sur les étapes du cycle complet d'une analyse de microbiologie que nous avons vu à la section 2.1.2. En effet lors de notre analyse des erreurs retrouvées au laboratoire, nous avons conclu à la section 3.4.6 que notre gestion des informations devait toujours se baser sur ce cheminement.

---

<sup>22</sup> Si *a* sous-relation de *b* et que nous avons X *a* Y alors nous avons automatiquement X *b* Y.

<sup>23</sup> Nous en avons vu une illustration de ceci avec l'Extrait de document 3-1 à la section Exemple 1 Des informations sur le choix du contenant ne se trouvait pas dans la section prélèvement mais dans la section transport et réception

Avec cette organisation des `plan_specification`, nous pouvons classer les logiques procédurales provenant des textes et de leurs extraits selon leur sémantique sans avoir à encoder leur logique.

### 8.3 L'encodage pour le traitement automatique

Nous avons vu qu'un protocole est transformé en un assortiment de logiques procédurales, représentées par des `plan_specification`. Ces logiques sont encodées par classes et non des instances, ceci découle du modèle même d'encodage d'OntoMic, hérité d'OBI.

OntoMic n'encode pas un protocole comme un algorithme, qui serait éventuellement appliqué lors du travail sur de véritables cultures. OntoMic procède à l'inverse, elle représente **en premier** de véritables cultures. Puis en **second lieu**, OntoMic décrit les caractéristiques qu'une véritable culture doit avoir pour qu'elle soit considérée comme ayant été faite en conformité aux exigences du protocole. Ces cultures, considérées conformes, sont des instances d'une classe spéciale : `planned process`.

Chaque culture a un déroulement dans le temps, durant lequel des éléments matériels sont consommés, transformés, produits et des informations sont utilisées ou produites. Ces interactions entre la culture et les éléments matériels et informations sont encadrées par les exigences du protocole. C'est le protocole qui dicte, par exemple, la température à laquelle une incubation doit se dérouler. C'est le protocole qui dicte quelles géloses doivent être ensemencées.

La prémisse fondamentale d'OntoMic est que la logique procédurale du protocole peut être encodée en mettant des règles, que nous appellerons des balises, sur ces interactions. C'est ce que nous appelions ci-haut décrire les caractéristiques nécessaires à une véritable culture pour être en conformité avec le protocole.

Au Chapitre 6 sur notre analogie des formulaires, nous avons introduit une notion équivalente où nous encodions un protocole par des contraintes sur les données utilisées et produites lorsqu'une culture est faite plutôt qu'en spécifiant des listes de gestes. Ces contraintes étaient encodées par des éléments spécifiques dans un formulaire qui serait **théoriquement** à remplir pendant le parcours d'une culture.

### 8.3.1 La classe qui représente les cultures, les `planned_process`

Pour OntoMic, chaque culture est, à la base, une instance de la classe `process` qui provient de BFO. Être un individu de la classe `process` n'impose qu'une seule contrainte: celle d'avoir un déroulement dans le temps.

Pour BFO, un `process` n'a pas de limites de « finesse ». Conceptuellement, un `process` comprend toutes les interactions possibles, même les interactions avec des éléments aussi fins que les particules atomiques. C'est à l'ontologie de définir quelles interactions d'un `process` doivent ressortir. C'est donc OntoMic qui déclare que, dans le déroulement d'une culture, nous nous intéressons à des groupes de bactéries et non pas à une cellule bactérienne individuelle.

Pour OntoMic, les cultures sont des `process` spéciaux, des `planned_process`, parce que leur déroulement est planifié et que par définition ils ont respecté la planification. Les `planned_process` sont issus d'OBI et leur déroulement est sous la gouverne<sup>24</sup> d'un `plan_specification`. Une culture de selles hypothétique où les étapes seraient choisies aléatoirement sera toujours une instance de la classe `process`, car elle a un déroulement dans le temps, mais elle ne sera pas un `planned_process`.

La classe `planned_process` est une `defined_class` et la règle qui la gouverne stipule que chaque instance de `planned_process` doit être reliée à (gouvernée par) **une** instance de `plan_specification`. En effet pour OntoMic, l'ensemble de la logique procédurale qui gouverne un `planned_process` est contenue dans **une** instance de `plan_specification`. Cette instance unique de `plan_specification` peut avoir des sous-parties, comme `plan_global_culture` qui a nécessairement des sous-parties `plan_étape_culture`.

OntoMic crée des sous-classes de `planned_process` en fonction du choix du `plan_specification`. Par exemple, la `lecture_gélose` comprend tous les `planned_process` qui sont sous la gouverne

---

<sup>24</sup> Le véritable lien est la réalisation d'une concrétisation comme expliqué à la section 7.2.2.5. Mais la nature précise de ce lien est peu importante pour OntoMic.

d'une instance de `plan_lecture_gélose`, une sous-classe de `plan_specification`. `SJ_culture_selles_V7` comprend les cultures faites sous la gouverne de `plan_selon_SJ_MIC-PON-072_V7`.

Les `plan_specification` sont des entités abstraites auxquelles on ne peut accéder que par le biais d'une matérialisation. Donc déclarer qu'un `planned_process` est sous la gouverne d'un `plan_specification` ne fait aucun encodage, seulement des liens vers des textes sources en passant par l'intermédiaire du `plan_specification`.

### 8.3.1.1 Les protocoles comme des `planned_process` balisés

Nous avons vu que la prémisse fondamentale d'OntoMic est que nous pouvons encoder la logique procédurale du protocole en mettant des règles, que nous appellerons des balises, sur les interactions entre la culture et des éléments matériels ou d'informations.

Pour encoder un protocole, nous débutons en créant une sous-classe de `planned_process`. Cette sous-classe est spécifique pour les cultures faites sous la gouverne du `plan_specification` que le protocole *denotes*. Il faut noter que cette sous-classe spécifique n'aura **pas** d'instances, elle ne servira que de point d'ancrage pour les balises. En effet, nous ne nous intéressons pas à représenter de véritables cultures, mais plutôt à comparer des jeux de balises issus de protocoles différents.

Pour encoder une balise, nous débutons par préciser si l'élément extérieur au `planned_process` sur lequel elle se porte est en entrée ou en sortie du processus. Puis nous spécifions la ou les classes qui regroupent les éléments en interaction avec le processus. Cette classe peut être une classe native, mais le plus souvent, il faudra créer une `defined_class`.

Pour donner un exemple de balises, reprenons l'étape `SJ_lecture_gélose-sang_V7` que nous avons vue à la section 6.1 sur notre analogie de formulaires et fabriquons une balise pour cette étape de la culture de selles. Cette balise stipulera qu'il faut en entrée de `SJ_lecture_gélose-sang_V7` une gélose-sang qui a subi une incubation adéquate.

Pour faire la balise, débutons en définissant la classe `SJ_lecture_gélose-sang_V7`, qui rassemble les lectures de gélose qui respectent la logique procédurale du protocole. Puis nous définissons la classe `gélose-sang_incubée` comme étant les géloses-sang en sortie d'une incubation.

Pour imposer la balise, nous faisons de `SJ_lecture_gélose-sang_V7` une `defined_class` avec une règle qui impose que chacune des instances de `SJ_lecture_gélose-sang_V7` ait en entrée une `gélose-sang_incubée`.

Il reste maintenant à donner la règle pour la `defined_class` : `gélose-sang_incubée`. Idéalement, les balises sont construites avec des éléments prédéfinis d'OntoMic. Cependant, il serait encombrant de prédéfinir `gélose_sang_incubée`, `gélose_MacConkey_incubée`, `gélose_CIN_incubée`. Sans compter qu'il faudra faire la même énumération pour les géloses en sortie d'ensemencement, l'étape qui précède l'incubation.

Nous pouvons grandement simplifier en définissant dans OntoMic : `dispositif_culture_incubé`. Cette classe est composée de tous les dispositifs de culture qui sont en sortie d'un processus d'incubation, peu importe de quel type de dispositif il s'agit. Nous pouvons donc définir `gélose-sang_incubée` comme l'intersection de `gélose-sang` et de `dispositif_culture_incubé`. En procédant ainsi la règle pour `gélose-sang_incubée` n'est construite qu'avec des classes générales d'OntoMic.

Nous pouvons nous demander que sont les instances de `gélose-sang_incubée`.<sup>25</sup> Pour bien comprendre la réponse à cette question, voyons le cheminement d'une gélose en gardant à l'esprit le principe qui veut qu'OntoMic, comme BFO, soit le plus près possible de la réalité.

Après la réception du spécimen, la technicienne prépare l'ensemencement en rassemblant les géloses nécessaires. Si aucune `gélose-sang` n'est singulière avant que la technicienne ne la retienne pour l'ensemencement, il en est tout autrement après. Non seulement la gélose retenue est devenue singulière, mais les processus demandent de souligner cette singularité. En effet, la technicienne doit assigner à la gélose un identificateur unique écrit sur son couvercle ou imprimé sur une étiquette collante.

---

<sup>25</sup> Il n'y a pas d'instances dans les classes, nous aurions pu écrire « quelles seraient les instances ». Mais l'utilisation du conditionnel peut, dans ce cas, porter à confusion.

Pour OntoMic, les instances de `gélose-sang_incubée` **seraient** des géloses-sang qui ont été retenues pour une culture. Mais nous ne nous intéressons pas aux instances de `gélose-sang_incubée`, tout comme nous ne nous intéressons pas aux instances `SJ_lecture_gélose-sang_V7`. Ce qui nous intéresse ce sont les balises sur les `planned_process` ou les règles des `defined_class` qui représentent le fait qu'un protocole culture demande une `gélose-sang_incubée` en entrée tandis qu'un autre protocole demande une `gélose_MacConkey_incubée` en entrée. Le fait que nous n'exigions pas seulement une `gélose-sang`, mais bien une `gélose-sang_incubée` est important pour l'ordonnancement des tâches, nous verrons pourquoi à la section 8.5.

Par notre analogie des formulaires, la structure du formulaire est analogue au jeu de balises qui encode la logique procédurale. La Figure 6-1 qui nous donne un formulaire pour `SJ_lecture_gélose-sang_V7` définit la balise sur la `gélose-sang` en entrée avec ce que nous avons appelé le champ d'application du formulaire.

### 8.3.1.2 Le fractionnement des `planned_process`

Encoder tout un protocole de culture par une liste de balises sur une sous-classe de `planned_process` serait trop lourd. Comme faire un seul formulaire pour tout le processus de culture serait aussi trop lourd. Les `planned_process` sont donc subdivisés en sous-tâches. Le processus parent est relié à son ou ses sous-processus par la relation *has\_part*.

Comme nous sommes à l'étape de présentation de l'encodage en langue naturelle nous supposons que *has\_part* a la capacité, comme le prévoit BFO, de relier deux classes. Cependant, par ce que nous avons vu à la section 4.5.3.1, nous savons que les relations entre classes ne sont pas implantées dans la version OWL de BFO. Nous verrons à la section 8.4.2.1 comment les relations entre classes sont implantées avec OWL.

Avec notre exemple `SJ_culture_selles_V7 has_part SJ_lecture_gélose-sang_V7` qui elle-même a *has\_part* `SJ_repérage_gélose-sang_V7`.

Les sous-processus ont la même exigence d'être gouvernés par un `plan_specification` que le processus global. Dans le cas d'un sous-processus au lieu d'avoir le document du protocole au complet comme texte source du `plan_specification` nous aurons un extrait du protocole.

### 8.3.1.2.1 Fractionnement et tâches conditionnelles

Avec ce qui précède, nous savons que l'énoncé suivant est vrai

`SJ_lecture_gélose-sang_V7` *has\_part* `SJ_repérage_gélose-sang_V7`

Par contre, cet énoncé est faux (l'énoncé est en ■ et se termine par un \*, car il contient une erreur de logique):

`SJ_lecture_gélose-sang_V7` *has\_part* `SJ_travail_S.auréus_V7*`

En effet, la culture de selles comporte toujours un repérage des colonies compatibles avec du *S. auréus*. Elle ne comporte évidemment pas une étape de travail sur le *S. auréus* si aucune colonie compatible avec cet organisme n'est retrouvée.

Nous remplaçons donc ce faux énoncé par le suivant, qui lui est toujours vrai

`SJ_travail_S.auréus_V7` *part\_of* `SJ_lecture_gélose-sang_V7`

Cet énoncé corrigé stipule que si une étape de travail sur le *S. auréus* existe, alors elle doit faire partie de la lecture de la gélose-sang.

### 8.3.1.3 Les processus\_modèle en OntoMic

Par analogie avec les patrons de formulaire, nous avons défini en OntoMic des processus\_modèle généraux, qui imposent des balises générales. Ces processus\_modèle se retrouvent du côté gauche, GLOBAL, de notre diagramme à la Figure 8-1. Pour `SJ_lecture_gélose-sang_V7`, la classe modèle sera `lecture_gélose`.

Tandis que `SJ_lecture_gélose-sang_V7` aura comme balise d'avoir une `gélose-sang_incubée` en entrée. Le processus\_modèle : `lecture_gélose` aura comme balise d'avoir en entrée un choix de n'importe quelle gélose incubée.

Donc, pour chaque sous-tâche du processus global de culture, nous définissons un processus\_modèle. Nous assignons à ces modèles des balises générales, jugées essentielles et **universelles**.

Comme le patron de formulaire pour l'étape de lecture d'une gélose devait pouvoir servir à créer un formulaire spécifique pour n'importe quel protocole.

Pour s'assurer que les balises d'un modèle sont correctes, il est utile de bien cerner quelles sont<sup>26</sup> les instances de la classe du modèle. La classe `SJ_lecture_gélose-sang_V7` a comme instances les lectures de gélose-sang pour les cultures de selles faites sous la gouverne de `SJ_MIC-PON-072_V7`<sup>27</sup>. La classe modèle `lecture_gélose` a comme instances les lectures de gélose pour toutes les cultures, peu importe s'il s'agit de selles, d'urine, de gorge, etc. et peu importe selon quel protocole ou par quel laboratoire.

Cette exigence d'universalité d'application est sévère, néanmoins certaines balises s'imposent d'emblée. Par exemple, celle qui, pour un processus de lecture de gélose, exige d'avoir une gélose incubée en entrée.

D'autres balises cependant sont le résultat d'une modélisation de tâche. Nous avons vu des exemples de ces balises avec notre analogie des formulaires. Le modèle qui sous-tend à la création de ces balises fractionne la tâche lecture selon chacun des types de colonies que le protocole juge pertinents. Seules des expériences empiriques d'encodages de protocoles peuvent démontrer l'applicabilité universelle du modèle.

#### 8.3.1.4 Conséquences du modèle

##### 8.3.1.4.1 `Planned_process` et atteinte des objectifs

En imposant des contraintes (des balises) sur l'appartenance à la classe créée pour un protocole, nous affirmons que toutes instances de la classe répondent aux contraintes. Donc une culture faite selon le protocole, mais qui comprend une erreur et ne respecte pas une balise n'est **pas** dans la

---

<sup>26</sup> Encore une fois il n'y a pas d'instances dans les classes, mais l'utilisation du conditionnel peut, dans ce cas, porter à confusion.

<sup>27</sup> Pour simplifier nous avons fait le raccourci de désigner le protocole comme source de gouvernance. Strictement selon le modèle d'OntoMic c'est le plan specification que le protocole *denotes* qui gouverne.

classe. Par exemple, une culture de selles faite à Saint-Jérôme où la recherche de la flore normale a été omise n'aura pas d'instance de `SJ_repérage_gélose-sang_V7`.

Mais comme nous ne nous intéressons pas aux instances des classes d'encodage, mais bien à en comparer les balises, cette particularité n'est pas restrictive pour OntoMic, elle est même utile pour la logique de l'encodage, car ainsi nous n'avons pas à gérer toutes les possibilités d'erreur. Par exemple, nous n'avons pas à encoder dans `SJ_lecture_gélose-sang_V7` les actions à prendre si l'incubation a été un échec parce que l'étuve est tombée en panne. Quand `SJ_lecture_gélose-sang_V7` exige d'avoir en entrée une `gélose-sang_incubée`, `SJ_lecture_gélose-sang_V7` peut présumer que l'incubation a été adéquate.

#### 8.3.1.4.2 Interrogation d'OntoMic et tâches facultatives

Nous avons vu qu'aucune balise appliquée à `SJ_repérage_gélose-sang_V7` n'indique un lien avec `SJ_travail_S.auréus_V7`. Il arrive que rien ne pousse sur les géloses, même pour une selle, un spécimen qui n'est jamais stérile<sup>28</sup>. Donc peu importe son adaptation pour un protocole particulier, `lecture_gélose` aura toujours *has\_part* `repérage_gélose`, mais on ne pourra pas imposer *has\_part* `travail_isolat_repéré`. La relation sera toujours renversée avec `travail_isolat_repéré part_of lecture_gélose`.

Une conséquence de cette absence d'embranchement conditionnel est que nous ne pouvons pas simplement suivre un enchaînement pour savoir ce qui suivra un `repérage_gélose`. Pour rechercher ce qui arrive avec un isolat de *S. auréus* repéré sur la gélose-sang par `SJ_repérage_gélose-sang_V7`, il faut faire une recherche dans toutes les tâches pour retrouver celles qui ont `SJ_isolat_S.auréus` en entrée.<sup>29</sup>

---

<sup>28</sup> Ceci peut survenir, car plusieurs bactéries de la flore intestinale ne peuvent croître dans les conditions de la culture de selles, des bactéries comme les anaérobies par exemple.

<sup>29</sup> Ces tâches ne sont pas nécessairement des `travail_isolat_repéré`. Par exemple, dans le cas où la colonie a une apparence compatible avec un organisme agressif, la tâche suivante est une tâche de sécurité qui est commune à tous les protocoles.

### 8.3.2 Présentation générale des `measurement_datum`

Pendant qu'une culture est faite, des données sont produites comme des comptes-rendus, des décomptes de colonies. De même, certaines données sont utilisées pendant le travail, comme ces mêmes décomptes, dont la valeur permet d'orienter le déroulement du travail.

Il faut donc des balises sur les données, utilisées, produites par les protocoles. Jusqu'à maintenant, nous n'avons vu dans les exemples précédents que des balises sur des entités matérielles parce qu'elles sont plus simples.

OntoMic utilise abondamment les balises sur les `measurement_datum`. Nous avons vu que les `measurement_datum` sont les résultats de mesures faites pendant l'exécution de processus. Nous ajoutons que pour que la mesure soit un `measurement_datum`, il faut que l'action de faire la mesure soit prévue selon un plan specification. Les `measurement_datum` sont donc toujours en sortie d'un `planned_process`. Cette définition n'empêche pas qu'un `measurement_datum` soit utilisé dans un `planned_process` autre que celui qui l'a produit<sup>30</sup>.

Une balise qui impose un `measurement_datum` en entrée d'un `planned_process` signifie que la valeur est utilisée pour un calcul ou pour une décision selon une règle, tandis qu'imposer un `measurement_datum` en sortie implique que la valeur pour la mesure doit être consignée dans un registre quelconque.

Mais souvent, l'encodage nécessite la construction de balises plus complexes que simplement imposer la présence obligatoire de la mesure en entrée ou en sortie. Par exemple pour une incubation nous pourrions mettre une balise de durée cible de 24 heures. À la section 6.1 nous avons vu une autre balise qui demande que le `measurement_datum` sur le décompte des colonies de *S. aureus* soit « forte prédominance » ou « culture pure » pour que l'isolat soit jugé pertinent et

---

<sup>30</sup> Nous avons vu l'importance pour les cultures des règles sur l'âge des patients, mais ces âges sont calculés par d'autres processus que ceux des cultures.

travaillé plus loin. Nous nommons **normes** cette combinaison d'un **measurement\_datum** et d'un **critère**.

Jusqu'à présent, nous avons présenté les **measurement\_datum** sans expliquer comment OntoMic en faisait la représentation machine. Cette représentation est nécessaire si nous voulons pouvoir comparer automatiquement des balises qui comprennent des **measurement\_datum**.

### 8.3.3 Les **measurement\_datum** comme des ICE d'IAO

#### 8.3.3.1 Les **measurement\_datum** et les **plan\_specification**

Avant d'expliquer la structure des **measurement\_datum**, il nous faut comprendre en quoi ces derniers sont fondamentalement différents des **plan\_specification**. Nous avons vu que les **plan\_specification** étaient des entités abstraites, mais même abstrait, un **plan\_specification** est une instance, un individu, il se veut la représentation dans l'ontologie d'une chose unique.

Bien que les **measurement\_datum** n'ont pas d'instances, car il n'y a pas de résultats dans OntoMic, ils ne sont pas uniques pour un protocole. Il y a conceptuellement une instance de **measurement\_datum** pour **chacune** des exécutions du protocole. Ici, le recours à notre analogie de formulaire est utile. Nous pouvons voir « un » **measurement\_datum** comme un **champ** à remplir dans un formulaire et non pas comme les **valeurs** entrées dans ce champ.

Par exemple, « un » **décompte\_colonie** est une **defined\_class** qui est sous-classe de la classe mère **measurement\_datum**. Bien qu'« un » **décompte\_colonie** est vide d'instances et que nous nous intéressons seulement qu'à ses règles, conceptuellement, il y aura une instance de **décompte\_colonie** pour chacune des cultures.

Donc OntoMic conçoit les **measurement\_datum** comme des classes et nous ferons l'abus de langage de désigner par « un » **measurement\_datum** la **defined\_class** qui définit et encadre la mesure.

Dans un formulaire, on doit spécifier par une étiquette ce qui fait l'objet de la saisie. Pour les valeurs numériques, le formulaire doit aussi spécifier les unités de mesure à utiliser. Pour les

valeurs catégorielles, le formulaire stipule indirectement une échelle de mesure par les libellés associés à chaque case de choix. Les `measurement_datum` devront aussi avoir les mêmes capacités de spécifier l'objet de la saisie, les unités de mesure ou l'échelle catégorielle.

Les `measurement_datum` sont représentés par IAO, car ils sont des éléments d'information qui sont intentionnellement créés. Ce sont des mesures qui sont faites sur les caractéristiques de quelque chose ; on ne peut pas mesurer la chose elle-même, on en mesure la masse, la taille, la quantité. Avec BFO on dira que l'on mesure une `quality` d'un `independent_continuant`.

Nous avons vu les cinq facettes des ICE à la section 5.2 soit l'étendue, la nature, le sujet, la production et l'utilisation. À la section 7.2.2, nous avons vu comment IAO gère ces facettes pour les `plan_specification`. Mais définir ces mêmes facettes pour un `measurement_datum` est plus difficile. Par exemple, la frontière entre la nature et le sujet d'un `measurement_datum` est subtile.

### **8.3.3.2 La nature des `measurement_datum`**

Mais, avant même de tenter de décrire la nature et le sujet d'un `measurement_datum`, il faut s'assurer que l'entité que nous cherchons à représenter est bien un `measurement_datum`. En effet, ce ne sont pas toutes les quantités qui sont des `measurement_datum`, prenons l'exemple de la Concentration Minimale Inhibitrice (CMI) pour une combinaison antibiotique – bactérie. Il n'y a pas de méthode qui mesure directement une CMI. La CMI est une quantité inférée à partir d'observations sur la croissance de la bactérie en présence de l'antibiotique. Une CMI est ce qu'OBI considère un `predicted_data_item`.

Maintenant que nous savons ce qu'est un véritable `measurement_datum`, nous pouvons définir sa nature. La nature d'une ICE est donnée par son `universal` et correspond généralement à une `quality` qui est mesurée. Rappelons-nous que, conceptuellement, nous devons pouvoir sortir un `universal` du cadre de notre ontologie sans qu'il perde son sens.

La classe `measurement_datum` est un `universal`, en effet dans n'importe quelle ontologie une mesure a un sens. À l'opposé il est difficile d'imaginer le sens de `SJ_décompte_colonie_compatible_S.aureus_gélose-sang` dans un contexte autre qu'une ontologie du domaine de la

microbiologie. Même la classe mère de SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang, décompte\_colonie\_gélose ne s'interprète pas en dehors du champ de la microbiologie. Ce ne sont pas donc pas des universal.

Mais le dénombrement d'un ensemble quelconque est valable, peu importe le contexte. Nous aurons la hiérarchie suivante<sup>31</sup> où les classes en ■ sont des universal. Nous avons utilisé une notation en escalier, allant de la classe la plus générale à l'extrême gauche à la plus particulière à l'extrême droite.

```
information_content_entity
  data_item
    measurement_datum
      dénombrement
        décompte_colonie_gélose
          SJ_décompte_colonie_compatible_S.auréus_gélose-sang
```

La nature de SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang et de décompte\_colonie\_gélose est donc d'être un dénombrement. Dans le cas précis des dénombrements, nous n'avons pas de quality qui fait l'objet de la mesure parce que BFO n'implante pas la notion d'ensemble, donc pas de notion de cardinalité.<sup>32</sup> Mais dans la plupart des cas, nous avons une correspondance comme entre la nature de la mesure et la quality. Comme avec temperature\_measurement\_datum et la quality : temperature ou entre length\_measurement\_datum et la quality : length.

### 8.3.3.3 Le sujet des measurement\_datum

Il reste à définir l'independant\_continuant qui porte la quality qui est mesurée : le sujet de la mesure. Il y a un équilibre à trouver entre le niveau de précision offert par son sujet et celui qui est

---

<sup>31</sup> La hiérarchie implantée par IAO est différente de celles illustrée. En effet IAO définit categorical\_measurement\_datum et scalar\_measurement\_datum comme deux universal distincts. Ceci est une erreur d'IAO, car la manière de compter n'est pas la nature du comptage.

<sup>32</sup> Ceci est vrai également pour les durées des processus qui ne sont pas couvertes par BFO.

offert par le processus de production de la mesure. Nous avons vu à la section 7.1.1 un exemple où l'on précisait la mesure en décrivant si elle avait été faite avant ou après une stimulation immunitaire. Donc dans cet exemple on amenait une précision sur les mesures en définissant comment elles étaient produites.

Prenons l'exemple de la mesure de turbidité d'une suspension de bactéries après agitation mécanique du milieu de culture liquide. La nature sera : mesure de turbidité, le sujet : milieu de culture liquide. Ainsi, la notion que la mesure est faite **après** agitation ne fait pas partie ni de la nature ni du sujet de la mesure et doit donc être décrite dans le processus de production de la mesure. Nous aurions pu faire autrement et définir comme sujet de la mesure : milieu de culture post agitation. Nous verrons à la section 8.5.2.2 qu'établir cette frontière entre sujet et production nous a amené des difficultés.

### **8.3.4 La valeur d'un measurement\_datum**

OntoMic ne s'adresse pas directement aux instances des `measurement_datum`, comme nous l'avons montré dans la section 8.3.3.1. Mais il faut quand même un mécanisme pour assigner des valeurs tangibles à un `measurement_datum`, car c'est avec ce mécanisme que les balises sont faites.

OBI n'a pas encore totalement stabilisé les `measurement_datum`. Une partie de ce qui est présenté ici est une extension pour OntoMic de ce qui existe dans OBI.

Pour encoder les valeurs d'un `measurement_datum`, nous utilisons les relations *has\_value* et *has\_category\_value*, pour les mesures scalaires et catégorielles respectivement.

#### **8.3.4.1 Pour les valeurs scalaires**

La relation *has\_value* n'est pas une relation d'individu à individu, contrairement à celles que nous avons vues jusqu'à maintenant. Elle est une « data property », c'est-à-dire une relation entre un individual et une valeur comme un chiffre.

Pour indiquer les unités de mesure, nous utilisons la relation *has\_unit\_label*, qui elle, est une relation classique individu à individu. En effet, pour OBI et OntoMic, les unités de mesure sont représentées par des instances de l'ontologie Unit-Ontology du groupe OBO. Ceci assure l'interopérabilité entre les ontologies en stabilisant les unités de mesure.

#### 8.3.4.2 Pour les valeurs catégorielles

La relation *has\_category\_value* est une relation classique, ses cibles sont des instances. Pour spécifier les valeurs admissibles pour une mesure catégorielle, OntoMic restreint les cibles de la relation *has\_category\_value* à une *échelle\_de\_mesure*. Par exemple, *SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang* prend ses valeurs parmi *échelle\_par\_prédominance*. Cette échelle comprend les instances « absence<sup>33</sup>, 'rare', 'non prédominant', 'prédominant', 'fort prédominant', 'culture pure'.

#### 8.3.4.3 Les unités de mesure et les *measurement\_datum*

La question des unités de mesure pose un problème intéressant pour les *measurement\_datum*. Est-ce que les unités de mesure font partie de la nature d'une mesure ? Selon ce que nous pouvons déduire de notre examen d'OBI et d'IAO, la réponse est non. Les unités ne font pas partie de la mesure, mais de sa concrétisation.

Donc un peu comme une instance de *plan specification* représente une sémantique abstraite et théorique, un *measurement\_datum* représente le résultat d'une mesure dans l'absolu, abstraite et sans unité. Lors de la concrétisation, on choisit des unités de mesure. Avec cette modélisation, 60 minutes et 1 heure peuvent être la même instance de *measurement\_datum*.

OWL ne fait pas d'arithmétique, donc OWL ne peut pas convertir les valeurs des mesures selon les unités. Cependant, OntoMic sera à même de détecter automatiquement si une norme tente

---

<sup>33</sup> Notons que *échelle\_par\_prédominance* fait un accroc à la logique pure avec l'instance 'absence'. Il est en effet illogique de quantifier une chose absente...mais cet accroc simplifie la modélisation.

d'appliquer un critère sur une mesure et que la norme et la mesure ne partagent pas les mêmes unités.

### 8.3.5 Les balises sur les `measurement_datum`

Pour encoder une restriction sur les valeurs admissibles OBI a défini<sup>34</sup> la classe `value_specification`, reprise par OntoMic. Une `value_specification` se définit exactement comme un `measurement_datum` avec les relations *has\_value*, *has\_unit\_label* et *has\_categorical\_value*. Elles diffèrent cependant par leur *is\_about*, en effet toutes les `value_specification` ont comme sujet un `data_item`. Cette cible est plus large que les `measurement_datum`, car nous pouvons vouloir mettre une norme sur autre chose qu'une mesure. Par exemple, nous pouvons vouloir mettre une norme sur une CMI, qui n'est pas un `measurement_datum`, comme nous l'avons vu à la section 8.3.3.1.

OntoMic définit des `critère`, qui sont des `value_specification` génériques que nous assemblons pour former une balise. Il y a des `critère` qui stipulent les unités de mesure et des `critère` qui stipulent les valeurs acceptables.

Si, par exemple, nous avons à faire une balise qui stipule une température d'incubation de 35<sup>0</sup> C. Nous débutons en adaptant avec le chiffre 35, la classe `critère_valeur_cible` pour faire `critère_cible_35`. Puis nous combinons avec la classe `critère_degré_Celsius`.

Par la suite, il faut spécifier la mesure qui est associée à la norme. Pour les températures d'incubation, dans la plupart des cas, le processus de culture en stipule une valeur attendue, mais en laisse la surveillance à un processus de contrôle externe. Il n'y a donc généralement pas de `measurement_datum` pour la température de l'étuve dans l'encodage d'un protocole de culture. Dans ce cas, la norme sera l'intersection de `critère_cible_35`, `critère_degré_Celsius` et un `temperature_measurement_datum` générique.

---

<sup>34</sup> Le rôle des `value_specification` dans OBI n'est pas stabilisé et a des interprétations différentes selon les auteurs.

## 8.4 Le schéma complet d'encodage illustré par la culture de selles

Maintenant que nous avons en main tous les éléments nécessaires pour encoder nos protocoles, nous allons voir avec des exemples comment nous implantons les constructions d'OntoMic en OWL.

### 8.4.1 Les constituants d'OntoMic

Revenons sur les constituants d'OntoMic avec une courte définition.

Les éléments de base de l'encodage

- Textes  
Regroupent `document` et `document_part` qui eux-mêmes représentent les protocoles et leurs extraits. Mais aussi par extension des jeux de données comme des feuilles de chiffrier.
- `plan_specification`  
Les représentations par des IRI des logiques procédurales
- `processus_modèle`  
Les tâches canoniques avec leurs balises essentielles
- `measurement_datum`  
Les données produites et utilisées par les `planned_process`
- critère  
Des critères sur des `measurement_datum`

Les éléments du vocabulaire standard

- `independant_continuant`  
Les `material_entity` comme les dispositifs de cultures, les agents microbiens et leur taxonomie, les instruments, etc.
- `dependant_continuant`  
Les `role` affectés au `continuant` comme contenir spécimen, les `quality` comme la couleur ou la forme d'un colonie, etc.

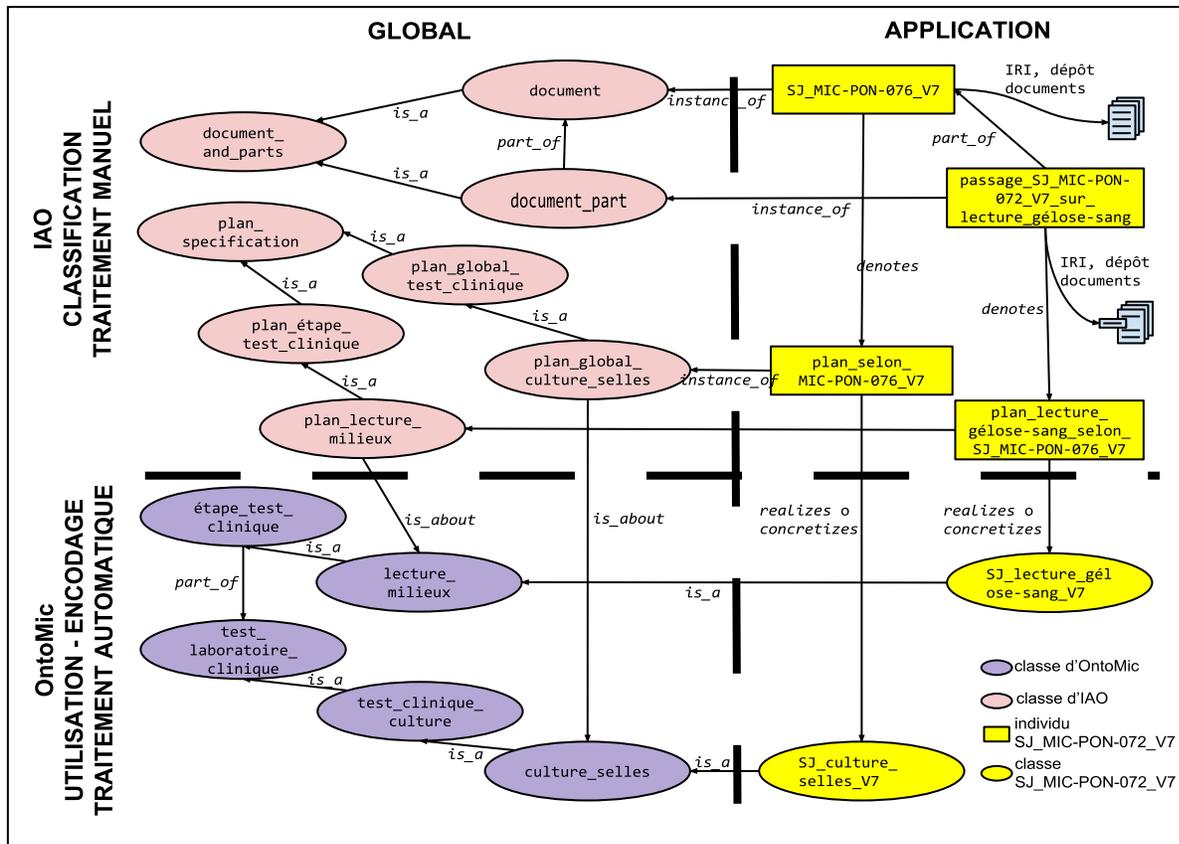


Figure 8-3, le schéma d'encodage des textes et les planned process

### Les éléments d'un encodage

- processus\_application  
La spécialisation des planned\_process généraux pour un protocole
- entités application  
Les entités autres que les planned\_process qui sont créées spécifiquement pour un protocole sont en principe quantité minimale.

## 8.4.2 Le passage des textes aux planned\_process

### 8.4.2.1 Les processus\_modèle et les plan\_specification

Le coin inférieur gauche du diagramme de la Figure 8-3 Figure 8-3, le schéma d'encodage des textes et les planned process illustre les processus\_modèle que nous avons utilisés pour notre exemple. Ces classes générales d'OntoMic ont été adaptées pour définir processus\_application

pour notre exemple. OntoMic impose une hiérarchie aux processus\_modèle symétrique à celle des plan\_specification.

OntoMic fait des classes de la hiérarchie des plan\_specification des defined\_class, en imposant par axiomes une relation *is\_about* avec leur processus\_modèle correspondant. Cependant comme il n'y a pas de relation entre classes avec OWL nous devons passer par une construction que nous allons illustrer par un exemple dans ce qui suit.

Pour les lectures de géloses, nous avons

plan\_lecture\_gélose  $\equiv$  plan\_étape\_culture **and** *is\_about* **some** lecture\_gélose

Ce qui veut dire que chaque instance de plan\_lecture\_gélose est en relation *is\_about* avec au moins une instance de lecture\_gélose.

À la section 4.3.2.1.2, nous avons vu que déclarer un individu membre d'une defined\_class est équivalent à affirmer que celui-ci obéit aux règles de la classe. Donc, en déclarant dans notre application que 'SJ\_plan\_lecture\_gélose-sang\_V7' est une instance de plan\_lecture\_gélose nous pourrions croire que nous imposons l'énoncé suivant :

'SJ\_plan\_lecture\_gélose-sang\_V7' *is\_about* **some** lecture\_gélose\*

Mais cet énoncé est faux, en fait, il ne s'agit même pas d'un énoncé OWL. En effet OWL ne permet pas l'usage de quantificateur dans les triplets. En effet, un axiome de restriction, comme celui que nous tentons de faire, impose une règle de peuplement à l'élément de gauche. Cet élément de gauche doit donc être une **classe**. En effet, s'il s'agit d'une instance nous ne pouvons pas imposer une règle de peuplement, l'individu est déjà désigné et unique.

La véritable interprétation est

Si 'SJ\_plan\_lecture\_gélose-sang\_V7' est une instance de lecture\_gélose  
alors il existe au moins un 'x' instance de lecture\_gélose tel que  
'SJ\_plan\_lecture\_gélose-sang\_V7' *is\_about* 'x'

Donc conceptuellement ‘SJ\_plan\_lecture\_gélose-sang\_V7’ est *is\_about* d'au moins une instance particulière de *lecture\_gélose* alors que ce que nous voulions exprimer est que le plan a comme sujet la classe *lecture\_gélose* en elle-même. Le compromis d’encodage des *is\_about* ci-haut est fonctionnel, il permet par exemple de faire la recherche des *plan\_specification* qui ont comme sujet la lecture de gélose.

Si nous voulons faire des recherches plus ciblées, par exemple sur les *plan\_specification* qui traitent des lectures de gélose-sang, l’encodage ci-haut est insuffisant. Il faut ajouter plus de précision à la relation *is\_about*. Pour *SJ\_plan\_lecture\_gélose-sang\_V7*, nous pourrions ajouter l’axiome suivant :

‘*SJ\_plan\_lecture\_gélose-sang\_V7*’ *is\_about some* *gélose-sang*\*

Encore une fois, ce qui précède n’est pas un énoncé OWL, car nous ne pouvons pas mettre d’axiome sur une instance. Pour imposer ce dont nous avons besoin, nous pouvons écrire :

*plan\_sujet\_gélose-sang*  $\equiv$  *plan\_specification* **and** *is\_about some* *gélose-sang*  
‘*SJ\_plan\_lecture\_gélose-sang\_V7*’ *instance\_of* *plan\_sujet\_gélose-sang*

Comme OWL permet de mettre le même individu dans plusieurs classes, l’énoncé ci-haut est admissible. Nous n’avons pas trouvé de méthode pour construire une classe comme *plan\_sujet\_gélose-sang* en ne faisant que des combinaisons de classes génériques déjà définies dans OntoMic. Nous ne pouvons pas faire comme pour *gélose-sang\_incubée* à la section 8.3.1.1. Si nous voulons seulement travailler avec des éléments prédéfinis, nous devons définir dans OntoMic une sous-classe de *plan\_specification* pour chaque dispositif de culture. L’autre stratégie est de définir lors de l’encodage une classe APPLICATION. Cette classe risque d’être redondante, avec celles définies par les autres encodages.

#### 8.4.2.2 Les processus application

Au quadrant inférieur droit de la Figure 8-3, nous retrouvons les *planned\_process\_application*. Ils sont conçus à partir des modèles pour encoder la logique de leur *plan\_specification*. Ce sont eux qui utilisent ce même *plan\_specification* par la concrétisation suivie de la réalisation. Ce lien

de concrétisation et de réalisation d'utilisation ne sert qu'à marquer quelle instance de `plan_specification` gouverne quel `processus_application`. Ce lien est implanté par l'axiome suivant :

`SJ_lecture_gélose-sang_V7`  $\equiv$  `lecture_gélose`  
**and realizes some** (*concretizes value* 'SJ\_plan\_lecture\_gélose-sang\_V7')

Cet axiome est fort, il affirme que tout processus de lecture de gélose qui réalise la concrétisation du plan 'SJ\_plan\_lecture\_gélose-sang\_V7' est une instance de `SJ_lecture_gélose-sang_V7`. Nous pouvons aussi exprimer l'axiome par: 'SJ\_plan\_lecture\_gélose-sang\_V7' ne peut être en lien de concrétisation et de réalisation avec un processus de `lecture_gélose` autre que `SJ_lecture_gélose-sang_V7`.

Nous avons aussi l'axiome suivant

`SJ_lecture_gélose-sang_V7`  $\equiv$  `lecture_gélose`  
**and realizes only** (*concretizes value* 'SJ\_plan\_lecture\_gélose-sang\_V7')

Le quantificateur **only** signifie que `SJ_lecture_gélose-sang_V7` ne peut être gouverné par quoi que soit d'autres que `SJ_plan_lecture_gélose-sang_V7`.

Cela n'empêche pas une autre étape du processus de culture d'avoir en entrée obligatoire 'SJ\_plan\_lecture\_gélose-sang\_V7'. Par exemple pour le processus `prescription`, nous voulons disposer des informations sur la préparation du patient pour un test. Ces informations se trouvent, par convention, dans les `plan_specification` pour l'étape `prélèvement`. Pour `OntoMic`, il y a une différence entre utiliser des informations et être gouverné par ces mêmes informations.

### 8.4.3 L'encodage en langage machine des logiques, les balises

Comme nous l'avons vu, les exigences des protocoles sont encodées en spécifiant les entités matérielles qui sont acteurs ou qui subissent le `planned_process`, ou encore en spécifiant les informations produites ou utilisées par le processus.

`OntoMic` implante ces notions avec les relations *has\_specified\_input* et *has\_specified\_output* issues d'OBI. L'entrant qui est désigné par *has\_specified\_input* est une entité qui n'est pas créée

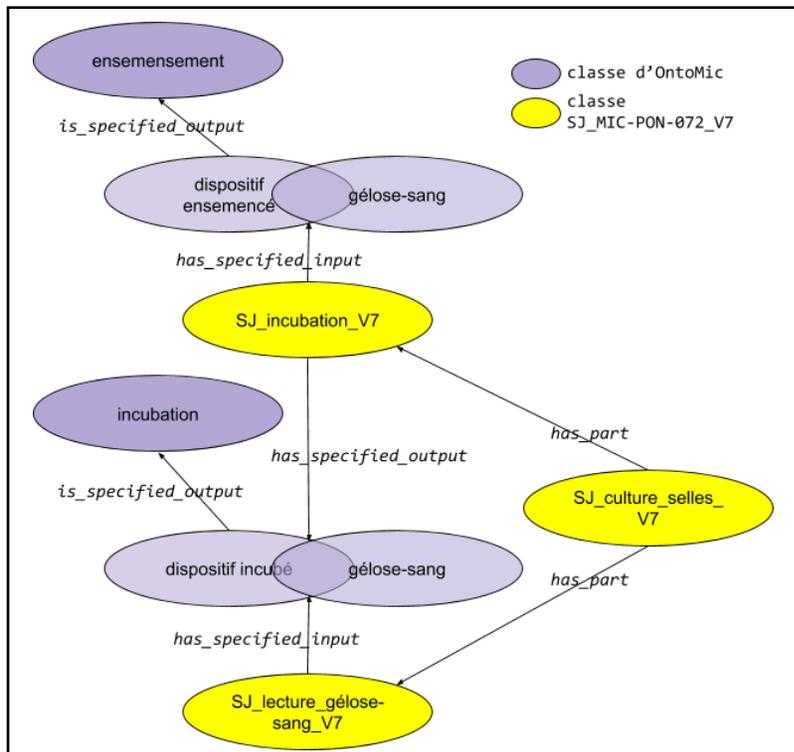


Figure 8-4., exemple d'une balise sur un élément matériel

par le processus et qui est essentielle à sa conduite. Un sortant est créé ou modifié par le *planned\_ - process* et sa présence à la fin d'un processus est considérée essentielle à la réussite du *planned\_ - process*.

Notons que les relations *has\_specified\_input* et *has\_specified\_output* ne peuvent être conditionnelles, les entités cibles de ces relations doivent **toujours** être présentes. Nous ne pouvons pas signaler que les *Salmonelles* sont des organismes que la culture de selles est à même de retrouver en écrivant *culture\_de\_selles has\_specified\_output Salmonelle*. En effet, la plupart des cultures de selles n'auront pas en sortie une *Salmonelle*.

Pour encoder un élément en sortie conditionnelle, il nous faut recourir à des axiomes complexes que nous n'illustrerons pas ici.

#### 8.4.3.1 Les balises sur les éléments matériels.

Le diagramme de la Figure 8-4 illustre des balises qui imposent des éléments en entrée.

Ces balises sont les plus simples. Comme nous l'avons vu, il s'agit de préciser la nature de l'élément et de spécifier s'il est en entrée ou en sortie. Nous avons vu, informellement, la balise qui imposait la présence en entrée de `SJ_lecture_gélose-sang_V7` d'une gélose-sang incubée. En voici sa représentation en OWL.

- 1) `dispositif_incubé`  $\equiv$  `dispositif_de_culture`  
**and** *is\_specified\_output\_of* **some** incubation
- 2) `gélose-sang_incubée`  $\equiv$  `gélose-sang` **and** `dispositif_incubé`
- 3) `SJ_lecture_gélose-sang_V7`  $\sqsubseteq$  `lecture_gélose`  
**and** *has\_specified\_input* **some** `gélose-sang_incubée`

Les énoncés 1) et 2) sont pour la `defined_class` pour le `material_entity` qui est en entrée. Le premier se veut très général, il définit tous les dispositifs de culture incubés, que ce soit une gélose, un milieu liquide dans un tube, etc. La présence de l'opérateur  $\equiv$  pour cet énoncé 1) signifie que tout dispositif de culture qui est en sortie d'une incubation quelconque est un dispositif incubé.

Par la suite, à l'énoncé 2), nous définissons par intersection `gélose-sang_incubée`. Ce recours à l'intersection permet de ne pas avoir à définir pour chaque gélose une version incubée.

Finalement, à l'énoncé 3) nous imposons la balise comme telle à `SJ_lecture_gélose-sang_V7`. Notons l'utilisation de l'opérateur  $\sqsubseteq$  au lieu de  $\equiv$ . Ceci signale que la balise est nécessaire, mais non suffisante pour définir `SJ_lecture_gélose-sang_V7`. Il peut y avoir des `lecture_gélose` qui ont une `gélose-sang_incubée` en entrée et qui pourtant ne sont pas des `SJ_lecture_gélose-sang_V7`.

#### 8.4.3.2 Les balises sur des `measurement_datum` scalaires

Nous savons que les températures d'incubation sont cruciales en microbiologie, construisons donc un exemple de classe pour la prise de température d'une étuve.

##### 8.4.3.2.1 La définition du `measurement_datum`

Directement d'OBI nous avons les classes et énoncés suivants :

measurement\_datum  
  scalar\_measurement\_datum  
    temperature\_measurement\_datum

quality  
  quality\_of\_single\_physical\_entity  
    temperature

4) temperature\_measurement\_datum  $\equiv$  scalar\_measurement\_datum  
  **and has\_unit\_label some** temperature\_unit  
  **and is\_quality\_measurement\_of some** temperature

La relation *is\_quality\_measurement\_of* provient d'OBI et est une sous-relation de *is\_about*. Nous avons déjà vu le concept de sous-relation à la section 8.2.1.1 avec la relation *denotes*. Cette définition d'une sous-relation spéciale n'apporte pas de précision par rapport à *is\_about*, car la nature des entités en relation est précisément définie. Ceci nous semble un accroc au formalisme de BFO.

Nous devons maintenant définir l'indépendant\_continuant qui porte la *temperature*. Dans ce cas, la température du métal à la surface extérieure de l'étuve n'est pas celle qui nous intéresse, mais bien la température à l'intérieur de la chambre d'incubation. Pour ce faire, nous utilisons les *universal atmosphere* et *incubation\_chamber* d'OBI et la relation *located\_in*. La relation *located\_in* signifie que la totalité de l'entité matérielle de gauche se trouve à l'intérieur de celle de droite. L'entité *étuve* est définie par *OntoMic*.

5) environnement\_d'incubation  $\equiv$  atmosphere *located\_in* **some**  
  (incubation\_chamber *located\_in* **some** étuve)

6) mesure\_température\_étuve  $\equiv$  temperature\_measurement\_datum  
  **and is\_about some** environnement\_d'incubation

L'énoncé 6) fait une combinaison avec la classe mère *temperature\_measurement\_datum*, donc *mesure\_température\_étuve* doit donc répondre aux conditions de l'énoncé 4) en plus d'avoir la condition *is\_about some* environnement\_d'incubation. Ceci est équivalent à l'énoncé suivant :

mesure\_température\_étuve  $\equiv$  scalar\_measurement\_datum  
    **and** *has\_unit\_label* **some** temperature\_unit  
    **and** *is\_quality\_measurement\_of* **some** temperature  
    **and** *is\_about* **some** environnement\_d'incubation

Il est à noter que l'axiome n'**impose pas** que la température mesurée soit bien celle d'un environnement d'incubation, il le **signale**. Nous pourrions mettre *mesure\_température\_étuve* en sortie d'un processus de prise de température d'un réfrigérateur et il n'y aurait pas d'erreur signalée ! Mais encore une fois, nous ne nous intéressons pas à valider des instances de *mesure\_température\_étuve*, mais à comparer les axiomes, les balises, entre différentes mesures. Nous avons un autre exemple d'une telle logique « incomplète » à la section 9.2.2.2.

#### 8.4.3.2.2 La construction de la norme avec le *measurement\_datum* et des critères

OBI définit les classes *value\_specification* et *scalar\_value\_specification*, c'est OntoMic qui ajoute les classes *critère\_valeur* et *critère\_unité*.

*value\_specification*  
    *categorical\_value\_specification*  
        *scalar\_value\_specification*  
            *critère\_valeur*  
                *critère\_35\_à\_37*  
                *critère\_unité*  
                *critère\_degré\_Celsius*

Les énoncés suivants donnent la formalisation en OWL des critères. 'degree\_Celsius' est une instance de la classe *temperature\_unit* qui provient d'une ontologie partenaire : Unit Ontology du groupe OBO.

*critère\_35\_à\_37*  $\equiv$  *critère\_valeur* **and** (*has\_value* **some** [ $\geq$  35.0 ,  $\leq$  37.0])  
*critère\_degré\_Celsius*  $\equiv$  *critère\_unité* **and** (*has\_unit\_label* **value** 'degree\_Celsius')

Nous savons que s'il y a une norme, cela implique nécessairement qu'il y a un processus quelconque de mesure, même si ce processus est en dehors du processus d'incubation. Nous appliquerons donc le critère à une mesure générique dans l'énoncé suivant :

norme\_sur\_temperature\_étuve\_35\_à\_37  $\equiv$  critère\_35\_à\_37  
**and** critère\_degré\_Celsius  
**and is\_about some** mesure\_température\_étuve.

Finalement nous appliquons la norme pour en faire une balise :

SJ\_incubation\_V7  $\sqsubseteq$  incubation  
**and has\_specified\_input some** norme\_sur\_temperature\_étuve\_35\_à\_37

La norme est modélisée comme une classe alors que logiquement il pourrait fort bien s'agir d'un individu. Mais comme nous le verrons à la section 9.2, ceci est nécessaire pour pouvoir se servir de l'inférence logique automatique pour comparer les normes. Nous pouvons concevoir ce recours à une classe comme si, à chaque fois qu'une incubation est faite, il se fait en même temps une application de la norme et qu'ainsi une instance de la norme est créée.

Finalement, même avec ce type de construction, nous ne pouvons pas bâtir la norme uniquement avec des éléments prédéfinis d'OntoMic. En effet, le langage OWL ne permet pas l'utilisation de paramètres à la place des nombres 35 et 37. Au moment de l'encodage, il faudra créer de nouveaux critères avec les bonnes valeurs en adaptant le modèle.

#### **8.4.3.3 Les balises sur des measurement\_datum catégoriels**

La Figure 8-5 illustre l'encodage de la balise sur le décompte de *S. aureus* qui nous provient de notre exemple d'encodage par formulaire de la section 6.1.

Dans le diagramme de la Figure 8-5 la direction des relations est importante. En effet, comme nous l'avons vu dans la description des tâches conditionnelles à la section 8.3.1.2.1, certaines relations doivent être renversées. Par exemple pour l'isolat de *S. aureus* et son décompte, si aucune colonie suspecte n'est retrouvée sur la gélose ni l'isolat ni son décompte n'existeront.

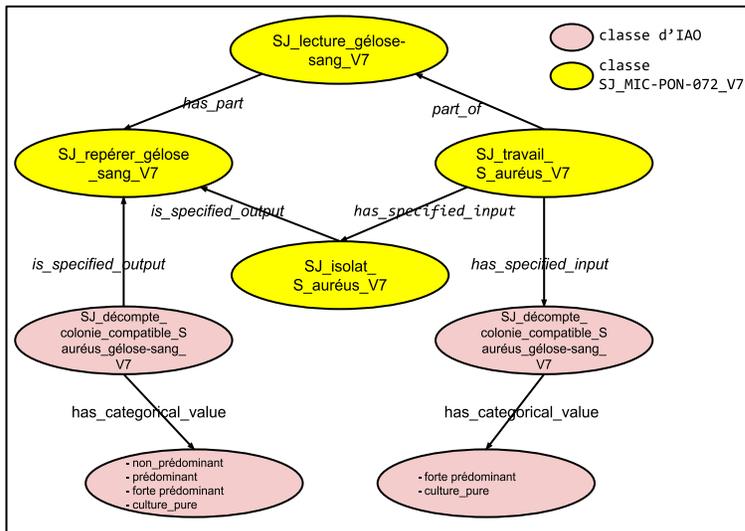


Figure 8-5, un exemple de balise sur une mesure catégorielle

La balise sur le *S. auréus* est différente de celle sur la température d'une étuve décrite à la section précédente. La première différence est qu'il s'agit d'une mesure catégorielle plutôt qu'une mesure scalaire. Une seconde différence est qu'il s'agit d'une norme sur une mesure produite à l'interne du processus. Cette mesure est différente aussi parce qu'elle pose plusieurs défis de représentation. Un de ces défis est que nous n'avons pas de représentation d'ensemble et de cardinalité en BFO. Un autre défi réside dans la représentation des éléments de « l'ensemble » dont nous voulons mesurer la cardinalité. Ces éléments ne sont pas des cellules bactériennes uniques, mais des colonies. Nous devons représenter les colonies en gardant un lien vers les bactéries qui les composent.

Commençons par définir ce qui est mesuré. Il s'agit d'un décompte de colonies, colonies qui ont une apparence commune qui est compatible avec celle de colonies de *S. auréus*. Pour définir ce décompte, il nous faut définir la notion de colonies, mais aussi celle de morphologie de colonie.

#### 8.4.3.3.1 La modélisation de l'entité colonie

Nous avons expliqué à la section 0 ce qu'était une colonie : un amas d'organismes qui découlent de leur prolifération. Nous avons omis dans cette définition qu'une colonie est le résultat de la multiplication à partir d'un **seul** exemplaire d'organisme.<sup>35</sup>

Une colonie est un **universal**, nous pouvons en observer en dehors du champ de la microbiologie, sur du fromage en train de moisir par exemple. Il ne s'agit donc pas d'une **defined\_class**. Nous n'avons donc qu'à donner une règle **nécessaire** à l'appartenance à la classe **colonie**, sans avoir à viser que cette règle soit **suffisante**. Traduit en terme axiomatique, cela veut dire que nous pouvons nous contenter de :

colonie  $\sqsubseteq$  « l'ensemble des instances qui obéissent à l'axiome »

plutôt que :

colonie  $\equiv$  « l'ensemble des instances qui obéissent à l'axiome ».

Nous nous intéressons non pas aux éléments individuels de la colonie, mais aux caractéristiques communes qu'ils partagent.

Nous pourrions tenter de définir chaque colonie comme une classe. Mais c'est logiquement impossible. De nouvelles classes s'ajouteraient pour chaque instance de culture, ce qui n'est permis ni par OWL ni par BFO. Nous avons opté pour une modélisation avec une structure déjà présente dans BFO.

Comme nous l'avons déjà vu à la section 4.4.2, BFO comprend la classe **objet\_aggregate**, dont les instances sont des assemblages de **member**. Ces **member** doivent être des instances **d'une** autre classe, donc des individus ayant certaines caractéristiques communes. La cardinalité n'est pas définie, par choix de BFO, pour un **objet\_aggregate**. La définition affirme que le retrait d'un individu de l'assemblage ne change pas l'essence de l'**objet\_aggregate**. Nous avons vu qu'**independant\_continuant** pouvait se transformer sans perdre son identité fondamentale, une

---

<sup>35</sup> Ce détail est important du point microbiologique, car il implique une certaine homogénéité génétique des organismes qui composent la colonie.

personne reste la même malgré la perte d'un bras. Le retrait d'un membre d'un `objet_aggregate` va plus loin, non seulement l'objet demeure le même, mais le changement n'est perceptible qu'au prix d'un effort. Par exemple, le retrait d'une seule pierre d'un amas n'est pas décelé sans un laborieux décompte.

Les colonies correspondent très bien à des `objet_aggregate`, le retrait d'une seule bactérie ne change pas la colonie. Pour qu'un changement soit visible, il faut que des milliers de bactéries meurent<sup>36</sup>. Aussi la notion de caractéristiques communes des membres d'un `objet_aggregate` permet de relier les colonies aux organismes qui les composent.

En modélisant les colonies comme des `objet_aggregate`, nous n'avons plus de limite aux nombres de colonies que nous pouvons représenter. Conceptuellement, chacune des colonies sur chacune des géloses dans le monde peut avoir son `individual` qui la représente.

Voici l'axiome nécessaire pour une colonie, notons l'utilisation de l'opérateur  $\sqsubseteq$

`colonie`  $\sqsubseteq$  `objet_aggregates`  
**and** *has\_member* **some** organisme  
**and** *has\_member* **only** organisme

Et avec l'adaptation de l'énoncé précédent pour le *S. auréus*

`colonie_S.auréus`  $\equiv$  `colonie`  
**and** *has\_member* **some** S.aureus  
**and** *has\_member* **only** S.auréus

Notons dans ce cas l'usage de  $\equiv$ , en effet l'axiome dit que `colonie_S.auréus` est d'abord une colonie. La condition qui suit n'a qu'à donner la caractéristique suffisante pour qu'une colonie soit une `colonie_S.auréus`.

---

<sup>36</sup> On voit ceci en clinique lorsque les bactéries au centre et au sommet de la colonie meurent, laissant ce qui est nommé une colonie ombiliquée.

Dans la définition de *colonie*, le mot clé **only** est présent seulement parce la classe sert de modèle, il est bien peu restrictif. Par contre dans la définition de *colonie\_S.aurés*, le quantificateur **only** a une grande importance ; il signifie que rien d'autre que des *S.aureus* peuvent être des *has\_member* de *colonie\_S.aurés*.

Mais il manque une partie de la logique à cet axiome. L'énoncé précédent n'exprime pas l'origine clonale, mais nous avons mentionné que nous n'avions pas à définir une condition suffisante pour la classe *colonie*. Comme OntoMic n'a pas recours à la caractéristique clonale d'une colonie, l'axiome choisi ne la reflète pas.

#### 8.4.3.3.2 La modélisation de l'aspect des colonies

Finalement, nous savons que le décompte ne se fait pas sur toutes les colonies, mais bien sur celles « qui ont l'air du *S. aurés* ». Nous avons vu à la section 6.2.1 que repérer sur une gélose les colonies suspectes d'être un organisme pathogène est une tâche cruciale en microbiologie. Nous ne pouvons pas faire l'économie de la modélisation d'un concept comme « ont l'air du *S. aurés* ».

Pour ce faire, nous définissons des classes pour « l'air de », que nous exprimons par le vocable : morphologie. Encore une fois, OntoMic fonctionne en classe. Nous ne définissons **pas une** instance « morphologie colonie » qui s'applique à **un groupe** de colonies. Au contraire, **chaque** colonie a **sa propre** instance de *morphologie\_colonie*. Cela reste près de la réalité, chaque colonie est unique, aucune n'est exactement pareille à une autre. Chaque colonie a donc sa morphologie, ce qui n'empêche pas de regrouper ces morphologies selon des caractéristiques communes.

quality  
morphologie\_colonie

morphologie\_colonie  $\sqsubseteq$  quality **and inheres\_in some** colonie

Cet énoncé veut dire que les instances de *morphologie\_colonie* sont des *quality* et que chacune est rattachée à une, ou à des, instances de *colonies*. Ce n'est pas tout à fait la correspondance une à une de la définition en langue naturelle. L'axiome est quand même satisfait si une seule instance de *morphologie\_colonie* est attachée à plusieurs instances de *colonies*. Nous pourrions implanter

un axiome plus complexe qui imposerait cette exigence d'une relation une à une, mais avoir que morphologie\_colonie est nécessairement attachée à une colonie est suffisant.

L'aspect des colonies dépend de l'organisme qui la compose, mais aussi de la gélose<sup>37</sup>. Il faut donc une option pour relier une gélose à la définition, ce qui se traduit par l'énoncé :

quality

morphologie\_colonie

morphologie\_colonie\_spécifique\_dispositif

morphologie\_colonie\_spécifique\_dispositif  $\sqsubseteq$  quality

**and inheres\_in some** (colonie *adjacent\_to* **some** dispositif\_de\_culture)

La relation de BFO : *adjacent\_to* impose le fait que, la morphologie est valable lorsque la colonie se trouve « adjacente » au dispositif de culture prévu. Encore une fois, ceci n'est pas totalement précis, la colonie se trouve à la surface (*adjacent\_to*) non pas du dispositif, mais du milieu de culture qui est **contenu dans** le dispositif. Dans ce cas, nous n'avons besoin que de relier la morphologie à un dispositif de culture, l'axiome est donc suffisant.

Une fois adapté pour le *S. auréus* sur gélose-sang nous avons :

morphologie\_colonie\_S.auréus\_sur\_gélose-sang  $\equiv$

morphologie\_colonie\_spécifique\_dispositif

**and inheres** (colonie *adjacent\_to* **some** gélose-sang)

Mais ceci ne nous dit pas quelles sont les caractéristiques de cette morphologie. Nous pourrions préciser cette morphologie en lui donnant des *has\_part* comme dans l'énoncé suivant :

---

<sup>37</sup> Souvent se confond en microbiologie l'apparence de la colonie et l'apparence du milieu de culture au pourtour de la colonie. Par exemple, une colonie est dite lactose positive lorsqu'elle induit un changement du milieu de culture par fermentation du lactose. Donc morphologie\_colonie\_spécifique\_dispositif devra être modifié pour comprendre l'apparence du milieu au pourtour de la colonie.

morphologie\_colonie\_S.auréus\_sur\_gélose-sang  $\sqsubseteq$  *has\_part* **some** yellow  
**and** *has\_part* **some** opaque  
**and** *has\_part* **some** circular  
**and** *has\_part* **some** smooth

Notons qu'il ne s'agit pas d'une instance 'yellow', mais du regroupement dans une **classe** yellow des teintes que prend les instances de morphologie\_colonie\_S.auréus\_sur\_gélose-sang. Nous affirmons que ces teintes ont la caractéristique commune de pouvoir être qualifiées de yellow. Les classes yellow, opaque, circular et smooth proviennent de l'ontologie Phenotype And Trait Ontology (PATO), toujours du groupe OBO.

Finalement, notons que

colonie\_S.auréus  $\equiv$  colonie  
**and** *has\_member* **some** S.aureus  
**and** *has\_member* **only** S.auréus

est différent de

colonie\_morphologie\_compatible\_S.auréus  $\equiv$  colonie  
**and** *bearer\_of* **some** morphologie\_colonie\_S.\_auréus\_sur\_gélose-sang

La première définition est pour la classe de toutes les colonies composées de S.auréus tandis que la deuxième est celle des colonies qui ont « l'air » d'être composée de S.auréus. Tout ce que nous pouvons affirmer sur ces deux classes c'est que leur intersection n'est pas vide, il existe dans le monde au moins une colonie composée de S.auréus qui en a l'apparence typique.

#### 8.4.3.3.3 La définition du measurement\_datum : décompte\_colonie

Nous avons la hiérarchie de classes suivante.

measurement\_datum  
  categorial\_measurement\_datum  
    dénombrement\_semi-quantitatif  
      décompte\_sur\_gélose\_semi-quantitatif  
        SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang

Rappelons qu'OntoMic définit les valeurs que peut prendre un categorial\_measurement\_datum par des échelles.

categorial\_label  
échelle  
  échelle\_semi\_quantitative  
  échelle\_prédominance

échelle\_par\_prédominance  $\equiv$  {'absence', 'rare', 'non\_prédominant',  
'prédominant', 'fort\_prédominant', 'culture\_pure'}

décompte\_sur\_gélose\_semi-quantitatif  $\sqsubseteq$  categorial\_measurement\_datum  
  **and** *has\_categorical\_label*   **some** échelle\_semi\_quantitative  
  **and** *is\_about*                   **some** gélose\_incubée  
  **and** *is\_about*                   **some** colonie

Nous ne pouvons pas définir la *quality* comme nous l'avons fait pour *temperature\_-measurement\_datum* à la section 8.4.3.2.1, car la notion de cardinalité n'est pas incluse dans BFO. Mais OntoMic peut se passer de la *quality* pour les décomptes. En effet, nous savons qu'il s'agit d'une mesure de cardinalité par la classe mère et nous savons par les relations *is\_about* qu'il s'agit d'un décompte sur des colonies sur une gélose.

SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang  $\equiv$   
  décompte\_sur\_gélose\_semi-quantitatif  
  **and** *has\_categorical\_label* **some** échelle\_predominance  
  **and** *is\_about* **some** colonie\_morphologie\_compatible\_S.auréus  
  **and** *is\_about* **some** gélose-sang\_incubéed

Il n'y aura bien sûr pas de décompte si nous ne retrouvons pas de colonies compatibles avec du *S. auréus* sur la gélose-sang. Alors, comme dans le cas de *SJ\_travail\_sur\_S.auréus\_V7* décrit à la section 8.3.1.2.1 nous aurons l'axiome « inversé »

SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose\_sang  $\sqsubseteq$   
  *is\_specified\_output\_of* **some** SJ\_repérage\_gélose-sang\_V7

#### 8.4.3.3.4 La construction de la norme

La démarche est comparable à la norme sur la température d'incubation. Dans le cas présent cependant, il s'agit d'une norme sur une entité mesurée par le processus. Nous avons donc à gérer la norme et la mesure sur laquelle elle s'applique.

value\_specification  
  scalar\_value\_specification  
    categorical\_value\_specification  
      critère\_semi-quantitatif  
      critère\_forte\_prédominance\_culture\_pure

critère\_semi-quantitatif  $\equiv$  categorical\_value\_specification  
  **and** *has\_category\_value some* categorical\_value\_specification

Avec l'axiome qui suit, nous retirons les instances d'échelle prédominance qui ne sont pas dans les valeurs ciblées par le protocole.

critère\_forte\_prédominance\_culture\_pure  $\equiv$  critère\_semi-quantitatif  
  **and** *has\_category\_value some* échelle\_prédominance \  
    {'absence', 'rare', 'non\_prédominant', 'prédominant'}

SJ\_norme\_décompte\_S.auréus\_gélose\_sang  $\equiv$   
  critère\_forte\_prédominance\_culture\_pure  
  **and** *is\_about some* SJ\_décompte\_S.auréus\_gélose\_sang

Pour imposer la norme nous avons

SJ\_travail\_S.auréus\_V7  $\sqsubseteq$  travail\_colonie\_intérêt  
  **and** *has\_specified\_input some*  
    (SJ\_décompte\_S.auréus\_gélose-sang  
      **and** critère\_forte\_prédominance\_culture\_pure)

## 8.5 L'ordonnement des tâches en OntoMic

Nous avons vu implicitement des moyens que prend OntoMic pour modéliser les contraintes sur le flot du travail entre les sous-tâches. Dans cette section nous allons expliquer cette méthode qui est inhabituelle, comme l'est l'encodage par balises.

Ni BFO ni OBI n'ont été conçus pour représenter des algorithmes. Des relations comme *precedes* et *immediately\_precedes* sont présentes dans BFO pour mettre un ordre sur des *process*. Mais ceci demeure rudimentaire. Ceci a conduit un groupe à développer un autre formalisme qui reprend les classes d'OBI, mais qui ajoutent un aspect algorithme (Giraldo et al., s. d.).

Cependant, nous croyons que les structures d'OBI sont suffisantes pour encoder les flots de tâches sans utiliser les relations de type *precedes*. Nous pensons que cet encodage comporte même des avantages certains par rapport à un encodage classique.

### **8.5.1 Approche de programmation fonctionnelle**

Les structures d'OBI sont suffisantes pour représenter l'ordonnancement si nous adoptons une approche analogue à la programmation fonctionnelle. Il est à noter qu'OBI implante également cette approche sans toutefois le mentionner explicitement.

#### **8.5.1.1 Déclenchement des *planned\_process***

Avec cette approche, nous définissons par balises les conditions de départ pour chaque *planned\_process* et il est réputé être prêt à être déclenché dès que ses conditions de départ sont remplies. Conceptuellement, un *planned\_process* se déclenche en produisant une instance de sa classe, donc, par notre analogie par formulaires, en imprimant le formulaire approprié pour la tâche et en donnant un identificateur unique à ce formulaire.

Nous ne nous intéressons pas à remplir ces formulaires, mais bien à les comparer. De façon analogue, nous ne nous intéressons pas aux instances de *planned\_process*, mais aux conditions d'activation, elles serviront de base de comparaison.

Nous avons vu une illustration de ce type de condition d'activation à la Figure 8-4 pour *lecture\_gélose-sang\_SJ\_V7* où nous exigeons en entrée une *gélose-sang* qui était elle-même en sortie d'une étape d'incubation.

#### **8.5.1.2 Précision sur le déclenchement, les *planned\_process* invités**

Il y a une différence entre l'encodage en OntoMic et notre analogie de formulaire. Avec notre analogie, c'est le formulaire que l'on est en train de remplir qui en appelle un autre selon une règle. Mais avec OntoMic, un *planned\_process* ne peut pas en « appeler » un autre et ceci peut causer des problèmes.

Voyons un exemple de ces problèmes avec la phrase « Pour le Staph aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure ». Il y a plusieurs méthodes pour faire ce dépistage de résistance et ici le texte mentionne l'utilisation d'une méthode standard qui nécessite 24 heures d'incubation. Il y a cependant des méthodes plus rapides, comme le TAAN, ces méthodes sont souvent onéreuses et l'on voudra en faire un usage judicieux.

Supposons le protocole modifié suivant :

« Pour le Staph.aureus, si forte prédominance ou culture pure faire le dépistage de la résistance à l'oxacilline par la méthode TAAN si le patient est hospitalisé et par la gélose oxacilline autrement. »

Au laboratoire, il y a un protocole pour le dépistage en gélose et un autre pour le dépistage par TAAN. Ces procédures sont des exemples d'outils généraux auxquels plusieurs protocoles de traitements de spécimens font appel.

Si chaque `planned_process` est conceptuellement réputé être prêt à être déclenché dès que ses conditions de départ sont remplies, il faudra pour les outils généraux faire une liste de conditions de départ dans chaque `planned_process`. Dans le cas de notre exemple, il faudra ajouter au `planned_process` pour le dépistage par TAAN le fait d'avoir un *S. aureus* en provenance d'une selle chez un patient hospitalisé.

L'obligation de faire une liste exhaustive des usages que nous faisons d'un outil dans la description de l'outil lui-même est contre-intuitive. Le plus souvent on présente seulement quelques exemples

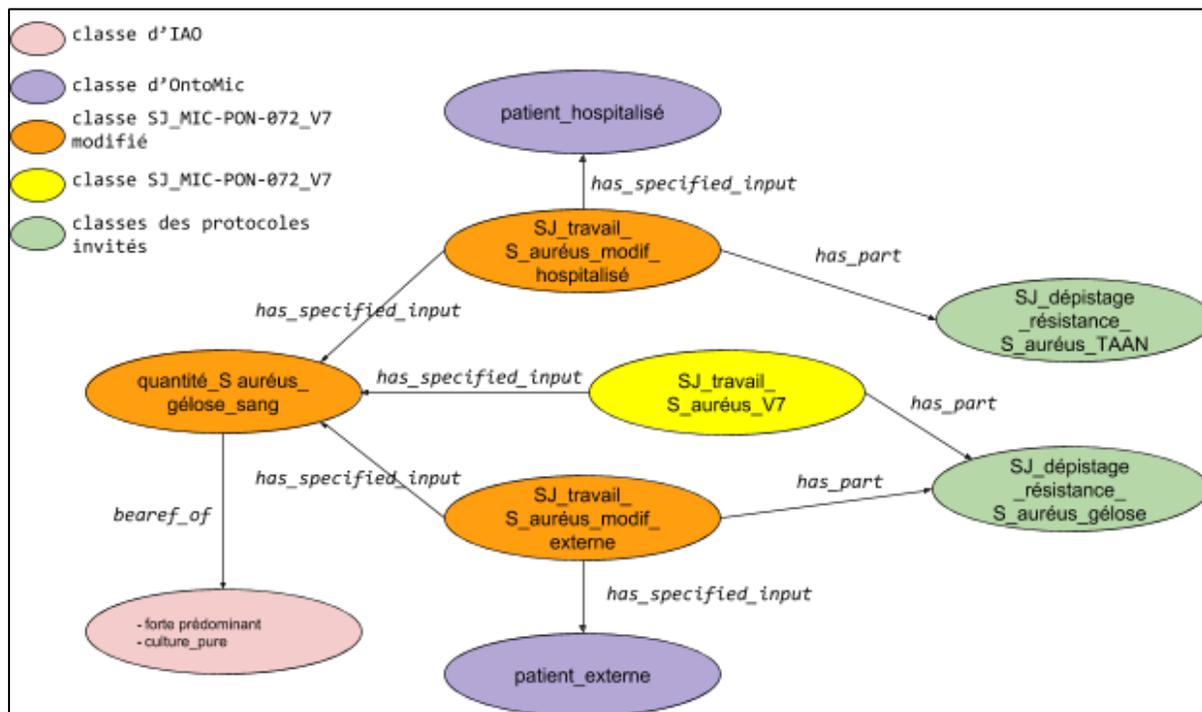


Figure 8-6, illustration du mécanisme de *planned\_process* invité

d'utilisations pour aider à la compréhension. Nous avons donc modifié notre règle sur le déclenchement des *planned\_process* pour implanter la notion de *planned\_process* invité.

Les *planned\_process* externes à un *test\_laboratoire\_clinique* doivent être « invités » par une relation *has\_part*. On modifie notre convention de déclenchement en stipulant que pour qu'un *planned\_process* se déclenche il faut que les conditions de déclenchement de toute sa chaîne ascendante de *part\_of* soient remplies.

C'est à la tâche qui invite de stipuler les conditions d'activation. Avec notre exemple, c'est au protocole appelant de spécifier que le patient doit être hospitalisé pour appeler la détection de la résistance par TAAN. Les conditions d'activation du processus invité demeurent inchangées et sont assez générales pour être applicable lors de toutes les invitations. Pour notre exemple le *planned\_process* outil : *SJ\_dépistage\_résistance\_S.auréus\_par\_TAAN* n'aura que l'exigence d'avoir un isolat de *S. auréus* en entrée, peu importe de quel milieu cet isolat provient, ou son décompte ou le statut du patient source.

#### 4) Gélose CIN :

Lire la gélose à 24 et 48 heures.  
Ensemencer les colonies avec centre rouge entouré d'une zone translucide (en œil de bœuf) suspectes de *Yersinia* sur MacConkey.  
Les colonies lactose négative sur MacConkey iront à l'identification complète.

Extrait de document 8-2, lecture de la gélose CIN selon le labo SJ

Source : SJ\_MIC-PON-072\_V7

Bien sûr, les mots « invité », « déclenché » ne sont utilisés que pour aider à comprendre la logique de flot, en réalité ces notions permettent de sélectionner les balises pertinentes lors de la comparaison de protocoles.

Avec notre exemple de protocole modifié nous avons le diagramme de la Figure 8-6.

## **8.5.2 Avantages de l'approche par programmation fonctionnelle**

### **8.5.2.1 Parallélisme**

La modélisation de flot par programmation fonctionnelle permet de représenter efficacement l'exécution en parallèle.

#### 8.5.2.1.1 Exemple d'une lecture intérimaire pendant une incubation

L'Extrait de document 8-2 montre l'exemple de la lecture de la gélose CIN. Pour cette gélose on demande une incubation totale de 48 heures, mais cette incubation doit être interrompue transitoirement pour faire une lecture à 24 heures. Pour modéliser ceci, il suffit de définir que la `lecture_gélose_CIN` accepte une gélose incubée 24 ou 48 heures. Bien sûr, ceci n'est pas tout à fait vrai, car pour être lue la gélose doit être retirée de l'incubateur et l'incubation est donc interrompue. Généralement, on ne tient pas compte de ce genre d'interruptions.

Voici l'encodage (simplifié) en OWL, pour les durées OBI utilise une la relation *is\_duration\_of* qui s'applique directement sur un `process`, cette relation n'est pas définie avec BFO, car il y a controverse sur la manière de représenter les durées.

```

mesure_durée_incubation  ≡  is_duration_of some SJ_incubation_CIN_V7
critère_cible_24          ≡  has_value      value 24
critère_cible_48         ≡  has_value      value 48
critère_heures            ≡  has_unit_label value 'hour'
norme_incubation_totale  ≡  critère_cible_48 and critère_heures
      and is_about some mesure_durée_incubation
norme_lecture_intérimaire ≡  critère_cible_24 and critère_heures
      and is_about some mesure_durée_incubation
SJ_lecture_gélose_CIN_V7 ≡  has_specified_input some gélose_CIN
      and has_specified_input some
      (mesure_durée_incubation
      and (norme_incubation_intérimaire or norme_incubation_totale))

```

Notons que dans ce cas nous avons dû relâcher la balise sur la gélose en entrée. En effet nous ne pouvons pas imposer gélose CIN incubée en entrée, car l'incubation peut ne pas être terminée avant de faire une lecture.

#### 8.5.2.1.2 Le Gram sur spécimen optionnel pour la culture de selles

Un autre exemple est : la lecture au microscope de la coloration Gram de la selle. Cette lecture est faite à Saint-Jérôme sur demande spéciale seulement. Elle peut se faire indépendamment de l'ensemencement, de l'incubation et de la lecture des milieux. En fait, la seule exigence est qu'elle doit être terminée avant la fin de la culture de selles et l'émission du rapport. Au lieu de décrire des embranchements complexes, nous encodons simplement le Gram comme nécessitant en entrée un spécimen et une requête avec une demande explicite de Gram. L'étape d'élaboration du rapport a, quant à elle, l'exigence d'avoir à sa disposition un résultat pour le Gram si la requête le demandait.

#### 8.5.2.2 Homologie avec l'organisation du travail en laboratoire

Il est intéressant de constater que, bien que ce modèle par programmation fonctionnelle soit bien théorique, il est plus proche de travail de laboratoire qu'un modèle par algorithme. En effet, dans un laboratoire, on ne travaille pas linéairement une seule culture, les interruptions du flot de travail sont fréquentes.

La technicienne a une liste des cultures à sa charge, elle passe la liste une à une. Lorsqu'elle travaille un spécimen, elle commence par lire le compte-rendu pour déterminer ce qui reste à faire. Puis elle effectue les tâches qu'elle peut faire de façon continue. Elle remet à plus tard les tâches qui brisent son flot de travail ou encore qui se font mieux par lot. Par exemple, les colorations de lames pour la microscopie se fait beaucoup plus efficacement par lots. La technicienne qui a travaillé le spécimen plus tôt durant la journée, celle qui fait la coloration de lame et celle qui lit la lame au microscope peuvent être toutes des personnes différentes.

### **8.5.2.3 Possibilité de faire des « super-protocoles »**

Nous pourrions vouloir faire des « super-protocoles », c'est-à-dire des protocoles qui ont préséance sur tout autre, par exemple pour la biosécurité. Ces super-protocoles n'auraient pas besoin d'être invités pour être déclenchés. Le super-protocole exige que le seul travail\_isolat\_intérêt possible pour des colonies ayant des caractéristiques compatibles avec un organisme dangereux est l'interruption du travail, la mise en quarantaine de la culture et la consultation avec le microbiologiste. Par traitement automatique, nous devrions être capables de déceler tout protocole qui demande un travail supplémentaire sans s'assurer d'avoir éliminé la possibilité de l'organisme dangereux.

## **8.6 Difficultés entre le modèle théorique et son application**

### **8.6.1 La frontière entre sujet et production des measurement\_datum**

Reprenons notre exemple de lecture de la gélose-sang telle que nous l'avons vue à la section 6.1. Cette tâche se sépare en deux sous-tâches soit le repérage des colonies d'intérêts et le travail sur ces colonies. Dans l'exemple il y avait deux types de colonies d'intérêts, celles qui sont compatibles avec de la flore fécale et celles qui sont compatibles avec du *S. auréus*.

Il y a donc deux décomptes SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang et SJ\_décompte\_colonie\_compatible\_flore\_fécale\_gélose-sang. Nous avons vu que la nature de ces deux mesures est d'être des décompte.

décompte

décompte\_colonie\_gélose

SJ\_décompte\_colonie\_compatible\_S.auréus\_gélose-sang

SJ\_décompte\_colonie\_compatible\_flore\_fécale\_gélose-sang

Il nous reste maintenant à en définir les sujets. Le sujet des `décompte_colonie_gélose` était initialement conçu pour être `ensemble_colonie_sur_gélose`. La première complication est que nous ne pouvons pas définir cet ensemble avec BFO. Nous avons donc convenu de remplacer `ensemble_colonie_sur_gélose` en définissant non pas un sujet pour `décompte_colonie_gélose`, mais deux : `colonie` et `gélose`.

Notre approche initiale était de ne jamais changer la nature et le sujet d'une mesure lorsque nous passons d'une mesure générale comme `décompte_colonie_gélose` à son adaptation pour un protocole particulier comme `SJ_décompte_colonie_compatible_S.auréus_gélose-sang` et `SJ_décompte_colonie_compatible_flore_fécale_gélose-sang`. Ce qui devait changer était uniquement le processus de production de la mesure.

Avec cette approche, les deux décomptes `SJ_décompte_colonie_compatible_S.auréus_gélose-sang` et `SJ_décompte_colonie_compatible_flore_fécale_gélose-sang` partagent les mêmes sujets `colonie` et `gélose`.

Cependant, cette approche est rapidement apparue irréaliste, car elle impose une lourdeur dans l'écriture des balises sur les `measurement_datum`. En effet, supposons que nous voulions comparer les protocoles pour savoir comment ils traitent le *S. auréus*. Il nous faut extraire toutes les tâches qui ont en entrée un `décompte_colonie` qui traite du *S. auréus*. Avec notre approche initiale, les sous-classes de `décompte_colonie_gélose` n'ont pas de marqueurs qui nous donnent le type précis de colonies, il faudra faire une chaîne d'implication qui passe par le processus de production du `décompte_colonie_gélose`.

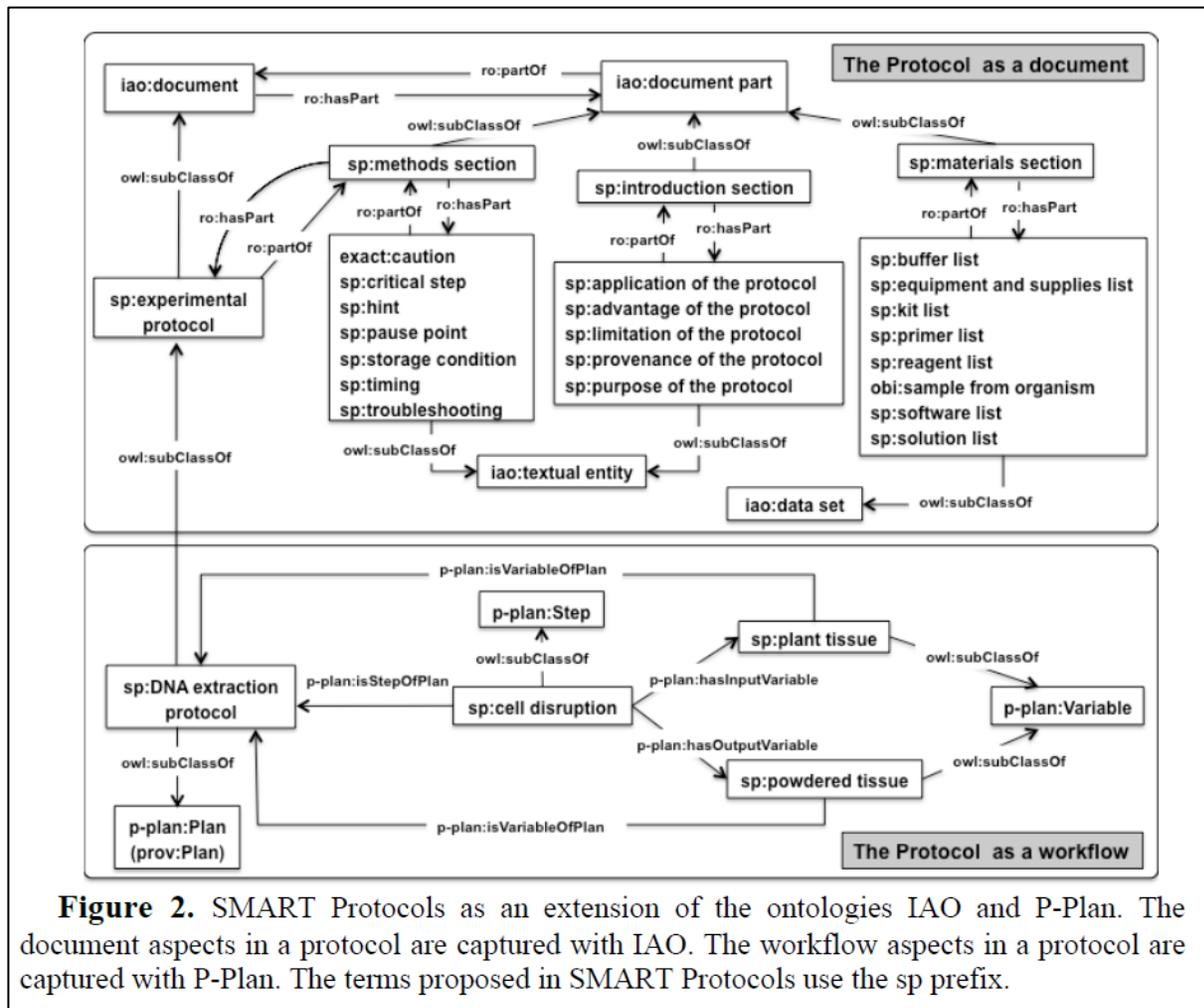
Cette chaîne d'implication est illustrée par la phrase : « Extraire toutes les tâches qui ont en entrée un décompte de colonie, décompte qui est lui-même en sortie d'un processus de repérage qui s'adresse au *S. auréus*. »

Nous avons donc dû abandonner notre principe de faire porter la précision sur le processus producteur. Le sujet de `SJ_décompte_colonie_compatible_S.auréus_gélose-sang` est donc `S.auréus` et `gélose-sang`. Notons que ceci est une répétition des informations au sein de la même ontologie. Une erreur de logique comme celle d'avoir en sortie d'un processus de repérage qui s'adresse aux levures un décompte qui a comme sujet `S.auréus` ne sera pas détectable automatiquement.

### **8.6.2 Plusieurs façons d'encoder un même protocole**

Reprenons l'exemple de la lecture intérimaire avant la fin de l'incubation de la gélose CIN que nous avons vue à la section 8.5.2.1. Notre choix d'encodage a été de donner en entrée de la tâche de lecture un critère sur la durée de l'incubation avec la possibilité pour déclencher la lecture à 24 et 48 heures, nous aurions pu faire un choix différent. Nous aurions pu représenter l'exigence du protocole pour une lecture intérimaire avec un encodage à l'étape d'incubation. Pour ce faire, définissons deux géloses en sortie de l'incubation soit une gélose intérimaire et une gélose finale, la lecture aura alors simplement à accepter chacune des deux géloses.

Ceci est troublant, car nous verrons plus loin que la comparaison automatique ne pourra pas considérer comme équivalentes ces deux manières d'encoder.



**Figure 2.** SMART Protocols as an extension of the ontologies IAO and P-Plan. The document aspects in a protocol are captured with IAO. The workflow aspects in a protocol are captured with P-Plan. The terms proposed in SMART Protocols use the sp prefix.

Figure 8-7, diagramme de l'ontologie SMART

Source: SMART Protocols: Semantic Representation for Experimental Protocols

## 8.7 Inspiration pour notre modèle

Notre découpage d'OntoMic en trois « couches » que nous avons vu à la Figure 8-1 a été inspiré de l'ontologie SMART protocol (Giraldo et al., s. d.). Cette ontologie voit les protocoles sur deux facettes : celle du texte lui-même et celle du workflow représenté par le texte. Le diagramme de la Figure 8-7 provient de l'article et illustre la vision des auteurs.

Cette modélisation avait cependant le désavantage de requérir une troisième ontologie et de ne pas avoir de représentation du « sens » des textes représentés par les plan\_specification.

Ce passage par des entités intermédiaires permet une meilleure représentation de la réalité. Par exemple, elle permet de gérer le fait que la répartition des informations dans les protocoles peut être d'une manière différente de celle d'OntoMic. En effet, notre modèle se base sur une vision longitudinale du processus de culture allant de la décision sur la nécessité du test, puis la prescription, pour se terminer à l'interprétation des résultats. Rien n'oblige les laboratoires à avoir la même organisation dans leurs documents. Un laboratoire pourrait, par exemple, avoir un recueil d'aide à l'interprétation des résultats qui s'adresse à toutes les cultures.

Également, nous n'avons pas choisi une approche d'ordonnement de tâches par algorithme comme nous l'avons vu à la section 8.5.

## **8.8 Conclusion sur le modèle OntoMic**

Plusieurs éléments étaient absents du modèle OBI, car les buts d'OBI et d'OntoMic sont différents, comme nous l'avons vu à la section 7.1.1. Voici une liste des modélisations qui ont été construites pour OntoMic :

- Modélisation des liens entre les textes et les encodages
- Constructions de balises à partir d'éléments prédéfinis d'OntoMic
- Refonte de la modélisation des mesures faites pendant les cultures
- Création de la notion de normes sur les mesures
- Stratégie d'ordonnement des tâches

Ces modélisations sont des innovations, la validité de la logique a été testée durant nos expériences d'encodages et d'inférences automatiques. Elles nécessitent cependant la création de plusieurs éléments pour l'encodage d'une seule balise et les protocoles comprennent des dizaines de balises.

Prenons toujours notre même exemple de la lecture de la gélose-sang tel que montré par l'Extrait de document 8-3, mais intéressons-nous maintenant au second paragraphe. Ce paragraphe est

Lire la gélose à 24 heures.

Noter sur la présence ou l'absence de la flore fécale.

Pour le Staph.aureus, faire un dépistage de la résistance sur la gélose oxacilline si forte prédominance ou culture pure.

Rechercher les colonies  $\beta$ -hémolytiques abondantes ou en prédominance, faire oxydase et indole spot sur ces colonies. Sous-cultiver les colonies + pour les 2 tests sur TSI (Aeromonas, Plesiomonas et Vibrio). Pour les colonies A/A et K/A, faire identification complète.

Extrait de document 8-3, traitement des isolats candidats selon le labo SJ

Source : SJ\_MIC-PON-072\_V7

« dense » en effet pour l'encoder il faudra créer plusieurs éléments : la représentation de l'isolat d'intérêt, la description des colonies d'intérêts ( $\beta$ -hémolytiques<sup>38</sup>), le décompte des colonies  $\beta$ -hémolytiques, la norme sur ce décompte, un travail sur colonies  $\beta$ -hémolytiques avec deux tests rapides (oxydase et indole spot), un résultat pour l'oxydase ainsi que pour l'indole spot, une norme sur chacun des résultats, une étape d'identification partielle avec le test TSI et un résultat pour le TSI avec sa norme. Treize éléments, sans compter que certains de ces éléments demanderont à leur tour la création d'autres éléments.

Pour qu'OntoMic soit réellement applicable pour l'encodage des protocoles il faut un instrument d'encodage. L'utilisateur doit pouvoir créer les éléments beaucoup plus rapidement qu'avec l'éditeur Protégé<sup>39</sup>.

---

<sup>38</sup> Une colonie  $\beta$ -hémolytique sur gélose-sang est entourée d'un halo clair transparent par rapport au reste du milieu qui est opaque. Ceci est due à la capacité des organismes qui composent la colonie de dégrader les globules rouges. C'est une illustration de l'abus de langage que nous avons vu à la section 8.4.3.3.2 qui attribue à la colonie l'apparence qui est en réalité celle du milieu de culture.

<sup>39</sup> Protégé un éditeur open-source pour les ontologies en OWL.

## Chapitre 9 L'utilisation des données encodées

La littérature sur les ontologies dans le domaine biomédical traite le plus souvent de l'encodage que de son utilisation pour des applications pratiques. Un article (Vita, Overton, Greenbaum, et al., 2013) élabore un peu sur l'utilisation d'un raisonneur avec Protégé pour faire du forage de données.

Il y a donc fallu développer nous-même une stratégie d'utilisation des données encodées. Plusieurs embûches ont été rencontrées que nous allons présenter dans cette section. Finalement nous allons voir comment ces embûches nous ont fait revoir notre stratégie, particulièrement en changeant le rôle des raisonneurs.

### 9.1 Faire des recherches avec OntoMic

OntoMic utilise le plus souvent des classes comme entités pour encoder, comme nous l'avons vu au chapitre précédent. Ceci rend les interrogations plus difficiles. En effet, nous utilisons des axiomes OWL sur ces classes construites pour l'encodage, mais OWL n'a pas de mécanisme d'interrogation d'axiomes. Faire de simples tâches comme établir des listes est difficile.

Prenons l'exemple de sélectionner\_dispositif\_culture, la tâche pour laquelle nous définissons les dispositifs pertinents pour la culture. Supposons, pour simplifier, que notre protocole SJ\_MIC-PON-072\_V7 ne demande que deux géloses : gélose-sang et MacConkey. Avec BFO, en langue naturelle, nous avons :

```
SJ_sélectionner_dispositif    has_specified_input  gélose-sang
SJ_sélectionner_dispositif    has_specified_input  gélose_MacConkey
```

Car pour BFO les relations entre classes existent. En OWL nous formulons la même logique avec l'axiome de restriction suivant.

```
7) SJ_sélectionner_dispositif_V7  $\sqsubseteq$  has_specified_output some gélose-sang
   and has_specified_output some gélose_MacConkey
```

Comme OWL n'a pas de mécanisme de recherche ou d'interrogation d'axiomes, comment faire pour lister automatiquement les dispositifs\_de\_culture ?

On serait tenté de faire

$SJ\_liste\_dispositif \equiv dispositif\_culture$   
**and** *is\_specified\_output* **some**  $SJ\_sélectionner\_dispositif$

Nous pourrions nous attendre à ce que le raisonneur arrive aux énoncés suivants (les énoncés sont en ■■■ car ils contiennent une erreur de logique) par vérification de la subsomption :

$gélose\_MacConkey \sqsubseteq SJ\_liste\_dispositif^*$   
 $gélose\_sang \sqsubseteq SJ\_liste\_dispositif^*$

Mais ces deux énoncés sont faux. Avant d'expliquer pourquoi ils sont faux, souvenons-nous que nos classes sont vides. Si nous faisons une analogie entre une classe et un vase, il n'y a pas besoin qu'il y ait des fleurs et de l'eau dans le vase pour étudier et aimer sa forme !

La classe  $SJ\_liste\_dispositive$  est à même de contenir seulement les géloses individuelles choisies pendant une culture selles à SJ. Mais il n'y a rien qui empêche qu'il y ait des instances de  $gélose\_MacConkey$  et de  $gélose\_sang$  qui ne soient pas dans  $SJ\_liste\_dispositif$ . Pour qu'une subsomption soit valide, il faut qu'elle demeure vraie pour **n'importe quelles** combinaisons d'instances. Ceci est fondamental, la subsomption ne demande pas seulement la relation d'inclusions soit vraie avec les instances présentes au moment de l'exécution du raisonneur. C'est ce que dans le monde des ontologies on nomme le raisonnement sur le « monde ouvert ».

Pour faire la recherche, nous devons utiliser des techniques comme des requêtes sur la sérialisation de l'ontologie. À l'Extrait de document 9-1 nous avons un exemple, simplifié, de sérialisation pour l'axiome 7) ci-dessus. Nous avons choisi ce format de sérialisation parce que c'est le plus facile à lire pour un humain :

```

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ontomic: <https://sites.google.com/levreauj/ontomic#> .

ontomic:SJ_sélectionner_dispositif_V7
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Class ;
    owl:intersectionOf (
      _:genid2
      _:genid4
    )
  ] .
_:genid2
  a owl:Restriction ;
  owl:onProperty ontomic:has_specified_output ;
  owl:someValuesFrom ontomic:gélose-sang.

_:genid4
  a owl:Restriction ;
  owl:onProperty ontomic:has_specified_output ;
  owl:someValuesFrom ontomic:gélose_MacConkey .

```

Extrait de document 9-1, encodage en triplets RDF d'une axiome.  
 Les triplets sont présentés avec la notation Turtle.

Nous voyons que faire des recherches lexicales ne sera pas aisé et ce sera toujours à la merci de variations de la sérialisation.

Il est possible, peu importe le langage de sérialisation du départ, de donner une représentation de l'ontologie avec de triplets RDF. En principe, mais cela reste à valider, le jeu de triplets RDF devrait être stable d'une sérialisation à l'autre et pour interroger le jeu de triplet RDF nous pouvons utiliser le SPARQL, un langage développé spécifiquement pour cette tâche.

Mais faire le code de ces recherches demande une expertise qui dépasse celle d'un utilisateur, même avancé, d'un éditeur d'ontologies comme Protégé. Même s'il y a des possibilités d'utiliser des modèles génériques de SPARQL comme l'ont fait une équipe qui a travaillé avec OBI (Vita et al., 2013). Ceci est un premier indice qu'il nous fallait s'éloigner d'un modèle où ce serait du

personnel technique spécialement entraîné qui ferait l'encodage et les recherches directement en manipulant l'ontologie. D'autres indices de ceci sont apparus lorsque nous avons tenté d'appliquer la subsomption pour comparer des protocoles, comme nous le verrons dans la section suivante.

## **9.2 L'utilisation des axiomes et de la subsomption**

Dans cette section nous allons introduire l'utilisation de la subsomption pour comparer les logiques, donc pour comparer les balises. Nous allons voir qu'il nous a fallu introduire un autre niveau de division des logiques procédurales et pourquoi il nous faut encadrer de façon très serrée comment sont fait les axiomes. Nous allons voir certains aspects du fonctionnement des axiomes et les conséquences sur l'inférence logique.

### **9.2.1 Les clauses**

Les textes complets et leurs liens avec leurs `plan_global_culture` sont moins importants, pour l'encodage, car la grande majorité des balises proviennent des extraits de texte et non du texte au complet. Donc, dans ce qui suit, nous ne nous intéresserons qu'aux `plan_étape_culture`.

Il est difficile d'encoder toute la logique procédurale d'un `plan_étape_culture` dans un seul axiome. Par exemple, dans l'Extrait de document 9-2, nous voyons qu'il faut faire un axiome qui désigne le choix de géloses, mais qu'il faut que cet axiome prenne en compte que la gélose CNA n'est pas faite dans tous les cas. Nous verrons à la section 9.2.4 pourquoi il est ardu de le faire dans un seul axiome.

Avec un écouvillon, déposer un inoculum léger du spécimen, dans l'ordre, sur les géloses suivantes :

- GS
- CNA (pour les enfants âgés de trois mois et moins)
- MacConkey Sorbitol
- MacConkey
- CIN

Ensemencer avec un inoculum moyen

- Gélose SS
- CCDA

Extrait de document 9-2, choix des géloses selon le labo CSL

Source : CSL\_MIC-PON-TSP-SELLE\_V7

Pour rendre plus étroite la correspondance entre les subdivisions des logiques procédurales et les axiomes nous avons défini les **clause**. Une **clause** est une entité de logique procédurale qui se représente par un seul axiome que nous voulons « simple ». Nous ajoutons qu'un seul extrait de texte *denotes* une **clause**. Ceci nous permettra de fractionner l'encodage et de garder un lien bidirectionnel entre extrait de textes et axiome.

Les **plan\_specification** sont donc un ensemble de **plan\_étape\_culture**, eux-mêmes à leur tour composés de **clauses**. Nous verrons au fur et à mesure de leur présentation les conditions pour qu'un axiome soit considéré assez « simple » pour être l'implantation d'une **clause**. De même nous verrons comment les axiomes des **clause** sont combinés pour faire un **plan\_étape\_culture**.

Reprenons l'exemple du protocole de la Cité de la Santé avec l'Extrait de document 9-2. Nous nous rappelons que ce protocole exige la recherche de *Streptocoque* du groupe B chez le nourrisson

Le texte de l'Extrait de document 9-2 est lié à **plan\_pour\_CSL\_sélectionner\_dispositif**, mais ce texte comprend plusieurs unités logiques, dont certaines ne sont pas reliées à la sélection des géloses. Il traite, entre autres, de l'ordre dans lequel on doit inoculer les dispositifs et de la quantité de selles qui doit être utilisée. Ces informations se rattachent à **plan\_pour\_CSL\_ensemencement**.

Quant à la sélection des dispositifs, nous voyons que gélose-sang (GS), MacConkey Sorbitol, MacConkey, CIN, SS et CCDA sont tous sélectionnés d'emblée sans condition supplémentaire.

Ceci forme donc un tout logique, c'est un exemple de **clause**. La sélection la gélose Cetrimide-Nalidixic-Acid (CNA), quant à elle, dépend de l'âge du patient, ce sera une autre **clause**.

Voici, en abrégé, l'axiome pour la sélection des dispositifs généraux.

```
CSL_sélectionner_dispositifs_généraux
  and has_specified_output some gélose-sang
  ...
  and has_specified_output some gélose_CIN
```

Pour le moment, intéressons-nous à l'ajout de la gélose CNA et nommons la tâche de sélection de cette gélose : CSL\_sélectionner\_dispositif\_culture\_pour\_Strep\_B. La clause sur la sélection de gélose CNA est encodée par :

```
critère_3_ou_moins  $\equiv$  value_specification
  and has_specified_value some [ $\leq$  3]
```

```
critère_mois  $\equiv$  value_specification
  and has_unit_label_value 'month'
```

```
CSL_mesure_âge_patient  $\equiv$  value_specification
  and is_quality_measurement_of some âge
  and has_unit_label_value 'month'
```

```
CSL_norme_âge_3_mois_ou_moins  $\equiv$ 
  critère_3_ou_moins
  and critère_mois
  and is_about âge
```

```
8) CSL_sélectionner_dispositif_culture_Strep_B  $\equiv$  sélectionner_dispositif_culture
  and has_specified_input some
    (CSL_norme_âge_3_mois_ou_moins and mesure_âge_patient)
  and has_specified_output some gélose_CNA
```

Dans le reste du chapitre nous allons reprendre l'exemple de la clause décrite par l'énoncé 8) ci-haut, avec des variantes, pour illustrer comment la subsumption fonctionne.

## 9.2.2 Particularité du raisonnement automatique sur les axiomes

### 9.2.2.1 Écriture des axiomes à comparer

À la section 4.5.2 nous avons vu qu'à droite des symboles  $\equiv$  ou  $\sqsubseteq$  se trouvent des éléments de la combinaison logique qui définissent une classe virtuelle. Lors de l'application du raisonneur et de la recherche de subsomption ce sont ces classes virtuelles qui seront examinées.

Admettons les classes virtuelles  $A'$  et  $B'$  avec les axiomes  $A \sqsubseteq A'$  et  $B \sqsubseteq B'$ , alors même si le raisonneur conclut que  $A' \sqsubseteq B'$  il ne pourra pas conclure que  $A \sqsubseteq B$ . En revanche, si  $A \sqsubseteq A'$  et  $B \equiv B'$  et que le raisonneur conclut que  $A' \sqsubseteq B'$  alors il pourra affirmer que  $A \sqsubseteq B$ .

Néanmoins pour éviter les résultats inattendus il est préférable de comparer par subsomption des classes qui sont définies par des conditions nécessaires et suffisantes, autrement dit il est préférable d'avoir l'opérateur  $\equiv$  et non  $\sqsubseteq$ .

Cette exigence pour l'opérateur  $\equiv$  signifie que les axiomes de  $A'$  et  $B'$  doivent être exhaustifs et lister toutes les conditions, toutes les clauses. Dans notre exemple d'encodage de l'Extrait de document 9-2 qui nous donne `CSL_sélectionner_dispositif`, nous avons vu que nous avons deux clauses. Lorsque nous comparons par subsomption `CSL_sélectionner_dispositif` avec une autre tâche de sélection nous ne pouvons pas savoir laquelle des deux clauses expliquent la conclusion du raisonneur.

Nous allons voir plus loin que ceci est une des motivations de notre passage à une approche granulaire. Nous allons voir cette approche à la section 9.4.

### 9.2.2.2 Limite sur la précision des classes pour les axiomes

Notons que, selon l'énoncé 8), pour être une instance de `CSL_sélectionner_dispositif_culture_-Strep_B`, il suffit d'être en relation avec une instance de `mesure_âge_patient` **quelconque** qui a une valeur de moins de 3 mois. En effet, rien n'exige que `mesure_âge_patient` soit effectivement l'âge du **patient qui est la source du spécimen**. Ceci est tout à fait analogue à un formulaire papier, où il y aurait une case qui redemande la saisie de l'âge même si cette information est déjà entrée

ailleurs. Un formulaire papier ne pourra pas détecter si les données entrées ne sont pas les mêmes. Cependant lorsque que l'on compare les formulaires, la présence de la case indique le besoin que nous avons de l'âge et d'y mettre un critère.

Nous pouvons imposer par axiome le bon choix de l'âge\_patient. Mais pour ce faire les énoncés sont fortement imbriqués, difficiles à comprendre et donc à entretenir. Rappelons que notre but n'est toujours pas de valider des instances de cultures, mais bien de comparer des axiomes, mettre une contrainte sur un âge quelconque est suffisant pour la comparaison.

### 9.2.2.3 Limite sur la précision des quantificateurs pour les axiomes

Le quantificateur **some** veut dire au minimum 1, il existe un autre quantificateur, **exactly**, qui stipule exactement un nombre. Il serait souvent plus précis de stipuler **exactly 1** plutôt que **some**. Par exemple l'Extrait de document 9-2 mentionne qu'il faut ensemer une gélose-sang, pas « un nombre quelconque, plus grand que zéro, de gélose-sang ». C'est une stratégie qu'OBI a prise pour son encodage.

Ceci pourrait avoir un impact pour l'encodage de certains protocoles où on enseme deux géloses identiques. Si la stratégie pour les listes est de faire des recherches dans un fichier de sérialisation, nous pouvons simplement ajouter deux énoncés identiques, qui se retrouveront dans la sérialisation. Une telle duplication n'est pas détectée par Protégé et pour le raisonneur l'énoncé redondant est sans effet. Nous pouvons aussi, dans ces cas précis, inscrire une contrainte avec une cardinalité « exactly 2 », mais cela complique les recherches dans le fichier de sérialisation.

### 9.2.2.4 L'inférence automatique et résultats inattendus

Parfois les résultats de l'inférence automatique sont, pour le moins, surprenants. Pour illustrer une de ces situations supposons que nous avons fait l'erreur de mettre une norme en 'month' sur une mesure en 'year' et voyons comment le raisonneur va détecter l'erreur. La relation *has\_unit\_label* est déclarée dans OBI comme ayant la caractéristique d'être « functional ». Ce qui veut dire qu'il ne peut y avoir plus d'une instance de *unit\_label* par instance critère ou mesure.

Alors si nous avons :

mesure\_âge\_patient  $\equiv$  value\_specification  
**and** *is\_quality\_measurement\_of* some âge  
**and** *has\_unit\_label\_value* 'year'

CSL\_norme\_âge\_3\_mois\_ou\_moins  $\equiv$   
critère\_3\_ou\_moins  
**and** critère\_mois  
**and** *is\_about* âge\_patient

Il semble évident qu'aucune instance ne pourra satisfaire en même temps à CSL\_mesure\_âge\_patient et à CSL\_norme\_âge\_3\_mois\_ou\_moins. Cette instance devrait avoir deux cibles 'month' et 'year' pour *has\_unit\_label* et ceci est interdit par la définition de la relation comme « functional ».

Mais le raisonneur ne signale...rien. Nous ne pouvons même pas savoir pourquoi il n'a rien trouvé. C'est seulement en déclarant un individu « x » comme étant instance de la mesure **et** de la norme que la cause de ce comportement surprenant apparaît.

Si nous avons

« x » *instance of* CSL\_mesure\_âge\_patient **and** CSL\_norme\_âge\_3\_mois\_ou\_moins  
« x » *instance of* norme\_âge\_3\_mois\_ou\_moins

Cela implique que

« x » has\_unit\_label 'month'  
« x » has\_unit\_label 'year'

Comme *has\_unit\_label* ne peut avoir deux cibles différentes, le raisonneur ajoute alors un nouvel énoncé inféré :

'month' *same\_individual\_as* 'year'

Au départ sans l'ajout de l'instance "x" le raisonneur ne mentionne pas que l'intersection de la norme et du critère est nécessairement vide, car il y a une **possibilité** que 'year' et 'month' représente la même instance. Cependant, au moment où nous ajoutons "x" alors l'équivalence entre les deux instances devient **obligatoire** et est donc inférée. Il faut activer dans Protégé une option

“all individual distinct” pour que le raisonneur nous signale que l’intersection est nécessairement vide.

Au-delà d’être une curiosité sur le raisonnement automatique, cet exemple révèle comment une erreur pourrait passer totalement inaperçue. Ceci est un autre indice que le recours au raisonnement automatique ne sera pas simple comme nous l’avons cru au départ.

### 9.2.3 Sévérité des clauses et subsomption

Nous allons étudier les clauses de façon indépendante sans nous soucier de comment elles seront combinées pour faire un `plan_specification`.

Dans les exemples suivants nous allons étudier l’effet de variantes de la `clause` sur l’ajout de la gélose CNA chez les nourrissons.

#### 9.2.3.1 Les contraintes

Pour notre exemple de variantes sur la `clause` nous allons aller plus loin dans le découpage d’une logique procédurale et diviser la `clause` en contraintes.

```
CSL_sélectionner_dispositif_culture_Strep_B ≡  
  sélectionner_dispositif_culture  
  and has_specified_input some  
    (CSL_norme_âge_3_mois_ou_moins and mesure_âge_patient)  
  and has_specified_output some gélose_CNA
```

Cette clause comprend deux contraintes, une sur l’âge et l’autre sur le choix de gélose, nous pouvons réécrire les axiomes comme suit :

```
CSL_sélectionner_dispositif_culture_Strep_B_contrainte_âge ≡  
  sélectionner_dispositif_culture  
  and has_specified_input some  
    (CSL_norme_âge_3_mois_ou_moins and mesure_âge_patient)
```

```
CSL_sélectionner_dispositif_culture_Strep_B_contrainte_gélose ≡  
  and has_specified_output some gélose_CNA
```

CSL\_sélectionner\_dispositif\_culture\_Strep\_B  $\equiv$   
sélectionner\_dispositif\_culture  
**and** CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_contrainte\_âge  
**and** CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_contrainte\_gélose

Nous n'avons pas écrit "contraintes" dans la fonte réservée aux éléments de l'ontologie. En effet le besoin de recourir à des subdivisions d'une granularité aussi fine n'est apparu que bien tard dans notre travail sur OntoMic et le statut des "contraintes" n'est pas encore très clair. Ces "contraintes" pourraient ne demeurer qu'une façon de voir les axiomes sans qu'elles ne deviennent des classe d'OntoMic.

Nous allons maintenant étudier l'effet de variations sur les contraintes. Dans ces exemples, *mesure\_âge\_patient* est toujours la même peu importe les variations sur les critères, nous ne l'avons pas incluse dans les axiomes pour simplifier

### 9.2.3.2 Variations sur une contrainte à la fois

#### 9.2.3.2.1 Variations sur la contrainte pour l'âge

Supposons une modification protocole qui donne la clause pour 6 mois ou moins :

CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_modif\_6\_mois  $\equiv$   
sélectionner\_dispositif\_culture  
**and** *has\_specified\_input* **some** CSL\_norme\_âge\_6\_mois\_ou\_moins  
**and** *has\_specified\_input* **some** gélose\_CNA

Alors en appliquant le raisonneur nous obtiendrons

9) CSL\_sélectionner\_dispositif\_culture\_Strep\_B  $\sqsubseteq$   
CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_modif\_6\_mois

En effet un individu qui est instance de *CSL\_sélectionner\_dispositif\_culture\_Strep\_B* sera automatiquement instance de *CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_modif\_6\_mois* parce que les restrictions sur *has\_specified\_input* se font sur des classes qui ont la caractéristique suivante.

CSL\_norme\_âge\_3\_mois\_ou\_moins  $\sqsubseteq$  CSL\_norme\_âge\_6\_mois\_ou\_moins

Nous pouvons exprimer l'équation 9) de façon imagée comme : plus il a de "choix" à droite de **some**, moins la contrainte est sévère, alors l'axiome sur la classe est plus "permissif" et la classe comprend plus d'individus.

Le protocole avec l'exigence sur la gélose CNA pour les 3 mois ou moins est moins sévère que celui avec une exigence à 6 mois, car il demande de faire l'ajout de la gélose CNA moins souvent. Et la subsomption nous donne la tâche de sélection avec le 3 mois incluse dans celle le 6 mois. Dans ce cas la sévérité croit dans la même direction que la chaîne d'inclusion.

#### 9.2.3.2.2 Variation de la contrainte sur le choix de la gélose

Malheureusement cette direction de l'accroissement de la sévérité par rapport aux inclusions n'est pas dans le même sens pour chaque type de clauses. En effet supposons une autre modification qui impose l'utilisation d'un second dispositif de culture

CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_modif\_second\_dispositif  $\equiv$   
sélectionner\_dispositif\_culture  
**and has\_specified\_input some** norme\_âge\_3\_mois\_ou\_moins  
**and has\_specified\_input some** gélose\_CNA  
**and has\_specified\_input some** bouillon\_Todd-Hewitt

Le raisonneur aura le comportement inverse que lors d'une modification de contrainte sur l'âge et l'axiome modifié, plus sévère, est inclus dans l'original.

CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_second\_dispositif  $\sqsubseteq$   
CSL\_sélectionner\_dispositif\_culture\_Strep\_B

En effet, pour qu'un individual soit instance de CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_second\_dispositif il doit répondre aux mêmes exigences que pour CSL\_sélectionner\_dispositif\_culture\_Strep\_B **plus** une autre. Chaque instance de CSL\_sélectionner\_dispositif\_culture\_Strep\_B\_second\_dispositif est donc automatiquement instance de CSL\_sélectionner\_dispositif\_culture\_Strep\_B.

La lourdeur de la tâche, c'est-à-dire dans ce cas le nombre de dispositifs de cultures à ensemercer croît dans la direction inverse de la chaîne d'inclusion, plus nous remontons dans la chaîne, moins la contrainte est sévère.

### 9.2.3.3 Variations de plus d'une contrainte à la fois

Imaginons une clause, encore plus sévère, qui combine les deux modifications comme dans l'axiome suivant :

```
CSL_sélectionner_dispositif_culture_Strep_B_modif_6_mois_et_second_dispositif ≡  
sélectionner_dispositif_culture  
and has_specified_input some norme_âge_6_mois_ou_moins  
and has_specified_input some gélose_CNA  
and has_specified_input some bouillon_Todd-Hewitt
```

Le raisonneur ne trouvera aucune hiérarchie de subsomption entre `CSL_sélectionner_dispositif_culture_Strep_B` et `CSL_sélectionner_dispositif_culture_Strep_B_modif_6_mois_et_second_dispositif`, car la sévérité du critère n'évolue pas dans le même sens pour la contrainte sur l'âge et la contrainte sur les milieux.

### 9.2.3.4 La stratégie de renversement des contraintes

Pour la contrainte sur l'âge, plus la contrainte a une plage de valeurs large plus la contrainte est sévère. Tandis pour les dispositifs, plus nous imposons des contraintes sur `sélectionner_dispositif_culture` en ajoutant des relations *has\_specified\_input*, plus la contrainte est sévère.

De façon arbitraire, mais qui nous semble naturelle, nous avons fait la convention qu'un `sélectionner_dispositif_culture` qui est inclus dans un autre est plus sévère. Plus nous allons profondément dans la chaîne de subsomption, plus la clause est sévère.

Pour réaliser ceci, nous appliquerons un changement aux contraintes de façon qu'au moment d'une comparaison, les contraintes dont la sévérité croît avec la cardinalité seront renversés.

Les critères avec contrainte renversé pour utilisations subsomption.

CSL\_sélectionner\_dispositif\_Strep\_B\_reversé  $\equiv$   
sélectionner\_dispositif\_culture  
**not** *has\_specified\_input* **some** norme\_âge\_3\_mois\_et\_moins  
**and** *has\_specified\_output* **some** gélose\_CNA

CSL\_sélectionner\_dispositif\_Strep\_B\_modif\_6\_mois\_reversé\_et\_second\_dispositif  $\equiv$   
sélectionner\_dispositif\_culture  
**not** *has\_specified\_input* **some** norme\_âge\_6\_mois\_et\_moins  
**and** *has\_specified\_output* **some** gélose\_CNA  
**and** *has\_specified\_output* **some** bouillon\_Todd-Hewitt

Alors nous aurons

CSL\_sélectionner\_dispositif\_Strep\_B\_modif\_6\_mois\_reversée\_et\_second\_ -  
dispositif  $\sqsubseteq$   
CSL\_sélectionner\_dispositif\_Strep\_B\_reversée

Le raisonneur a donc pu conclure, ce qui n'était pas le cas sans le renversement des contraintes. Il y a des mécanismes de marquage d'axiomes avec OWL qui permettent d'indiquer quels sont les énoncés à renverser. Bien sûr cela implique que nous devons transformer les axiomes avant d'exécuter le raisonneur.

#### 9.2.3.4.1 Comparaisons d'axiomes qui n'ont pas le même jeu de contraintes

Supposons une modification où nous ajoutons la gélose CNA et la recherche de Strep B pour tous les patients, alors la gélose CNA se trouvera dans la clause de milieux généraux comme suit :

CSL\_sélectionner\_dispositifs\_généraux\_avec\_CNA\_pour\_tous  
sélectionner\_dispositif\_culture  
**and** *has\_specified\_output* **some** gélose\_CNA  
**and** *has\_specified\_output* **some** gélose-sang  
...  
**and** *has\_specified\_output* **some** gélose\_CIN

L'application de la subsomption par le raisonneur nous donnera

CSL\_sélectionner\_dispositifs\_généraux\_avec\_CNA\_pour\_tous  $\sqsubseteq$  CSL\_sélectionner\_ -  
dispositifs\_généraux

Si nous suivons notre convention sur la sévérité `CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_tous` devait répondre également à l'énoncé suivant, car il demande d'ensemencer la gélose CIN en plus des autres géloses pour tous les patients

`CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_tous`  $\sqsubseteq$  `CSL_sélectionner_dispositifs_Strep_B*`

Cependant ce dernier énoncé est faux, car les deux classes ne sont pas comparables. La contrainte d'âge dans `CSL_sélectionner_dispositifs_Strep_B` bloque l'inclusion. Renverser la contrainte d'âge n'aide en rien. Pour que le renversement des contraintes fonctionne, il nous faut changer le contrainte "faire pour tous" pour le faire chez les 0 à 135 ans.

```
CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_135_ans_et_moins
  sélectionner_dispositif_culture
  and has_specified_output some gélose_CNA
  and has_specified_output some gélose-sang
  ...
  and has_specified_output some gélose_CIN
  and has_specified_input some norme_âge_135_ans_et_moins
```

Alors le raisonneur conclura que

```
CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_135_ans_et_moins_-
renversé  $\sqsubseteq$ 
CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_tous
```

Avec ce critère d'âge étendu, nous aurons :

```
CSL_sélectionner_dispositifs_généraux_avec_CNA_pour_135_ans_et_moins_-
renversé  $\sqsubseteq$ 
CSL_sélectionner_dispositif_Strep_B_renversée
```

Si nous voulons à priori pouvoir comparer n'importe quelles paires de `sélectionner_dispositif`, il faudra que toutes les paires aient le même nombre de contraintes, donc dans notre cas ajouter la contrainte inopérante d'avoir un âge entre 0 et 135 ans pour tous les `sélectionner_dispositif` qui n'ont pas de contraintes d'âge. Cette solution est non seulement inélégante, mais lourde à maintenir à jour. En effet, si nous ajoutons à OntoMic un nouveau type de critère, sur les mois de l'année par

exemple, il faudra retourner dans les axiomes et ajouter des contraintes inopérantes comme d'avoir un mois entre janvier et décembre. Nous retrouvons ici le problème de manque d'évolutivité que nous avons vu à la section 3.6.1.1 avec le modèle rationnel.

### 9.2.3.5 Étude séparée des contraintes.

Il y a une autre stratégie possible où au lieu de tenter de comparer l'axiome au complet, nous nous intéressons à la comparaison des contraintes.

```
CSL_sélectionner_dispositif_Strep_B ≡  
  sélectionner_dispositif  
  and CSL_sélectionner_dispositif_Strep_B_contrainte_âge  
  and CSL_sélectionner_dispositif_Strep_B_contrainte_gélose
```

En effet CSL\_sélectionner\_dispositif\_Strep\_B contrainte âge et CSL\_sélectionner\_dispositif\_Strep\_B\_contrainte\_gélose deviennent des classes à part entière sur lesquelles le raisonneur appliquera la subsomption.

Cette stratégie demande la création d'encore plus d'entités et nous avons déjà vu à la section 8.8 que l'encodage avec OntoMic produisait déjà beaucoup d'entités. Cette stratégie laisse à l'utilisateur la tâche de se retrouver dans les directions d'inclusion et la sévérité selon le type de contraintes.

### 9.2.3.6 Conclusion sur la sévérité et la subsomption

Par ces exemples nous voyons bien qu'utiliser la subsomption pour comparer les **clause** n'est pas simple. De plus, dans ces exemples, nous n'avons fait varier que les limites d'âge et les géloses à ensemercer, nous aurions pu mettre d'autres variantes.

Mais il y a des difficultés qui se présentent avant même de faire la comparaison des clauses. En effet, faire le découpage des logiques procédurales en **clause** et agencer ces dernières n'est pas sans ambiguïtés comme nous le verrons à la section suivante.

## 9.2.4 Découpage de la logique, plusieurs avenues de modélisation

À la section 6.6.3, nous avons montré qu'avec notre analogie des formulaires, nous avons deux choix pour représenter la logique sur l'utilisation de la gélose CNA. Celui où nous faisons deux formulaires séparés : l'un pour les plus de 3 mois et l'autre pour les 3 mois et moins. Un autre choix où nous ajoutions simplement un calcul d'âge sur le formulaire général.

Avec OntoMic, nous aurons aussi plusieurs manières d'encoder, que nous allons illustrer ici.

### 9.2.4.1 Une seule instance de sélectionner\_dispositif

Le premier groupe sont des méthodes qui modélisent la sélection des géloses comme une tâche unique. Pour une culture en particulier nous n'aurons qu'une seule instance de sélectionner\_dispositif. Les clause sont combinées par des unions ou intersections.

#### 9.2.4.1.1 Avec des clause qui donnent une intersection vide

Ceci est analogue à avoir des formulaires totalement séparés pour les 3 mois et moins et pour les plus de 3 mois.

```
CSL_sélectionner_dispositif_culture_3_mois_ou_moins ≡  
  sélectionner_dispositif_culture  
  and has_specified_input some norme âge 3 mois ou moins)  
  and has_specified_output some gélose GS  
  and has_specified_output some gélose MC  
  and has_specified_output some gélose CNA
```

c) Enfants < 5 ans :

Pour la période de janvier à mai inclusivement, faire la recherche du Rotavirus sur une seule selle. Si le résultat est positif, ne pas faire de culture ni de recherche de C.difficile. Si le résultat est négatif, faire les tests demandés. Le reste de l'année, procéder comme spécimen de routine.

Extrait de document 9-3, recherche de Rotavirus l'hivers selon le labo SJ

Source : SJ MIC-PON-072 V7

```
CSL_sélectionner_dispositif_culture_plus_de_3_mois
  sélectionner_dispositif_culture
  and not has_specified_input some norme_âge_3_mois_ou_moins)
  and has_specified_output some gélose_GS
  and has_specified_output some gélose_MC
```

```
CSL_sélectionner_dispositif_culture_première_forme ≡
  CSL_sélectionner_dispositif_culture_plus_de_3_mois
  or
  CSL_sélectionner_dispositif_culture_3_mois_ou_moins
```

Avec cette stratégie, si nous avons besoin de tenir compte d'une autre condition il y a un problème de multiplication du nombre de **clause** nécessaires. Nous avons un exemple d'une autre condition avec l'Extrait de document 9-3. Cet extrait qui indique de faire une autre technique chez les enfants durant les mois d'hiver.

Si nous modélisons par des formulaires ou des **clause** à intersection vide, il nous faudra une multitude d'encodages soit : 3 mois ou moins en hivers, plus de 3 mois mais moins de 5 ans en hivers, plus de cinq ans peu importe la saison, 3 mois ou moins en dehors de l'hiver et plus de 3 mois et moins de cinq ans en dehors de l'hiver. En plus de cet aspect multiplicatif, ce type d'encodage est aussi difficile à gérer, car lorsque l'on ajoute une nouvelle norme il faut faire un retour sur ce qui est déjà encodé.

9.2.4.1.2 Avec un axiome sélectionner\_dispositif\_culture général et un pour chaque cas particulier

Cette méthode est analogue à un formulaire général avec une section du formulaire spécialisée pour les 3 mois ou moins.

CSL\_sélectionner\_dispositif\_culture\_pour\_tous ≡  
sélectionner\_dispositif\_culture  
*has\_specified\_output* **some** gélose\_GS  
*has\_specified\_output* **some** gélose\_MC

CSL\_sélectionner\_dispositif\_culture\_3\_mois\_ou\_moins ≡  
sélectionner\_dispositif\_culture  
*has\_specified\_input* **some** norme\_âge\_3\_mois\_ou\_moins  
*has\_specified\_output* **some** gélose\_CNA

CSL\_sélectionner\_dispositif\_culture\_plus\_de\_3\_mois ≡  
sélectionner\_dispositif\_culture  
*has\_specified\_input* **some** norme\_âge\_plus\_de\_3\_mois

CSL\_sélectionner\_dispositif\_culture\_seconde\_forme ≡  
CSL\_sélectionner\_dispositif\_culture\_pour\_tous  
**and**  
(CSL\_sélectionner\_dispositif\_culture\_3\_mois\_ou\_moins  
**or**  
CSL\_sélectionner\_dispositif\_culture\_plus\_de\_3\_mois)

Avec cette méthode nous devons ajouter un CSL\_sélectionner\_dispositif\_culture\_plus\_de\_3\_mois pour les plus de 3 mois, qui ne sélectionne aucune gélose. Avec cet énoncé « vide » des tests avec Protégé ont montré que CSL\_sélectionner\_dispositif\_culture\_première\_forme et CSL\_sélectionner\_dispositif\_culture\_seconde\_forme sont équivalents.

Cette seconde méthode est plus élégante, chaque nouvelle clause se traduit par l'addition de deux contraintes sans nécessairement réécrire les premières clauses.

#### 9.2.4.2 Méthode à plusieurs instances de sélectionner\_dispositif\_culture

Avec cette méthode un premier sélectionner\_dispositif\_culture génère les deux géloses GS et MC pour tous les patients sans contraintes. Puis un second sélectionner\_dispositif\_culture, qui ne vise que les âges de 3 mois ou moins, génère le milieu CNA. Avec cette méthode, il n'y a pas d'intersection ou d'union sur les clause. Chacune des clause est une sous-classe de sélectionner\_dispositif\_culture à part entière. Les cultures n'auront pas nécessairement le même nombre d'instances de sélectionner\_dispositif\_culture.

Cette seconde méthode est plus intuitive, la **clause** génère le ou les dispositifs qui lui sont associés, sans interaction avec les autres règles. En particulier sans le recours à des règles vides. Également, ce type de règles correspond mieux à notre schéma général où un **planned\_process** se déclenche et produit une instance de sa classe, lorsque les conditions de départs sont remplies.

Malheureusement, du point de vue strictement logique, la formulation avec seule instance décrite à la section précédente et celle avec un nombre variable d'instances ne sont pas comparables par raisonneur. Il s'agit encore d'une autre restriction à la « liberté » de l'encodage si nous voulons pouvoir appliquer l'inférence logique.

### 9.2.5 La présence de contraintes qui interfère pendant la subsomption

La comparaison par subsomption peut être bloquée par des énoncés sur les **plan\_specification** qui sont sans lien avec les **clause** examinées. Nous avons vu à la section 7.2.2.5 comment les **planned\_process** marquent leur utilisation d'un **plan\_specification**. Ceci se traduit par les énoncés suivants :

SJ\_sélectionner\_dispositif\_culture

*realizes some (concretizes value 'SJ\_plan\_specification\_sélectionner\_gélose')*

CSL\_sélectionner\_dispositif\_culture

*realizes some (concretizes value 'CSL\_plan\_specification\_sélectionner\_gélose')*

Si nous définissons ainsi les tâches avec leur lien d'utilisation, alors nous ne pouvons plus comparer par subsomption, car les deux classes sont reliées à des individus différents et donc ne seront jamais incluses l'une dans l'autre.

Avec un écouvillon, déposer un inoculum léger du spécimen, dans l'ordre, sur les géloses suivantes :

- GS
- CNA (pour les enfants âgés de trois mois et moins)
- MacConkey Sorbitol
- MacConkey
- CIN

Ensemencer avec un inoculum moyen

- Gélose SS
- CCDA

Extrait de document 9-5, choix des gélose selon le labo CSL

Source : CSL\_MIC-PON-TSP SELLE\_V7

Milieux :

- Gélose-sang
- Mc Conkey
- Hektoen
- Gélose Campy CSM
- McConkey-Sorbitol avec cefixime et tellurite
- CIN

Extrait de document 9-4, choix des géloses selon le labo SJ

Source : SJ\_MIC-PON-072\_V7

## 9.2.6 Difficultés d'obtenir des généralisations avec la subsomption.

Nous avons vu que pour que la subsomption soit utilisable, il faut que les clauses à comparer soient encodées de la même façon et que nous nous soyons débarrassés des énoncés qui interfèrent. Dans ce qui suit, nous allons montrer un autre problème avec la subsomption qui survient lorsque nous voulons comparer deux protocoles, non pas selon leur sévérité, mais en termes d'équivalences.

Comparons la sélection des géloses pour le protocole de Cité de la Santé à l'Extrait de document 9-5 et celle pour le protocole de Saint-Jérôme à l'Extrait de document 9-4. Une personne à l'aise avec la microbiologie clinique qui compare manuellement ces deux listes va conclure que, sans compter la différence avec la gélose CNA, les deux listes sont équivalentes, car elles comprennent toutes deux des géloses qui ont été choisies pour accomplir le même but.

La gélose MacConkey Sorbitol, choisie par CSL, est une version de la gélose McConkey-Sorbitol avec cefixime et tellurite, choisie par SJ, elles servent toutes deux à isoler le *E. coli* O157:H7. La

gélose SS de CSL et la gélose Hektoen sont toutes deux optimisées pour les *Salmonelle* et *Shigelle*.  
 Finalement les géloses CCDA et Campy CSM sont optimisées pour les *Campylobacter*.

Si nous définissons la notion de géloses optimisées pour un groupe d'organismes, nous pourrions dire

gélose_MacConkey_Sorbitol	$\sqsubseteq$ gélose_optimisée_E.coli_O157:H7
gélose_MC-Sorbitol_céfexime_tellurite	$\sqsubseteq$ gélose_optimisée_E.coli_O157:H7
gélose_SS	$\sqsubseteq$ gélose_optimisée_Salmonelle-Shigelle
gélose_Hektoen	$\sqsubseteq$ gélose_optimisée_Salmonelle-Shigelle
gélose_Campy_CSM	$\sqsubseteq$ gélose_optimisée_Campylobacter
gélose_CCDA	$\sqsubseteq$ gélose_optimisée_Campylobacter

Nous aimerions que le raisonneur puisse bâtir lui-même un sélectionner\_dispositif\_culture\_ - généralisé

sélectionner\_dispositif\_culture\_généralisé\_SJ\_et\_CSL  $\equiv$   
 sélectionner\_dispositif\_culture  
*has\_specified\_output* **some** gélose\_GS  
*has\_specified\_output* **some** gélose\_MacConkey  
*has\_specified\_output* **some** gélose\_CIN  
*has\_specified\_output* **some** gélose\_optimisée\_E.coli\_O157 :H7  
*has\_specified\_output* **some** gélose\_optimisée\_Salmonelle-Shigelle  
*has\_specified\_output* **some** Gélose\_optimisée\_Campylobacter

À ce moment nous aurons

SJ\_sélectionner\_dispositif\_culture  $\sqsubseteq$   
 sélectionner\_dispositif\_culture\_généralisé\_SJ\_et\_CS

CSL\_sélectionner\_dispositif\_culture  $\sqsubseteq$   
 sélectionner\_dispositif\_culture\_généralisé\_SJ\_et\_CS

Nous n'avons pas trouvé de méthode qui génère des axiomes de généralisation. Ceci est un enjeu fort important, car des substitutions entre dispositifs de culture avec le même objectif sont fréquentes. Ces protocoles généralisés devront donc être faits à la main.

## 9.3 Inférence logique et flexibilité dans l'encodage

Nous avons noté à la section 6.7 que nous espérons que le passage par une ontologie nous libérerait de la nécessité d'un encodage rigide. Malheureusement, le raisonneur ne peut pas inférer sur deux logiques « équivalentes », mais encodées différemment.

Nous devons donc imposer une structure d'encodage, ce qui de prime abord est très limitatif. Mais les mêmes types d'éléments d'information se retrouvent souvent dans les protocoles. Nous pouvons créer des canevas pour ces éléments fréquents et pour un élément sans canevas rien n'empêche un utilisateur expert de créer ad hoc la logique d'axiomes. Dans la littérature il y a plusieurs références sur l'application de canevas pour la création d'énoncés (O'Connor, Halaschek-Wiener, & Musen, 2010) (Jupp, Horridge, Iannone, et al., 2011) (Vita et al., 2013).

## 9.4 Stratégie granulaire

Dans les sections précédentes, nous avons vu les problèmes auxquels nous faisons face lorsque nous voulons faire des requêtes sur les informations encodées où si nous voulons les comparer avec l'inférence automatique.

Nous n'avons pas encore montré comment les informations encodées sont entreposées. Dans la section suivante, nous allons voir que notre conception de départ, toute simple, a rencontré elle aussi des difficultés.

### 9.4.1 L'inférence globale

Initialement, l'entreposage apparaissait simple, toutes les informations encodées étaient simplement ajoutées à une copie locale d'OntoMic, le fichier de sérialisation OWL devenait ainsi le dépôt d'information.

Bien que ceci soit tout à fait possible pour l'entreposage. Nous sommes alors aux prises avec le problème des axiomes non comparables que nous avons vu à la section 9.2.3 et à celui des contraintes de blocages vues à la section 9.2.5.

Mais en plus, lorsque nous tentons d'utiliser un raisonneur nous nous heurtons à une contrainte sévère de temps réponse. En effet, sur la version complète d'OBI, sans les ajouts pour OntoMic, le raisonneur prend quatre minutes d'exécution sur un ordinateur portable de puissance modeste. Chaque ajout d'axiomes nécessite une réinitialisation du raisonneur avec le même temps de calcul.

Nous souhaitons un moteur de comparaison en temps réel. Par exemple, si un remplacement de matériel doit être fait en urgence, nous ne voudrions pas que retracer les impacts prennent plusieurs minutes de calcul.

Il y a bien sûr des stratégies d'optimisation possible. Par exemple avec un raisonneur qui serait capable de fonctionner de façon incrémentielle, en ne rebâtissant pas tout à chaque lancée. Une exploration de cette avenue est intéressante et pourra faire partie de plans futurs.

Mais un raisonneur incrémentiel irait plus vite parce qu'il ne rebâtira pas toutes les hiérarchies, il ne ferait que tester les nouveaux axiomes qui ont été ajoutés – modifiés depuis le dernier lancement. Avec un raisonnement analogue, pourquoi lors de l'analyse d'une nouvelle clause, devrions-nous refaire toutes les subsomptions du dépôt ? Ne pourrions-nous pas simplement faire la subsomption entre les clauses pertinentes ?

Dans le modèle d'inférences globales sur tout le dépôt, nous n'orientons pas les subsomptions. Le raisonneur est lancé et nous obtenons une hiérarchie inférée pour toutes les classes. Nous répondons aux questions en examinant la hiérarchie inférée pour les classes pertinentes.

#### **9.4.2 L'inférence granulaire**

Par opposition à l'approche par inférences globales, l'inférence granulaire ne fait que comparer les classes pertinentes selon des critères préétablis.

Pour illustrer ceci, reprenons notre exemple de discordances entre la requête, le répertoire d'analyse et le protocole pour le contenant vu à Exemple 1 du Chapitre 3.

SJ\_sélectionner\_contenant\_selon\_protocole  $\equiv$   
sélectionner\_contenant  
**and has\_specified\_output some** (pot\_vissé\_stérile  
**or** pot\_vissé\_Cary-Blair  
**or** dispositif\_écouvillon)

SJ\_sélectionner\_contenant\_selon\_requête  $\equiv$   
sélectionner\_contenant  
**and has\_specified\_output some** pot\_vissé\_Cary-Blair

SJ\_sélectionner\_contenant\_selon\_répertoire\_analyse  $\equiv$   
sélectionner\_contenant  
**and has\_specified\_output some** pot\_vissé\_stérile

Après avoir lancé le raisonneur, nous aurons :

SJ\_sélectionner\_contenant\_selon\_requête  $\sqsubseteq$   
SJ\_sélectionner\_contenant\_selon\_protocole

SJ\_sélectionner\_contenant\_selon\_répertoire\_analyse  $\sqsubseteq$   
SJ\_sélectionner\_contenant\_selon\_protocole

Il faut donc une stratégie pour identifier les clause pertinentes, les incorporer à OntoMic et lancer le raisonneur. Pour le classement des clause

plan\_specification  
  plan\_étape\_test\_clinique  
    plan\_prélèvement  
      plan\_sélection\_contenant\_transport  
        clause\_choix\_contenant\_transport

SJ\_sélectionner\_contenant\_selon\_protocole           *instance\_of*  
  clause\_choix\_contenant\_transport

SJ\_sélectionner\_contenant\_selon\_requête           *instance\_of*  
  clause\_choix\_contenant\_transport

SJ\_sélectionner\_contenant\_selon\_répertoire\_analyse   *instance\_of*  
  clause\_choix\_contenant\_transport

Les énoncés pour une clause sont sauvegardés comme une ontologie autonome. Pour travailler avec ces clause encodées comme des mini-ontologies, il suffit de rechercher les mini-ontologies

pour **la** classe de **clause** qui nous intéresse, de les fusionner avec OntoMic et appliquer le raisonneur.

Notre encodage se prête bien à cette stratégie granulaire. En effet, les **clause** sont indépendantes les unes des autres parce que leurs axiomes sont bâtis à partir d'éléments d'OntoMic et ne comprennent **pas** de références à des éléments créés pour d'autres **clause**.

Nous avons fait le test pour les **clause** données en exemple. L'inférence logique fonctionne sur ce sous-groupe d'axiomes, pour cette comparaison simple.

## 9.5 Conclusion sur l'utilisation des données encodées

Au départ nous croyions que le raisonnement automatique serait à même de gérer la complexité de l'encodage des logiques procédurale. Notre plan était d'encoder en nous assurant que les axiomes traduisent fidèlement la logique, mais sans trop nous soucier de l'agencement de l'axiome, surtout sans nous soucier que ces agencements suivent tous le même patron. Notre conception, fort naïve à la lumière de ce que nous savons maintenant, était que le raisonneur pourrait faire abstraction des différences d'encodages.

Nos expériences nous ont montré que plus l'axiome est complexe moins il y avait de chances que le raisonneur arrive à un résultat utile, à moins que les axiomes à comparer ne soient tout à fait stéréotypés. Si nous nous attardons à bien cerner les capacités des raisonneurs, cette difficulté qu'ils ont avec les axiomes complexes saute aux yeux. En effet pour une paire de classes A et B, tout ce qu'un raisonneur peut déclarer pour cette paire c'est :  $A \sqsubseteq B$ ,  $B \sqsubseteq A$ ,  $A \equiv B$ , A sans lien avec B. Il est évident qu'un raisonneur ne pourra forer dans des axiomes complexes et nous informer sur les différences.

Donc la prémisse que nous avons au départ était erronée, il faut gérer la complexité en amont du raisonneur et utiliser ce dernier que pour des comparaisons simples. Nous avons vu que notre plan était de considérer le classement des logiques procédurales et leurs encodages comme des tâches séparées. Le classement étant plus du fait d'IAO et l'encodage celui d'OntoMic. Mais avec le recours aux **clause** et aux contraintes, cette dichotomie est brisée. En effet, prendre la décision de

séparer en deux clause CSL\_sélectionner\_dispositif\_culture\_pour\_tous et CSL\_sélectionner\_dispositif\_culture\_3\_mois est à la fois une décision d'encodage et de classement.

## Chapitre 10 Conclusion

Le travail sur ce mémoire a été un travail de balancier entre deux pôles : le pôle « problème » qui est l'ensemble des difficultés rencontrées depuis plusieurs années au laboratoire SJ, l'autre pôle étant « modèle », l'étude et le développement d'une solution par ontologie, élégante, mais qui nécessite beaucoup de travail.

Pour nous attaquer au côté « problème » de notre balancier, nous aurions pu poursuivre le long de la voie déjà amorcée et travailler à une nouvelle version de la gestion documentaire. Ceci aurait vraisemblablement donné des résultats probants de façon beaucoup plus immédiate. Par contre, nous savons que cette solution est vouée tôt ou tard à se heurter à ses limites. De plus, il y avait peu de chance dans cette voie de trouver de réelles innovations. L'exploration d'une solution, peu connue, mais prometteuse, était plus appropriée pour un mémoire, et aussi, avouons-le, beaucoup plus intéressante.

Mais il fallait au minimum, démontrer qu'une solution par ontologie avait le potentiel de s'attaquer au problème, qu'il ne s'agissait pas seulement d'une modélisation élégante. C'est pourquoi une recherche d'erreurs dans les avoires informationnels a été faite. Le but était de montrer que le problème est bien présent, mais aussi, de fournir des cas de figure pour démontrer la réelle utilité d'OntoMic.

Donc il fallait se lancer dans l'élaboration d'OntoMic, comme point de départ nous avons la version « core » d'OBI, une version qui ne contient que les classes essentielles. Puis nous avons choisi un protocole à encoder, le protocole de culture de selles de Saint-Jérôme.

Une première ébauche d'encodage a été faite, avec notre regard d'aujourd'hui, cette ébauche était assez confuse. Cette confusion venait du fait qu'OBI n'a pas le même but qu'OntoMic et que par conséquent le modèle d'OBI présente des lacunes pour des tâches clés d'OntoMic. Il nous a donc fallu faire plusieurs développements comme nous l'avons vu à la section 8.8 qui conclut le chapitre sur le modèle OntoMic.

Une fois le modèle d'OntoMic stabilisé, nous avons pu reprendre l'encodage du protocole du laboratoire SJ. Nous n'avons pas trouvé dans ce protocole des éléments de logique que nous n'avons pas pu modéliser dans des structures d'OntoMic. Le problème fut, et est encore, plutôt l'inverse : comment choisir parmi les multiples manières d'encoder ? Ces choix sont difficiles, surtout si nous devons toujours tenir compte des impacts sur l'inférence automatique.

L'étape suivante envisagée était d'encoder le protocole de la Cité de la Santé et par la suite de travailler avec l'inférence automatique pour les comparer. Mais le protocole de la Cité de la Santé n'est pas du tout du même style que celui de Saint-Jérôme, ce dernier étant beaucoup plus synthétique et facile à encoder. De plus, il était déjà apparu que la comparaison par inférence ne serait pas si simple à implanter. Nous avons donc décidé de cesser de mettre des efforts sur l'encodage de nouvelles données pour nous concentrer sur la recherche des bonnes stratégies d'encodage.

Nous avons réalisé que la comparaison par inférence automatique ne découlerait pas simplement de soi une fois les modélisations faites adéquatement. En effet, au départ, nous avons la conception naïve que si l'encodage était logiquement adéquat alors l'inférence automatique fonctionnerait. Ce n'est pas du tout le cas et un travail important est à faire pour rendre l'inférence automatique vraiment utile.

C'est à ce moment du développement que nous avons compris qu'il n'était pas réaliste que les utilisateurs aient à manipuler l'ontologie directement. À ce moment également s'est imposée l'idée de la construction de l'encodage à partir d'éléments prédéfinis, avec entre autres la construction de balises par intersections de critères généraux. Finalement, c'est aussi à ce moment que la notion de canevas pour l'encodage s'est imposée.

Aussi il nous a fallu nous demander s'il était toujours vraiment utile d'encoder en l'absence de comparaison automatique. En effet, nous pourrions simplement classer les extraits de textes et procéder à des comparaisons manuelles. Après réflexion, nous croyons que l'encodage est toujours utile même si l'objectif de comparaisons automatiques reste lointain. Voici quelques-unes des raisons que nous avons trouvées :

- L'encodage est peut-être la meilleure façon d'exprimer une logique procédurale pour une interprétation par un humain.
- L'encodage est facile une fois un vocabulaire standard défini et le besoin d'un tel vocabulaire est indéniable. En effet, encoder est essentiellement faire des combinaisons d'éléments du vocabulaire. Ce qui est difficile c'est d'encoder pour le raisonneur, pas d'encoder pour la compréhension humaine.
- La logique d'encodage pourra servir de mots clés pour des recherches afin de répondre à des questions comme : « quelles sont les géloses qui sont incubées dans une atmosphère enrichie en CO<sub>2</sub> à Saint-Jérôme ? »

Mais intellectuellement, l'attrait du raisonnement automatique demeurerait toujours aussi grand. Nous avons tenté des pistes de solutions, toujours en travaillant avec Protégé. Il est cependant apparu qu'il nous fallait travailler les ontologies par programmation plutôt qu'avec un éditeur comme Protégé. Entamer un développement dans cette direction n'était pas réaliste dans le cadre de ce mémoire. D'autant que nous avons toujours à tenir compte du côté « problème » de notre balancier.

Du côté « problème », nous avons formalisé le vocabulaire et la méthode de traitement des informations dans les textes. Notre méthode de classement des extraits de textes selon les étapes des cultures est, du point de vue de la microbiologie, une innovation, bien qu'informatiquement cela soit classique et assez simple. Du côté d'OntoMic, il n'y a pas vraiment de développement à faire pour appliquer cette méthode de classement sur le terrain, nous sommes donc beaucoup plus près de résultats tangibles.

Finalement, la tâche de rédaction de ce mémoire a pris beaucoup plus de temps que nous l'avions estimé. Cela a été, et de loin, la tâche la plus ardue de notre travail. En effet, tout est à expliquer, et plusieurs de ces explications sont pour des concepts presque aussi nouveaux pour l'auteur que pour le lecteur ! Le plus difficile a été d'expliquer, de façon suffisamment claire nous l'espérons, le principe de l'encodage avec OntoMic, principe qui est puissant, mais fort peu intuitif.

## 10.1 Réflexion sur notre travail<sup>40</sup>

Quand j'explique à mon entourage le sujet de mon travail, je leur dis que je fais de la « modélisation de connaissances ». Tout le long de ce travail, j'ai été frappé par la difficulté de passer de l'univers du raisonnement humain à celui des machines.

J'ai été frappé par la quantité d'informations implicites, assumées ou carrément manquantes dans des jeux de connaissances que le jugement humain trouve fort complets. Et même pire, il semble que, souvent, les efforts faits pour clarifier génèrent en fait de nouvelles informations, sur lesquelles nous aurons..., de nouvelles imprécisions.

Depuis ce constat, j'ai fait mienne cette citation du philosophe du siècle dernier Bertrand Russell (Russell, 1918) qui se lit comme suit :

« Everything is vague to a degree you do not realize till you have tried to make it precise. »

Par la suite, j'ai été confronté aux limites de l'inférence logique sur les ontologies. Au départ, je voyais l'inférence comme un moyen de passer du formalisme du cerveau humain à celui d'une machine. Je croyais que les capacités de raisonnement sur les ontologies permettraient aux deux de se rencontrer, mais il manque aux raisonneurs une facette centrale du raisonnement humain.

En effet, les raisonneurs OWL déduisent de nouveaux liens sur les éléments existants de l'ontologie tandis que, sans s'en rendre compte, les humains font un pas de plus, ils créent de nouvelles entités en plus de faire des liens.

Si, par exemple, nous voulons comparer deux protocoles de cultures quant aux géloses nécessaires. Un humain fera aussitôt une généralisation, il ne comparera pas simplement les listes, mais il se posera des questions sur ce qui a motivé le choix des géloses. L'humain fera la comparaison et

---

<sup>40</sup> Cette section sera au « je », car il s'agit de ma réflexion toute personnelle sur le résultat de mon travail et sur ce mémoire.

signalera que certaines différences n'en sont pas vraiment, car il s'agit de variantes de choix parmi des géloses sélectionnées en vue du même but. Les raisonneurs sont incapables de faire des comparaisons en incluant des généralisations.

Les raisonneurs sont également incapables de séparer la tâche de comparaison par unité logique, la comparaison se fait en bloc.

Ma réaction devant les limites des raisonneurs a été celle d'un programmeur « classique ». J'ai systématisé la logique en la brisant en fragments de plus en plus petits et en mettant une classification rigide sur ces fragments. Je suis donc passé du processus complet de culture, à un processus subdivisé en ses grandes étapes, puis encore subdivisé en sous-tâches des grandes étapes, puis encore subdivisé en clauses, pour finir en contraintes élémentaires. Puis j'ai bâti des canevas d'imbrications de ces contraintes élémentaires pour que le raisonneur puisse fonctionner.

Le modèle pour OntoMic obtenu avec cette approche est plus évolutif qu'un modèle par base de données, mais il est aussi rigide dans ses choix d'encodage des informations. Il est aussi fort décevant de constater que les capacités d'inférences automatiques ne deviennent qu'un moyen automatique de vérifier que  $< 3$  et plus restrictif que  $< 6$ ... Cette stratégie est peut-être celle qui donnera des résultats tangibles le plus rapidement, mais ce n'est peut-être celle qui sera la meilleure à terme.

À la lumière de ce travail, je comprends pourquoi le domaine de l'intelligence artificielle rencontre plus de succès avec des approches agnostiques par rapport au raisonnement humain qu'avec des modèles comme celui sur lequel j'ai travaillé. Même dans un domaine aussi encadré que celui de la microbiologie clinique, la modélisation et le raisonnement automatique s'avèrent ardu.

## **10.2 Le développement futur d'OntoMic**

Nous retrouvons avec le développement futur le même balancier entre deux pôles que lors de notre travail pour ce mémoire.

### **10.2.1 Concentration sur le pôle « problème »**

Si notre axe de développement est de mettre en pratique une solution pour les problèmes de gestion des avoirs informationnels alors la première étape est de compléter le vocabulaire d'OntoMic. C'est un long travail et qui devra nécessairement fait en collaboration avec d'autres microbiologistes.

Pour impliquer la communauté des microbiologistes, il nous faudra une démonstration des capacités d'OntoMic à s'attaquer à des problèmes de terrain. Un de ces problèmes est l'absence de standardisation des désignations d'organismes dans les SIL. Nous pourrions faire la démonstration qu'OntoMic peut servir à aligner les tables d'organismes des SIL et ainsi amorcer l'interopérabilité des rapports issus de différentes institutions. Entre autres, OntoMic pourrait donner des définitions précises pour les désignations d'organismes qui ne correspondent pas à un taxon dans la taxonomie internationale.

Avec cette orientation, la comparaison automatique est laissée de côté pour un moment et les axiomes servent essentiellement à la compréhension humaine et de mot clé pour les recherches. Mais faire ces recherches n'est justement pas possible avec le modèle actuel d'OntoMic, car il n'y a pas de mécanisme d'interrogation d'axiomes en OWL.

Plusieurs stratégies peuvent être envisagées pour s'attaquer à ce problème, des requêtes dans le fichier de sérialisation de l'ontologie, des requêtes sur le jeu de triplets RDF qui représente l'ontologie, la modification des axiomes et une approche par programmation. Quoiqu'il en soit ce problème doit être résolu sans quoi OntoMic ne satisfera pas les besoins sur le terrain.

Bien d'autres développements sont à faire, comme le traitement des tables du SIL pour les règles et les messages. Car nous soupçonnons que c'est dans ces tables que nous trouverons le plus de discordances avec les protocoles. Mais ceux-ci devront suivre après la stabilisation du vocabulaire et de la méthode d'interrogation.

### **10.2.2 Concentration sur le pôle « modèle »**

Nous avons vu une approche, que nous avons nommée « classique » qui se base sur un fractionnement des logiques et en organisation des fragments.

Mais une autre manière d'aborder le problème serait de développer un raisonneur spécifique. Un raisonneur qui, à partir de la hiérarchie de classes d'OntoMic, va tenter de générer des généralisations comme le ferait un être humain. Le raisonneur spécifique devrait aussi être capable d'exploiter la structure de BFO pour, par exemple, faire les comparaisons par sous-unités logiques.

Un tel raisonneur spécifique dépendra nécessairement d'un programme qui manipulera l'ontologie en plus d'un raisonneur traditionnel. Nous savons qu'une approche par programmation est également envisageable pour faire du forage de données.

### **10.3 Plan de développement**

En conservant notre approche « bipolaire », l'étape qui suit le dépôt de ce mémoire est la préparation d'une version d'OntoMic pour l'alignement des tables d'organismes. Et une série de présentations à mes collègues pour, espérons-le, recruter une équipe de microbiologistes intéressés à développer le vocabulaire.

Par la suite et possiblement en parallèle avec le développement du vocabulaire, une exploration du traitement par programmation d'OntoMic. Les travaux sur le traitement par programmation d'OntoMic auront comme but premier de faire un instrument de forage de données, mais ils pourront également servir de base pour le travail sur un raisonneur spécifique.

## Glossaire<sup>41</sup>

<i>adjacent_to</i> .....	95	Gélose .....	33
Antibiogramme.....	24	Gram .....	47
Axiome .....	99	Guide de prélèvement .....	28
Balise.....	147	<i>has_category_value</i> .....	160
Basic Formal Ontology (BFO).....	85	<i>has_part</i> .....	95
<i>bearer_of</i> .....	96	<i>has_participant</i> .....	96
Classement .....	107	<i>has_specified_input</i> .....	166
clause .....	194	<i>has_specified_output</i> .....	167
Code test.....	28	<i>has_unit_label</i> .....	160
Colonie .....	122	<i>has_value</i> .....	159
Colonie d'intérêt.....	122	Identification.....	23
Coloration.....	47	independent_continuant.....	90
Compte-rendu.....	24	individual.....	77
critère.....	143	Inférence logique .....	96
defined_class .....	88	Information Artifact Ontology (IAO) .....	135
dependant_continuant .....	90	Information Content Entity (ICE).....	106
Dépôt – comparaison.....	114	Informations spécimen.....	22
<i>derives_from</i> .....	95	<i>inheres_in</i> .....	96
<i>derives_into</i> .....	95	Isolat d'intérêt.....	122
Détection d'antigène .....	23	Laboratoire de Santé Publique du Québec (LSPQ) .....	46
Dispositif de culture .....	32	Localisation anatomique .....	22
disposition .....	92	<i>located_in</i> .....	95
document .....	145	<i>location_of</i> .....	95
Document secondaire .....	30	Logique procédurale .....	116
document_part .....	145	Maître - réutilisation .....	114
Domaine .....	45	material_entity.....	94
<i>E coli</i> O157:H7.....	45	measurement_datum.....	135
Encodage .....	107	member .....	98
Ensemencement.....	49	Méthode de cueillette.....	22
Entreposage .....	108	Microbiologie clinique.....	14
Étendue.....	106	Milieu de culture.....	32
function.....	92		

---

<sup>41</sup> Le numéro de page à droite indique la page où est défini le terme. Cette liste ne comprend que les entités qui sont définies dans le texte et réutilisées, Lorsque le vocable existe dans le langage courant **et** dans OntoMic il sera écrit dans la fonte habituelle (Times New Roman), s'il n'existe que dans OntoMic la fonte (Arial) réservée à l'ontologie-ci est utilisée.

Nature .....	106	quality .....	91
object_aggregate .....	98	Quantificateur .....	101
occurent .....	90	realizable_entity.....	91
<i>occurs_in</i> .....	95	<i>realized_in</i> .....	96
<b>only</b> .....	166	<i>realizes</i> .....	96
Ontologie .....	76	Repérage .....	107
Ontology for Biomedical Investigations (OBI).....	84	Répertoire d'analyses .....	27
Open Biomedical Ontologies Foundry (OBO Foundry ou OBO).....	82	Requêtes .....	26
Ordonnance .....	22	Resource Description Framework (RDF) ..	99
Organisme clinique .....	35	role .....	92
Organisme d'intérêt.....	122	Sémantique.....	116
<i>part_of</i> .....	95	Shiga Toxin producing <i>E Coli</i> (STEC).....	45
<i>participate_in</i> .....	96	SoftLab.....	31
plan_specification .....	135	Spécimen.....	21
planned_process .....	148	Subsorption .....	97
Prescription.....	22	Substance biologique .....	21
process .....	94	Sujet .....	106
process_boundary .....	95	Système d'Information de Laboratoire (SIL) .....	26
processus_modèle .....	152	TAAN .....	23
Production .....	106	temporal_region.....	95
Protégé.....	190	universal.....	88
Protocole.....	30	Utilisation.....	106, 108
		Web Ontology Language (OWL) .....	99

# Bibliographie

- Arp, R., Smith, B., & Spear, A. D. (2015). *Building ontologies with Basic Formal Ontology*. Cambridge, Massachusetts: Massachusetts Institute of Technology.
- Bandrowski, A., Brinkman, R., Brochhausen, M., Brush, M. H., Bug, B., Chibucos, M. C., Clancy, K., Courtot, M., Derom, D., et al. (2016). The Ontology for Biomedical Investigations. *PLoS ONE*, *11*(4), e0154556. <https://doi.org/10.1371/journal.pone.0154556>
- Federhen, S. (2012). The NCBI Taxonomy database. *Nucleic Acids Research*, *40*(D1), D136-D143. <https://doi.org/10.1093/nar/gkr1178>
- Giraldo, O., García, A., & Corcho, O. (s. d.). SMART Protocols: SeMAntic RepresenTation for Experimental Protocols (p. 12). Présenté à 4th Workshop on Linked Science 2014 - Making Sense Out of Data (LISC2014), Riva del Garda, Trentino, Italia. Consulté à l'adresse <http://ceur-ws.org/Vol-1282/>
- Jupp, S., Horridge, M., Iannone, L., Klein, J., Owen, S., Schanstra, J., Wolstencroft, K., & Stevens, R. (2011). Populous: a tool for building OWL ontologies from templates. *BMC Bioinformatics*, *13*(Suppl 1), S5. <https://doi.org/10.1186/1471-2105-13-S1-S5>
- Leber, A. L. (Éd.). (2016). *Clinical microbiology procedures handbook* (4th edition). Washington, DC: ASM Press.
- Morosoff, P., Rudnicki, R., Bryant, J., Farrell, R., & Smith, B. (2015). Joint Doctrine Ontology: A Benchmark for Military Information Systems Interoperability. Dans *STIDS 2015 Semantic Technology for Intelligence, Defense, and Security*. Fairfax, VA, USA.
- O'Connor, M. J., Halaschek-Wiener, C., & Musen, M. A. (2010). Mapping Master: A Flexible Approach for Mapping Spreadsheets to OWL. Dans P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, & B. Glimm (Éd.), *The Semantic Web – ISWC 2010* (Vol. 6497, p. 194-208). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-17749-1\\_13](https://doi.org/10.1007/978-3-642-17749-1_13)
- Russell, B. (1918). THE PHILOSOPHY OF LOGICAL ATOMISM [with Discussion]. *The Monist*, *28*(4), 495-527.
- Scientific Nomenclature - Emerging Infectious Disease journal - CDC. (s. d.). Consulté 31 mai 2018, à l'adresse <https://wwwnc.cdc.gov/eid/page/scientific-nomenclature>
- SNOMED CT - Summary | NCBO BioPortal. (s. d.). Consulté 25 mai 2018, à l'adresse <http://bioportal.bioontology.org/ontologies/SNOMEDCT/?p=summary>
- SNOMED Home page. (s. d.). Consulté 21 octobre 2018, à l'adresse <http://www.snomed.org/>
- Soldatova, L. N., Nadis, D., King, R. D., Basu, P. S., Haddi, E., Baumlé, V., Saunders, N. J., Marwan, W., & Rudkin, B. B. (2014). EXACT2: the semantics of biomedical protocols. *BMC Bioinformatics*, *15*(Suppl 14), S5. <https://doi.org/10.1186/1471-2105-15-S14-S5>
- Vita, R., Overton, J. A., Greenbaum, J. A., Sette, A., OBI consortium, & Peters, B. (2013). Query enhancement through the practical application of ontology: the IEDB and OBI. *Journal of Biomedical Semantics*, *4*(Suppl 1), S6. <https://doi.org/10.1186/2041-1480-4-S1-S6>

What LOINC is — LOINC. (s. d.). Consulté 24 mai 2018, à l'adresse <https://loinc.org/get-started/what-loinc-is/>

# Annexe 1 Exemple de requête

PRÉSENTEZ-VOUS AVEC CE FORMULAIRE ET VOTRE CARTE D'ASSURANCE MALADIE AU CENTRE DE PRÉLÈVEMENT ENTRE 6 H 30 ET 14 H 30 DU LUNDI AU VENDREDI (sauf les jours fériés)



**REQUISITION LABORATOIRE**

Centre de prélèvements  
480, 5e rue, Saint-Jérôme, J7Z 0H6  
(près des Galeries des Laurentides)

**IMPORTANT :** tout formulaire non identifié ou non signé sera refusé

Info: www.cdsjlabo.org

**PRESCRIPTEUR**

Clinique : \_\_\_\_\_

Date de l'ordonnance : \_\_\_\_\_

Nom, prénom : \_\_\_\_\_

Signature : \_\_\_\_\_

N° permis : \_\_\_\_\_

CV  OC # \_\_\_\_\_

**CLIENT**

N° RAMQ : \_\_\_\_\_  
(obligatoire)

Nom, prénom : \_\_\_\_\_  
(obligatoire)

Renseignements cliniques : \_\_\_\_\_

**BIOCHIMIE**

4 Electrolytes (Na K Cl) ●

414 Créatinine (peut calculer filtration glomérulaire) ●

406 Glucose ● 

415 Cholestérol ●

416 HDL/LDL Chol/Trig ● 

417 Triglycérides ● 

492 Apolipoprotéine A ●

493 Apolipoprotéine B ●

427 Bilirubine totale ●  
Bilirubine directe référée si Bilirubine totale anormale

423 ALP ●

424 GGT ●

419 ALT (GPT) ●

421 CK totale ●

426 Lipase ●

441 PSA total ●

453 CEA ●

442 CA 125 ●

598 CA 15-3 ●

595 CA 19-9 ●

446 Cortisol AM (8-10 h) ●

447 Cortisol PM (16-20 h) ●

546 Facteur rhumatoïde ●

591 Insuline ● 

444 Prolactine (β-11 h) ● 

**BIOCHIMIE (SUITE)**

437 Hémoglobine glyquée ● 

480 Microalbuminurie ●

501 Analyse d'urine ●

430 Protéines totales ●

431 Albumine ●

435 Calcium ●

434 Phosphore ●

429 Acide urique ●

462 Calcium ionisé ●

436 Magnésium ●

461 Immunoglobulines (IgA, IgG, IgM) ●

953 CTFF (TIBC) (par + coefficient séparation) ●  
Ne prendre aucun supplément de fer dans les 24h précédentes

467 Ferritine ●

532 Parathormone (PTH intacte) ●

450 TSH (si TSH anormale T4L réflexée) ●

452 Anticorps anti-thyroïdiens (Anti-TPO et Anti-Thyroglobuline) ●

469 FSH ●

471 LH ●

445 Estradiol (œstrogénés) ●

596 Testostérone biodisponible calculé ●

530 Testostérone totale ●

Autre test

**PRÉLÈVEMENT**

APPOSER L'ÉTIQUETTE CODE-BARRE DE LA REQUÊTE ICI

Clinique, centre ou service de prélèvement : \_\_\_\_\_

Date et heure prélèvement : \_\_\_\_\_

N° d'accréditation: P0707 : \_\_\_\_\_

**DIVERS**

101 Groupe sanguin, Rh ● 

201 Culture d'urine ●

Date et heure prélèvement : \_\_\_\_\_

Antibiotiques reçus: NON  OUI

**HÉMATOLOGIE**

603 FS (Formule sanguine) ● 

601 FSC (Formule sanguine complète) ● 

605 Réticulocytes ● 

606 ANA ●

607 ENA ●

608 Anti-dsDNA ●

609 Anti-tissus ●

683 Anti-transglutaminase et gliadine ●

**COAGULATION**

721 PT (RIN) ● 

Inscrire type anticoagulant : \_\_\_\_\_

719 PTT (suivi héparine) ● 

Inscrire type anticoagulant : \_\_\_\_\_

Autre test

Tube de sang à prélever; Boisson : jaune ● lavande 3.0 ml ●  Pot culture : stérile ● milieu Cary-Blair ou entériques ● milieu SAF ● stérile bien fermé adhésif ●

Écouvillon : Amie ● charbon ●  Milieu de culture Strepto B ● 

GC-1662 (SRN 14000041) LAB-FOR-139 révision octobre 2016

Milieu de culture APTIMA : prélèvement vaginal ● urine ● 

