



HAL
open science

Contribution à la mise en oeuvre d'un outillage unifié pour faciliter la qualification d'environnements normés

Anthony Gelibert

► **To cite this version:**

Anthony Gelibert. Contribution à la mise en oeuvre d'un outillage unifié pour faciliter la qualification d'environnements normés. Modélisation et simulation. Université Grenoble Alpes, 2016. Français. NNT : 2016GREAM059 . tel-01649311

HAL Id: tel-01649311

<https://theses.hal.science/tel-01649311v1>

Submitted on 27 Nov 2017

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Anthony GELIBERT

Thèse dirigée par **Ioannis PARISSIS** et
coencadrée par **Denis GENON-CATALOT** et **Sébastien JEAN**

préparée au sein du **Laboratoire de Conception et
d'Intégration des Systèmes (LCIS)**
dans **l'École Doctorale Mathématiques, Sciences et
Technologies de l'Information, Informatique (MSTII)**

Contribution à la mise en œuvre d'un outillage unifié pour faciliter la qualification d'environnements normés

Thèse soutenue publiquement le **27 octobre 2016**,
devant le jury composé de :

M. Ioannis PARISSIS

Professeur à l'Institut Polytechnique de Grenoble

Membre

M. Denis GENON-CATALOT

Maitre de conférences à l'Université Grenoble Alpes

Membre

M. Sébastien JEAN

Maitre de conférences à l'Université Grenoble Alpes

Membre

M. Camille SALINESI

Professeur à l'Université Paris 1

Rapporteur

M. Vincent CHAPURLAT

Professeur à l'Institut Mines-Telecom

Rapporteur

Mme. Laure PETRUCCI

Professeur à l'Université Paris 13, Directrice du LIPN

Présidente



Remerciements

Un doctorat est un cheminement tout autant personnel que collectif. Or, si ce sont bien mes idées qui sont présentées dans ce mémoire, elles n'auraient jamais pu voir le jour sans le concours de toutes les personnes ayant croisé ou partagé ma route durant ces cinq années.

Ainsi, mes tout premiers remerciements vont à Cécile, Téo, Lola et Nino. Ils m'ont chaque jour rappelé les raisons de mon engagement dans cette aventure, m'ont aidé à relativiser dans les périodes de doute et supporté dans les périodes difficiles.

Je remercie également mes parents : Serge et Rachel, pour leur soutien et leurs conseils. Cela restait définitivement une affaire de « 0 » et de « 1 », je l'ai enfin compris.

Lorsque je fais le bilan de ces travaux, je ne peux qu'exprimer ma gratitude à Denis GENON-CATALOT et Gérard SANTAILLER. Ces travaux doctoraux ont débuté par leur rencontre et je ne saurais leur témoigner toute la reconnaissance qu'ils méritent pour leur patience et leur engagement à mes côtés durant ces quelques années.

J'ai une pensée toute particulière pour Nocosium... Ce fut une fabuleuse aventure et je suis fier d'y avoir contribué.

Je tiens à remercier tout particulièrement Sébastien JEAN pour tous les travaux menés ensemble depuis mon DUT, pour son soutien indéfectible et l'optimisme forcé qu'il place en toute chose...

Je remercie Robert FRATTINO pour m'avoir aidé à remettre de l'ordre dans ma démarche et mes idées, ce fut un travail salvateur.

Je voudrais remercier Ioannis PARISSIS d'avoir dirigé ces travaux ; les Pr. Camille SALINESI et Vincent CHAPURLAT d'avoir accepté d'en être les rapporteurs et enfin la Pr. Laure PETRUCCI de faire partie du jury.

Enfin, je remercie tous les amis et collègues ayant croisé ma route que ce soit au LCIS, au sein de Nocosium ou à l'IUT de Valence. Dans le désordre et en en oubliant très certainement (qu'ils me pardonnent) : François, Lionel, Fabien, Nicolas, Jean-Paul, Michel, Amalric, Emmanuel, Julien, Jérémy, Sandrine, Amaury, François, Guillaume, Benjamin, Nathanaël...

Au terme de ces travaux doctoraux, ce n'est pas tant le chemin parcouru, mais la voie s'ouvrant maintenant à moi que je regarde. La vie d'un chercheur est une longue route et ce document en est seulement l'un des premiers pavés...

Notes préliminaires

Les travaux de doctorat présentés dans ce mémoire ont commencé en fin d'année 2011 dans le cadre du dispositif CIFRE (« Conventions Industrielles de Formation par la REcherche »¹), mis en œuvre par l'Association Nationale de la Recherche et de la Technologie (ANRT), au sein de la société Nocosium [Noc15].

Nocosium était alors une jeune TPE (« Très Petite Entreprise ») de la région lyonnaise, spécialisée dans la réalisation d'environnements confinés médicaux. Ce domaine « métier » précis a donc constitué le champ d'expérimentation principal de ces travaux et ceux-ci se sont nourris de ses besoins.

Des difficultés ont malheureusement conduit la société Nocosium à déposer le bilan en début d'année 2016, la liquidation s'étant opérée dans les mois suivants.

Ainsi, la vision industrielle exprimée dans ces travaux est celle d'une TPE aujourd'hui éteinte... Malgré cela, les échanges et les perspectives ont de nombreuses fois confirmé la pertinence et la réalité du besoin identifié. Ainsi, si l'entreprise support de ces recherches est aujourd'hui défunte, les objectifs restent tout à fait d'actualité.

1. Bourse CIFRE N°289/2011.

Résumé long

Les blocs chirurgicaux, les laboratoires en biotechnologies et autres salles blanches industrielles sont des lieux critiques abritant des processus complexes et dont la mise en œuvre peut générer de nombreux risques, que ce soit pour l'environnement, pour les « utilisateurs » ou pour les processus réalisés. Cela conduit inévitablement à un traitement spécifique par le législateur (qu'il soit national ou international) afin d'en encadrer au maximum la conception, la réalisation et l'utilisation. On peut alors clairement employer le terme « d'environnement normé ».

Par essence, ces environnements extrêmement complexes requièrent de très nombreuses compétences pour correspondre aux exigences du législateur et aux attentes des utilisateurs. Malheureusement, les personnes et entreprises capables de maîtriser cette complexité sont relativement rares et ainsi, tout aussi surprenant que cela puisse paraître, la conception et réalisation des environnements normés diffèrent souvent relativement peu de celle des environnements « classiques ».

Dans cette thèse, nous proposons une approche unifiée visant à faciliter la qualification des environnements normés pour en améliorer la conception générale (*safety-by-design*). Nous nous inscrivons clairement dans les travaux d'Ingénierie des Exigences (RE), en définissant un cadre de pensée souple et générique, traitant l'environnement dans sa totalité, d'une phase d'analyse préalable pour en appréhender les spécificités, en passant par la qualification de sa conception, pour s'achever par une traçabilité de son fonctionnement.

Dans cette approche, nous étudions plus particulièrement les moyens :

- de représenter les contextes normatifs des environnements normés sous la forme d'un graphe unique, enrichi de toutes les situations analysées, ceci dans le but d'obtenir une manière efficace et originale de connaître les contraintes liées à un objectif particulier ;
- de représenter la modélisation du contexte normatif d'un environnement normé sous la forme d'un métamodèle initial puis d'une suite d'aspects permettant de modéliser l'environnement final sous forme de « calques » vérifiables successivement.

La démarche est illustrée dans cette thèse au travers du domaine des environnements confinés médicaux. Après une étude approfondie de ce domaine d'application, et notamment des conséquences potentielles d'une mauvaise conception, nous présentons l'une de nos expérimentations dans ce contexte particulier.

La démarche proposée est toutefois générique et pourrait donc s'appliquer à l'ensemble des environnements normés.

Long abstract

Operating theatres, biotechnology laboratories and other industrial clean rooms are critical places where complex processes take place, which induce several risks for environment, users or achieved processes. This critical conditions leads the law-maker to consider independently these spaces so as to best control their design, realization and using. Thus these spaces can clearly be called as “standardized environment”.

These highly complex environments inherently require multiple skills to meet the law-maker requirements and users expectations. Unfortunately, people and companies able to manage this complexity are relatively rare. Therefore, surprisingly, design and building of standardized environments have very few differences with classical ones.

We propose in this thesis a unified approach which aim is to help qualification of standardized environments so as to enhance their general design (safety-by-design). We clearly join Requirements Engineering (RE) works, by defining a flexible and generic way of thinking, considering environment in its entirety, from a preliminary analysis phase, throughout qualification design, reaching the operation traceability.

By using this approach, we will concentrate particularly on ways to:

- Represent normative context of standardized environment under a single graph, augmented with all analyzed situations, in order to obtain in an efficient and original way the restrictions related to this specific purpose.
- Represent normative context modeling under a first metamodel, followed by a series of complementary aspects. This allows the representation of final environment with sequentially “layers”, validated individually.

The illustration of this approach is found in this thesis through the confined medical spaces field. Following an in-depth study of this application field, above all on potential consequences of a bad design, we will present one of our experiments in this specific situation.

The suggested methodology is generic and could therefore find an application to the whole standardized environment field, despite the fact the technological transfer is currently limited to the Nocosium Company.

Table des matières

Table des matières	1
1 Introduction générale	7
1.1 Introduction	7
1.2 Contexte	9
1.2.1 L'hôpital	10
1.2.2 Vers un hôpital « numérique »	10
1.2.3 Dépasser ce cadre institutionnel	12
1.3 Objectifs de la thèse	13
1.4 Contenu du mémoire	14
I Contexte et état de l'art	17
2 Contexte et besoin industriel	19
2.1 Introduction	19
2.2 Prise de recul	21
2.3 Les environnements confinés médicaux	23
2.3.1 Définition	23
2.3.2 Maintien du confinement	25
2.3.3 Arrangement normatif	26
2.3.4 Infections nosocomiales	27
2.4 Conclusion	29
3 Les environnements normés	31
3.1 Positionnement industriel	32
3.1.1 Conception	32
3.1.2 Réalisation : le découpage en lots	34
3.1.3 Conclusion	36
3.2 Ingénierie des Exigences	37
3.2.1 Motivation	37
3.2.2 Définition et objectifs	38
3.2.3 Contraintes	39
3.2.4 Activités	40
3.2.5 Historique	42
3.2.6 Travaux connexes	50
3.3 Conclusion	52

4	Problématique	55
4.1	Les différentes problématiques	56
4.1.1	Analyse préalable	56
4.1.2	Qualification	57
4.1.3	Supervision et audit	58
4.2	Synthèse finale	59
II	Contributions	61
5	Approche unifiée	63
5.1	Schéma générique	63
5.2	Phase 0 — Analyse du cadre normatif	65
5.2.1	Constitution de l'arrangement normatif	65
5.2.2	Expression des contraintes	67
5.2.3	Informatisation des contraintes	68
5.3	Phase 1 — Aide à la qualification	68
5.3.1	Questionnaire et sélection des contraintes	70
5.3.2	Vérification	70
5.4	Phase 2 — Qualification de l'usage	72
5.4.1	Plateforme de traçabilité	73
5.4.2	Outil d'audit	74
5.5	Synthèse	74
6	Analyse du cadre normatif	77
6.1	Expression des contraintes	77
6.1.1	Motivation	78
6.1.2	Principe	78
6.1.3	Détails	79
6.1.4	Conclusion	84
6.2	Transformation des contraintes	84
6.2.1	Principe	85
6.2.2	Les normes sous forme de graphe	87
6.2.3	Nettoyage des dépendances	88
6.2.4	Inversion des dépendances	90
6.2.5	« Linéarisation » des contraintes	91
6.3	Conclusion	96
7	Aide à la qualification	99
7.1	Principe	99
7.2	Questionnaire	100
7.2.1	Création	100
7.2.2	Forme	100
7.2.3	Usage	102
7.2.4	Du graphe des normes à l'outil d'aide à la qualification	103
7.3	Outil de qualification	103
7.3.1	Extraction des étapes	103
7.3.2	Orchestration des étapes	104
7.4	Conclusion	111

III	Mise en œuvre	113
8	Graphes de normes	115
8.1	Rappel de l'objectif	115
8.2	Sélection de la technologie	116
8.2.1	Contraintes	116
8.2.2	Première hypothèse : SGBD Relationnels	117
8.2.3	Seconde hypothèse : NoSQL	118
8.3	NEO4J	122
8.3.1	Un peu d'histoire	122
8.3.2	Modalités d'interaction avec NEO4J	123
8.3.3	Le langage CYPHER	124
8.4	Mise en œuvre : API et implémentation	127
8.4.1	La notion de norme	128
8.4.2	L'interaction avec le serveur	129
8.4.3	Stratégie de parcours de la base de données	129
8.4.4	Gestionnaire de scénarios	130
8.4.5	Conclusion	131
8.5	Résultats	131
8.5.1	Représentation visuelle	133
8.5.2	Construction d'un graphe	135
8.5.3	Graphe complet	137
8.6	Conclusion	140
9	Outil d'aide à la qualification	143
9.1	Rappel de l'objectif	143
9.2	Sélection de la technologie	145
9.3	KerMeta 2	149
9.3.1	Introduction	149
9.3.2	Syntaxe	150
9.4	Mise en œuvre et résultat	153
9.4.1	Outillage	153
9.4.2	Architecture générale	154
9.4.3	Fonctionnement général	155
9.4.4	Résultat	155
9.5	Conclusion	156
10	Illustration	157
10.1	Présentation	157
10.1.1	Cas réel étudié	158
10.1.2	Motivations	158
10.1.3	Structure de l'environnement étudié	158
10.2	Arrangement normatif considéré	160
10.2.1	Code de la Santé Publique	161
10.2.2	Décret n° 2000-1316 du 26 décembre 2000	161
10.2.3	Bonnes Pratiques de Pharmacie Hospitalière	161
10.2.4	Bonnes Pratiques de Préparation	162
10.2.5	Recommandations pour la manipulation des médicaments cytotoxiques	162
10.2.6	NF S 90-351	162

10.2.7	ISO 14644-1	162
10.2.8	ISO 14698-1	162
10.3	Analyse normative	163
10.3.1	Code de la Santé Publique (CSP)	163
10.3.2	Bonnes Pratiques de Pharmacie Hospitalière	164
10.3.3	Bonnes Pratiques de Préparation (BPP)	165
10.3.4	Recommandations pour la manipulation des médicaments cytotoxiques	166
10.3.5	ISO 14644-1	166
10.3.6	NF S 90-351	166
10.3.7	ISO 14698-1	168
10.3.8	Conclusion	168
10.4	Questionnaire	168
10.5	Traitement du graphe	172
10.5.1	Nettoyage	173
10.5.2	Inversion	173
10.5.3	Tri topologique	173
10.6	Architecture informatique	176
10.7	Interrogation du graphe	177
10.8	Modélisation	178
10.9	Conclusion	187
11	Transfert des contributions	189
11.1	Contexte	189
11.2	Mise en œuvre « complète »	190
11.2.1	Schéma	190
11.2.2	Détails	190
11.2.3	Conclusion	196
11.3	Adaptation aux contraintes industrielles	196
11.3.1	Schéma	196
11.3.2	Détails	196
11.3.3	Conclusion	198
11.4	Conclusion générale	198
IV	Conclusion, limites et perspectives	201
12	Conclusion	203
13	Limites	207
13.1	Limites à notre analyse du cadre normatif	207
13.1.1	Représentation réticulaire des normes	207
13.1.2	Bilan	211
13.2	Aide à la qualification : approche théorique	211
13.2.1	« Aspect-Oriented Modeling » ou « Métamodèle fixe » ?	212
13.2.2	Complexité et passage à l'échelle	213
13.2.3	Industrialisation	213
13.3	Limites à notre mise en œuvre avec Neo4J	214
13.4	Conclusion	215

14 Perspectives	217
14.1 Auditer un environnement normé	217
14.2 Model@run.time	219
14.3 Intégration à la MNB	220
V Annexes	223
A Modélisation réticulaire avec un SGBDR	225
A.1 Adaptation à notre objectif	225
A.2 Pertinence de cette représentation	229
A.2.1 Filtrer les redondances	229
A.2.2 Inversion des dépendances	230
A.2.3 Extraction de chemins de longueur variable	232
A.3 Conclusion	236
B URCC : codes sources complémentaires	237
B.1 Application Java	237
B.2 Script de traitement de la liste des fichiers	237
B.3 Métamodèle du bâti	237
B.4 Aspect pour l'instrumentation	239
Bibliographie	243
Table des figures	263
Table des codes sources	267
Liste des acronymes	269

Introduction générale

1.1 Introduction

Les sociétés modernes — et la société française ne fait pas exception — sont devenues extrêmement friandes de normes, de conventions, de spécifications, etc. Les activités et productions techniques humaines se sont ainsi vues largement encadrées et nécessitent toujours plus de formations et de compétences afin d'être réalisées dans les « règles de l'art » ; ce respect conditionnant dans de nombreux cas le droit d'exercer une activité ou de commercialiser un produit (marquages divers, professions réglementées, etc.). Les diplômes, normes, standards, recueils de bonnes pratiques, etc. visent ainsi à fournir des garanties et à poser des garde-fous sur les méthodes de conception et la qualité des produits finis.

Cette évolution est motivée par de nombreux facteurs. Nous allons aborder les plus intéressants, pour la nature des travaux présentés dans cette thèse.

- Les différentes révolutions industrielles ont ainsi profondément marqué nos civilisations et le degré de technicité actuellement requis pour de très nombreuses opérations interdit tout dilettantisme.
- La juridicisation et judiciarisation¹ de la société a également eu ses répercussions [Rou08]. Ainsi — bien que les dernières études sur le sujet identifient le domaine médical comme peu atteint par le phénomène [Rou09, Lau13] —, de nombreuses grandes crises sanitaires ont émaillé les dernières décennies et forcé les autorités à se positionner face aux pressions sociétales. Le respect obligatoire de normes et de standards précis constitue l'une des réponses apportées.

1. « *La juridicisation est l'extension du droit et des processus juridiques à un nombre croissant de domaines de la vie économique et sociale (...) et la judiciarisation est la tendance des justiciables à confier au système judiciaire la gestion de tous leurs différends et le règlement de tous les problèmes sociaux.* » (Centre de traduction et de terminologie juridiques (CTTJ), Faculté de droit, Université de Moncton).

— Le « marché » pousse lui aussi à la normalisation et à la réglementation de différentes activités. En effet, de nombreuses professions voient d'un très mauvais œil l'arrivée d'une concurrence féroce, tirant les prix vers le bas. Dans les environnements normés (c.-à-d., les environnements règlementés par des normes), des différences importantes de tarifs traduisent très souvent une méconnaissance des contraintes normatives imposées ou une différence importante dans la qualité de la prestation². Lorsque ce genre de problème devient systémique — par exemple, à l'ouverture d'une activité à la concurrence internationale — les partenaires sociaux en appellent alors à la réglementation des activités pour harmoniser les tarifs et lutter contre le *dumping*³ tarifaire. La définition de normes communes (contraignant ainsi les entreprises en termes de résultats [et donc bien souvent de moyens]) ou encore la réglementation de certaines professions (contraignant les diplômes requis) sont des réponses couramment apportées à ce genre de situation.

La conjonction de ces facteurs conduit à l'apparition d'une « société des normes » où tout est règlementé. La Figure 1.1 est une affiche de l'Agence Française de NORmalisation (AFNOR) sur la normalisation des sports et des activités de plein air, une illustration intéressante de cette nouvelle société.



FIGURE 1.1 – Illustration de la « normalisation » de notre société par l'AFNOR

Appliqué aux environnements (au sens « bâti » du terme), cela se traduit

2. Dans tous les cas, c'est l'utilisateur final le principal perdant.

3. **Définition :** Pratique qui consiste à diminuer le prix d'une marchandise lorsqu'elle est vendue sur des marchés extérieurs.

par des contraintes toujours plus poussées sur leur conception, construction et utilisation. C'est d'autant plus vrai pour les environnements très fortement encadrés comme les centrales nucléaires, les environnements médicaux ou salles blanches industrielles (nanotechnologies, etc.) auxquels nous nous intéressons dans ces travaux.

Intuitivement, on imagine ce type de construction réalisée par des experts spécifiquement formés sur le sujet (comme exprimé précédemment). Malheureusement, la réalité est bien plus contrastée et dépendra de la pression sociale sur le sujet. Ainsi, pour les centrales nucléaires, sujet brûlant, les précautions deviennent draconiennes. À l'inverse, dans les bâtiments tertiaires⁴, longtemps considérés comme de peu d'importance, les exigences sont bien plus faibles, la situation évolue et des problématiques telle l'efficacité énergétique font actuellement bouger les lignes.

Les sociétés responsables des constructions entendent monnayer chèrement leurs compétences, leur « Business Model » reposant sur une intensive gestion du savoir (*knowledge management* en anglais). Cependant, leur démarche est totalement *ad hoc* et requiert une connaissance fine de l'environnement ciblé (dépendant du type précis d'environnement), du projet à réaliser (dépendant du client) et ne tolérant pas l'évolution sans nouvelle intervention de l'expert ainsi qu'une nouvelle analyse de conformité. Ces experts maîtrisent également les caractéristiques pertinentes à suivre durant l'usage pour garantir la bonne marche et la qualité de l'environnement.

Il nous semble légitime de nous interroger sur cet état de fait. N'y a-t-il pas moyen de définir une solution « générique » qui permettrait pour un type particulier d'environnement normé d'en apprécier la conformité réglementaire ? De la même manière, est-il possible de le faire pour un projet particulier ? Et enfin, peut-on articuler ces différentes analyses pour les unifier et poursuivre ces vérifications jusqu'à l'usage réel ?

1.2 Contexte

Les travaux présentés dans cette thèse se nourrissent des besoins exprimés par les entreprises spécialisées dans la conception d'environnements normés, pour proposer une approche globale à la qualification des environnements réalisés.

De par la nature des financements employés pour cette thèse (cf. les notes préliminaires), le domaine de la santé était le plus approprié et parmi toutes les thématiques accessibles, le « monde hospitalier » nous a semblé le plus intéressant : importance, volume financier, contexte normatif et profonde mutation en cours du paysage technologique.

4. « Le tertiaire dans le bâtiment correspond aux bâtiments occupés par les activités du secteur tertiaire (commerces, bureaux, santé, enseignement, infrastructures collectives destinées aux sports, aux loisirs, aux transports, CHR — cafés/hôtels/restaurants —, et tous les E.R.P — établissements destinés à recevoir du public). » (ClimaMaison, <http://www.climamaison.com/lexique/tertiaire.htm>).

1.2.1 L'hôpital

L'hôpital a une longue histoire : son rôle et sa place dans notre système de santé ayant largement évolué au fil des siècles [Mol05]. Peu à peu, il s'est vu confier un rôle si important qu'il est parfois remis en cause et qu'il constitue l'une des nombreuses interrogations et craintes sur la pérennité de notre modèle de santé [Ins12].

De fait, les comparaisons internationales mettent en lumière que l'hôpital a, en France, une place plus grande que dans la moyenne des pays comparables en matière de résultats de santé. Nos hôpitaux représentent plus de 35 % de nos dépenses de santé contre 29 % en moyenne dans les pays de l'OCDE et nous avons 17,88 Equivalent Temps-Plein (ETP) hospitaliers pour 1000 habitants contre 10,99 en Allemagne [II12]. Le terme « d'hospitalo-centrisme » est utilisé pour décrire — et déplorer — cette situation. La pression économique joue un rôle dans cette interrogation, l'hôpital étant considéré comme plus coûteux que des formes de prise en charge ambulatoires ou moins techniques.

De nombreuses pistes ont été abordées pour réduire les dépenses hospitalières [Ass14] et, lors des nombreux efforts entrepris, l'une des constatations régulières est la fragilité du système d'information du dispositif hospitalier, particulièrement faible en matière d'organisation des soins [Ins12].

L'adaptation des systèmes d'information n'a par ailleurs pas toujours été menée à son terme. Pourtant, l'existence d'un système décisionnel est devenue un élément clé dans la capacité des établissements à accompagner les réformes et à maîtriser leur équilibre financier. Les équipes qui pilotent les pôles doivent pouvoir disposer d'indicateurs fiables et partagés. Il importe de pouvoir travailler sur un seul et même système, auquel accèdent tant le directoire que le chef de pôle ou n'importe quel médecin ou cadre. Cette attention au pilotage interne de l'établissement grâce à un système d'information cohérent est confirmée dans les orientations nationales [Min14] et constitue un objectif majeur de l'action gouvernementale dans le domaine.

1.2.2 Vers un hôpital « numérique »

Afin d'améliorer l'organisation des soins, le développement et la modernisation des Systèmes d'Information Hospitalier (SIH) sont devenus un enjeu majeur. En prévision des prochaines étapes du développement des SIH au service d'une meilleure prise en charge des patients, la Direction Générale de l'Offre de Soins (DGOS) a lancé fin 2011 le programme « Hôpital numérique » [Min14], plan stratégique de développement et de modernisation des SIH pour la période 2012-2017.

Le programme « Hôpital numérique »

Pour la mise en œuvre de la Stratégie Nationale de Santé (SNS), les technologies numériques constituent un levier majeur à la modernisation des acteurs de l'offre de soins.

Dans ce contexte, le programme « Hôpital numérique », politique nationale relative aux SIH pilotée par la DGOS, vise à :

- fixer les objectifs et les priorités à 5 ans (2012-2017) ;

- mobiliser l'ensemble des acteurs concernés ;
 - accompagner les établissements de santé dans leur transformation par les technologies de l'information et de la communication.
- Le programme « Hôpital numérique » s'appuie sur les ambitions suivantes :
- coordonner l'ensemble des acteurs (établissements, industriels, Agences Régionales de Santé (ARS), ministères) autour d'une feuille de route commune pour les SIH ;
 - amener l'ensemble des établissements de santé à un niveau de maturité de leurs systèmes d'information, permettant une amélioration significative de la qualité et de la sécurité des soins dans des domaines fonctionnels prioritaires ;
 - soutenir les projets innovants.

Le palier de maturité ciblé doit permettre le partage et l'échange d'informations autour d'une prise en charge coordonnée du patient avec les différents acteurs du territoire et l'amélioration significative de la qualité, la sécurité des soins et la performance dans des domaines fonctionnels prioritaires liés aux processus de soins, sur un socle assurant la sécurité des données.

Le socle commun, ou palier de maturité cible, a été défini en concertation avec des représentants des institutions et des opérationnels du terrain, et est constitué :

- De domaines constituant les « prérequis » :
 - identités – mouvements,
 - fiabilité – disponibilité,
 - confidentialité ;
- De domaines fonctionnels :
 - les résultats d'imagerie, de biologie et d'anatomopathologie ;
 - le dossier patient informatisé et interopérable ;
 - la prescription électronique alimentant le plan de soins ;
 - la programmation des ressources et l'agenda du patient ;
 - le pilotage médico-économique.

Afin d'accompagner l'ensemble des établissements de santé dans l'atteinte du palier de maturité minimum, le programme propose d'agir simultanément sur quatre axes stratégiques :

Axe 1 : Gouvernance. Comblent les manques de la gouvernance des Systèmes d'Information (SI) et favoriser dans les SI des professionnels de santé et cadres dirigeants.

Axe 2 : Compétences. Renforcer les compétences relatives aux SIH.

Axe 3 : Offre. Stimuler et structurer l'offre de solutions.

Axe 4 : Financement. Financer un socle de priorités, subordonné à l'atteinte de cibles d'usages.

Synthèse

Le programme « Hôpital numérique » est un projet ambitieux visant à augmenter l'usage de l'informatique dans les établissements de santé pour améliorer le partage et l'échange d'informations entre les différents acteurs du système de santé. Cela doit permettre une amélioration significative de la qualité, la

sécurité des soins et la performance des processus de soins, sur un socle assurant la sécurité des données.

Conformément à ces objectifs, les indicateurs [Min12] produits par le programme — et sur lesquels les établissements de santé doivent se concentrer — adoptent une orientation résolument « médicale ». Ils sont peu nombreux, très précis, se focalisant exclusivement sur le système d'information et principalement sur sa capacité à traiter les données médicales pour les rendre accessibles et manipulables par l'ensemble du monde hospitalier (patients, praticiens, administratifs, direction. . .).

1.2.3 Dépasser ce cadre institutionnel

Le programme « Hôpital numérique » est une étape importante pour l'amélioration des hôpitaux, afin qu'ils deviennent des établissements moins coûteux, plus rassurants pour les patients et plus aisément gérables par les décideurs. L'une des pistes actuellement explorées est l'amélioration du Système d'Information des hôpitaux. Cette initiative vise à numériser le matériel documentaire médical existant pour permettre une meilleure consultation de celui-ci ainsi qu'une meilleure prise de décision. En effet, on peut aisément imaginer la quantité d'informations et la qualité du tableau de bord accessible par croisement des sources de données tant pour le patient que pour la direction, un excellent exemple d'application de l'exploration de données (*data mining* en anglais) [WLK⁺14].

À mon sens, c'est une première étape, ouvrant de nouvelles perspectives captivantes. Parmi celles-ci, nous pourrions compléter les informations médicales par d'autres données, issues d'autres domaines, plus « techniques », mais pouvant avoir une interprétation médicale dans certains contextes (infections nosocomiales, animation des Comité de Lutte contre les Infections Nosocomiales (CLIN), etc.).

Mais surtout, en prenant un peu de recul, ces nouveaux hôpitaux au Système d'Information adéquat et performant n'ouvrent-ils pas la porte à de nouvelles méthodes de conception ? Car, indubitablement, il faut appréhender la problématique hospitalière globalement, dès sa conception. Le passé fournit de très nombreux exemples de projets « catastrophiques », mal analysés et donc mal conçus, ayant entraîné surcoûts, retards et problèmes lors de l'usage réel. Un projet hospitalier doit prendre en compte les aspects de santé publique, réglementaire, d'urbanisme, économiques (nationaux et locaux), écologiques, etc.

Historiquement, en France, cette conception est réalisée :

- pour le domaine public en conformité avec la loi de 1980 [Fra10], dite loi MOP, processus impliquant un éclatement des missions d'études [Fra06] et générant une complexité du mécanisme de la construction. Cela engendre trop souvent le non-respect des objectifs initiaux (prix, délai, mais surtout, pour les activités hyperspécialisées, des écarts lourds rendant au final le processus hospitalier moins efficient dans son ensemble) ;
- pour le domaine privé, par de grands preneurs d'ordre. Leur métier de base étant toutefois loin du processus de santé, ils s'associent les compétences utiles par la suite. Là encore, souvent, pour les activités hyperspécialisées, les mêmes écarts lourds pénalisent les établissements et obligent d'autre part, les pouvoirs publics à mettre en place les contrôles et suivis des « performances » [Hau14].

Devant ces perspectives, nous envisageons une approche novatrice tirant parti de toutes les nouvelles pistes offertes par l'informatisation des environnements normés (soutenu en cela par les pouvoirs publics) pour répondre à l'extrême complexité de conception des domaines concernés. En empruntant une piste originale dans cet axe de recherche, nous pourrions *in fine* permettre de réaliser des économies d'échelles importantes tout en offrant une voie de recherche innovante.

1.3 Objectifs de la thèse

Les environnements normés sont extrêmement complexes et nécessitent un nombre très important de compétences variées pour être réellement pertinent et de la qualité attendue par leurs utilisateurs. Malheureusement, les personnes et les entreprises capables d'appréhender et de mettre en œuvre cette complexité sont relativement peu nombreuses et entendent monnayer cette compétence qu'ils appliquent au cas par cas, en fonction du type de réalisation et des données précises du projet (l'évolution de l'un de ces deux facteurs requérant une nouvelle analyse du projet).

Dans le cadre de leurs activités, ces entreprises s'interrogent sur la possibilité d'isoler une partie de leur travail pour « l'informatiser », afin de pouvoir statuer automatiquement sur la conformité réglementaire d'un projet d'environnement et de poursuivre ensuite cette qualification « à l'usage ». Elles cherchent également à savoir s'il est possible d'adapter automatiquement la démarche, parallèlement à l'évolution des normes et des contraintes réglementaires.

Ces travaux de doctorat s'inscrivent dans la thématique de l'Ingénierie des Exigences et amènent à étudier la faisabilité conceptuelle et technique d'une approche unifiée — générique — qui débiterait par l'appréhension des contraintes réglementaires liées à un type d'environnement, se poursuivrait par une aide à la mise en conformité d'un projet précis et déboucherait sur une analyse de conformité continue à l'usage. De par la nature des activités de l'entreprise avec laquelle ces travaux ont été conduits, les environnements normés médicaux ont constitué une cible privilégiée d'étude.

En somme, cela nous conduit à nous interroger sur la faisabilité scientifique d'une telle démarche, et ce à différents niveaux. D'un point de vue général, peut-on définir une approche générique d'Ingénierie des Exigences, qui suivrait un projet de sa conception jusqu'à son usage pour en analyser la conformité réglementaire ? Comment raisonner à la fois en terme générique (un type d'environnement normé) et en terme précis (un projet particulier avec ses contraintes spécifiques) ? Enfin, dernière interrogation, et non des moindres, peut-on concevoir une approche « unifiée », cohérente d'un bout à l'autre et qui ne perdrait qu'un minimum d'information entre ses différentes phases ?

Dernier point méritant d'être mentionné, ces objectifs doivent être mis en perspective avec le contexte d'exécution de ce doctorat. En effet, ces travaux ont été financés par une convention CIFRE dans une TPE, et les propositions et recherches effectuées devaient être évaluées sous le double prisme de leur intérêt « académique » (est-ce une réelle contribution au domaine scientifique abordé ?) et « industriel » (est-ce une solution crédible aux problèmes « réels » rencontrés ?).

1.4 Contenu du mémoire

Après cette introduction, ce document est divisé en cinq grandes parties, distinguant : la problématique, les contributions, les mises en œuvre, les conclusions, limites et perspectives, et enfin les annexes et tables.

La première partie présente les concepts généraux et introduit la problématique de cette thèse.

Le Chapitre 2 détaille le contexte industriel ayant motivé ces travaux doctoraux, nous aboutissons à un schéma général présentant le flot de conception/réalisation visé.

Le Chapitre 3 comporte deux parties ; dans la première, nous commençons par présenter l’approche industrielle actuelle répondant à ce besoin, tandis que dans la seconde, nous abordons le regard universitaire (thématiques et travaux pertinents) sur ces pratiques. L’objectif n’est pas de dresser un panorama complet d’un domaine de recherche, mais plutôt de nous concentrer sur les travaux de recherches connexes et pertinents.

Enfin, le Chapitre 4 réalise une synthèse de la problématique visée par ces travaux.

La deuxième partie présente les contributions de cette thèse sous la forme de trois chapitres complémentaires. Nous débutons par des présentations génériques et conceptuelles pour aborder ensuite la concrétisation des principes définis.

Ainsi, le Chapitre 5 présente notre approche globale pour l’aide à la qualification d’un projet d’environnement normé. Les éléments présentés sont totalement génériques et peuvent s’appliquer à n’importe quel type d’environnement. Nous aboutissons à une démarche en trois étapes : analyse des arrangements normatifs, aide à la qualification d’un projet et aide à la qualification durant le fonctionnement.

Les Chapitres 6 et 7 présentent nos choix conceptuels pour, respectivement, l’analyse de l’arrangement normatif puis l’aide à la qualification d’un projet. Dans ces chapitres, nous examinons les disciplines informatiques choisies pour réaliser l’approche définie. Nous abordons ainsi pour l’analyse de l’arrangement normatif la théorie des graphes [HBF12] et, dans le cas de la qualification d’un projet, la « modélisation orientée aspects » [SSK⁺06]. Nous aboutissons alors à des mises en œuvre possibles, prêtes à être réifiées.

La troisième partie présente les mises en œuvre de ces contributions et les illustre sur un exemple concret d’environnement normé médical.

Nous débutons dans les Chapitres 8 et 9 par une réification des propositions des deux précédents chapitres. Pour la théorie des graphes, cela nous amènera à présenter notre implémentation de l’analyse d’arrangement normatif à l’aide d’une base de données « orientée graphe » [RWE15] (dans le cas présent NEO4J [Neo15b, Gru15, VWA⁺15]). Pour la « modélisation orientée aspects », nous présenterons nos travaux basés sur l’intergiciel KerMeta [Tri15, JBF11, MFJ05].

Le Chapitre 10 présente une illustration de notre approche sur un cas concret d’environnement normé médical, une Unité de Reconstitution Centralisée des Cytotoxiques (URCC).

Le Chapitre 11 abordera les transferts technologiques réalisés durant cette thèse. Dans un premier temps, nous détaillerons notre proposition technique pour la qualification de fonctionnement d’un environnement normé. Ensuite,

nous présenterons une mise en œuvre alternative de notre approche, éludant l'étape de qualification d'un projet pour laquelle il n'existe pas d'outillage adapté au contexte industriel de ces travaux.

La quatrième partie dresse un bilan de ces travaux doctoraux dans les Chapitres 12, 13 et 14. Ainsi, après une conclusion générale, nous discutons des limites et faiblesses identifiées. Cela nous permet d'esquisser les perspectives et futures recherches envisagées par la suite.

Enfin, la cinquième partie contient l'annexe A présentant une modélisation réticulaire alternative basée sur un Système de Gestion de Base de Données Relationnelles (SGBDR) suivie de la bibliographie, des différentes tables (figures, algorithmes et codes sources) et de la liste des nombreux acronymes employés dans ce mémoire.

Première partie

Contexte et état de l'art

Contexte et besoin industriel

Cette thèse de doctorat porte sur des travaux où les thématiques industrielles et universitaires sont intimement liées. L'objectif industriel a pavé la voie aux recherches engagées, mais ce sont ces mêmes recherches qui ont permis d'atteindre et de poursuivre des contributions innovantes.

2.1 Introduction

Les centrales nucléaires, les blocs chirurgicaux ou les salles blanches industrielles sont des lieux critiques abritant des processus complexes et comportant de nombreux risques, que ce soit pour l'environnement, pour les utilisateurs ou les processus. Cela conduit inévitablement à un traitement spécifique par le législateur (qu'il soit national ou international) afin d'encadrer au maximum leurs conception, réalisation et utilisation. On emploie alors le terme « d'environnement normé ». Néanmoins, pour diverses raisons précédemment évoquées, la réalisation de ces environnements « spécifiques » n'a malheureusement rien d'exceptionnel et abouti à des constructions très en deçà des attentes des utilisateurs finaux.

L'Hôpital Numérique présenté dans la Section 1.2.2 est un des nombreux exemples d'environnements normés en cours d'évolution (voir, même, de révolution) et pour lequel l'informatique peut permettre d'améliorer très nettement la conception et l'usage. L'un de ces axes d'amélioration, est celui d'une approche de conception/réalisation « transversale » et « globale ».

Proposer une telle réponse, c'est être capable de réaliser un environnement « clé en main »¹ : correspondant au besoin final, supportant des conditions d'exploitations très agressives tout en respectant les réglementations applicables.

Mais plus que cela, l'objectif de la transversalité et des réalisations « clé en main » est d'obtenir plus de cette mutualisation que la seule somme de ces

1. NdR : par « clé en main », il faut comprendre « environnement complet vu comme un seul objet et potentiellement qualifiable comme tel ».

parties indépendantes. L'optimisation des coûts, la maîtrise du processus de réalisation font partie de ces gains espérés.

Il y a une multitude d'aspects à prendre en compte :

Scientifique : lutte contre les infections nosocomiales (le cas échéant), métrologie, etc.

Technique : réalisation d'environnement répondant aux contraintes de fonctionnement.

Industriel : maîtrise des coûts et des process.

On comprend bien (presque « intuitivement ») que sans une informatisation massive du processus et du produit, de tels objectifs sont inaccessibles.

Cette informatisation a déjà eu lieu, et nul n'ignore que les environnements normés actuels sont réalisés par ordinateur : étude, plans, suivi, etc. Ce qui est recherché, c'est d'étendre cette informatisation à l'encadrement normatif et à sa vérification :

- qualification normative assistée par ordinateur ;
- déploiement d'un ensemble précis d'équipements informatiques lors de la réalisation ;
- traçabilité et audit du fonctionnement à l'aide de systèmes informatiques.

La Figure 2.1 représente ce besoin d'informatisation en isolant les deux parties concernées du cycle de vie de l'environnement (sa conception et son utilisation) et en indiquant, pour chacune de ces parties, l'objectif recherché.

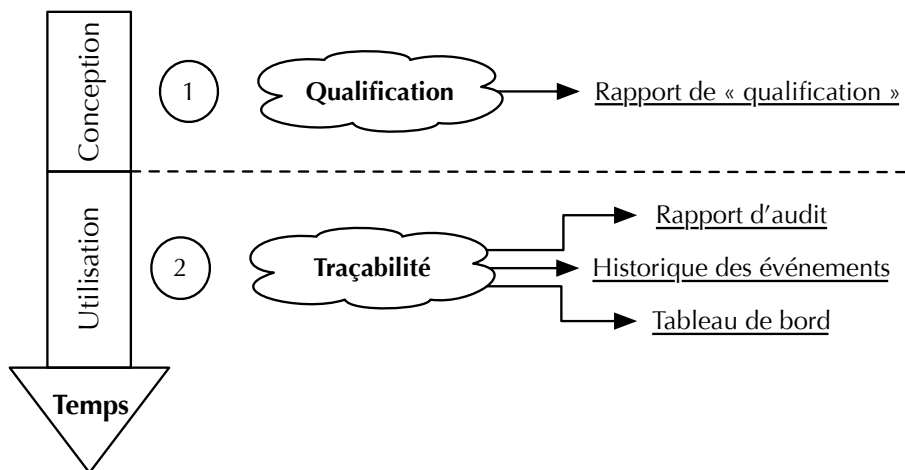


FIGURE 2.1 – Objectif industriel

Les besoins exprimés dans ce schéma peuvent se formuler relativement simplement.

Lors de la conception, l'objectif est d'être capable de juger « sur papier » si les éléments requis pour assurer la conformité réglementaire de l'environnement sont réunis tant pour sa mise en service que pour son exploitation. Cet

accompagnement devrait se traduire par la génération d'un « rapport de qualification » permettant d'apporter un soutien, une assistance aux concepteurs de l'environnement ciblé.

Une fois la mise en service effectuée, lors de son utilisation, il y a trois objectifs interdépendants :

- le premier est d'être capable d'analyser en temps réel l'environnement (sous la forme d'un « tableau de bord ») ;
- le suivant est d'enregistrer tous les événements « pertinents » survenant durant l'exploitation ;
- et le dernier est d'utiliser ces enregistrements pour la réalisation « d'audit ».

L'objectif n'est pas de transformer l'environnement normé en cockpit d'avion afin de pouvoir en réaliser la qualification normative, c'est impossible. L'esthétique extérieure attendue ainsi que le procédé se déroulant dans la zone imposent de nombreuses contraintes fonctionnelles, incompatibles avec le déploiement d'équipements lourds et il faudra savoir jouer intelligemment de l'intégration en cloison, de l'encombrement, etc. pour remplir les objectifs.

Ces aspects, bien qu'indépendants des travaux de recherche d'un doctorat en informatique, sont primordiaux dans le cadre d'un transfert industriel de ces travaux.

Nous cherchons donc à savoir si une approche globale, permettant de suivre un environnement de sa conception jusqu'à son utilisation, est envisageable et, si elle l'est, de quelle manière. Une fois ces questions traitées, tout un champ d'exploration s'offre à nous concernant la capitalisation des informations entre les différentes phases du cycle de vie de l'environnement ciblé ainsi qu'entre les analyses de différents environnements normés.

2.2 Prise de recul

Si cette brève introduction a situé le contexte industriel de ces travaux, il nous faut savoir nous en éloigner pour en améliorer le traitement. En effet, le cadre de pensée offert par le besoin industriel (Figure 2.1) n'est pas suffisant pour appréhender correctement l'ensemble du problème et y répondre de manière pertinente.

Pour saisir ces limitations, le plus simple est de prendre le besoin exprimé par sa dernière étape : l'analyse *in situ* de l'environnement. Lors de celle-ci — apportant des éléments de réponse concernant le maintien de la conformité réglementaire de la cible durant son usage — plusieurs questions se posent : « Quels sont les éléments à vérifier dans l'environnement ? », « Comment effectuer ces vérifications ? », et surtout « Quelles sont exactement les vérifications à effectuer ? »

Une réponse évidente à la première question pourrait être « tout ». C'est-à-dire : procéder à une traçabilité systématique de l'ensemble des éléments techniques présent dans l'environnement suivie d'une « exploration des données » (*data mining* en anglais) pour qualifier la conformité réglementaire de la cible.

Cette solution ne nous semble pas crédible pour plusieurs raisons :

- La première est économique. Nous n’estimons pas réaliste de vouloir « tout » tracer dans un environnement, sans se préoccuper ni de la complexité d’acquisition des données ni de l’intérêt réel des données obtenues. Cette acquisition pouvant être très coûteuse — en matériel, logiciel, frais de maintenance et supervision, compétences techniques, etc. —, quadrupler un budget pour des données « prêtes à jeter » nous semble pour le moins excessif.
- La deuxième est technique. Nous pensons impossible — notamment dans les environnements fortement contraints « temporellement » (telles les centrales nucléaires où chaque processus doit s’exécuter dans un temps bien défini) —, l’idée de pouvoir tout enregistrer, analyser et tracer « indistinctement » sans mettre en péril le système global que ce soit par l’explosion des traitements requis ou la complexification du système pour supporter une telle inspection. On pourrait imaginer faire cela « hors ligne », sur des serveurs indépendants de la production, mais cela ne répondrait pas totalement au besoin industriel, celui-ci visant un retour immédiat.
- La dernière est pragmatique. Cette position revient à repousser en aval (c’est-à-dire, à l’usage) une problématique qui aurait pu (dû) être traitée en amont, lors de la conception. La prise de recul est alors maximale et il devient possible de raisonner à très haut niveau sans s’encombrer de détails superflus, ceux-ci pouvant être intégrés plus tard.

La phase d’analyse du projet (déterminant la conformité réglementaire « sur plan ») nous semble la position idéale pour relever ces informations et recueillir la liste précise et exhaustive des équipements à analyser ainsi que des moyens disponibles.

Cela nous conforte dans la vision initiale, développée dans l’introduction de cette thèse. L’impératif à respecter est la cohérence de la méthodologie pour qu’elle agisse de manière « unifiée » sans perte d’information entre la conception et l’usage de l’environnement. En résumé, il ne faut pas analyser l’environnement pour aider sa qualification puis analyser son usage pour aider au maintien de cette qualification, mais analyser l’environnement et utiliser le produit de cette analyse comme guide pour aider au maintien de la conformité.

Malheureusement, le recueil des données du projet n’est pas suffisant. La dernière question « quelles sont exactement les vérifications à effectuer ? » demeure...

Répondre à cette question requiert plus qu’une simple description du projet, aussi exhaustive et précise soit-elle. Apporter une réponse véritablement pertinente demande une parfaite maîtrise du type d’environnement, de ses contraintes normatives et des implications de celle-ci ; dans les faits, cela requiert même une maîtrise de l’activité à réaliser dans l’environnement afin d’effectuer les choix les plus judicieux pour l’utilisateur final.

Pour travailler sans une maîtrise préalable parfaite, l’une des approches classiques passe par l’Intelligence Artificielle [Nil14], plus particulièrement par l’apprentissage automatique (*machine learning* en anglais) [MRT12] et notamment, dans notre cas, du sous-domaine de l’« apprentissage supervisé »² (*supervised learning* en anglais).

2. L’apprentissage supervisé est une technique d’apprentissage automatique où l’on cherche à

Dans ce champ d'études, notre sujet peut s'apparenter à des problèmes de discrimination et de régression, pour lesquels il existe de très nombreuses solutions (machines à vecteurs de supports, etc.). Dans le cadre de cette thèse, cela ne nous a pas semblé pertinent pour deux raisons ; la première est la nature des environnements ciblés, ces solutions étant, par principe, plus adaptées à la recherche de corrélations entre des facteurs multiples qu'à la certification de causalité ; la seconde est la nécessité de disposer d'une base de données d'exemples sur lesquels entraîner notre moteur, inexistante actuellement.

Par conséquent, nous estimons nécessaire de faire précéder la phase d'analyse d'un projet précis par une analyse du type de projet à réaliser. Cette étape préliminaire doit, pour un type d'environnement précis, permettre de connaître les vérifications à opérer selon la phase d'avancement du projet.

Ainsi, confrontés à un projet concret, nous disposerons de toutes les informations nécessaires pour en juger de la qualité intrinsèque, que ce soit par l'analyse normative préalable qui nous a fourni le cadre de pensée nécessaire, que par les données du projet qui nous fournissent les éléments concrets à vérifier.

En somme, le besoin industriel présenté en Figure 2.1 ne peut se concevoir qu'avec le recul présenté en Figure 2.2 et finalement, c'est une approche « unifiée » pour répondre à ce besoin-ci qui est réellement pertinente et innovante.

Cette problématique est applicable à l'ensemble des environnements normés, peu importe le domaine métier réellement concerné.

Afin de faciliter la compréhension de notre contribution et en raison du financement de ces travaux (cf. les notes préliminaires et l'introduction de cette thèse), nous avons fait le choix d'une expérimentation dans le domaine médical.

Nous allons maintenant procéder à une présentation du contexte particulier des environnements confinés médicaux.

2.3 Les environnements confinés médicaux

La présente section vise à fournir au lecteur une meilleure compréhension des environnements confinés médicaux ainsi que des risques liés à leurs mauvaises conceptions, réalisations ou utilisations.

2.3.1 Définition

La littérature propose plusieurs définitions du concept « d'environnements confinés ». Selon les sources, la terminologie et la sémantique peuvent différer et il convient de bien différencier les aspects liés au « maintien du confinement », de ceux liés au « travail en environnement confiné ».

Bien que la seconde approche semble découler de la première, celle-ci couvre également le travail dans des environnements « naturellement » confinés (puits, fosses, égouts, etc.) pour lesquels le confinement est une contrainte intrinsèque. Les implications sont très différentes ; dans une fosse, on va chercher à garantir la survie du travailleur en le protégeant de l'anoxie, tandis que dans une salle

produire automatiquement des règles à partir d'une base de données d'apprentissage contenant des « exemples » (en général des cas déjà traités et validés).

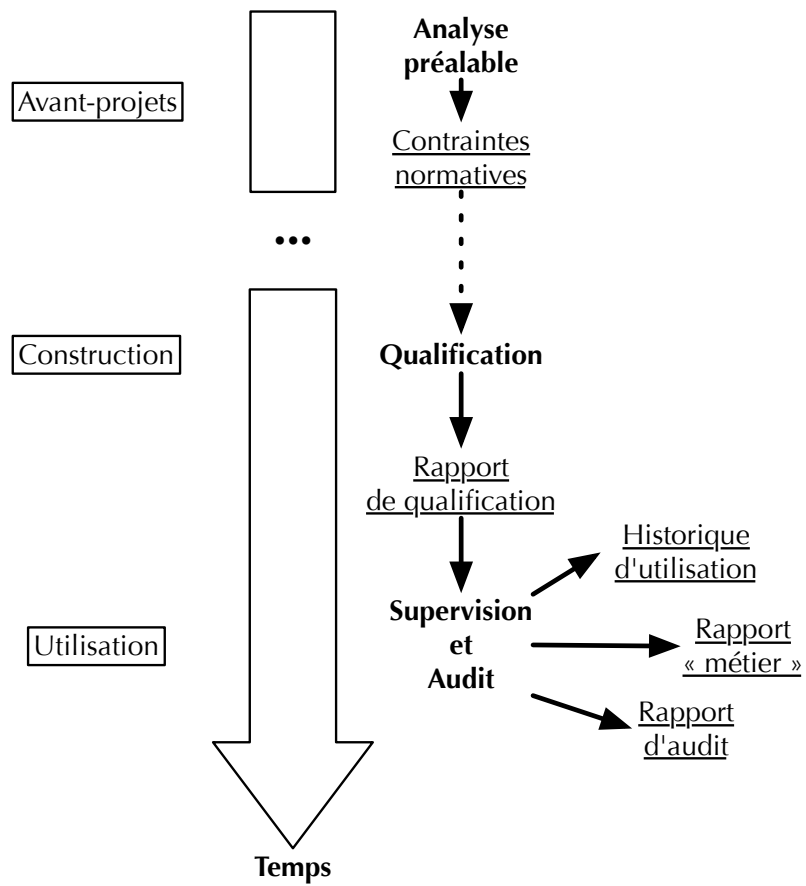


FIGURE 2.2 – Objectif industriel « réel »

d'opération, il faut garantir le confinement tout en sécurisant le patient et l'équipe médicale.

Dans ces travaux, nous nous intéresserons au maintien d'un confinement « artificiel » en milieu médical. Voici quelques définitions :

- L'association TEKHIN [Ass15], centre de compétences des techniques et méthodes en milieu confiné, définit le confinement comme « *une zone de travail contrôlée, au volume limité par un moyen physique, dans laquelle on empêche tout transfert non maîtrisé entre l'environnement (dont les intervenants font partie), et l'intérieur de la zone confinée (qui abrite un produit ou un process)* ».
- L'Association Française de NORmalisation (AFNOR) préfère la locution « zone à environnement maîtrisé en établissements de santé » et, s'appuyant pour cela sur les définitions de l'International Organization for Standardization (ISO) [Int99a, Int04], définit :
 - Un **environnement maîtrisé** comme « *une zone définie où l'on maîtrise les sources de contamination à l'aide de moyens spécifiés* ».
 - Une **zone propre** comme « *un espace dédié dans lequel la concentration des particules en suspension dans l'air est maîtrisée et qui est construite et utilisée de façon à minimiser l'introduction, la production et la rétention des particules à l'intérieur de la pièce, et dans lequel d'autres paramètres pertinents, tels que la température, l'humidité et la pression sont maîtrisées comme il convient* ».

Comme indiqué dans ces définitions, pour garantir le confinement, des méthodologies particulières sont nécessaires.

2.3.2 Maintien du confinement

Bien que le terme « zone propre » s'applique à un très grand nombre de domaines médicaux et que la réalisation d'un tel espace puisse par conséquent prendre de nombreuses formes ; le principe général reste sensiblement le même. Ce seront surtout les aspects techniques, les « paramètres de conception », qui différencieront, créant ainsi une combinatoire de solutions possibles, envisageables, adaptées, souhaitables, autorisées, etc. extrêmement complexe et nécessitant une réelle expertise pluri-disciplinaire (aéroulque, thermique, micro-biologique, etc.).

Ce principe général de confinement est celui de « l'asepsie progressive » [Hoe94] (ou « marche en avant »), c'est-à-dire la mise en place d'un « *ensemble de mesures propres à empêcher tout apport exogène de micro-organismes ou de virus, conçu par une série de barrières successives pour limiter le risque de contamination de la plaie opératoire* ». Popularisé par les travaux d'Hœt dans les années 90, cette approche est la pierre angulaire de toute la chaîne de protection contre la biocontamination.

En une vingtaine d'années, de très nombreuses techniques ont vu le jour pour améliorer ces barrières et limiter au maximum la biocontamination. Parmi ces techniques traditionnelles, les plus emblématiques sont :

- l'usage de gradients de pression avec imposition d'un sens particulier de circulation de l'air ;
- la séparation la plus claire et ferme possible des zones propres et sales ;

- la diminution drastique du nombre de surfaces horizontales³ ;
- d'importantes formations sur le lavage des mains avec des approches multi-paradigme (formation, informatisation. . .) ;
- l'amélioration des techniques de sas. Pour rappel, un sas est une zone de nettoyage et de désinfection, constituant un verrou aéraulique « physique » et servant très souvent de lieu de changement de tenue. Les sas existent sous forme de locaux ou zones de circulation (dans cette dernière situation, cela requiert la mise en place d'un mécanisme d'interlockage⁴).

2.3.3 Arrangement normatif

Après avoir présenté l'une des conséquences possibles d'une rupture de confinement, nous pouvons revenir au problème de la conception et réalisation des environnements normés.

Comme nous l'avons présenté en introduction, la différence entre les environnements « normés » et les environnements « traditionnels » repose sur le volume de contraintes à respecter et le degré de contrôle à maintenir lors de leur exploitation.

Or, répondre à de telles obligations et fournir un tel contrôle n'est pas anodin, et nécessite une maîtrise parfaite de l'environnement ciblé et de son objectif fonctionnel. Cela fait intervenir un grand nombre de corps de métier ou de disciplines — architecture, métrologie, contrôle d'accès, maîtrise d'œuvre, etc. — avec des phases de qualification souvent très complexes. Pour répondre à ces impératifs, les autorités ont procédé à la normalisation de différentes méthodes et usages pour chaque type d'environnement. Ces normes ne sont pas toutes applicables et certaines sont seulement des recommandations, des conseils ou encore des guides de bonnes pratiques. Les recommandations de l'Organisation Mondiale de la Santé (OMS) en sont l'exemple [Int10].

Ainsi, en considérant l'ensemble des caractéristiques d'un environnement — comme son utilisation, sa fabrication, sa fréquentation ou encore sa situation géographique — on aboutit à la définition d'un « arrangement normatif », c'est-à-dire l'ensemble des normes à respecter pour un environnement particulier.

Les arrangements normatifs sont très volumineux et peu de professionnels en appréhendent la totalité (légalement, techniquement et scientifiquement), impactant ainsi directement le coût et la qualité de l'installation finale. Dans ces conditions, il n'est pas rare de voir des bâtiments partiellement reconstruits immédiatement après leur livraison afin de répondre à des contraintes légales ou d'utilisation qui n'avaient pas été prises en compte initialement.

Nous allons maintenant aborder les conséquences potentielles d'écarts aux contraintes normatives.

3. Afin de limiter les possibilités de biocontamination, on préférera employer des plans légèrement inclinés, qui éviteront la stagnation des liquides et permettront ainsi de faciliter le nettoyage et la décontamination.

4. Par « mécanisme d'interlockage », nous entendons tout mécanisme permettant de bloquer un ensemble de portes lorsqu'un autre ensemble de portes est ouvert, sans oublier des configurations bien plus complexes faisant intervenir des délais de blocage particuliers, des temporisations, etc.

2.3.4 Infections nosocomiales

Les éléments suivants sont largement inspirés par la « définition des infections nosocomiales » fournie par le Centre de Coordination de Lutte contre les Infections Nosocomiales (CCLIN) Sud-Est [CCL15] et la « définition des infections associées aux soins » du Ministère de la santé, de la jeunesse et des sports [Com07].

Contexte

Jusqu'à la fin des années 80, la France était particulièrement en retard dans la lutte contre les infections nosocomiales.

Mais, la création en 1988 des Comités de Lutte contre les Infections Nosocomiales (CLIN) dans chaque établissement puis l'institution, en 1992, du Comité Technique national des Infections Nosocomiales (CTIN) et de cinq Centres de Coordination interrégionaux de la Lutte contre les Infections Nosocomiales (CCLIN) et la création d'équipes d'hygiène dans les hôpitaux, ont contribué à faire largement évoluer la situation.

La chaîne de commandement est relativement simple :

À l'échelon national : la cellule « Infections Nosocomiales » du ministère, commune à la Direction Générale de la Santé (DGS) et à la Direction des Hôpitaux (DH), est chargée de coordonner l'ensemble du dispositif.

De son côté, le CTIN est une instance de proposition, de coordination et d'évaluation, constituée d'experts hospitaliers, proposant des objectifs prioritaires et des méthodologies standardisées de surveillance et de prévention au ministre.

La surveillance en incidence des IN (sur une période donnée) est coordonnée dans le cadre du Réseau national d'Alerte, d'Investigation et de Surveillance des Infections Nosocomiales (RAISIN), issu du partenariat entre les 5 CCLIN et l'Institut de Veille Sanitaire (INVS) depuis 2001.

À l'échelon inter-régional : cinq CCLIN ont été mis en place. Ils appuient techniquement les établissements hospitaliers et opèrent au niveau des inter-régions (Est, Ouest, Paris-Nord, Sud-Est, Sud-Ouest).

Ils sont chargés de mettre en place la politique définie au niveau national et d'animer la coopération inter-hospitalière (réseau de surveillance, formation, documentation, études...)

À l'échelon des établissements hospitaliers : chacun de ceux-ci dispose d'un CLIN chargé d'organiser et coordonner la surveillance, la prévention et la formation continue en matière de lutte contre les infections nosocomiales.

Les CLIN sont composés de médecins, de pharmaciens, d'infirmières et de directeurs d'établissement, souvent assisté de personnels d'hygiène hospitalière. Il peut s'agir d'une infirmière hygiéniste ou d'une équipe opérationnelle en hygiène hospitalière pour les établissements plus importants.

Définition

Historiquement, les pathologies infectieuses étaient classées en deux catégories : infections communautaires et infections nosocomiales (également nommées

« infections acquises en établissement de santé »). Toutefois, la diversification des parcours de soins, des intervenants dans la dispensation des soins, des structures et des systèmes de soins, ainsi que la survenue parfois tardive de l'infection rendaient caduque cette classification.

Ainsi, certaines infections hospitalières précoces d'acquisition communautaires étaient classées comme nosocomiales et certaines colonisations sans signe clinique l'étaient également (incitant alors la prescription d'antibiotique souvent inutile). À l'inverse, certaines infections postopératoires tardives, bien que probablement associées aux soins, ne sont pas classées comme nosocomiales.

Devant ces constats, le rapport [Com07] « Définition des infections associées aux soins » du Ministère de la santé, de la jeunesse et des sports publié en 2007 visait à définir le champ de l'ensemble des Infections Associées aux Soins (IAS) et à présenter une actualisation des définitions de l'Infection Nosocomiale (IN).

Le concept élargi des IAS s'intéresse aux épisodes infectieux indépendamment du lieu de soins.

Définition 1 *Une infection est dite « associée aux soins » si elle survient au cours ou au décours d'une prise en charge (diagnostique, thérapeutique, palliative, préventive ou éducative) d'un patient, et si elle n'était ni présente, ni en incubation au début de la prise en charge.*

Lorsque l'état infectieux au début de la prise en charge n'est pas connu précisément, un délai d'au moins 48 heures ou un délai supérieur à la période d'incubation est couramment accepté pour définir une IAS. Toutefois, il est recommandé d'apprécier dans chaque cas la plausibilité de l'association entre la prise en charge et l'infection.

Pour les infections du site opératoire, on considère habituellement comme associées aux soins les infections survenant dans les 30 jours suivant l'intervention ou, s'il y a mise en place d'un implant, d'une prothèse ou d'un matériel prothétique dans l'année qui suit l'intervention. Toutefois, et quel que soit le délai de survenue, il est recommandé d'apprécier dans chaque cas la plausibilité de l'association entre l'intervention et l'infection, notamment en prenant en compte le type de germe en cause.

Le terme d'IN est seulement réservé aux IAS contractées dans un établissement de santé. Celles-ci peuvent toucher les patients, la famille, les visiteurs, mais aussi le personnel soignant et les autres professionnels de santé de l'établissement.

Conséquences

Les maladies nosocomiales sont l'une des problématiques majeures du secteur de la santé aux conséquences lourdes, pour le patient comme pour la société.

D'après les rapports [Min09, Ins07, Vas06], et bien que la mesure précise du nombre de décès directement dus à une infection nosocomiale reste un exercice délicat, on estime que 6,6 % des décès qui interviennent chaque année à l'hôpital ou à la suite d'une hospitalisation surviendraient en présence d'une infection nosocomiale. Au total, ces infections seraient donc la cause de 9.000 décès par an, dont 4.200 concernent des patients pour lesquels le pronostic vital n'était pas engagé à court terme à leur entrée à l'hôpital. Pour comparaison, les accidents de la route ont fait 3.427 morts pour l'année 2013 en France [Obs15]. Nous sommes

face à une problématique extrêmement critique qui, si l'on se projète sur des années voire des décennies, pourrait s'avérer capitale dans le futur. Le principal poste d'IN concernant la pose de sondes urinaires sur les personnes âgées, le vieillissement de la population constitue une circonstance aggravante à ne pas négliger.

Pour la société, le surcout financier est important. Les différentes études disponibles font état d'une échelle de couts très large, allant de 340 euros à 40.000 euros. En appliquant une fourchette de surcout moyen de 3.500 à 8.000 euros par infection aux 750.000 infections nosocomiales annuelles, on atteint un montant de dépenses de 2,4 à 6 milliards d'euros. Ainsi, une diminution de 10 % du nombre d'infections conduirait à une économie de 240 à 600 millions d'euros, soit jusqu'à six fois plus que l'effort de prévention consenti par les établissements hospitaliers. Ce rapide calcul montre combien le cout de la non-qualité est supérieur à celui de la prévention.

2.4 Conclusion

Dans ce chapitre, nous avons présenté le besoin industriel, et l'avons ensuite raffiné pour en produire une synthèse exhaustive, à laquelle nous tâcherons d'apporter une réponse dans cette thèse de doctorat (Figure 2.2). Une fois ce contexte « général » posé, nous avons évoqué l'importance du besoin industriel dans le domaine médical.

Nous allons maintenant présenter l'état de l'art actuel concernant ce besoin industriel. Nous débuterons par celui de la réalisation des environnements normés pour ensuite présenter les travaux universitaires s'inscrivant dans cette dynamique et sur cette thématique.

La conception-réalisation d'environnements normés

Après avoir présenté le contexte industriel de ces travaux, attachons-nous à observer les réponses apportées et les pistes de recherches universitaires connexes.

Ces travaux — de par la thématique industrielle et la forme de notre réponse — sont à l'intersection de plusieurs domaines de recherches distincts. La conception/réalisation d'environnements normés est une activité industrielle en pleine expansion et pour laquelle les réponses technologiques se multiplient, empruntant de nombreuses directions (Figure 3.1).

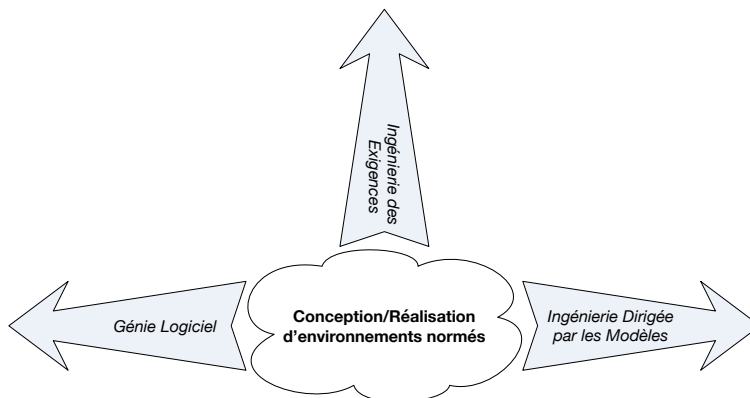


FIGURE 3.1 – Positionnement thématique de ces travaux

Cependant, nous ne prétendons pas apporter d'originalité dans l'ensemble de ces domaines. Notre principale contribution s'inscrit en Ingénierie des Exigences et nous sommes seulement des « utilisateurs originaux » des autres technologies et concepts.

Ainsi, il n’y a aucun intérêt (et nous n’y avons pas vocation) à lister de manière exhaustive tous les travaux industriels et universitaires ayant trait à l’une ou l’autre de ces thématiques. Au contraire, nous allons essayer de dresser une synthèse cohérente des us et coutumes de l’industrie, puis de décrire les travaux universitaires pertinents s’y rattachant, en lien avec nos propres travaux et dont l’apport enrichit la pratique courante.

3.1 Conception et réalisation : positionnement industriel

Tout aussi surprenant que cela puisse paraître, la conception et la réalisation des environnements normés médicaux diffèrent relativement peu de celle des environnements « classiques ». La conception utilise des outils « traditionnels » avec un découpage en « lots » réalisés ensuite par des corps de métiers spécialisés pour le sujet concerné (mais sans spécialité aucune pour le type d’environnement normé concerné).

Bien qu’à première vue séduisante, cette approche présente de nombreuses limitations dans le cas des environnements normés médicaux. C’est ce que nous allons maintenant aborder dans les sections suivantes.

3.1.1 Conception

Si, historiquement, la conception de bâtiments était réalisée manuellement par les différents opérateurs, l’apparition d’outils informatiques complexes et spécifiques a permis de grandement rehausser la qualité des conceptions.

L’avènement des logiciels de Conception Assistée par Ordinateur (CAO)¹, de Dessin Assisté par Ordinateur (DAO) ou encore d’Ingénierie Assistée par Ordinateur (IAO), dédiés aux bâtiments (par exemple, AutoCAD [Aut15a], Revit [Aut15c] ou encore Allplan [Nem15a]) a permis de réaliser numériquement les opérations concernées afin de disposer, outre les avantages classiques de l’informatique (cout, archivage, collaboration, mise en forme, etc.), d’opérations plus spécifiques réalisées directement par les outils (calculs de volumes, de débits, de pression, d’encombrement...).

D’un point de vue plus général, si les premières solutions étaient plutôt « atomiques » — des outils permettant d’améliorer la réalisation d’une opération particulière du processus de conception (dessin, représentation 3D, calculs de fluides, etc.) —, le projet est maintenant clairement d’utiliser l’outil informatique pour représenter « globalement » le bâtiment afin de pouvoir l’interpréter et l’analyser dans sa totalité.

Cela a permis l’émergence de la Maquette Numérique du Bâtiment (MNB) (*Building Information Model (BIM)* en anglais)² [Wik15a], présentée dans la section suivante.

1. Les anglophones distinguent deux catégories sur ce sujet : Computer-Aided Design (CAD) et Computer-Aided Architectural Design (CAAD).

2. Il existe un très grand nombre de traductions de l’acronyme original anglais « BIM » : Modélisation des Données du Bâtiment, Bâti et Informations Modélisés, Modèle d’Information unique du Bâtiment, ou encore Maquette Numérique du Bâtiment. Ce dernier terme nous paraît le plus représentatif de l’acronyme anglais, tout en étant juste en langue française.

La Maquette Numérique du Bâtiment

La MNB est une méthode de travail, un processus, utilisant une maquette numérique 3D intelligente comme élément central des échanges entre les différents intervenants dans la construction. Initiée par l'architecte, cette maquette est ensuite rendue accessible aux différents bureaux d'études dans le but d'être complétée, voire modifiée techniquement. Chaque participant, chaque acteur, enrichit alors le projet de son propre savoir-faire et de sa propre expérience.

Initialement, la maquette sera la plus souvent divisée sous la forme de plusieurs maquettes « métiers » (une maquette « structure », une maquette « fluides »...). Celles-ci sont ensuite fusionnées dans une « master maquette » pour détecter et résoudre les conflits éventuels (réseaux en interférence par exemple). On obtient ainsi une vision générale et complète du projet ciblé afin de pouvoir réaliser des estimations fines sur son chiffrage, son « réalisme », et même d'en faire des démonstrations et des présentations. Ce travail d'assemblage et d'analyse des conflits se nomme la « synthèse ».

Ainsi conçue, la maquette sert à produire les plans d'exécution qui seront diffusés au chantier. Les entreprises de construction utilisent également la maquette pour réaliser leurs métrés, planifications ou phasages. Durant les travaux, la maquette est maintenue à jour par les concepteurs et les constructeurs, de sorte qu'à la fin du chantier le modèle du bâtiment soit exactement conforme au bâtiment tel qu'il a été construit.

Cette maquette, livrée au Maître d'Ouvrage avec les clés de son bâtiment lui donne la possibilité de gérer informatiquement son bien immobilier, dans le but de réaliser des travaux ultérieurs, de gérer son patrimoine, d'intégrer des systèmes domotiques ou de réaliser diverses simulations (ensoleillement, flux, acoustique...).

Ce modèle « global » est donc la convergence d'un certain nombre d'intérêts :

- Les promoteurs immobiliers souhaitant faire la promotion et la commercialisation de leur projet au plus vite. La MNB offre de nombreuses opportunités, telles que la possibilité de faire « visiter » des locaux aux futurs acquéreurs par des techniques de « réalité virtuelle ».
- Les bailleurs de fonds souhaitant affiner drastiquement la qualité des estimations financières réalisées sur le cout d'un bâtiment préalablement à sa construction. Ils espèrent ainsi pouvoir faire intervenir l'ensemble des acteurs devant contribuer à la création du bâtiment pour comprendre et analyser leur contribution (et surtout le cout de celle-ci).
- Les maîtres d'œuvre, ingénieurs, architectes, etc. souhaitant disposer d'une vision globale de leur projet et pouvoir ainsi y mener un certain nombre d'analyses automatiques (cout, temps de construction, etc.). Les modèles ainsi produits pourraient permettre de voir la réalisation d'un bâtiment comme la réalisation d'un énorme « Lego » réutilisable entre chaque projet.

Il existe une vaste gamme de logiciels de CAO [Wik15c], s'inscrivant de manière plus ou moins large dans la MNB et fournissant une quantité variable de simulations. On pourra citer (parmi les plus connus) : AutoCAD [Aut15a] (avec son extension « Chauffage, Ventilation, Climatisation » : MEP [Aut15b]), Revit [Aut15c] (avec son extension CVC : MEP [Aut15d]), Allplan [Nem15a] (avec sa version « Engineering » [Nem15b]).

Néanmoins, il ne faudrait pas penser que la MNB présente seulement des avantages et dispose d'un consensus global parmi tous les acteurs impliqués.

Actuellement, les sceptiques y voient surtout une nouvelle intrusion de l'informatique dans le BTP ou le génie civil, et la volonté des grands donneurs d'ordres et grosses structures d'étouffer les petits acteurs en leur imposant des outils et des méthodes hors de leur portée.

En effet, les logiciels de CAO permettant de réaliser la MNB (ainsi que leurs mises à jour fortement recommandées pour rester « dans la course ») sont relativement onéreux, et requièrent une certaine expertise nécessitant formation et usage continu afin d'être réellement efficace (rendant les utilisateurs « spécialistes » de la solution concernée). Par conséquent, cela crée une relation de dépendance entre l'entreprise utilisatrice et l'éditeur du logiciel, jugée néfaste (voir parasitaire) par les détracteurs de la MNB.

De plus, imposer aux différents corps de métier de fournir la maquette de leur contribution pour synthèse finale, rend obligatoire l'usage d'une solution de MNB même pour des acteurs n'y voyant ou n'en ayant pas réellement d'utilité³. Par conséquent, ces corps de métiers (quelle que soit leur taille ou leur activité) doivent également investir dans le matériel, les logiciels et les compétences nécessaires pour répondre aux donneurs d'ordres, et ce, quelles que soient leur taille et leur capacité de financement. Pour de grosses structures, c'est anodin, mais en revanche très difficile pour des TPE ou des PME dont l'activité n'a traditionnellement aucun lien avec l'informatique.

La principale inquiétude de ces opposants est actuellement de voir l'usage de maquettes numériques rendu obligatoire par le législateur.

Tout aussi contrastée que soit l'introduction de la MNB dans l'univers du BTP (et plus généralement du génie civil), elle permet d'anticiper un certain nombre d'évolutions futures et d'opportunités pour la conception d'environnements normés : automatisation d'analyses « métiers », « normatives » ou « financières », retour visuel immédiat sur le projet considéré, simulation interactive, etc.

En revanche, il en va tout autrement de la phase de réalisation de l'environnement conçu, qui quant à elle, reste dans la plus pure tradition du bâtiment ; ce que nous allons maintenant discuter.

3.1.2 Réalisation : le découpage en lots

La découpe par « lot », est un exemple typique de réponse « verticale » à un problème d'ingénierie.

Dans ce type d'approche, on privilégie une analyse identifiant des « grandes fonctions » et des « grands blocs logiques », qui seront réalisés indépendamment par des « experts » du domaine concerné⁴. Si le cahier des charges a été suffisamment étayé, on espère ainsi couvrir correctement les objectifs. L'accent est mis sur l'isolation entre les différents acteurs, celle-ci garantissant théoriquement une « séparation des préoccupations », chacun apportant sa contribution en se concentrant sur son domaine « métier » spécifique.

3. On notera au passage que les solutions de CAO ne sont pas totalement interopérables.

4. Le terme est cependant assez imprécis, tous les corps de métiers ne possèdent pas la notion « d'expert » et tous les projets n'ont pas les moyens nécessaires, donc bien souvent on fera simplement appel à des professionnels « classiques ».

On espère obtenir ainsi le meilleur résultat dans chaque discipline, une philosophie que l'on retrouve notamment dans le « compagnonnage » en France.

La Figure 3.2 illustre cette réflexion... on escompte que la somme des lots correspondra à l'intégralité du problème à résoudre. Chaque partie possède son propre responsable, qui doit remplir le cahier des charges précisé au moment de l'analyse.

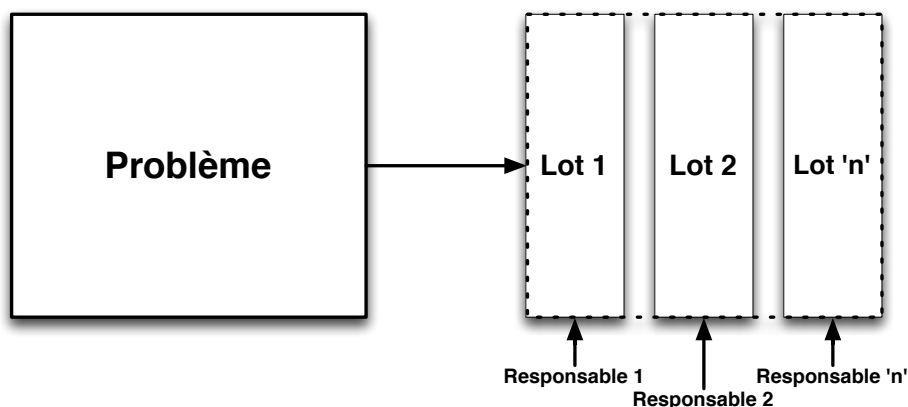


FIGURE 3.2 – Analyse « verticale » d'un problème

Dans le domaine du bâtiment, cela se traduit par la réalisation du « gros œuvre » puis à la découpe du « second œuvre » en un ensemble de lots décrivant les grandes « structures » du bâtiment : sols collés, cloisons sèches, menuiserie, contrôle d'accès, etc. Chaque lot est réalisé par le professionnel correspondant : « menuisier », « société de sécurité », « frigoriste », etc. La Figure 3.3 illustre (rapidement) ce découpage en fournissant quelques exemples de seconds œuvres fréquemment réalisés.

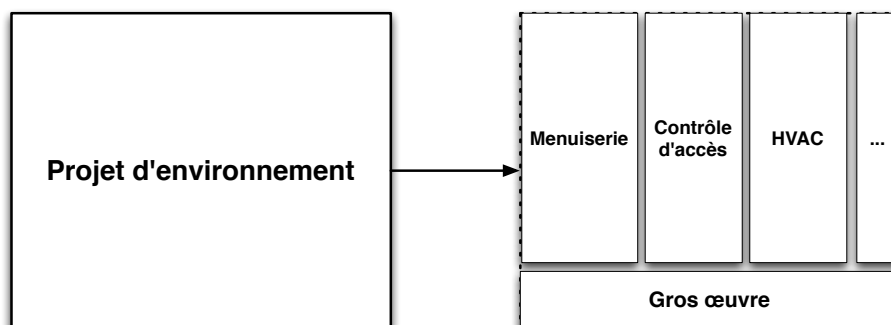


FIGURE 3.3 – Réalisation traditionnelle d'un environnement

Limites

Cette approche comporte malgré tout certaines limites, en voici les plus importantes. Tout d'abord, n'oublions pas qu'une division des responsabilités peut

conduire à une production de qualité inégale pouvant fragiliser l'intégralité de la solution. *Un défaut des pneumatiques d'une voiture complexifie le contrôle de sa trajectoire et augmente sa distance de freinage, impactant alors sa consommation et sa tenue de route, pouvant ainsi conduire à un accident.*

Ensuite, une mauvaise réalisation ou une mauvaise compréhension de l'analyse fonctionnelle du problème peut conduire à de très mauvais résultats, tant en termes de qualité générale de la solution que de réponse à l'ensemble des besoins exprimés par le client. On se retrouve face à des éléments de bonne facture, mais impossibles à utiliser conjointement. *Si le maçon et le menuisier ne s'entendent pas sur les mesures des fenêtres; après la construction du mur, la pose des huisseries peut s'avérer impossible.*

On n'oubliera pas l'impact financier d'un partage des responsabilités. Chaque professionnel traitant son problème séparément, aucune synergie n'est possible sur les charges tierces (non liées au domaine métier : administratif, etc.) et chacun réalise indépendamment sa marge et ses négociations. *Ce sont les problèmes bien connus de sous-traitance de production et de cout des intermédiaires pour les achats des entreprises.*

Enfin, il y a la question de la maîtrise d'œuvre. Le principal rempart contre les problèmes identifiés précédemment est le maître d'œuvre, responsable de la qualité du travail effectué. Mais encore faut-il que celui-ci soit un expert du domaine concerné et, dans le cas du confinement médical, ce n'est que très rarement possible. . . On pourra alors arguer qu'une expertise n'est pas nécessaire quand on dispose de suffisamment de métriques pertinentes pour vérifier la qualité de la réalisation. Encore faut-il que la définition de telles métriques soit possible. . . *Mais dans notre domaine, que veut dire « adopter une approche prophylactique » pour des menuisiers, des électriciens, etc. ? Comment faire respecter des obligations de résultats « globaux » à des professionnels traitant un problème vertical ? Comment définir la responsabilité des acteurs en cas de problème « global » (rupture de confinement, etc.) ?*

3.1.3 Conclusion

Dans cette section, nous avons présenté la situation industrielle concernant la conception/réalisation d'environnements normés.

La conception est actuellement en pleine mutation. La tendance étant la réalisation de « maquettes numériques » devant représenter le bâtiment et l'accompagner jusqu'à sa destruction (prévue dans la maquette). Cette maquette constitue une « évolution naturelle » dans la conception de bâtiment ; celle-ci étant passée du dessin « physique » au dessin « numérique » (2D), puis à la représentation 3D, et maintenant une forme de 4D voire 5D (avec une représentation 3D évoluant au cours du temps et contextualisée par un certain nombre de ressources documentaires, etc.). Si cette évolution ne se fait pas sans rencontrer un certain nombre d'oppositions (principalement de la part des petits acteurs et/ou ceux faiblement informatisés), elle pave la voie aux prochaines innovations. En effet, dans sa version actuelle, le MNB est un outil principalement « contemplatif » ou « gestionnaire », n'apportant pas de réelle plus-value « métier » ou « normative ». Il reste par conséquent encore beaucoup à explorer, et disposer de

telles représentations, ouvre des perspectives intéressantes dans le contexte de notre problématique.

La réalisation reste quant à elle totalement « classique », un découpage vertical permet une division du projet en lots qui seront affectés à des entreprises spécialisées dans l'opération considérée. Nous avons discuté des limitations d'une telle approche, spécifiquement dans le contexte des environnements, et de l'intérêt d'une approche « globale » (« horizontale ») pour améliorer, dans le domaine médical, la prophylaxie des lieux. Cela requiert une approche différente du projet, mais également la réalisation d'outils capables de conseiller les donneurs d'ordres et d'assister les utilisateurs durant le fonctionnement de l'environnement. Nous sommes de nouveau très proches des objectifs présentés dans le Chapitre 2.

En conclusion, les objectifs fixés dans cette thèse s'inscrivent dans le prolongement des mutations que connaît actuellement le domaine de la conception/-réalisation d'environnements normés. Dans la mesure où l'industrie commence à fournir les éléments et le « terreau » nécessaires à nos objectifs, nous allons maintenant en voir l'appropriation universitaire, au travers des travaux de recherches directement liés et connexes aux nôtres.

3.2 Ingénierie des Exigences

D'un point de vue « général », on pourrait résumer l'objectif industriel exprimé dans le Chapitre 2 à la vérification de conformité d'un produit à ses spécifications.

Dans le cas présent, on souhaite aider à la qualification d'un projet d'environnement normé (à la conception et l'exploitation) ; l'environnement normé est « le produit » et l'arrangement normatif à respecter constitue « les spécifications ».

C'est un problème « typique » d'Ingénierie des Exigences [Zav97, vL00, AW05, ZC05].

L'Ingénierie des Exigences est la branche de l'Ingénierie des Systèmes qui concerne l'étude des objectifs du monde réel dont les implications affectent des systèmes logiciels, des services que fournissent des systèmes logiciels, et des contraintes reposant sur des systèmes logiciels. Elle s'intéresse également à l'influence mutuelle entre ces facteurs et les spécifications du comportement des systèmes, ainsi qu'à leur évolution dans le temps à travers des familles de systèmes [Zav97].

Dans notre cas, nous appliquons cela non pas à un produit « logiciel », mais à un produit « physique » avec un appui logiciel pour réaliser les opérations de vérification et plus généralement d'aide à la décision.

Dans cette section, nous allons donc présenter au lecteur le domaine « général » de l'Ingénierie des Exigences pour nous intéresser ensuite aux travaux de recherche s'appliquant aux bâtiments et en lien avec notre problématique.

3.2.1 Motivation

Dans l'industrie du développement logiciel, le respect des exigences a toujours été une question difficile avec des résultats plus que discutables.

Dès les années 70 (soit depuis plus de 40 ans), Bell et Thayer [BT76] remarquaient que la mauvaise qualité de la formulation des besoins et exigences (inadéquation, incohérence, ambiguïtés, etc. [Mey85]) impactait très largement le produit final. Ils concluaient qu'on ne peut pas simplement « traiter » les exigences, mais qu'il faut mettre en place une véritable « ingénierie » autour de celles-ci. On peut considérer cette conclusion comme la première brique d'une nouvelle discipline. . .

Par la suite, dans les années 80 et 90, de nombreuses études se sont attachées à quantifier ce phénomène ; c'est-à-dire à mesurer l'impact concret de mauvaises spécifications (quel qu'en soit le motif) sur le produit logiciel (son cout, sa qualité, son adéquation avec la demande initiale, etc.).

Ainsi en 1981, Boehm [Boe81] estimait que les corrections tardives des exigences pouvaient entraîner un surcout d'un facteur 200 par rapport à une détection lors, justement, d'une phase d'ingénierie amont visant à traiter celles-ci.

Plus tard, dans les années 90, Lutz [Lut93] a étudié les erreurs logicielles des projets Voyager et Galileo de la NASA pour conclure que la principale cause des anomalies de sécurité était liée à des erreurs dans les exigences fonctionnelles (ou d'interface).

En 1995, une étude américaine [Cla95] portant sur plus de 8000 projets dans plus de 350 sociétés a révélé qu'un tiers des projets sont restés inachevés et que la moitié n'a que partiellement abouti (fonctionnement partiel, surcout ou retard important. . .). Dans plus de 50 % des cas, de mauvaises définitions des besoins et exigences ont été déclarées comme principales causes de la mauvaise conclusion du projet.

En 1997, une étude européenne [IR96] portant sur plus de 3800 entreprises de 17 pays concluait également que les principaux problèmes logiciels résident dans la spécification et la gestion des exigences.

Mettre en place une réelle « Ingénierie des Exigences » semble donc nécessaire pour traiter correctement ce problème.

3.2.2 Définition et objectifs

La première définition de l'Ingénierie des Exigences est issue d'une publication de Ross et Schoman [RS77], en 1977 :

Requirements definition is a careful assessment of the needs that a system is to fulfill. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed.

Les exigences couvrent donc un très large panel de contraintes : contexte d'utilisation, fonctionnalités attendues, contraintes liées à l'exécution de ces fonctionnalités dans le contexte d'utilisation prévu et les moyens disponibles. . . qu'il faut arriver à formaliser dans un langage pertinent ⁵. De plus, ces différentes contraintes peuvent évoluer lors des différentes mises à jour du produit ciblé ou lors de la mise en œuvre de variantes du produit initial [Zav97].

5. C'est à dire dans un langage capable de décrire précisément le comportement attendu du logiciel.

Cela aboutit à un domaine d'Ingénierie particulièrement complexe...

3.2.3 Contraintes

Examinons les contraintes de l'Ingénierie des Exigences à l'aune de ses objectifs et de sa définition primitive.

Tout d'abord, le terme « contexte d'utilisation » est relativement générique. Il peut s'agir d'un contexte « technique » (logiciel associé, équipement dans lequel s'intégrer, loi de la physique...) tout autant qu'« informel » (usage dans le cadre d'une entreprise, par des personnes précises, en respectant un certain contexte politique, etc.).

Le terme « fonctionnalités attendues » est tout aussi flou et peut dissimuler des objectifs « haut niveau » (p.ex., aller sur la lune) tout autant que très concrets et opérationnels (p.ex., rafraichir l'affichage de la vitesse instantanée toutes les 5 secondes). De plus, tous les objectifs n'étant pas « fonctionnels », il existe également de très nombreux critères non fonctionnels tels que le cout, les performances, la robustesse, l'interopérabilité, l'accessibilité, etc. Ces derniers sont d'ailleurs bien souvent conflictuels : on obtient rarement les solutions les plus performantes sans faire de compromis sur les couts financiers ou énergétiques ; la robustesse s'obtient souvent au détriment de la souplesse et l'accessibilité est un objectif non trivial dans bien des situations.

Enfin, l'ensemble de ces éléments peut être décrit de manière formelle ou informelle selon le contexte.

Ce qu'il faut comprendre, c'est que le « produit » n'est pas nécessairement logiciel et, par conséquent, toutes ces notions liées à « l'environnement » d'utilisation du produit sont à prendre dans leur sens le plus large, tous les paramètres doivent être inclus... rien ne doit être négligé.

Avec un tel éventail d'aspects à traiter, le nombre d'interlocuteurs, de « sachants ⁶ », est très élevé : clients, utilisateurs, décideurs, fournisseurs, experts « métier », ingénieurs, développeurs logiciels, agents de maintenance... Chacun d'eux dispose d'un vocabulaire, de connaissances, de compétences, d'expériences différentes et c'est l'union de tous ces points de vue (bien souvent conflictuels) qui permettra d'obtenir une liste précise et exhaustive des besoins et exigences réels à satisfaire.

Notre problématique illustre parfaitement cette complexité, avec un produit « physique » très particulier (un bâtiment) devant respecter :

- *des normes tout autant formelles qu'informelles ;*
- *produites par des instances différentes, aux responsabilités et objectifs parfois contradictoires ;*
- *portant autant sur des éléments très précis (p.ex., contraindre l'amplitude thermique d'une pièce) que totalement « abstraits » (p.ex., la conception de l'environnement doit simplifier au maximum le travail de décontamination) ;*
- *le tout avec des contraintes d'exploitation très dures (vies humaines en jeu) et particulièrement sensibles dans l'opinion publique (cf. grands scandales sanitaires de ces dernières années) ;*

6. Terme que l'on pourra rapprocher de celui de « stakeholder » en anglais.

- dans des contextes sociaux, économiques et politiques nationaux (politique de lutte contre les infections nosocomiales et réduction des dépenses publiques) et locaux potentiellement contradictoires ;
- et enfin utilisé par des acteurs (le personnel médical) très éloignés des décideurs et autres acteurs de la production des besoins et exigences.

3.2.4 Activités

Pour formaliser de manière efficace les besoins et exigences, dans un contexte aussi complexe, plusieurs activités ont été identifiées, celles-ci étant fortement intriquées et forment un « tout » cohérent :

L'analyse du domaine consiste à bâtir le produit et son contexte. Il s'agit de répondre à une série de questions, parmi lesquelles :

- Quel est le système (le produit) « existant » ? Quelles sont ses limitations, ses avantages, ses principales pistes d'amélioration, etc. ?
- Qui sont les « sachants » ? Il faut les identifier, les interroger. . .
- Quelles sont les contraintes « économiques », « sociales » et « politiques » pesant sur le système ?

Cette étape est très « générique », dans le sens où elle s'attache au produit avec une certaine distance.

L'élicitation vise à parfaire la définition des besoins et exigences.

Pour cela, il faut confronter les exigences et les objectifs avec la définition du système pour s'assurer de leur adéquation (p.ex., à l'aide de scénarios d'interaction⁷).

C'est également proposer des systèmes alternatifs, s'interroger sur des approches non conventionnelles, pour arriver à de nouvelles interactions entre le système et son environnement, avec potentiellement différents compromis d'exigences non fonctionnelles. Si les différents systèmes partagent un contexte et des objectifs similaires, leurs exigences pourront varier ; certains choix pouvant aboutir à des contraintes supplémentaires ou au contraire lever certaines barrières.

Enfin, c'est également s'assurer d'un vocabulaire commun entre les différents acteurs afin de ne pas omettre un besoin ou intégrer des exigences incorrectes sur des abus de langages ou des incompréhensions (terme usité dans deux domaines avec deux sens différents, termes différents pour désigner le même concept, etc.).

Au terme de cette activité, nous disposons d'un vocabulaire commun permettant d'exprimer différentes variantes d'un système final s'intégrant dans le contexte défini et respectant les objectifs, besoins et exigences.

La négociation et l'accord entre les différents sachants est suant à lui nécessaire afin de déterminer quel est le meilleur système parmi tous ceux définis.

Lors de ces discussions, on cherchera à évaluer les risques des différentes solutions et à comparer les différents compromis effectués.

7. On pourra associer ces scénarios aux « cas d'utilisation » UML [SR07], c'est d'ailleurs leur origine.

La spécification précise des besoins et exigences à respecter.

Une fois le système visé sélectionné, l'objectif est alors de le spécifier de la manière la plus précise et formelle possible.

L'analyse des spécifications permet ensuite de s'assurer de n'avoir pas laissé d'anomalie dans celles-ci.

Une nouvelle série de questions doit guider cette activité :

- L'ensemble des cas d'utilisation a-t-il été traité ?
- Tous les objectifs trouvent-ils des réponses ?
- Toutes les exigences fonctionnelles sont-elles respectées ?
- Les attentes exprimées concernant les exigences non fonctionnelles sont-elles réalistes ?
- Le projet est-il crédible ? Techniquement ? Financièrement ? Dans les temps impartis avec les moyens alloués ?

Cette phase est particulièrement importante au regard des énormes surcoûts engendrés par un mauvais traitement des exigences (cf. Section 3.2.1) ?

La documentation de l'ensemble du processus et du déroulement de chacune des activités permet de disposer d'une forme de « base de données » des différents choix et compromis réalisés durant la définition du contexte et du système visé. Lors de travaux ultérieurs, il est alors possible de s'y appuyer afin de réévaluer les limitations, reconsidérer certains choix, etc.

C'est une activité de « gestion des connaissances » (*knowledge management*, en anglais) où l'on cherche à capitaliser les expériences passées pour nourrir les réalisations futures.

L'évolution de l'ensemble de ces éléments au cours du temps et des différentes « mises à jour » du produit considéré recouvre plusieurs thématiques différentes :

- Suivre l'évolution des exigences, afin de pouvoir réévaluer la pertinence des solutions précédemment écartées à l'aune de ces nouvelles contraintes.
- Détecter « l'obsolescence normative » d'un produit lors de l'évolution de son contexte normatif.
- Suivre l'évolution des choix et compromis effectués afin de parfaire le travail documentaire.

La vérification du respect des exigences par le produit réalisé regroupe quant à elle toutes les actions visant à s'assurer de la conformité du produit avec ses exigences, tant durant la réalisation du produit lui-même que durant son usage. Selon la forme et les outils employés pour la rédaction des spécifications, cette activité peut s'automatiser (à divers degrés).

Ainsi, par l'emploi d'outil de vérifications formelles, il est possible de prouver mathématiquement le respect par un logiciel de certaines exigences. La portée de ces vérifications dépendra de la capacité de l'outillage et du type d'élément (code informatique, circuit électronique...) à vérifier. Par exemple, dans le cas de circuits électroniques, il est possible de s'assurer du temps de réponse le plus long possible pour un système, quel que soit le « chemin » emprunté dans le circuit ; pour des logiciels, cela pourra

être l'assurance de l'absence d'état indéterminé ou interdit, quel que soit le chemin d'exécution.

Cette dernière activité est peut-être la meilleure et la plus accessible des illustrations de surcouts annoncés dans la Section 3.2.1. En effet, même en admettant un système aux spécifications précisément définies et développé avec des outils de vérification formelle tant pour sa partie matérielle que logicielle, celui-ci ne pourra jamais faire mieux que les spécifications formulées. Si ces spécifications sont erronées, incomplètes, mal interprétées, etc., cela aboutira forcément à des écarts qui entraîneront des surcouts d'autant plus importants que les contraintes portées sur la réalisation et sa validation auront été importantes.

Nos objectifs s'inscrivent totalement dans ces activités, nous visons précisément à analyser le domaine, spécifier les contraintes pesant sur un environnement normé, vérifier la conformité du produit avec ses spécifications et traiter les problématiques liées à l'évolution des normes encadrant l'environnement.

Maintenant que nous avons défini l'Ingénierie des Exigences en termes de problématiques, d'objectifs et d'activités, détaillons rapidement son évolution depuis son avènement. Quelles ont été les différentes réponses apportées⁸ ?

3.2.5 Historique

D'un point de vue purement « informatique », la principale question de l'Ingénierie des Exigences est la modélisation du système et son contexte. C'est-à-dire, la représentation (et, à fortiori, le langage de représentation) employée pour le produit final, ses exigences et son environnement.

La modélisation est essentielle, car elle servira de support pour la description, la spécification, la documentation, la comparaison et l'évaluation des différents systèmes possibles.

Mais, ce qu'il faut bien comprendre, c'est que la modélisation représente le système tout autant que le système détermine la modélisation. Si la première relation peut sembler relativement évidente, la seconde pourrait surprendre.

C'est pourtant bien le cas. . . Comme nous l'avons précisé, dans cette ingénierie les notions de « système », « d'exigence » et « d'environnement » sont totalement « génériques » et couvrent une extrême diversité de situation, requérant par conséquent des termes et des approches spécifiques.

Ainsi, si la modélisation peut éventuellement faire appel à un langage « générique », le vocabulaire employé, quant à lui, sera forcément « spécifique » à la situation.

On comprend donc que le langage de modélisation fait le système tout autant que ce dernier le contraint.

Cela confirme notre assertion initiale. . . Dans ce domaine d'ingénierie, la question primordiale est celle de la définition du langage de modélisation ; c'est

8. Comme nous l'avons précisé précédemment, le domaine de l'Ingénierie des Exigences est extrêmement vaste et regroupe des travaux dépassant largement le champ de l'informatique (psychologie, sociologie, ingénierie des procédés, etc.). Notre domaine de compétence et d'étude étant l'informatique nous avons choisi de nous concentrer sur les approches techniques dans cet historique.

par là qu'il faut commencer, avant tout autre chose (d'où les activités d'analyse du domaine et d'élicitation du besoin).

Cette problématique se résume principalement à quatre questions :

- Que faut-il exactement représenter ?
- Comment peut-on représenter ces éléments ?
- Comment faire cela de manière formelle ?
- Comment raisonner sur cette représentation ?

Les deux premières questions accompagnent la définition d'un vocabulaire pertinent pour aborder le problème rencontré :

- La première question vise à définir la liste des concepts pertinents pour le problème traité. Ce que l'on cherche à obtenir, c'est une réponse au triptyque : quoi, comment, pourquoi⁹ ?
- La seconde question vise à définir les relations existant entre ces concepts, quels sont les liens existants entre chaque entité. Il faut maintenant mettre en perspective ces différents modèles, les combiner, etc.

La troisième question traite du « langage », comment représente-t-on ces concepts et relations pour pouvoir les utiliser ? Est-ce une spécification technique « formelle », « informelle », « semi-formelle », etc. ?

La dernière question est celle de l'outillage. Nous disposons d'un langage, d'un vocabulaire, quels sont les outils pour les manipuler durant les différentes activités à réaliser (élicitation, spécification, analyse...).

Au cours des dernières décennies, les différents travaux engagés se sont attachés à apporter et à débattre sur de nouvelles réponses et approches à ces questions [vL00].

Approches primitives

Structured Analysis and Design Technique (SADT) fut la première approche concrète d'Ingénierie des Exigences, proposée par Ross [Ros77] comme une mise en œuvre de sa définition de la gestion des exigences (cf. Section 3.2.2).

Ce langage permettait de représenter et lier différents modèles, analysés à l'aide de règles de cohérence/consistance. L'ontologie (la boîte à outils conceptuels) disponible était riche et permettait de représenter des données, des opérations ainsi que des concepts plus avancés comme les événements, les agents ou les « triggers ».

Il disposait également d'un aspect hiérarchique permettant de définir par subdivisions successives des concepts (modèles) composites.

C'était un langage « semi-formel » graphique. Si la manière d'exprimer les concepts (données et opérations) était formellement définie, les exigences étaient quant à elles décrites en langue naturelle et manuellement vérifiées.

SADT fut enrichi par plusieurs contributions... Par exemple, Bubenko [BJ80] introduisit des techniques de modélisation pour les entités et les événements avec la possibilité d'exprimer des contraintes formelles sur ceux-ci (notamment temporelles).

SADT ouvrit la voie à de nombreuses autres techniques semi-formelles à la fin des années 70 [Che76, DeM79, WS80], mais celles-ci s'avéraient moins ambitieuses, moins expressives et beaucoup plus spécialisées que leur prédécesseur.

9. « *What ? How ? Why ?* » en anglais.

Dans la décennie suivante, RML [GMB82] enrichit nettement SADT en y introduisant de nouveaux concepts forts comme la généralisation, l'agrégation ou encore la classification pour les données, les opérations et les contraintes. Pour ces dernières, il permettait également l'expression de contraintes formelles, notamment pour les références temporelles.

Les « agents »

L'évolution suivante fut la prise en compte, la constatation qu'un système en devenir n'est pas « inerte » (« statique »), mais plutôt la conjonction de plusieurs sous-systèmes actifs, coopérants, dont le fonctionnement doit être contraint pour atteindre les objectifs fixés¹⁰.

La première mise en œuvre concrète de cette notion « d'agent » fut celle de Feather [Fea87] permettant de représenter des agents, leurs interactions ainsi que le moyen de raisonner sur leurs choix comportementaux et responsabilités individuelles dans le cadre des objectifs collectifs. Cette notion de « responsabilité » devient essentielle dans cette approche et « responsabiliser » des agents (qu'ils appartiennent au futur produit ou à son environnement) devient alors un point central du processus d'Ingénierie des Exigences. Cela devient alors une forme d'engagement contractuel pour ceux-ci [FP87, JS93, KMQ93].

Les « objectifs »

Raisonnement en termes d'objectifs, c'est s'intéresser au « pourquoi » et non plus au « qui, quoi, comment » comme ce fut le cas dans les approches précédentes. On s'interroge sur la formalisation des raisons pour lesquelles le respect des exigences exprimées permettra d'atteindre les objectifs de plus haut niveau réellement poursuivis par le système [HTC74, Mum81, BL91, RG92, RS05].

Le principal intérêt des approches « orientées objectifs » (*Goal-Oriented Requirements Engineering (GORE)* en anglais) est de fournir une métrique simple pour juger de la complétion des exigences. On dispose de suffisamment de contraintes lorsque celles-ci permettent d'atteindre les objectifs (explicitement formulés dans les modèles) poursuivis [Yue87].

Deux approches coexistent :

Des méthodes « formelles » [DFVL91] avec des objectifs sous la forme de graphes structurés « ET/OU » (travaux issus de l'intelligence artificielle [CL72]) :

- L'opération « ET » permet d'exprimer un objectif de « haut niveau » comme le respect d'un ensemble d'objectifs plus simples.
- L'opération « OU » permet d'exprimer un objectif de « haut niveau » comme le respect d'un objectif parmi un ensemble d'objectifs plus simples.

Des méthodes « qualitatives » [MCN92] avec l'introduction de la notion d'objectifs « légers » et d'une définition « hiérarchique » des objectifs.

Dans cette vision, on considère qu'atteindre totalement un objectif est « improbable ». Ainsi, on vise plutôt la réussite d'un ensemble de sous-objectifs permettant de considérer l'objectif principal atteint de manière

10. Cette évolution peut également être considérée comme l'ajout d'une question « qui » au triptyque précédemment évoqué (cf. Section 3.2.5).

suffisamment certaine. Chaque sous-objectif est alors pondéré (positivement ou négativement) en fonction de son impact sur l'objectif global.

Méthodes « formelles »

L'approche formelle (c.-à-d., de graphes ET/OU) a donné naissance à la méthodologie KAOS¹¹ [vLDDD91, DVLF93, vL08, VL⁺09] pour l'élicitation, la spécification et l'analyse des objectifs, besoins, exigences, scénarios et responsabilités. Cette méthodologie est actuellement l'une des plus connues en Ingénierie des Exigences, enseignée à l'échelle mondiale.

De très nombreux travaux ont eu lieu sur la méthodologie KAOS afin d'en enrichir le fonctionnement initial, parmi ceux-ci :

- une couche optionnelle de vérification permettant d'utiliser différents types de raisonnements formels ;
- la formalisation des objectifs et des exigences dans une logique temporelle [MP92, Koy92] permettant de prouver la correction et la complétion de la décomposition effectuée [DvL96] ;
- la détection formelle de conflits entre des objectifs [VLDL98] ainsi que la gestion de contraintes environnementales générales pouvant nuire à leur réussite [LL98] ;
- l'inférence des exigences (exprimées sous la forme de préconditions, post-conditions ou « triggers » dans KAOS) directement depuis des scénarios d'utilisation [vLW98].

D'autres approches formelles sont apparues parallèlement à KAOS, comme i^* (prononcer « i étoile »¹²) un langage destiné aux phases préliminaires de modélisation du domaine [Yu97, YGMM11]. Le nom « i^* » se rapporte à la notion « d'intentionnalité distribuée », au cœur de ce framework.

L'approche i^* fut originalement développée pour représenter et raisonner sur des organisations et leurs systèmes d'informations, composés d'agents hétérogènes avec des objectifs différents — souvent concurrents —, mais interdépendants. Cette méthode est donc tout autant orientée « agents » qu'« objectifs ».

La différence majeure avec KAOS réside dans le support des « intentions » des différents acteurs.

Méthodes « qualitatives »

Les méthodes qualitatives (introduisant la notion « d'objectifs légers ») ont donné naissance à la méthodologie Non-Functional Requirements (NFR) pour la conception et l'analyse de décompositions alternatives des objectifs. On peut voir cette méthode comme une alternative simplifiée de la méthode formelle pour des problèmes simples, et comme une méthode complémentaire pour les objectifs de haut niveau qui ne peuvent être formalisés.

La méthode NFR illustre parfaitement l'intérêt des approches « orientées objectifs » pour l'expression des objectifs non fonctionnels permettant ainsi d'éliciter correctement les contraintes non fonctionnelles. Cet avantage a été illustré

11. Un acronyme pour « Knowledge Acquisition in autOmedated Specification » ou « Keep All Objectives Satisfied ».

12. « *i star* » en anglais.

par Nixon [Nix93] par la comparaison ses performances de différentes variantes du système durant les phases d'Ingénierie des Exigences et de conception.

Par corolaire, ce type de méthode non formelle est également particulièrement pertinent dans les travaux où l'on dispose seulement d'une connaissance imparfaite du domaine et où le travail d'Ingénierie des Exigences vise précisément à assurer le « meilleur compromis » entre exhaustivité des exigences et atteinte des objectifs non fonctionnels [CD15] (par exemple, dans le domaine de la sécurité [AKM94, Lev95]). Enfin, parmi les nombreuses contraintes non-fonctionnelles, on notera celle de l'interopérabilité [MDCV15, BDC15, CDB15] qu'il ne faudrait pas négliger.

Liens « agents ↔ objectifs »

Des modèles basés sur les concepts mixtes d'agents et d'objectifs peuvent être créés en utilisant des liens spécifiques entre les deux représentations.

Par exemple, dans KAOS les agents peuvent être assignés à des objectifs par des liens ET/OU de responsabilité, permettant ainsi de disposer d'une autre perspective sur le système et son environnement. Une nouvelle fois, des préconditions, postconditions et « triggers » permettent d'exprimer les exigences liant un agent et son objectif.

Des travaux de recherches complémentaires [YM94, Yu97] ont permis d'introduire des dépendances entre les agents (sur une tâche ou une ressource). On peut ainsi définir une forme de plan de travail de l'agent pour atteindre ses objectifs et étudier les éventuels conflits généraux que cela peut générer.

Points de vue et conflits

Peu importe finalement l'approche, qualitative ou formelle, les recherches ont montré que la modélisation des « objectifs » est le bon niveau « d'abstraction » pour traiter les éventuels conflits.

Pour isoler et traiter ces conflits, une procédure typique [Rob89, BBHL95] consiste à identifier les exigences antagonistes pour traiter les objectifs contradictoires. Celles-ci peuvent alors être débattues (phase de négociation), les objectifs modifiés et les exigences régénérées, autant de fois que nécessaire.

Avec du recul, les conflits d'exigences sont principalement le fait de différends de « point de vue » entre les différents « sachants » [Eas94].

La question est alors de savoir comment aborder et traiter ces différents points de vue de manière à intégrer le savoir obtenu dans la modélisation du système et son environnement.

Deux approches ont émergé :

Centralisée [NMS89, ZJ93] où les points de vue sont traduits sous forme d'assemblages logiques pour une analyse globale. La vérification de conflits passe par une forme de conjonction des différents graphes.

Distribuée [NKF94] où les points de vue ont des règles spécifiques de consistance associées. La vérification de conflits passe par une analyse des règles entre points de vue.

Attention, un conflit identifié durant la modélisation du système ou son environnement n'est pas nécessairement un « problème » qui doit être immédiatement traité. Lors de l'élicitation, la confrontation des différents points de vue permet d'obtenir des informations très intéressantes même si celles-ci sont partiellement conflictuelles. Il ne faut pas chercher à régler les problèmes de manière prématurée, ce n'est qu'en ayant une vision globale et la plus exhaustive possible que la phase de négociation sera pertinente.

Devant une telle diversité d'information à représenter et traiter, il est rapidement devenu évident qu'il fallait disposer d'un langage de spécification « multi paradigme », aucun langage « mono paradigme » existant ne pouvant se targuer d'être aussi versatile. Les différents travaux visant à intégrer des langages différents « dans un tout cohérent » semi-formel [RBP⁺91] ont conduit à la naissance du standard UML [BRJ96, RJB04].

Génération par les scénarios

Bien que les approches « orientées objectifs » soient redoutablement efficaces, l'élicitation des objectifs reste toujours une opération délicate et l'exhaustivité extrêmement difficile à garantir. L'utilisation de « scénarios » permet alors de « concrétiser » la réflexion et d'éviter de faire travailler de manière « abstraite » les « sachants » [WPJH98, JBC98].

De manière pratique, un scénario est une séquence d'interaction entre le système et son environnement en se concentrant sur la réalisation d'un objectif « implicite » (non formulé). En d'autres mots, le « sachant » doit décrire son utilisation du système « tâche après tâche ».

De très nombreuses utilisations ont été explorées pour les scénarios : élicitations [PTA94], identification des cas « exceptionnels » [Pot95], complétion de modèles existants après un travail « abstrait » [RBP⁺91, RG92], validation des exigences [Sut97], etc.

Nonobstant ces différents avantages et toutes les utilisations possibles des scénarios, cette approche reste inévitablement partielle. Elle souffre de plusieurs limitations issues des concepts et outils qu'elle manipule.

Tout d'abord, comme pour les « tests unitaires » en développement logiciel, il est impossible de garantir une couverture totale de toutes les possibilités du système (ceci entraînant par conséquent l'impossibilité de garantir l'absence formelle d'erreurs).

Ensuite, les scénarios étant des suites d'évènements représentés par des graphes d'états, l'analyse du déroulement possible de plusieurs graphes concurrents (chaque acteur prenant part à un scénario obéissant à sa propre logique) entraîne une explosion combinatoire de par l'énumération de toutes les combinaisons possibles.

De plus, les scénarios peuvent être « sur spécifiés » et, sans y prendre garde, des choix prématurés peuvent être effectués sur la nature du système et son interaction avec l'environnement [AMR⁺14].

Enfin, pour bâtir les scénarios, les « sachants » décrivent leur manière d'utiliser le système dans le contexte d'un environnement qu'il leur est bien connu. Par conséquent, ils font appel à des propriétés « implicites » du système qu'ils s'attendent à trouver et qu'ils n'explicitent pas forcément dans leur description

(on a toujours tendance à « négliger » les évidences). Il va donc falloir « pousser » les « sachants » à décrire ces propriétés implicites, au risque d'amener à la sur-spécification précédemment mentionnée.

Réutilisation des exigences

Par nature, une exigence est liée à une tâche spécifique dans un domaine bien précis.

Cependant, il ne semble pas aberrant d'imaginer que dans un même domaine, des tâches similaires mènent à des exigences identiques. Par ailleurs, et dans une logique identique, des domaines connexes partageront un certain nombre de tâches communes.

D'un point de vue plus technique, si les domaines sont suffisamment proches, un travail d'Ingénierie des Exigences pourra conduire à des environnements (quasi)identiques et les systèmes décrits présenteront de nombreuses similitudes.

Des recherches sur la définition, la spécification, de tâches, exigences, environnements ou domaines « réutilisables » présentent donc un certain intérêt (déjà bien connu dans le cadre des briques logicielles) [RW91].

D'autres travaux ont utilisé des méthodes d'apprentissages supervisées pour détecter des similarités d'exigences au niveau structurel [MS92] ou sémantique [MvL97]. Pour les tâches, la classification et la caractérisation de « types » de tâches ont été investiguées.

Documentation des exigences

Tous les éléments décrits dans ce chapitre doivent être documentés, tant dans leur modélisation finale que dans le processus qui a conduit à leur production [NKF94]. Ce sont des artefacts indispensables pour la communication, l'inspection, la négociation et l'évolution des représentations du domaine et du système.

Le système et le domaine étant voués à évoluer, c'est par conséquent une évidence (et une nécessité) qu'il faut, plutôt qu'agir à posteriori, anticiper ces futurs travaux et documenter le travail effectué dès son origine.

Il ne faut enfin pas oublier que les auteurs d'un choix (d'un compromis) et leurs motivations sont aussi importants que le choix lui-même et doivent forcément être décrits [GF95].

Travaux théoriques

Parallèlement à tous ces travaux orientés « langage et technique de modélisation », une partie de la communauté s'est recentrée sur les fondamentaux : définir clairement et précisément les termes, les concepts ainsi que les fondements théoriques de l'Ingénierie des Exigences [Jac95, PM95, Zav97].

Des recherches plus « fondamentales » ont donc été engagées pour clarifier la nature réelle des exigences ; trop d'amalgames étant rencontrés dans la littérature (notamment entre « exigences » et « spécifications logicielles »).

La première distinction effectuée [Jac95, PM95] fut celle entre « propriété du domaine » (indicatives) et « exigences » (optatives). C'est un point essentiel pour

certaines produits afin de bien faire comprendre la distinction existante entre les éléments non-négociables du domaine (les lois de la physiques, la législation, etc.) et les exigences. Les premières ne sont absolument pas des exigences, ce sont des « propriétés ».

La seconde distinction [Jac95, PM95] est faite entre « les exigences » du système considéré et les « spécification » du logiciel l'implémentant :

Les « exigences » sont formulées en terme d'objets du monde réel dans un langage « métier ». Elles doivent être compréhensibles et négociables par des « sachants » souvent très éloignés des contingences informatiques.

Elles désignent les relations requises entre des objets dans l'environnement suivi et contrôlé par le logiciel.

Les « spécifications » sont formulées en terme d'objets manipulés par le logiciel dans un langage accessible aux développeurs informatiques. Elles désignent les relations entre les entrées et les sorties d'objets informatiques.

Enfin une dernière distinction [VLDL98] mérite d'être formulée entre « exigences » et « hypothèses/postulats » (*assumptions*, en anglais) :

Les « exigences » sont imposées par le logiciel.

Les « hypothèses/postulats » sont imposés par les agents présents dans l'environnement.

Autres pistes de recherches

Partant de ces éléments, de nombreuses pistes de recherches ont émergé, tant dans le domaine des langages et des méthodologies que dans celui de l'outillage. Nous allons présenter les principales pistes en lien avec nos objectifs.

L'exploration des synergies entre recherches en « Ingénierie des Exigences » et recherches en « architecture logicielle » est une thématique complexe, la versatilité des exigences au cours de la vie d'un produit se mariant effectivement mal avec la nécessaire stabilité des architectures logicielles pour garantir la rentabilité du produit final.

L'extraction (par analyse sémantique) des paramètres précis d'un produit, directement depuis les exigences.

Dans le cas de lignes de produits, lorsque le produit comporte plusieurs variantes mineures dépendant simplement de quelques choix effectués par le client, l'objectif sera d'automatiser le travail et de réutiliser au maximum les modélisations existantes [DS06, ACR⁺15]. L'extraction des paramètres « particuliers » de cette « instance » du produit « générique » devrait pouvoir s'automatiser.

L'exploration des synergies entre recherches en « Ingénierie des Exigences » et recherches en « spécifications formelles » se donne quant à elle comme objectif d'améliorer largement les capacités de réflexion sur la modélisation du système et son environnement en intégrant les travaux issus des spécifications formelles dans les langages d'Ingénierie des Exigences [Lam00], notamment toutes les possibilités offertes par l'outillage d'analyse formelle [DAC99].

La poursuite des travaux engagés sur la documentation du système et son environnement, afin de bâtir des modèles capables de combiner des raisonnements semi-formels, formels et qualitatifs pour les exigences non fonctionnelles, conduit

à des compromis entre expressivité/précision pour enrichir l'analyse et simplicité d'utilisation pour faciliter l'utilisation des outils mis au point.

Historiquement, le concept « d'agent » s'est souvent vu limité à deux composants : le logiciel et son environnement. En réalité, il existe de nombreux agents (logiciels, humains, machines, sociétés, etc.) devant coopérer pour mener à bien les objectifs fonctionnels et non fonctionnels ciblés. Ainsi, ignorer des agents, c'est occulter un pan complet du système et se préparer, avec certitude, à de graves déconvenues futures [Lev95, BMPC98].

Une autre piste consiste en la mise au point de techniques, de langages, pour raisonner sur des alternatives au système actuel et anticiper les futurs changements plausibles.

Trop souvent les modélisations des alternatives d'un système n'ont aucun lien « formel » avec l'original, c'est dommageable. Il y a un intérêt évident à conserver cette information lorsque l'on fait évoluer les exigences ou les objectifs, afin (par exemple) de pouvoir automatiser la génération des alternatives précédemment écartées pour les réévaluer.

La mise en œuvre des techniques « itératives » pour des conceptions et des vérifications « incrémentales » des exigences [SB14].

Au lieu de raisonner sur le système complet dans son environnement final, il s'agit d'offrir le moyen de travailler sur des modélisations partielles et valider « étape par étape » la représentation. L'objectif est, notamment, d'isoler au plus tôt les conflits et de faciliter le travail documentaire sur les choix et les compromis effectués.

Enfin, même si ce survol historique a uniquement traité de « l'Ingénierie des Exigences », il ne faut pas négliger le domaine connexe de la « réingénierie des exigences », c'est-à-dire l'analyse et la réécriture des documents utilisés afin de disposer de matériaux plus « formels » et adaptés aux différentes activités décrites.

3.2.6 Travaux connexes

Nous allons maintenant aborder les travaux d'Ingénierie des Exigences connexes à nos objectifs. Ces travaux se divisent principalement en deux catégories : ceux centrés sur la représentation des exigences réglementaires (avec ce que cela comporte comme opérations complémentaires) et ceux centrés sur l'automatisation de la vérification des exigences réglementaires.

Cependant, à notre connaissance, il n'existe pas de travaux appartenant aux deux catégories et couvrant des « bâtis » et encore moins de leur conception à leur utilisation tel que présenté dans nos objectifs.

Par conséquent, nous définissons plutôt ici le « paysage » dans lequel s'inscrivent nos recherches.

Représentation des exigences normatives

Les travaux visant à la représentation des exigences réglementaires et légales sont relativement peu nombreux et concentrés sur des domaines très ciblés¹³.

13. Attention, nous ne parlons pas ici de tous les recueils de « bonnes pratiques » orientés « métiers ». Nous visons plutôt des documents s'inscrivant dans le cadre de l'Ingénierie des

Dans le domaine médical, l'un des principaux domaines concernés par ce type de travaux, on pourra citer les « Health Insurance Privacy Accessibility Act of 1996 (HIPAA) Omnibus Rule » [fCRO13] de l'Office for Civil Rights (OCR) de l'U.S. Department of Health and Human Services (HHS).

L'HIPAA Omnibus Rule est une synthèse enrichie et renforcée du précédent texte, l'Health Information Technology for Economic and Clinical Health (HITECH) Act [fCRO09], dont l'objectif était de stimuler l'adoption du Dossier Médical Personnel (DMP) et des technologies le supportant¹⁴. L'HITECH fixait alors les contraintes légales entourant le DMP et renforçait le cadre précédemment posé par les « HIPAA Privacy Rule » [fCRO02] et les « HIPAA Security Rule » [fCRO03].

D'autres activités sont concernées comme la réglementation du transport au Canada [BCS⁺12], la réglementation nucléaire finlandaise [URMT11, RMTV11] ou française [San13, SB14], le domaine médical restant le plus exploré et usité; les textes HIPAA étant souvent employés pour illustrer les approches « génériques ».

Parmi ces approches, on pourra citer les importants travaux de Breaux et al., autour de sa méthode dite « semantic parameterization » [BVA06] visant à reformuler les exigences exprimées en langage naturel dans des catégories formelles. Plusieurs méthodes de formalisation ont ainsi été proposées : ICM [BAD08], FBRAM [BABD09]... Des travaux ont également été menés dans l'étude comparative inter-juridictionnelle et la détection de conflits [GB12, GB13].

On pourra également citer ceux de Anton et al., autour des exigences légales : priorisation [MOA09], taxonomie et formalisation [MA09, MOHA10, YA10], évolution [MAS12a], détection de conflits [MAS11, MAS⁺12b]

Avec un outillage plus important, on peut citer les travaux de Mylopoulos et al. [KZB⁺07, KZB⁺08] utilisant le framework d'extraction d'informations lexicales Cerno [KZC⁺09], la vérification d'exigences modélisées en i^* [Yu97, YGMM11] et Nomos [SJI⁺12] pour la révision et la justification/démonstration de conformité à la régulation, enfin plus récemment la mise au point du framework Desiree pour la formalisation des exigences [LBG⁺16].

Les différents travaux de recherche sont relativement proches, les objectifs poursuivis et les normes employées étant relativement similaires, et ne diffèrent principalement que par l'approche adoptée.

Dans ces travaux, la vérification des contraintes s'effectue manuellement, en utilisant les contraintes préalablement extraites [MSOA11], ou par l'utilisation d'approches à contraintes [MLHS⁺11, SMD⁺12].

Automatisation de la vérification des exigences normatives

Pour automatiser la vérification des exigences normatives, l'approche de prédilection est le Model-Driven Engineering (MDE). Ce type d'approche comporte peu d'acteurs et une faible littérature scientifique.

Dans cette catégorie, on notera principalement les travaux de Panesar-Walawege et al [PWSB11, PW12] sur la norme CEI 61508¹⁵ et ceux de Zoughbi

Exigences; c'est-à-dire, posant des contraintes sur le système final devant répondre au besoin exprimé.

14. L'une des mesures les plus importantes est la finalisation de la clause de notification des découvertes de failles de sécurité

15. La CEI 61508 est la principale référence européenne de la sécurité fonctionnelle. Elle

et al [ZBL07,ZBL11] autour de la RTCA DO-178B¹⁶. Ces deux travaux partagent la même vision sur la façon d’aborder la problématique : le développement d’un profil spécifique à la norme considérée puis un travail avec des experts « métiers » pour instancier ce profil et modéliser les différentes activités prescrites par les normes.

Ce type de travaux repose sur une validation plus « syntaxique » que « sémantique » des exigences.

En effet, dans sa thèse de doctorat, Panesar-Walawege [PWSB11,PW12] propose l’outil CRESCO pour la génération de dépôts de « preuves » de sureté prenant en entrée des modèles conceptuels de domaines sous la forme de profils UML. Le profil est utilisé comme un « guide » des différentes « preuves » à fournir pour attester de sa conformité à la norme visée. Un outil graphique est également fourni pour manipuler et collecter les documents nécessaires à la certification.

Poursuivant sur cette voie, dans des travaux plus récents, de la Vara et Panesar-Walawege [dIVPW13,dIVRA⁺16] définissent le métamodèle « Safety-Met » visant à fournir un cadre générique pour la structuration et la gestion des preuves de conformité pour les contraintes de sureté. Ces travaux font suite à l’étude d’un ensemble de normes de domaines différents : aéronautique (DO-178C), automobile (ISO 26262), ferroviaire (EN 50128), etc.

Pour en revenir à nos propres objectifs, ceux-ci restent assez éloignés de ces travaux « syntaxiques » et des normes considérées. Les réglementations visées par SafetyMet sont très proches les unes des autres et ne concernent pas réellement le même type de contraintes. Ces dernières définissent des niveaux de sureté (« safety integrity levels »), et sont organisées autour d’activités, de documents en entrée et en sortie. Il y a donc peu de textes, chacun imposant un grand nombre de contraintes documentaires, de « preuves » à fournir.

3.3 Conclusion

Dans ce chapitre, nous avons choisi d’aborder l’état de l’art de nos objectifs et leur contexte sous les prismes industriels et académiques.

Dans le monde industriel (Section 3.1) :

La conception est actuellement en pleine mutation. La tendance étant la réalisation de « maquettes numériques » devant représenter le bâtiment et l’accompagner jusqu’à sa destruction (prévue dans la maquette). Ces maquettes sont principalement des outils de consultation et de gestion, les outils d’analyses restant relativement simples et fournissant peu d’aide à la décision.

propose une approche générique de toutes les activités liées au cycle de vie de sécurité de systèmes électrique/électronique/électronique programmable (E/E/PE) relatifs à la sécurité qui sont utilisés pour réaliser des fonctions de sécurité [FMD15].

16. Les normes ED-12C et DO-178C (Software considerations in airborne systems and equipment certification) développées en commun et éditées respectivement par EUROCAE et RTCA Inc., fixent les conditions de sécurité applicables aux logiciels critiques de l’avionique dans l’aviation commerciale et l’aviation générale. Elles précisent notamment les contraintes de développement liées à l’obtention de la certification d’un logiciel d’avionique [Wik15b].

La réalisation reste quant à elle totalement « classique », un découpage vertical permet une division du projet en lots qui seront affectés à des entreprises spécialisées dans l'opération considérée. Nous avons discuté des limitations d'une telle approche, spécifiquement dans le contexte des environnements normés, et de l'intérêt d'une approche « globale » (« horizontale ») pour améliorer, dans le domaine médical, la prophylaxie des lieux.

D'un point de vue académique (Section 3.2), le principal domaine de recherche dans lequel s'inscrivent ces travaux est l'Ingénierie des Exigences, un domaine multidisciplinaire, particulièrement vaste, s'attachant à la conformité d'un produit (au sens large du terme) avec ses exigences (fonctionnelles et non fonctionnelles).

Cela couvre un grand nombre d'activités : l'analyse du domaine, l'élicitation, la négociation sur la description réalisée, la spécification précise du système et son environnement, l'analyse des spécifications, la documentation de l'ensemble du processus et de ses livrables, l'évolution de l'ensemble au fil des années et la vérification de conformité du système produit avec les exigences décrites.

Un certain nombre de travaux s'attachent aux contraintes règlementaires (tant dans leur représentation que leur validation automatique), néanmoins aucun ne s'attache aux environnements normés et les différences entre les systèmes et les exigences étudiés trop importantes pour permettre de transposer les recherches réalisées.

En l'état de nos connaissances, il n'existe aucune solution industrielle s'attachant à la qualification normative d'environnements confinés médicaux (bien que la MNB soit un premier pas intéressant); et, du côté de l'Ingénierie des Exigences, il ne semble pas exister de travaux de recherches s'attachant à suivre le cycle de vie d'un environnement normé (avec tout ce que cela peut contenir comme singularités, au regard de produits industriels plus classiques) de sa conception à son utilisation.

De plus, les pistes de recherches actuellement explorées en Ingénierie des Exigences, sur la base historique décrite précédemment, sont tout à fait en phase avec nos objectifs et s'y inscrivent totalement.

En conséquence, il existe un axe de recherche original visant à répondre aux objectifs présentés dans le Chapitre 1, dans le contexte présenté au Chapitre 2.

Problématique

L'approche présentée dans la Figure 4.1 (rappel de la Figure 2.2 à la page 24) comporte trois grandes étapes : l'analyse préalable, la qualification d'un projet d'environnement et la supervision/l'audit de cet environnement lors de son fonctionnement.

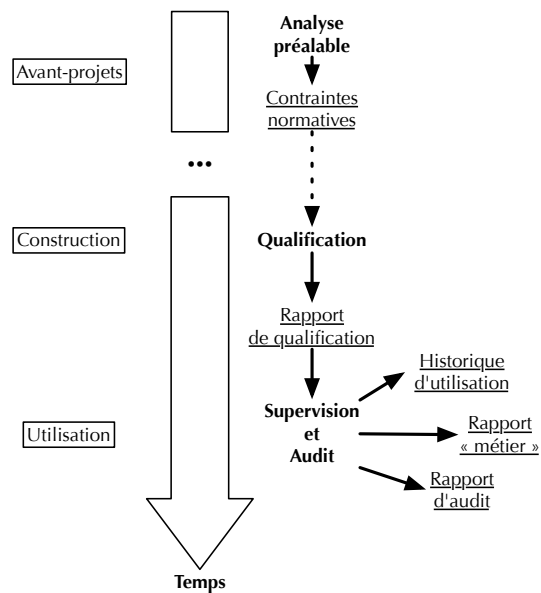


FIGURE 4.1 – Rappel de la problématique réelle

Par conséquent, nous faisons face à une problématique particulière : l'élaboration d'une approche globale et unifiée permettant de répondre au besoin

« complet », tout en respectant les contraintes et problématiques « locales » posées dans chacune des étapes du processus. C'est ce double niveau de difficulté, cet « aller-retour » entre le général et le particulier, qui constitue l'une des principales aspérités de cette thèse. v ggf

Afin de simplifier la compréhension du lecteur, nous avons choisi de présenter séparément les contraintes de chacune d'entre elles puis de présenter les contraintes globales d'intégration dans une synthèse finale.

4.1 Les différentes problématiques

Nous allons maintenant parcourir les différentes étapes de la Figure 4.1 et présenter, pour chacune d'entre elles, les contraintes devant être traitées.

4.1.1 Analyse préalable

L'analyse du cadre normatif est une opération lourde et complexe ne tolérant pas le dilettantisme et requérant une importante préparation.

On s'évertue qu'à la question : « Quelles sont les contraintes liées à la réalisation et l'exploitation d'une centrale nucléaire ? », nous puissions répondre :

- durant la conception, il faut vérifier X_1 , Y_1 et Z_1 .
- durant l'usage, il faut vérifier X_2 , Y_2 et Z_2 .
- durant un audit, il faut vérifier X_3 , Y_3 et Z_3 .

(avec X_n , Y_n et Z_n désignant des opérations précises et pas seulement la mise à disposition de documents ou des contraintes en langage naturel invérifiables automatiquement).

En d'autres termes, pour un « type¹ » précis d'environnement normé, il s'agit de produire la liste des contraintes applicables à chaque étape de son cycle de vie.

Ce qu'il faut bien prendre en compte c'est la dimension « prospective » et « générique » de cette étape. L'approche retenue ne doit pas être applicable qu'à l'instant « t » ou dépendre d'un domaine métier particulier. Bien au contraire, l'analyse préalable doit permettre de croiser et réutiliser les analyses afin de disposer d'une « matière » préexistante lors du traitement de types d'environnements connexes². De plus, il ne faut pas oublier que cette étape constitue le point de lancement de notre approche unifiée. Les choix effectués pour répondre à cette première problématique impacteront totalement le reste de la démarche.

1. Quand on aborde le problème sous le seul angle des environnements normés, la définition de « type » reste assez floue. Cela dépendra très fortement du domaine d'activité concerné, certains imposant une très grande précision.

Ainsi, dans le contexte des environnements normés médicaux, le « type » doit être assez précis. Par exemple, répondre « une salle d'opération » n'est pas suffisant pour être analysé ; en revanche, « une salle d'opération en chirurgie digestive » apporte les informations nécessaires et cette précision n'est pas anodine, les contraintes techniques seront totalement différentes pour une « salle d'opération en chirurgie cardiaque » ou une « salle d'opération en chirurgie orthopédique ».

2. En d'autres termes, si j'ai analysé l'environnement A et que je traite l'environnement B (surensemble strict de l'environnement A), je dois pouvoir réutiliser immédiatement le travail précédent

La problématique de cette première étape est donc de représenter de manière efficace les arrangements normatifs (y compris des contraintes pouvant prendre toutes les différentes formes exposées précédemment), dans une perspective de réutilisabilité maximale (que ce soit par mise à jour des analyses précédentes ou par enrichissement avec d'autres analyses), tout en offrant les éléments nécessaires pour permettre la qualification ultérieure (quelle soit de conception, opérationnelle ou d'audit).

4.1.2 Qualification

La précédente étape nous a fourni un canevas de travail, nous abordons maintenant l'aide à la qualification³ d'un projet précis.

Dans celle-ci, considérant un projet « X » d'un type d'environnement normé « Y », l'objectif est double : s'assurer que le projet est conforme du point de vue de sa conception avec ses contraintes normatives et s'assurer qu'il présentera toutes les garanties pour l'analyse opérationnelle et l'audit (capteurs, actionneurs, systèmes tiers, etc.). Cette étape faisant le lien avec la phase précédente, la description du projet doit être « compatible » avec la représentation des contraintes devant s'y appliquer.

Décrire un projet demande de pouvoir représenter trois types d'information :

La topologie des locaux avec les pièces (superficie, volume, disposition, etc.), les murs, les portes, etc.

Cette première dimension peut être assimilée à une « ossature » pour le projet, elle indique l'objet même sur lequel travailler et fournit les données primordiales pour estimer la conformité réglementaire⁴.

L'instrumentation de ces locaux avec les capteurs, actionneurs et systèmes tiers prévus dans l'environnement.

Cette seconde dimension décrit les outils prévus sur site pour réaliser les qualifications opérationnelles et l'audit du projet. Néanmoins, elle contribue elle-même à enrichir la liste des vérifications et des contraintes à évaluer⁵.

L'arrangement normatif à respecter d'après le contexte précédemment défini.

Cette dernière dimension est fortement liée à l'étape précédente et ses résultats.

3. Remarque subsidiaire, si les réifications de notre flot de conception peuvent aller jusqu'à la certification ; pour notre part, nous ne pensons pas cela possible dans le seul cadre d'un doctorat.

4. En effet, de nombreuses normes encadrent la disposition des locaux, leur volume, la configuration des systèmes tiers (comme la ventilation) en fonction de l'activité et de l'espace concernés, etc. Il est donc primordial de disposer de ces informations pour évaluer la conformité normative du projet.

5. Par exemple, un réfrigérateur pour matériel médical doit être à une température précise, bornée par des normes précises. Or, la mesure de cette température fait intervenir des capteurs étalonnés (c.-à-d., dont la justesse a été qualifiée précisément par des organismes certifiés avec des procédures précises), devant être qualifiés régulièrement.

Par conséquent, la demande de qualification des capteurs fait partie des vérifications à réaliser durant l'usage de l'environnement.

La problématique de cette deuxième étape comporte donc plusieurs volets. Le premier est de traiter correctement les éléments fournis par l'analyse préalable afin de les remettre en perspectives avec la description de l'environnement fournie par l'utilisateur, c'est un travail « conceptuel ». Le deuxième est de générer les éléments nécessaires aux vérifications opérationnelles et à l'audit. Enfin, le dernier est de fournir l'outillage capable de réaliser les deux opérations précédentes : décrire l'environnement existant, valider les contraintes vérifiables à la conception et générer les vérifications mentionnées, c'est un travail « technique ».

4.1.3 Supervision et audit

Cette troisième et dernière phase sort du cadre classique de l'Ingénierie des Exigences, celui-ci s'arrêtant plus traditionnellement à la validation du produit lors de sa conception et traitant plutôt des évolutions des contraintes de conception/réalisation ou des évolutions du besoin. Dans cette thèse, en revanche, nous souhaitons poursuivre la qualification de conception par la vérification du maintien de cette qualification à l'usage.

Ainsi, en utilisant les informations fournies dans le rapport de qualification, la plateforme de traçabilité doit vérifier en temps réel l'ensemble des contraintes exprimées. Ces contraintes pourront être très variables : documentaires (par ex., fournir les bons documents au bon moment), impossibles à automatiser, mais devant générer des alertes pour les techniciens (par ex., faire étalonner des capteurs ou remplacer des systèmes externes), totalement automatisables (par ex., vérifier la température d'une pièce), etc.

Dans cette phase, nous visons trois objectifs :

1. Réaliser le tableau de bord de l'environnement normé. C'est-à-dire, fournir un retour rapide (en temps réel) sur le fonctionnement actuel aux utilisateurs.
2. Conserver un historique de l'ensemble des données relevées dans l'environnement. Ce sont ces données qui ont permis de réaliser le tableau de bord.
3. Réaliser à la demande un audit de l'environnement. Une forme bien plus poussée de l'analyse réalisée en temps réel utilisant l'ensemble des informations disponibles et visant à identifier l'origine d'une anomalie (ou à minima sa première occurrence).

La problématique de cette dernière étape est de faire le lien entre une vérification « statique » et « purement logicielle » sur le projet d'environnement normé et cette vérification « dynamique » et « hybride matérielle/logicielle » sur l'environnement normé en cours de fonctionnement.

Par conséquent, la plateforme doit être en mesure de s'interfacer avec l'ensemble des sous-systèmes présents sur site pour confronter les données acquises auprès de ceux-ci avec les contraintes issues du rapport de qualification (en réalisant si nécessaire une réinterprétation de ceux-ci).

Cette dernière phase est ainsi à la croisée entre l'Ingénierie des Exigences, les réseaux de capteurs et l'exploration de données.

4.2 Synthèse finale

Nous arrivons au terme de ce chapitre présentant la problématique abordée par nos travaux de recherche.

Dans les sections précédentes, nous avons présenté les contraintes et problématiques « locales » propres à chacune des étapes de notre objectif :

- lors de l'analyse préalable, la représentation « efficace » de l'arrangement normatif encadrant un type d'environnement particulier ;
- lors de la qualification d'un projet d'environnement, la combinaison des contraintes de l'analyse préalable avec une description du projet afin de générer un rapport sur le projet et les éléments nécessaires à la vérification à l'usage de la cible ;
- lors de la qualification de l'environnement en fonctionnement, la confrontation des données issues de réseaux de capteurs avec les éléments du rapport de qualification pour identifier au plus tôt les anomalies.

D'un point de vue global, c'est l'enchaînement sans perte d'information de ces différentes étapes qui constitue la contrainte et la problématique principale. Nous devons proposer un flot de conception faisant le lien entre l'Ingénierie des Exigences et des domaines de recherche plus « opérationnels » tels que la sûreté de fonctionnement, les réseaux de capteurs, pour une analyse en « quatre temps » : générique, sur un projet d'environnement, sur le flot de données issu de l'instrumentation de cet environnement et enfin sur l'historique de toutes données concernant cet environnement. Toute la difficulté est donc de ne perdre aucune information durant l'ensemble du processus, de travailler seulement par enrichissements successifs afin de disposer de l'ensemble des informations pertinentes lors des dernières analyses.

Nous allons maintenant présenter notre réponse à cette problématique. Nous commencerons par l'approche unifiée dans le Chapitre 5, puis, dans les Chapitres 6 et 7, nous décrirons plus en avant nos propositions pour la mise en œuvre des deux premières étapes.

Deuxième partie

Contributions

Approche unifiée

Dans ce chapitre, nous débuterons par une présentation générale, avec un schéma global de principe présentant les flux de données, les outils et les résultats produits. Nous détaillerons ensuite chaque grande phase de ce flot de conception afin de préciser ce schéma. Nous veillerons cependant à rester suffisamment génériques pour traiter l'ensemble des environnements normés.

Dans ce contexte, nous nous situons à un niveau « conceptuel » ; c'est-à-dire, avec une méthodologie pouvant s'appliquer à l'ensemble des environnements normés et réalisables de diverses manières.

5.1 Schéma générique

La Figure 5.1 présente une première version simplifiée de notre approche¹. Ce chapitre explicite ce schéma afin d'en produire une version plus détaillée, notre proposition complète. Cette étape intermédiaire nous semble nécessaire pour mieux apprécier les motivations et les fondements de notre proposition ainsi que les problématiques que devront affronter les mises en œuvre concrètes.

Nous avons choisi de présenter notre méthodologie sous la forme de trois phases numérotées de « 0 » à « 2 ». Afin d'isoler clairement le traitement « avant-projet », commun à l'ensemble des projets d'un type d'environnement, mais insuffisant pour la qualification d'un projet particulier, nous avons choisi de le numéroté « 0 ». L'analyse réelle d'un environnement débute donc avec la phase « 1 ».

La phase « 0 » dite d'*analyse du cadre normatif* vise à assimiler le contexte normatif d'un projet afin d'être capable d'analyser un projet particulier. D'un

1. Cette proposition est naturellement à mettre en perspective avec le besoin exprimé dans la Figure 2.2.

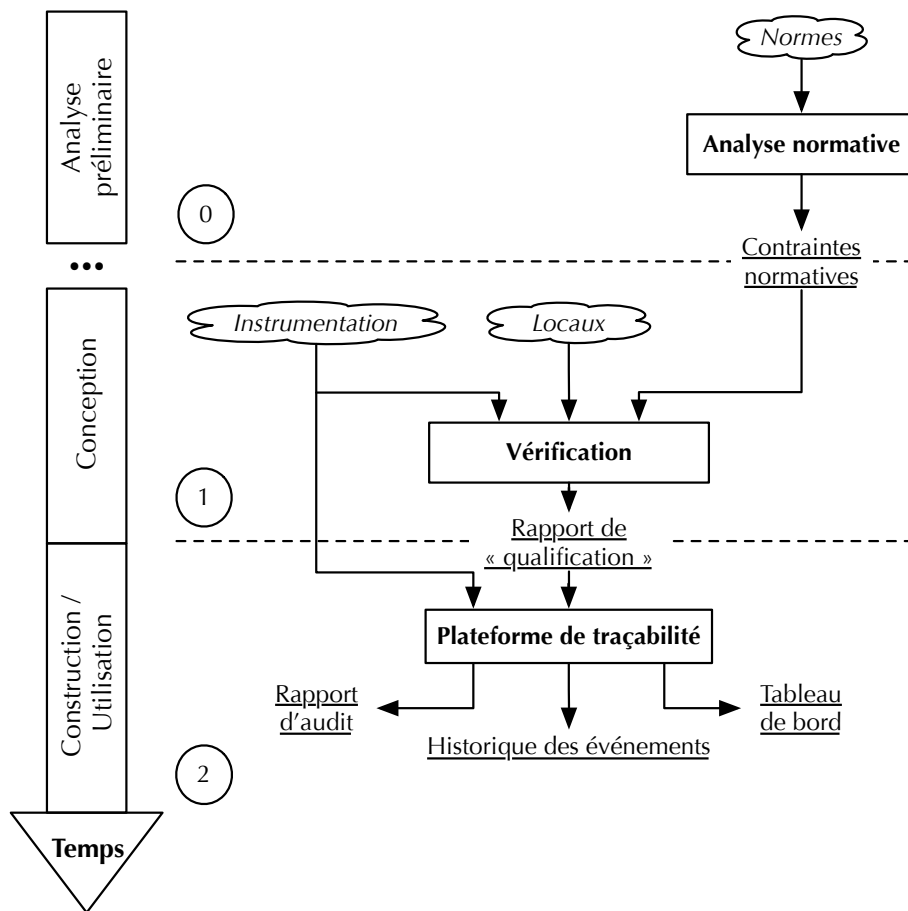


FIGURE 5.1 – Version générique de la méthodologie proposée

point de vue macroscopique, cela consiste à prendre l'ensemble des textes normatifs (au sens large du terme) liés à un sujet et d'en extraire les contraintes réglementaires applicables.

La phase « 1 » dite d'*aide à la qualification* vise à utiliser ces contraintes pour qualifier la conformité d'un projet en se basant sur sa description. D'un point de vue macroscopique, cela consiste à utiliser les contraintes identifiées et les données du projet pour produire un rapport de qualification.

La phase « 2 » dite de *traçabilité* vise à utiliser le rapport de qualification du projet comme un guide pour la vérification en temps réel de son fonctionnement et pour son analyse « à froid » dans un contexte d'audit. D'un point de vue macroscopique, cela revient à utiliser le rapport de qualification comme guide à la fois des contraintes à vérifier en temps réel, mais également des systèmes concernés et du moyen d'accéder à ceux-ci.

Dans la suite de ce chapitre, nous allons détailler ces différentes phases. Nous aborderons chacune d'elle par un rappel de son fonctionnement général suivi d'une description de ses objectifs.

Enfin, nous proposerons une synthèse finale avec une architecture permettant de traiter l'ensemble des objectifs exprimés.

5.2 Phase 0 — Analyse du cadre normatif

Pour mémoire, la première phase consiste à réaliser les opérations présentées dans la Figure 5.2 avec les objectifs présentés en Section 4.1.1.

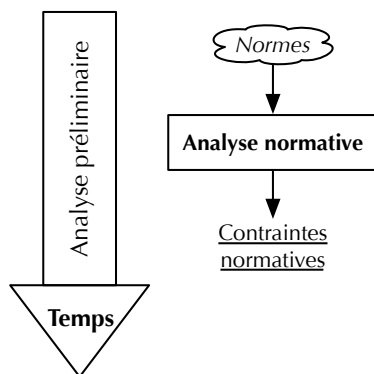


FIGURE 5.2 – Analyse du cadre normatif : principe général

La Figure 5.3 détaille notre proposition d'architecture détaillée pour répondre complètement à l'objectif exprimé.

5.2.1 Constitution de l'arrangement normatif

On débute par une phase de constitution de l'arrangement normatif encadrant la cible.

L'« arrangement normatif » est l'ensemble des textes normatifs s'appliquant à un type d'environnement. C'est un volumineux ensemble de documents, lié à un

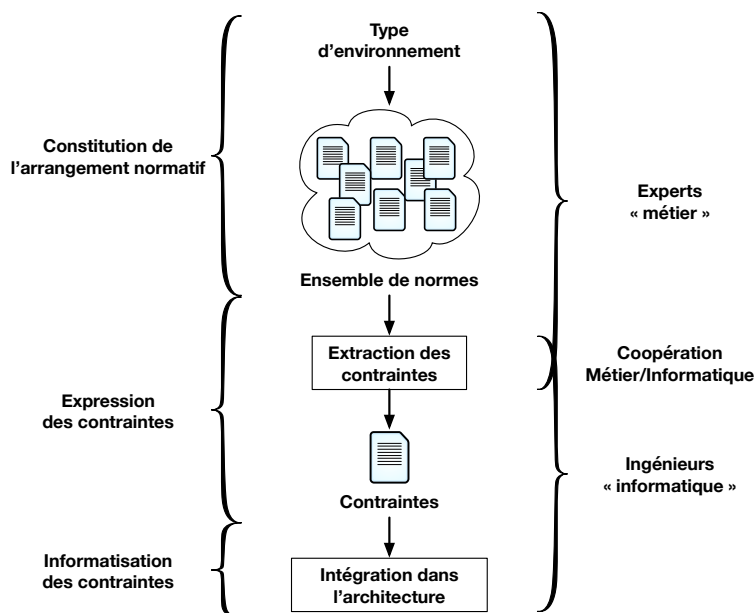


FIGURE 5.3 – Déroulement de l'analyse du cadre normatif

très grand nombre de facteurs : domaine d'activité concerné, type et taille de la réalisation, implantation géographique, etc. On trouve par conséquent des textes internationaux, nationaux, ou même régionaux (contraintes écologiques, zones touristiques, etc.). Dans le cadre de notre analyse, nous nous plaçons d'un point de vue général, il faut prendre en compte l'ensemble des professions concernées et le cycle de vie complet de l'environnement.

Enfin, tous ces textes ne sont pas applicables² et certains sont seulement des recommandations, des guides de bonnes pratiques ou encore des avis techniques. Les recommandations de l'Organisation Mondiale de la Santé (OMS) en constituent un bon exemple [Int10].

C'est une opération complexe, traditionnellement réservée à des spécialistes (juristes, assureurs, experts « métier », etc.), la multiplicité des approches et des compétences est la clé de la réussite de cette étape et d'elle découle la pertinence du reste de notre approche. En effet, tel le laser qui lorsqu'il dévie de quelques millimètres à son émission, produit un point à plusieurs mètres de sa cible lorsque celle-ci s'éloigne, les « angles morts » de l'arrangement normatif conduiront à l'oubli de pans entiers d'analyses à effectuer. C'est extrêmement critique.

Deux conclusions s'imposent, la première est d'utiliser des algorithmes sophistiqués pour la constitution de l'arrangement normatif et la seconde est que celle-ci doit permettre de s'adapter facilement aux évolutions de l'arrangement normatif.

2. « L'« applicabilité » est l'autorité qui s'attache aux dispositions d'un contrat, d'un texte de Loi ou d'un règlement administratif, pour régir une situation juridique donnée. » Dictionnaire du droit privé de Serge Braudo (<http://www.dictionnaire-juridique.com/definition/applicabilite.php>)

Dans ces travaux, nous prenons le parti de ne pas aborder la question des algorithmes. En effet, les textes à étudier ne sont absolument pas formels et leur étude est un travail d'extraction de contraintes « métier » à partir de textes normatifs rédigés en langage naturel. C'est un domaine de recherche à part entière³ et pour lequel de nombreux travaux sont actuellement menés [CW14]. Cela ne nous semble pas pertinent de nous appesantir sur ce point et d'effleurer un sujet qui mériterait d'être traité en profondeur. Nous nous bornerons à indiquer que dans le domaine juridique, l'approche traditionnelle de ces thématiques est l'usage d'ontologies⁴ que l'on nourrit par une analyse des textes en langages naturels [Per82, dRAdOCTQ07, Dea15]. Ces analyses sont plus ou moins simplifiées par les contraintes que l'on fait peser sur le formalisme des textes étudiés ; par exemple, certains producteurs de spécifications techniques (p. ex., l'Internet Engineering Task Force (IETF) [Alv04] pour les normes de télécommunications) utilisent des mots-clés clairement définis pour désigner les contraintes, les conseils, etc... [Bra97, RKB13]. Dans notre cas, ce type de prérequis est totalement inapplicable.

En revanche, nous nous devons d'être extrêmement vigilants sur le second point. C'est l'un des objectifs majeurs de ces travaux, faciliter la réutilisation de la qualification d'environnements lorsque des éléments de l'analyse évoluent (nouveaux bâtiments, nouvelles normes, etc.). Nous avons donc été particulièrement vigilants à proposer une méthodologie pouvant s'adapter aux changements.

Dans la suite de ces travaux, nous ne nous sommes pas appesantis sur ce point et les extraits d'arrangement normatif manipulé dans cette thèse sont issus des travaux de feu la société Nocosium. En somme, l'unique chose à retenir sur ce préalable dans le cadre de ces travaux doctoraux est que, conformément au schéma 5.3, nous disposons à l'issue de cette étape de l'ensemble des éléments normatifs régissant le type d'environnement considéré.

5.2.2 Expression des contraintes

Dans cette deuxième étape, il faut analyser l'arrangement normatif pour en extraire l'ensemble des contraintes normatives.

Bien que ce travail soit le fruit d'experts « métier » (domaine d'activité, juriste, avocats, assureurs), un dialogue avec des experts « informatiques » est requis pour s'assurer d'une parfaite compréhension mutuelle et d'une retranscription correcte des contraintes.

Le principal objectif est de déterminer comment analyser et représenter les normes pour faire communiquer les experts « métiers » et les informaticiens. C'est un savant dosage entre le besoin d'expressivité des premiers, la capacité de compréhension mutuelle et les possibilités de modélisation informatique.

De plus, pour un domaine d'activités, les différents arrangements normatifs selon le type précis d'environnement se recoupent plus ou moins largement. Ainsi,

3. Le Traitement Automatique du Langage Naturel (TALN) (*Natural Language Processing (NLP)* en anglais) [Kid13].

4. **Définition :** *An ontology is an explicit specification of a conceptualisation in the context of knowledge sharing, I use the term ontology to mean a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. [Gru93]*

bien que ce soit deux sujets aux contraintes différentes, il y a plus de points communs entre une « salle d'opération en chirurgie cardiaque » et une « salle d'opération en chirurgie orthopédique » qu'avec une « centrale nucléaire ». Il y a une certaine porosité entre les sujets et celle-ci doit être exploitée au maximum afin d'accélérer tant cette phase d'analyse préliminaire que l'intégration de nouvelles analyses.

Toute réalisation de notre méthodologie doit donc répondre à cette épineuse question de la formalisation de ces contraintes « métiers ». Dans le Chapitre 6 nous proposons une réponse à base d'un tableau multidimensionnel.

5.2.3 Informatisation des contraintes

Cette dernière étape doit informatiser les contraintes, c'est-à-dire les rendre manipulables par le reste de l'infrastructure.

À ce degré d'abstraction — nous ne cibons aucun type particulier d'environnement normé ni aucune implémentation —, il ne nous semble pas opportun de contraindre ou d'expliciter bien plus cette étape.

Ce qu'il faut seulement en retenir c'est que si la précédente étape exprimait les contraintes en termes purement « génériques », ne tenant absolument pas compte de l'informatisation du processus et de son automatisation, cette étape-ci doit procéder à cette intégration. Le format choisi doit pouvoir être employé dans la phase 1 décrite dans la section suivante.

Nous reviendrons sur cette informatisation bien plus en détail dans le Chapitre 6.

5.3 Phase 1 — Aide à la qualification

Pour mémoire, la deuxième phase consiste à réaliser les opérations présentées dans la Figure 5.4.

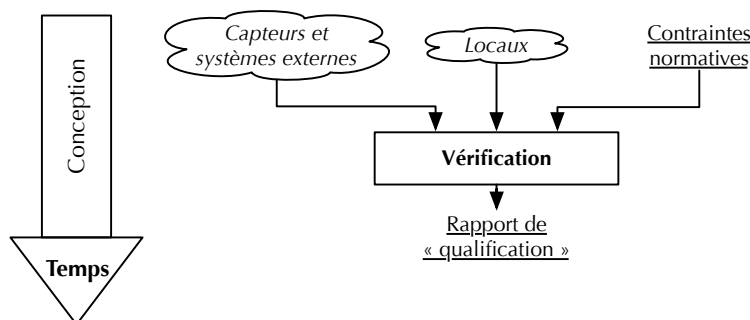


FIGURE 5.4 – Aide à la qualification : principe général

Nous allons maintenant décrire le déroulement de la phase d'aide à la qualification (illustrée dans la Figure 5.5).

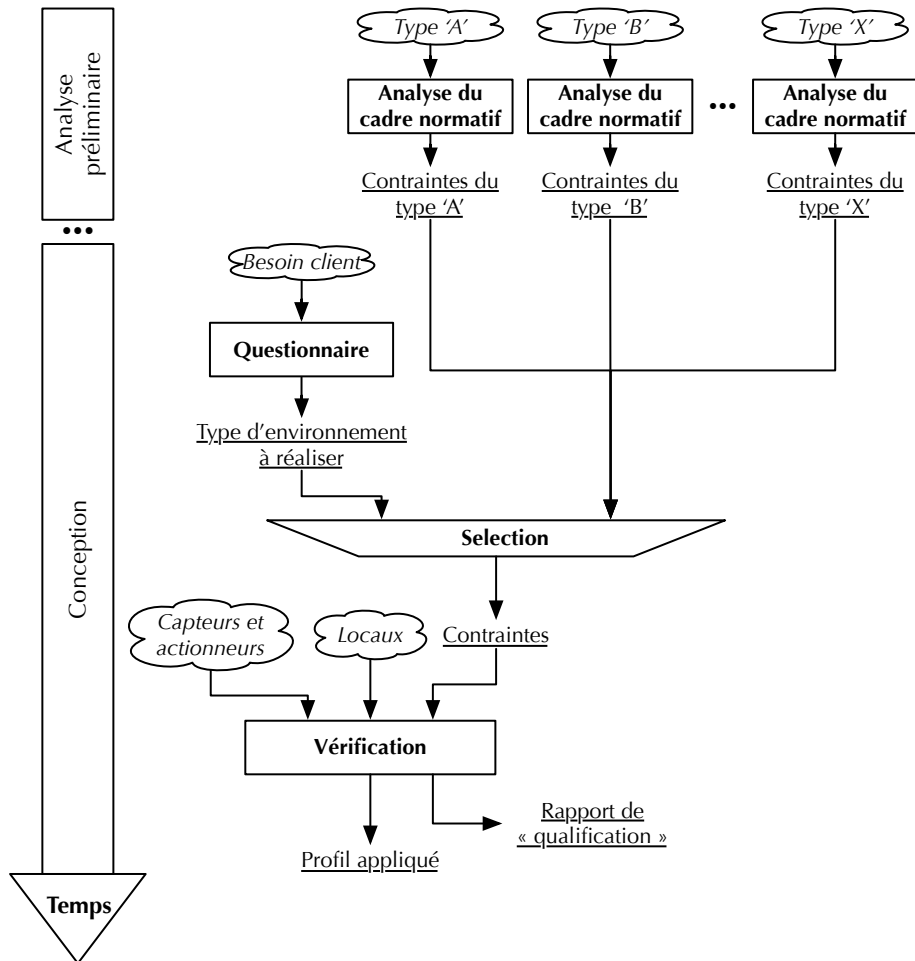


FIGURE 5.5 – Déroulement de l'aide à la qualification

Afin d'identifier le lien entre la qualification d'un projet particulier et l'analyse du type d'environnement « générique » (présenté précédemment), le lien avec la phase précédente a été conservé dans une version simplifiée pour faciliter la lecture du schéma.

5.3.1 Questionnaire et sélection des contraintes

Un point abordé plusieurs fois auparavant et sur lequel nous souhaitons revenir ici est la spécificité d'une analyse normative. Le type d'activité à mener dans l'environnement va très fortement conditionner sa vérification et celle-ci n'est que partiellement transposable d'un type à l'autre. Pour reprendre l'exemple précédemment cité, une « salle d'opération en chirurgie orthopédique » qualifiée ne permettra pas de réaliser des opérations digestives (ou tout du moins, ne sera pas qualifiée pour de telles activités). Et de manière identique, ce n'est pas parce que l'on sait qualifier pour la chirurgie orthopédique que l'on peut automatiquement faire de même pour de la chirurgie digestive.

Par conséquent, avant toute tentative de qualification d'un projet particulier, il faut assurer un vocabulaire commun entre l'utilisateur (qui va décrire son environnement) et l'outil de vérification qui doit en valider la conformité. Cette étape, nommée « élicitation des besoins » (« *requirements elicitation* » en anglais) en « Gestion des Connaissances » (et par corolaire en « Ingénierie des Exigences »), cherche à aider l'utilisateur à formaliser ses connaissances dans un langage commun [Zav97, ZC05].

Dans notre méthodologie, nous proposons deux opérations complémentaires. La première est d'associer aux contraintes issues de l'analyse du cadre normatif d'un type d'environnement son nom ou des étiquettes permettant de l'identifier facilement. La seconde est de réaliser parallèlement à l'analyse préliminaire un questionnaire qui servira précisément à orienter l'utilisateur vers le vocabulaire approprié. Ainsi un jeu de questions bien choisit comme « Quel type d'activité ? », « Pour quel type de patient ? » ou « Quel est le type d'organisme concerné, public ou privé ? » permet d'aiguiller les choses et d'assurer une compréhension mutuelle obligatoire pour la pertinence de l'analyse.

Ce questionnaire doit respecter la double contrainte d'être compréhensible par l'utilisateur devant modéliser le projet et suffisamment complet pour le guider vers son objectif « réel ». Le terme « réel » est impropre, le questionnaire vise plutôt à harmoniser le vocabulaire de l'utilisateur (qui peut adopter des termes propres à sa société ou son expérience personnelle) avec celui des experts « métiers » et autres juristes ayant procédé à l'analyse normative.

Une fois ce questionnaire rempli, il devient possible (et cohérent) de sélectionner les contraintes à vérifier.

5.3.2 Vérification

Au même titre que l'informatisation des contraintes dont il est le principal utilisateur (Section 5.2.3), l'outil de vérification dépend très fortement des choix technologiques effectués pour représenter le bâti, l'instrumentation et les contraintes. À ce titre, il y a un lien évident entre le formalisme adopté pour

l'expression individuelle des différentes composantes et la nature de l'outil de vérification. L'un ne va pas sans l'autre.

Au niveau d'abstraction auquel se place ce chapitre, il y a donc peu à dire sur le fonctionnement de l'outil de vérification.

En revanche, on peut discuter de la nature des artéfacts produits par cet outil. En effet, il nous apparaît pertinent de préciser ce qui était auparavant identifié grossièrement sous le vocable « Rapport de « qualification » ». Ce rapport possède en réalité deux composantes distinctes aux finalités différentes.

La première est le rapport de qualification en lui-même, c'est-à-dire l'analyse du projet avec les conseils, remarques et limitations identifiées. Ce document — devant être généré automatiquement — étudie l'application des contraintes sur la description faite de l'environnement. C'est le document finalisé permettant d'avoir le retour sur le projet courant tel que décrit dans le besoin industriel.

La seconde est ce que nous nommons le « profil appliqué ». Cette composante constitue le lien entre la vérification « projet » et la vérification « opérationnelle », elle comporte l'ensemble des contraintes ne pouvant être vérifiées à priori, mais au contraire requérant un travail ultérieur, lors de l'usage de l'environnement. Son formalisme, une fois de plus, dépend de choix d'implémentation, mais cette fois entre l'outil de vérification et la plateforme de traçabilité réalisant la qualification de l'usage.

Mais plus que la simple reprise des contraintes « opérationnelles », tel qu'elles peuvent être formulées dans la phase précédente, le profil appliqué représente une version « réexprimée » de toutes ces contraintes. C'est-à-dire une version compréhensible et manipulable directement par la plateforme de traçabilité à une vitesse permettant l'analyse en temps réel de l'environnement.

Cette « réingénierie » est rendue nécessaire par le niveau d'abstraction auquel se situent les contraintes normatives traditionnelles. En effet, une règle stricte comme « les deux portes d'un sas ne peuvent être ouvertes simultanément » n'a que peu de lien avec le monde « informatique » : qu'est-ce qu'un sas ? Qu'est-ce qu'une porte ? Comment savoir si une porte est ouverte ?

En utilisant les descriptions fournies par l'utilisateur, l'outil de vérification peut automatiser bon nombre de vérifications durant l'exploitation en réexprimant les contraintes durant la qualification.

Ainsi, dans des environnements médicaux, cela pourrait mener à la définition de contraintes explicites comme « la valeur du capteur de pression P1 doit toujours être supérieure à celle du capteur P2 » ou « les indicateurs d'ouverture O1 et O2 ne peuvent être actifs au même moment ». . . Évidemment, les informations requises doivent être fournies par l'utilisateur dans sa description des locaux et de l'instrumentation pour identifier précisément P1, P2, O1 et O2.

La génération du « profil appliqué » est un problème typique de « réingénierie des exigences » (l'une des grandes thématiques de l'Ingénierie des Exigences [Zav97,vL00]). Cela consiste à extraire automatiquement les paramètres pertinents d'une description abstraite pour produire une description concrète manipulable par la plateforme de traçabilité.

5.4 Phase 2 — Qualification de l'usage

Pour mémoire, la troisième phase consiste à réaliser les opérations présentées dans la Figure 5.6.

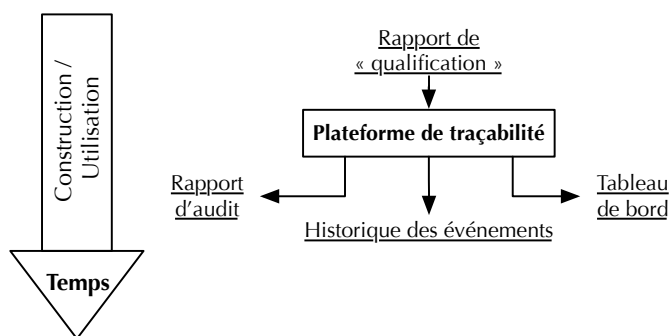


FIGURE 5.6 – Qualification de l'usage : principe général

Nous allons maintenant décrire le déroulement de la phase de qualification de l'usage (illustrée dans la Figure 5.7).

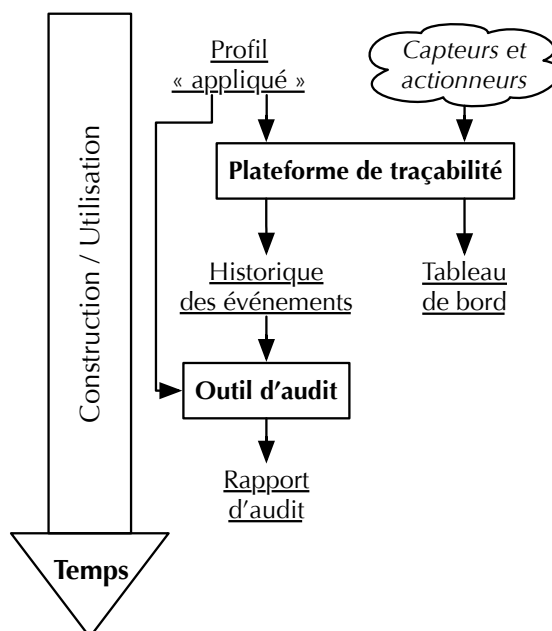


FIGURE 5.7 – Déroulement de la qualification de l'usage

5.4.1 Plateforme de traçabilité

La plateforme de traçabilité est l'outil responsable de la qualification en continu de l'environnement. Cet outil doit remplir deux fonctions « en continu », c'est-à-dire sans interruption de l'activité réalisée dans l'environnement normé et naturellement sans impacter le processus en cours.

La première fonction est la constitution d'un « tableau de bord » (*dashboard* en anglais) de l'environnement normé. Celui-ci doit fournir un retour rapide aux utilisateurs sur le fonctionnement de l'environnement, c'est-à-dire sur son respect des contraintes normatives fondamentales. Pour atteindre cet objectif, la plateforme de traçabilité doit vérifier « en temps réel » les règles exprimées dans le « profil appliqué » pour alerter les utilisateurs en cas d'anomalie de fonctionnement.

Cette fonction pose de très nombreuses questions dont les réponses pourront seulement être apportées par les réifications complètes de l'approche. En effet, plusieurs points dépendent des choix réalisés pour exprimer l'environnement et ses contraintes. Nous allons maintenant donner quelques exemples de questions importantes.

Les contraintes s'appliquent sur les données acquises et le profil doit donc décrire ces méthodes d'acquisition. Selon la forme choisie pour le profil appliqué, l'automatisation de la mise en place de cette acquisition sera très variable.

Si l'analyse normative a produit l'ensemble des contraintes à respecter, comment prioriser celles-ci ? Les priorités peuvent découler des normes à respecter ; mais, fréquemment, elles proviennent plutôt de l'expérience des experts à l'œuvre. Leur intervention étant très en amont de l'analyse d'un projet concret, ils doivent avoir intégré cette notion de priorité dès le départ.

Si une contrainte normative est « techniquement » importante, mais que celle-ci est constamment violée par les utilisateurs, sans aucun doute pour de très bonnes raisons, doit-on poursuivre l'émission d'alertes ? Comment faire comprendre aux utilisateurs les motivations de ces alertes ? Comment ne pas donner l'impression d'une « surveillance », d'un « flicage » des utilisateurs, mais plus d'une aide à la rigueur réglementaire. . .

Par exemple, le maintien d'une vitesse précise de l'air dans un bloc chirurgical est très important pour les filtres retenant les particules (ceux-ci garantissent leurs performances seulement pour une certaine vitesse d'écoulement de l'air), mais les praticiens peuvent être gênés par ce mouvement d'air surtout si les bouches d'aérations sont mal situées. Que faire dans de tel cas ? Faut-il déranger le praticien, quitte à nuire à la qualité de sa prestation et donc à la sécurité du patient, pour respecter les préconisations techniques ou inversement ?

De très nombreuses questions se posent et pour notre part, nous avons fait certains choix lors de nos expérimentations ; ceux-ci sont décrits dans les chapitres correspondants, plus loin dans cette thèse.

La seconde fonction est la constitution d'un historique de l'environnement. C'est simplement l'archivage de l'ensemble des données récupérées par la plateforme et utilisées (ou non) pour vérifier les règles contenues dans le profil appliqué.

L'historique des événements permettra par la suite de pratiquer les audits souhaités sur l'ensemble du cycle de vie de l'environnement.

5.4.2 Outil d'audit

L'outil d'audit est le dernier maillon de notre méthodologie. Il travaille sur l'historique des événements et l'ensemble des contraintes du profil « appliqué » pour identifier les anomalies et tenter d'isoler leur cause.

Cette démarche diffère de celle de la plateforme de traçabilité. Dans cette dernière, on dispose seulement des informations courantes pour évaluer un ensemble de règles devant nous permettre d'identifier des anomalies. Lors de l'audit, l'ensemble des événements de l'environnement est disponible et il s'agit plutôt d'un travail d'exploration des données pour retrouver la première occurrence d'une anomalie et tenter d'isoler l'origine de celle-ci ; voire, si la quantité de données observées le permet, tenter d'établir des corrélations entre les anomalies.

La finalité de cet audit est de superposer l'historique des anomalies constatées à celui des événements tiers, dépendant du type d'environnement normé concerné. Par exemple, cela permet de comparer l'historique des anomalies à la survenue d'infections nosocomiales.

5.5 Synthèse

Ce chapitre a passé en revue la version générique de notre méthodologie. Comme présentée, nous l'estimons adéquate pour la grande majorité des environnements normés. Nous avons pris garde à limiter au maximum les présupposés sur le type d'environnement normé manipulé.⁵

Tout au long de ce chapitre, nous avons détaillé l'architecture « simpliste » initialement proposée, la Figure 5.8 présente la version complète de notre méthodologie, intégration des Figure 5.3, 5.5 et 5.6.

Dans ce flot de conception général, nous avons fait le choix de nous concentrer sur des contributions précises (en l'occurrence, les deux premières phases) plutôt que d'aborder grossièrement l'ensemble de l'architecture. Cela nous a semblé plus raisonnable. Ainsi, nous avons concentré nos efforts sur la faisabilité d'une qualification d'un projet d'environnement normé.

Nous allons maintenant détailler l'incarnation concrète de cette architecture. Nous procéderons en deux temps ; nous débuterons par une présentation des algorithmes, concepts informatiques, et structure des données échangées (Chapitre 6 et 7) pour ensuite présenter une implémentation concrète pour le domaine médical (Part. III).

5. Bien que nous ayons souvent pris exemple sur le domaine « médical » afin de faciliter la compréhension du lecteur

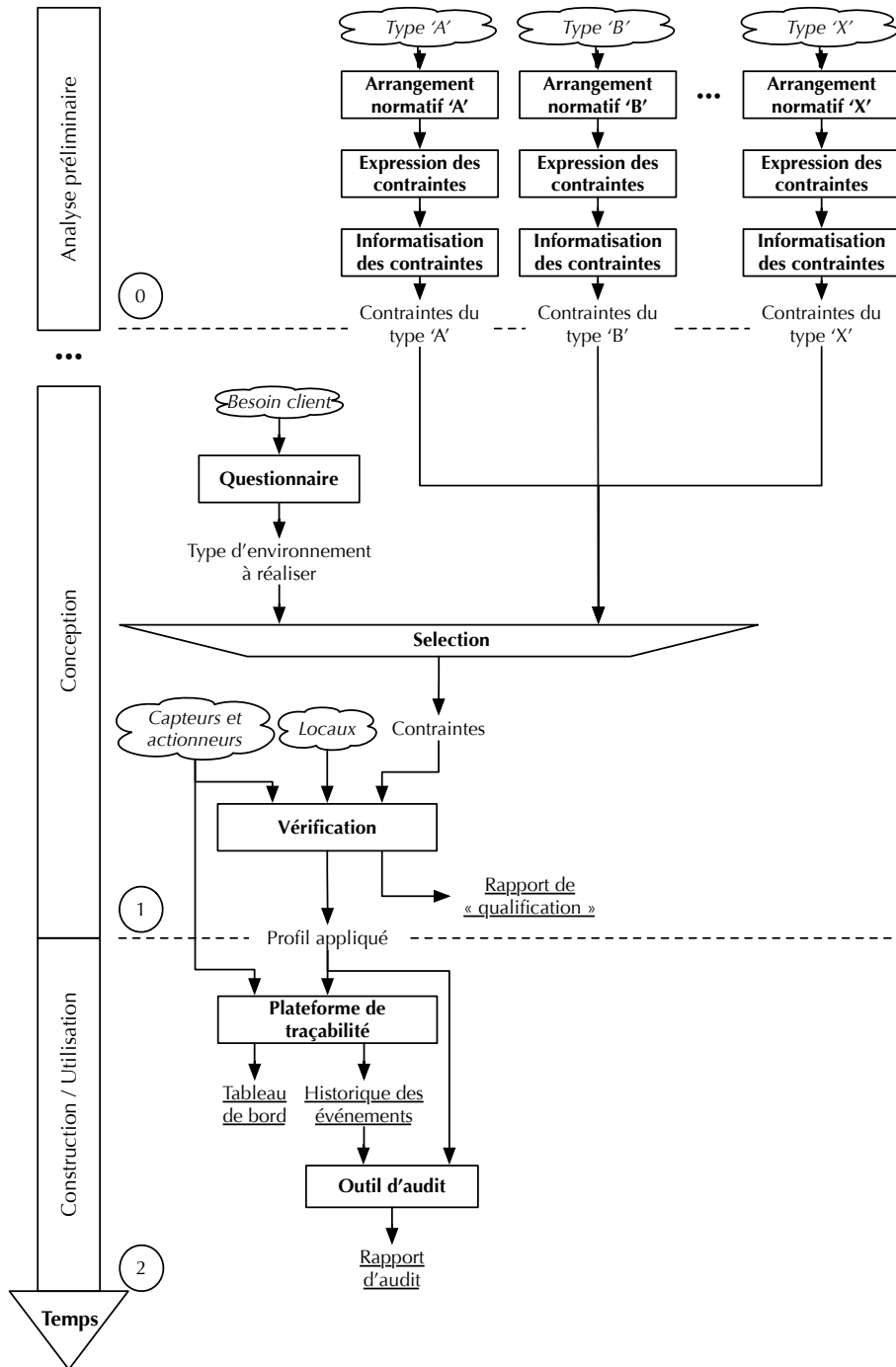


FIGURE 5.8 – Version détaillée de la méthodologie proposée

Analyse du cadre normatif

Dans le chapitre précédent, nous avons présenté notre méthodologie en termes génériques ; sans aucun choix technique ni aucune mise en œuvre concrète pour des cas réels d'environnements normés. Cela nous a permis d'explicitier notre approche et de la détailler suffisamment pour en justifier la mise en œuvre.

Nous avons fait le choix de cette présentation incrémentale (l'architecture générique puis les choix techniques et enfin la mise en œuvre concrète) afin de simplifier la reprise de ces travaux par d'autres chercheurs et d'autres publics. Ainsi, en séparant clairement ce qui est purement conceptuel (totalement générique) de ce qui tient plus de choix d'ingénierie ou du domaine d'activité concerné, nous permettons à quiconque de s'approprier et de débattre de nos choix. Cela nous permet également d'expérimenter différentes façons d'aborder les problématiques et de comparer ces mises en œuvre. Comme explicité dans le Chapitre 4, les contributions de ce doctorat sont tant dans l'approche proposée (explorant une piste relativement nouvelle de travail en Ingénierie des Exigences) que dans la manière de la mettre en œuvre (que nous allons maintenant détailler dans les chapitres suivants).

Dans ce chapitre, nous allons commencer par l'analyse du cadre normatif. Pour mémoire, cette phase est la suite de trois opérations : constitution de l'arrangement normatif, expression puis informatisation des contraintes.

6.1 Expression des contraintes

Pour mémoire, il n'existe actuellement aucune solution permettant d'automatiser la constitution de l'arrangement normatif ni aucun logiciel mature capable de fournir, pour un type d'environnement normé particulier, l'ensemble des textes concernés.

Nous débuterons donc par l'expression des contraintes, c'est-à-dire l'analyse de l'arrangement normatif obtenu.

6.1.1 Motivation

Pour mémoire, l'expression des contraintes doit remplir deux objectifs :

- assurer une compréhension commune des contraintes entre les rédacteurs (experts « métiers ») et les lecteurs (informaticiens) responsables de leur numérisation ;
- réaliser les analyses et exprimer les contraintes afin qu'une norme traitée pour un certain type d'environnement puisse être réutilisée dans d'autres circonstances.

L'approche nous ayant semblé la plus appropriée est la définition d'une typologie des contraintes normatives, sous la forme d'une grille de lecture particulière à employer lors de l'analyse d'une norme. Son objectif est de fournir une méthode de travail aux « experts métiers » afin qu'ils puissent classer les contraintes et permettre ainsi aux experts informatiques de les traduire plus simplement dans l'étape suivante. Cette grille doit répondre aux deux problématiques exprimées, en étant suffisamment explicites pour convenir à l'ensemble des parties et suffisamment robuste pour que l'analyse d'une norme dans le contexte d'un type d'environnement particulier puisse être utilisée dans un autre contexte.

D'autres manières d'aborder l'expression des contraintes ont été envisagées et étaient possibles au vu des spécifications formulées dans le Chapitre 5, cependant aucune ne nous a réellement convaincus.

En effet, l'expression des contraintes est une tâche à l'équilibre incertain, fragile... savant mélange entre des compétences très différentes. Demander un travail plus important de la part des experts « métiers » afin d'empiéter sur la phase de « numérisation » des contraintes, c'est pervertir le rôle de ceux-ci et commencer à placer des préalables importants sur les compétences des experts employés. À l'opposé, demander un travail plus important de la part des informaticiens, c'est prendre le risque d'une mauvaise compréhension des normes (ou des contraintes) par ceux-ci et d'entacher les analyses effectuées ultérieurement.

En revanche, l'utilisation d'une typologie des contraintes normatives permet la production de documents en langage naturel sur lesquels plusieurs itérations entre les experts métiers et les informaticiens sont possibles. Le cout de ces itérations (financier ou humain) est à nuancer ; l'analyse du type d'environnement normé est réalisée une seule fois et peut se permettre d'être relativement lourde, les gains potentiels justifiant très largement cet investissement.

6.1.2 Principe

La carte euristique présentée en Figure 6.1 donne le cadre d'analyse général à employer ainsi que les informations à fournir pour chaque norme et chaque contrainte.

Dans notre typologie, chaque norme est décrite par trois « composantes ». Nous allons les énumérer rapidement avant de les détailler dans les sections suivantes. Les exemples employés seront ceux du domaine médical pour faciliter la compréhension :

1. La première composante « Métadonnées » identifie clairement la norme. Cette dimension est primordiale dans notre approche de la numérisation des contraintes (voir Section 6.2) et elle doit être traitée avec extrêmement

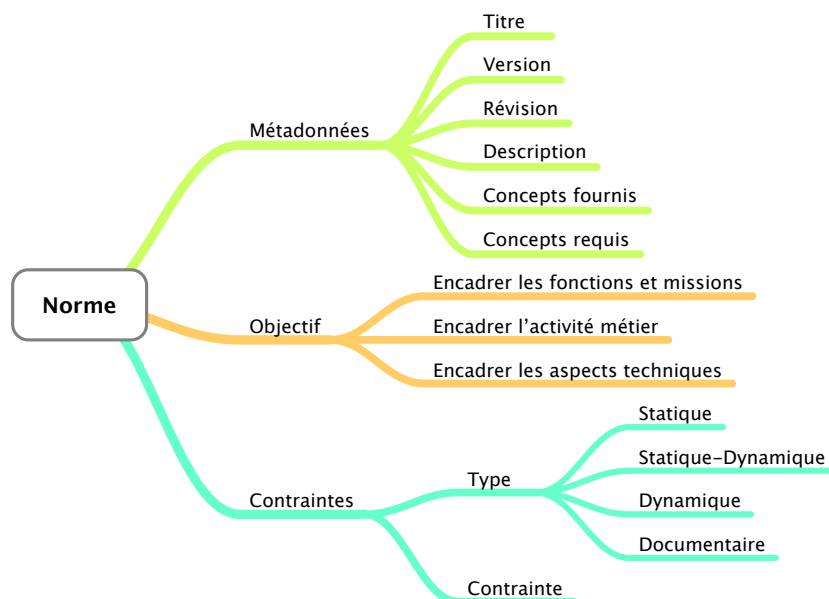


FIGURE 6.1 – Carte euristique pour l'analyse d'une norme

de précautions. Elle permettra de mettre en perspective les différentes analyses d'une même norme puis d'établir des liens de dépendances conceptuelles entre les documents.

2. La deuxième composante « Objectif » traite de l'objet poursuivi par la norme. Cette composante permet lors de la numérisation des contraintes de comprendre le lien existant entre la norme traitée et l'environnement à réaliser, mais également de mesurer le potentiel de « réutilisabilité » du document.
3. La dernière composante « Contraintes » contient l'ensemble des contraintes à vérifier pour la norme considérée. Pour chacune d'elle, on retiendra son type et naturellement la contrainte en elle-même. Le type permettra principalement aux informaticiens de savoir comment les représenter dans le système futur et quelle est la portée de la vérification : qualification de projet ou d'usage.

Ce prisme de lecture se veut évolutif au maximum, rien n'interdit de le faire évoluer et de le compléter à l'avenir.

6.1.3 Détails

Avant de détailler les différentes composantes, il faut bien comprendre les synergies devant exister entre les étapes, celles-ci sont inhérentes à notre approche.

Ainsi, cette grille de lecture s'articule avec notre approche de la numérisation des normes et des contraintes. Par conséquent, si toutes les informations demandées permettent de simplifier les relations entre experts « métiers » et

informaticiens, elles n'en demeurent pas moins toutes parties prenantes de la numérisation des normes et des contraintes abordée à la Section 6.2.

Identification de la norme

Dans un premier temps, les experts « métiers » doivent clairement identifier la norme pour permettre une numérisation et une intégration correcte dans la suite de notre mise en œuvre.

Cette identification demande aux experts de fournir, pour chaque texte, les informations suivantes :

Titre Titre complet du texte normatif. Ce nom doit correspondre à celui fourni par l'organisme éditant le document ¹.

Versión Version du document. Ce numéro de version doit correspondre à celui fourni par l'organisme éditant le document.

Dans la mesure où il n'y a pas de consensus sur le « versionnage » des normes et des textes règlementaires au niveau mondial et entre les disciplines, ce champ n'a pas une forme prédéfinie ².

Description Description du document. C'est un champ principalement contextuel permettant aux personnes consultant l'analyse de comprendre l'objectif général du document.

Concepts fournis Liste des concepts « fournis » par la norme.

Il est entendu par « concepts fournis », l'ensemble des notions « métiers » non triviales fournis par la norme et pouvant être manipulé par d'autres normes.

Concepts requis Liste des concepts « requis » pour la norme.

Il est entendu par « concepts requis », l'ensemble des notions « métiers » non triviales requis pour pouvoir appliquer la norme et vérifier ces contraintes.

Révision Ce champ permet d'identifier l'analyse d'une norme à l'aide d'un numéro de révision.

Les motivations de ce champ sont simples ; si le couple « nom et version » permet d'identifier une norme de manière unique, il faut nécessairement y adjoindre un numéro de révision pour isoler une « analyse » de la norme et ainsi tracer l'évolution des analyses.

La Figure 6.2 illustre ces métadonnées.

Le lecteur pourra s'interroger sur l'absence d'un champ indiquant le caractère « applicable » du texte. Dans la mesure où l'applicabilité d'un texte est rarement binaire (jamais/toujours), mais dépend plutôt d'une multitude de facteurs, nous avons choisi d'aborder cette question sans passer par un champ (cf. Chapitre 6.2).

Pour le bon déroulement de la numérisation des normes, il est primordial que le triplet « nom, version et révision » soit unique parmi l'ensemble des textes déjà traité, c'est ce qui permettra la numérisation puis une indexation et une recherche efficace des textes lors de leur exploitation.

1. Dans le cas où ce nom est peu usité dans les projets réels, charge à la mise en œuvre de notre méthodologie de fournir des « alias » pour les normes.

2. Pour illustration, lors de nos expérimentations, nous avons trouvé des révisions (1, 2, 3...), des années, des textes ou des assemblages (année et révision).



FIGURE 6.2 – Les métadonnées d’une norme sous forme graphique

Objectifs de la norme

Comme exposée, la composante « objectif » de la norme permet de comprendre son interaction avec l’environnement normé. Encadre-t-elle le type de processus pouvant se dérouler dans l’environnement ? Ajoute-t-elle des contraintes supplémentaires dans certaines situations ? Détaille-t-elle des principes techniques (méthodologie, outil, note de calcul...) ? Ce sont ces questions qui guideront l’« expert métier » dans ses choix d’objectifs et permettront ensuite de numériser correctement la norme concernée.

Lors de nos recherches et de nos expérimentations, seuls les trois types d’objectifs présentés dans la suite nous ont paru pertinents dans les cas « réels ». Nous ne nions pas la possible existence de types supplémentaires, mais cette sélection nous a donné pleinement satisfaction lors de nos expérimentations.

Fonctions et missions

Le premier objectif est l’encadrement des « fonctions et missions » remplies pas un environnement. Cette réglementation est primordiale, elle fournit un cadre légal aux activités menées et en borne l’étendue. Les documents présents dans cette catégorie sont spécifiques, cette dernière regroupe principalement des textes législatifs : codes, lois, décrets, etc.

Dans le domaine médical, il s’agit principalement du Code de la Santé Publique (CSP) [Min] accompagné de ses décrets. Par définition, les domaines d’activités indépendants de la santé n’emploieront pas ce code, mais le principe reste identique.

Dans notre méthodologie, ces textes servent à réaliser une partie du questionnaire pour la qualification d’un projet (cf. Section 5.3.1) et permettent de vérifier l’adéquation entre les objectifs de l’utilisateur et les possibilités offertes par le type d’environnement envisagé. Par corolaire, cela permet de définir le type d’environnement à réaliser en fonction de questions sur l’usage envisagé. La catégorisation permet ainsi de compiler rapidement la liste des normes nécessaires à la rédaction du questionnaire et de préparer la liste des questions pertinentes.

Activité « métier » et risques « spécifiques »

Le second objectif est l’encadrement de l’activité « métiers » ou des risques « spécifiques » des contraintes complémentaires induites par le contexte spécifique du projet (cytotoxiques, produits dangereux, produits inflammables, etc.).

Lors de nos expérimentations, nous avons constaté un lien direct entre les normes de première catégorie et celles-ci, les premières statuant sur ce qu’il est autorisé de faire et les secondes sur les conditions de cette exécution.

Dans le domaine médical, le CSP définit des activités et il délègue à des documents tiers — et notamment, à des guides de bonnes pratiques — les modalités exactes d'exploitation de l'environnement concerné.

Technique

La dernière catégorie contient les normes techniques (ISO, NF, etc.).

Ces normes sont traditionnellement mentionnées dans les normes « métiers » afin de spécifier ce qui n'était que conceptuel auparavant. Par exemple, les normes « métiers » parleront simplement de filtres « haute-performance » (ou High Efficiency Particulate Air (HEPA)) et c'est la norme NF EN 1822 qui va définir précisément ce qu'est un tel filtre (caractéristique, qualification, etc.).

Étant des documents techniques de références, ceux-ci sont très souvent réutilisés (c'est leur raison d'être) et dans des contextes très variés (toutes les normes mentionnant les filtres HEPA réfèrent la norme associée). L'intérêt d'une catégorie spéciale pour ces normes est d'isoler ces éléments à très forte réutilisabilité pour gagner du temps sur les analyses futures.

Types de contraintes

Une fois la norme décrite, les experts peuvent passer à son traitement et à l'expression des contraintes imposées par celle-ci.

Pour l'expression des contraintes, notre approche de la méthodologie générique demande deux informations. La première est la contrainte en elle-même, pour laquelle nous n'imposons aucun formalisme (le langage naturel nous semble le plus approprié), et la seconde est le « type » de la contrainte.

Le « type » se définit par la nature de l'élément à vérifier et permet aux informaticiens de connaître l'étape durant laquelle sera réalisée cette vérification (qualification de projets ou qualification de fonctionnements) et par corolaire, en fonction de la mise en œuvre effectuée, de quelle manière réaliser cette vérification.

Nous proposons quatre types de contraintes et nous rappelons une nouvelle fois l'aspect « empirique » de ce choix. En aucun cas, nous n'avons une garantie formelle sur l'exhaustivité de ces types et sur leur capacité à représenter tous les types de contraintes que l'on pourrait rencontrer dans la réalité. Mais cette sélection nous a donné parfaite satisfaction lors de nos expérimentations.

Les contraintes « statiques »

Ces contraintes ont principalement pour objet de vérifier la présence ou l'absence d'un élément dans l'environnement. Plus généralement, cela concerne l'ensemble des vérifications immuables ne dépendant pas d'un facteur lié à l'exploitation de l'environnement et pouvant évoluer avec celui-ci.

Il s'agit typiquement du genre de vérification à réaliser en « amont », c'est-à-dire lors de la qualification d'un projet d'environnement. La mise en œuvre de cette vérification sera extrêmement dépendante de la manière de modéliser l'environnement physique et son instrumentation³.

3. On entendra par instrumentation l'ensemble des capteurs, actionneurs et autres systèmes complexes présents dans un bâtiment pour en suivre et modifier le fonctionnement (sondes de températures, d'hygrométrie, contrôle d'accès, système de diffusion sonore...).

Pour illustrer ce type de contraintes, un bon exemple est le renouvellement d'air d'une pièce. Celui-ci demande un système de ventilation adéquat, mais également de vérifier la cohérence entre le taux de renouvellement d'air souhaité, les volumes des pièces, la puissance de soufflage employée, les surfaces d'introduction d'air, etc.

Les contraintes « dynamiques »

Ces contraintes ont pour objet la vérification d'éléments totalement liés à l'exploitation de l'environnement. Ici, on traite des contraintes seulement vérifiables *in situ* et strictement impossibles à valider sur le projet seulement.

Ce sont des vérifications « aval », dont la forme dépendra une nouvelle fois de la mise en œuvre. Dans notre cas, ce seront seulement des éléments à intégrer dans le profil appliqué.

Pour illustrer ce type de contraintes, on peut citer l'exemple de la certification dite « COFRAC » de capteurs déployés dans un environnement.

Le COmité FRnçais d'ACcréditation (COFRAC) est « l'unique instance chargée de délivrer les accréditations aux organismes intervenant dans l'évaluation de la conformité en France » [Min08].

Une fois accrédité par le COFRAC, un organisme pourra — à l'aide d'une série de tests normalisés — déclarer conforme à une norme particulière un capteur, lui permettant ainsi d'être utilisé dans des environnements réels requérant cette certification.

La certification du capteur est temporaire. La dérive du capteur étant inévitable, sa « précision » va diminuer et sa conformité devra être évaluée régulièrement. La fréquence de ces évaluations dépendra de nombreux facteurs, dont la nature du capteur, la norme concernée et le cas d'utilisation.

Ainsi, par définition, la certification n'est pas permanente et la conformité du capteur (et donc la pertinence des mesures effectuées) lors de l'usage de l'environnement ne peut être vérifiée « statiquement ».

Contraintes « documentaires »

Dernier type de contrainte, celui-ci concerne les obligations purement « documentaires » d'un type d'environnement.

Ce type de contrainte nous semble extrêmement difficile (si ce n'est impossible) à vérifier « sémantiquement », c'est-à-dire d'assurer l'adéquation totale entre les informations imposées par une norme donnée et les documents fournis lors de la description du projet. En revanche, vérifier seulement que la liste de documents produits est conforme à la liste de documents requis nous semble tout à fait réaliste.

Pour illustrer ce type de contraintes, on peut citer l'exemple des pharmacies hospitalières, où la direction est tenue de désigner un « pharmacien responsable » qui établira les « fiches de poste »⁴ des différents pharmaciens de l'unité. Une mise en œuvre de notre méthodologie pourra simplement établir une liste de documentations à produire avec parmi celles-ci une « déclaration du pharmacien

4. **Définition :** *La fiche de poste est un outil de gestion qui a un triple but : la clarification des rôles et des responsabilités ; l'amélioration du déroulement du recrutement dans le cas d'un poste à pourvoir ou de la mutation dans le cas d'un poste existant ; la possibilité de vérifier la bonne adéquation des résultats obtenus en fonction des objectifs définis [Les16].*

responsable » et des « fiches de postes ». Des travaux de recherches s'appuient déjà sur ce type d'approche de vérification de conformité par comparaison d'une liste de livrables attendus avec une liste de livrables produits [PW12].

Les contraintes « mixtes »

Certaines contraintes normatives ne sont pas strictement « statiques » ou « dynamiques » mais se traduisent plutôt par plusieurs contraintes techniques de différents types.

Pour illustrer ce type de contraintes, un bon exemple est celui du contrôle d'accès dans le domaine médical. La norme NF S 90-351 [Age13] impose, pour toutes les classes de risque supérieures ou égales à 3, un contrôle d'accès obligatoire de la zone à environnement maîtrisé. Il faut donc vérifier la présence d'un système de contrôle d'accès (contrainte « statique ») et générer la configuration de l'instrumentation (spécifications permettant d'accéder à ce système et de connaître son état courant) et les règles à vérifier (la spécification des sas est une contrainte « dynamique »).

Le lecteur pourra s'interroger sur la pertinence de cette catégorie s'il existe des catégories individuelles « statique », « dynamique » et « documentaire ». Cette apparente redondance permet aux experts de distinguer clairement les contraintes « pures » et celles pour lesquelles il y a un lien évident entre les différentes vérifications.

6.1.4 Conclusion

Dans cette section, nous avons détaillé notre mise en œuvre de l'expression des contraintes. Cette approche est fortement dépendante de notre manière d'aborder la transformation des contraintes (cf. Section 6.2) pour leur exploitation lors de la qualification d'un projet.

De manière synthétique, cette méthodologie demande aux experts « métiers » de documenter la norme (nom, version, concepts, type) et ensuite de lister les contraintes en les regroupant par catégories.

6.2 Transformation des contraintes

La dernière étape de l'analyse préalable avant la qualification d'un projet réel consiste en la « numérisation » des normes et leurs contraintes.

La mise en œuvre présentée dans ce chapitre poursuit notre philosophie générale, c'est-à-dire une numérisation agnostique de la représentation et de la vérification des contraintes, mais tirant profit de l'analyse des normes pour être capable de déterminer — pour un type particulier d'environnement normé⁵ —, les textes normatifs impliqués, pour quelle raison et dans quel objectif.

Ce chapitre ne traite pas (malgré les apparences) seulement de la transformation des contraintes, il aborde également la question des corpus normatifs. Ainsi, si la section précédente s'intéressait à l'analyse de normes prises séparément,

5. Avec toutes les précautions déjà mentionnées sur la définition d'un « type particulier d'environnement normé »

nous traitons ici de la numérisation de ces analyses et par voie de fait, à la mise en relation (en perspective) de ces analyses dans le cadre de corpus normatifs complets.

6.2.1 Principe

Lors de nos expérimentations sur la liste d'informations nécessaires pour identifier et comprendre les implications d'un texte normatif, une « mécanique normative » générale est apparue. C'est cette mécanique que nous évoquons dans ce chapitre et qui fait l'objet de cette section.

Nous retrouvons là l'une des principales contributions de ce doctorat, présenté dans la Section 1.3 de l'introduction de cette thèse : une approche originale de la représentation des corpus normatifs, de leur constitution et du moyen de les mettre facilement à disposition pour être employés, par exemple, à la vérification de projets.

Le corpus normatif : un graphe de normes ?

Lors de nos expérimentations, nous avons constaté de très grandes similarités entre les analyses de corpus normatifs.

Localement, on constate tout d'abord des similitudes dans les analyses des normes elles-mêmes (cela nous a permis de définir la typologie précédente). Les normes ont des fonctionnements internes similaires avec un certain nombre de contextes de vérification possibles et des informations requises identiques pour identifier les textes.

Globalement, avec un peu plus de recul, on constate des similitudes dans l'analyse de groupes de normes. Il existe une « articulation » similaire des normes dans chaque corpus. Cette articulation (cette « mécanique normative » pour reprendre notre précédente expression) est principalement due à la mise en relation de normes aux objectifs similaires (cf. Section 6.1.3) qui, une fois mises en perspective, forment un tout cohérent.

Dans la Figure 6.3, nous avons représenté la forme générale « classique » des contraintes pesant sur un environnement normé :

les normes « fonctions et missions » fournissent un cadre légal aux activités menées, et en bornent l'étendue. En encadrant les activités menées, ces normes font référence (implicitement ou explicitement) à d'autres documents devant règlementer ces activités.

Les normes « activités et risques spécifiques » contraignent, directement ou indirectement, l'environnement et son instrumentation.

Les normes « techniques » contraignent également directement l'environnement et son instrumentation.

En d'autres termes, pour une thématique donnée, le corpus normatif contient un premier ensemble de textes définissant le cadre légal d'activité et référençant ainsi un second ensemble de textes encadrant l'activité et ses dérivées. Pour mener à bien sa mission, ce dernier contraint l'environnement et son instrumentation⁶.

6. Ce dernier terme est à prendre dans son sens le plus générique : capteurs, actionneurs, systèmes tiers, etc.

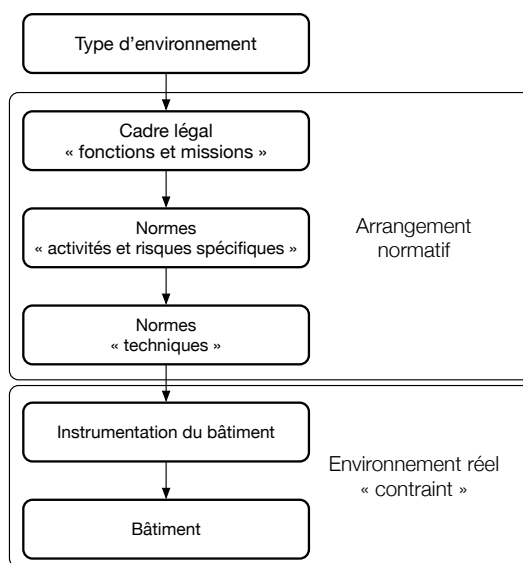


FIGURE 6.3 – Forme « générale » des contraintes pesant sur un environnement normé

Le contenu de ces ensembles étant fortement interconnecté par des références de concepts, etc. une modélisation sous forme de « graphe » nous a semblé une piste de réflexion intéressante et une manière originale de traiter la question normative. Dans ce graphe, chaque nœud est un texte normatif et les arêtes représentent les références implicites ou explicites entre les textes.

Un autre constat — important pour la suite de ce chapitre — est la nature particulière de ce graphe. Nous ne sommes manifestement pas en présence d’une structure quelconque. Ainsi, en incluant le « type d’environnement normé », « l’environnement physique » et « l’instrumentation », une structure très particulière se dessine. Le graphe a toujours les deux mêmes extrémités, le « type d’environnement » qui dépend de normes « fonctions et missions » et « l’environnement physique » dont dépendent les normes « activités, risques spécifiques et techniques » directement ou indirectement (par le biais d’autres nœuds).

De par le volume de texte existant pour chaque objectif, une sorte de « diamant » (losange) apparaît (cf. Figure 6.4).

Par corolaire, si nous considérons le graphe « non orienté », il existe alors-toujours — par définition — un (et certainement plusieurs) chemin entre le nœud représentant « l’environnement physique et son instrumentation » et celui représentant « le type d’environnement normé » considéré. Ce ou ces chemins peuvent être considérés comme la succession de textes règlementaires à respecter pour réaliser un environnement valide (ordonnée par les références mutuelles entre les textes). Une interprétation alternative est de considérer ce chemin comme la succession de contraintes et de concepts « métier » à intégrer dans l’environnement pour produire un environnement « valide ». Chaque norme constituant une « étape » supplémentaire vers la conformité. Cette observation est la clé de voute de notre contribution à la modélisation d’environnements

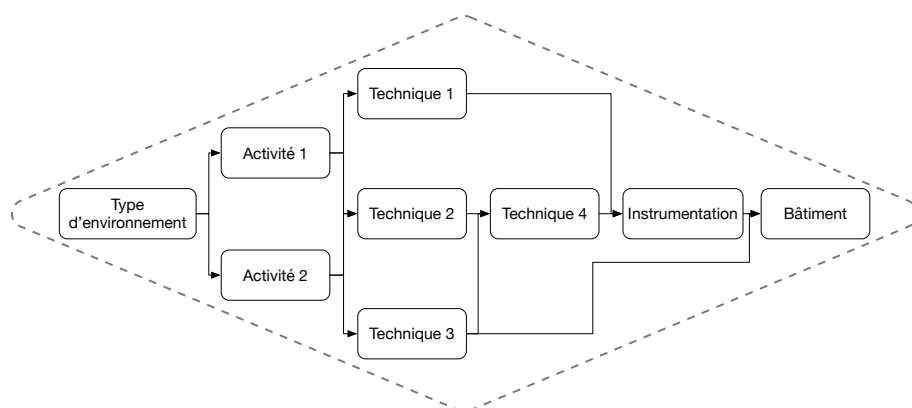


FIGURE 6.4 – Forme « générique » d'un graphe de normes

normés.

Nous allons maintenant examiner dans le détail notre mise en œuvre de la modélisation des normes, nous aurons alors totalement présenté notre contribution sur la représentation et la mise à disposition pour un type particulier d'environnement normé de son contexte normatif détaillé. Les sections suivantes seront réservées aux traitements effectués sur ce graphe pour le rendre manipulable par un outil d'aide à la qualification et nous pourrons alors aborder notre vision de la qualification de projet dans le Chapitre 7.

6.2.2 Les normes sous forme de graphe

Les métadonnées des normes permettent de produire un graphe illustrant les dépendances conceptuelles entre chaque texte.

Cette structure réticulaire nous a clairement semblé la plus adaptée et sa forme émerge naturellement des métadonnées acquises lors de l'analyse normative par les experts (comme décrit précédemment). De plus, cette approche constitue une approche relativement nouvelle et innovante de la question normative.

Ainsi, notre graphe de normes est constitué de nœuds représentant les textes normatifs et de relations représentant les liens conceptuels entre ceux-ci.

Plus précisément, chaque nœud est identifié par son triplet « nom, version et révision » (présenté en Section 6.1.3), il contient l'ensemble des métadonnées exprimées par les experts et ses relations sont issues des exigences conceptuelles identifiées. Ces relations désignent tant les liens conceptuels existant entre des normes que l'application de contraintes par la source sur la destination de la relation. Ces relations peuvent être « obligatoires » ou « facultatives » selon les résultats de l'analyse. Faire porter cette distinction par les relations et non par les nœuds permet d'autoriser des normes facultatives dans certains contextes et obligatoires dans d'autres. De cette manière, on évite des doublons artificiels dans la modélisation des normes, simplement présents pour représenter des contextes différents d'utilisation d'un texte.

Conformément à nos observations en introduction de ce chapitre, l'ajout dans ce graphe de deux nœuds supplémentaires, *instrumentation* et *bâtiment* (le premier dépendant du second), permet de représenter l'intégralité d'un type d'environnement. Par définition, nous obtenons ainsi un « graphe orienté acyclique » (*Directed Acyclic Graph (DAG)* en anglais) ayant pour racine le cas d'utilisation souhaité (c'est-à-dire, le type d'environnement normé) et pour feuille le bâtiment, avec de très nombreux chemins entre ces deux extrémités.

Nous pouvons même aller plus loin, en prenant un recul supplémentaire sur ce graphe. Chaque cas d'utilisation se révèle être un DAG à la racine différente, mais pouvant partager des nœuds ou des feuilles avec ceux issus d'analyses précédentes. Cela permet ainsi de maximiser la réutilisabilité des analyses et s'assurer qu'une norme « analysée » pourra resservir dans un très grand nombre de contextes sans pour autant nécessiter de reprise.

Ces DAG auront enfin une forme très particulière et similaire, ressemblant tous à des losanges : une racine (le cas d'utilisation), des nœuds intermédiaires nombreux (les différentes normes) et une unique feuille (le bâtiment) — car, par définition, les normes imposent des contraintes sur l'instrumentation, les locaux ou la documentation.

La Figure 6.5 illustre cet ensemble de DAG aux formes similaires.

6.2.3 Nettoyage des dépendances

La Figure 6.5 est capitale pour comprendre cette étape.

Tout d'abord, on y observe la concordance de notre modélisation avec nos attentes : des nœuds pour les différents types de normes, un ensemble de « diamants » pour chaque cas d'utilisation et enfin des nœuds partagés entre les différents scénarios, différant éventuellement par leur applicabilité légale.

Ensuite, on y voit le très grand nombre de relations et une certaine forme de redondances entre celles-ci. Par exemple, le nœud « Norme métier 2 » possède deux dépendances : « Instrumentation » et « Bâtiment » tandis que le nœud « Instrumentation » dépend de « Bâtiment ».

Appliquée dans un contexte réel, cette modélisation semble assez classique, c'est le cas typique d'une norme des contraintes sur le bâtiment et sur son instrumentation, un cas extrêmement courant.

Cependant, si la modélisation de la Figure 6.5 est très proche de la réalité et des dépendances conceptuelles exprimées dans les textes, elle n'est pas la plus pertinente dans notre mise en œuvre de la modélisation des normes.

En effet, si l'instrumentation dépend du bâtiment (parce qu'elle s'applique à des éléments de celui-ci) alors les concepts du bâtiment sont en quelque sorte « enrichis » par de nouvelles contraintes et de nouvelles possibilités (les capteurs, actionneurs, systèmes tiers, etc.). Par conséquent, nos relations possèdent une forme de transitivité : si $A \rightarrow B \rightarrow C$ alors $A \rightarrow C$ par l'intermédiaire de B. Il n'est donc pas nécessaire de modéliser deux relations, mais une seule, la seconde ($A \rightarrow C$) étant implicite.

Une seule précaution s'impose, tenir compte de la nature de la relation : $A \rightarrow B \rightarrow_{\text{optionnelle}} C$ et $A \rightarrow C$ ne sont pas redondants.

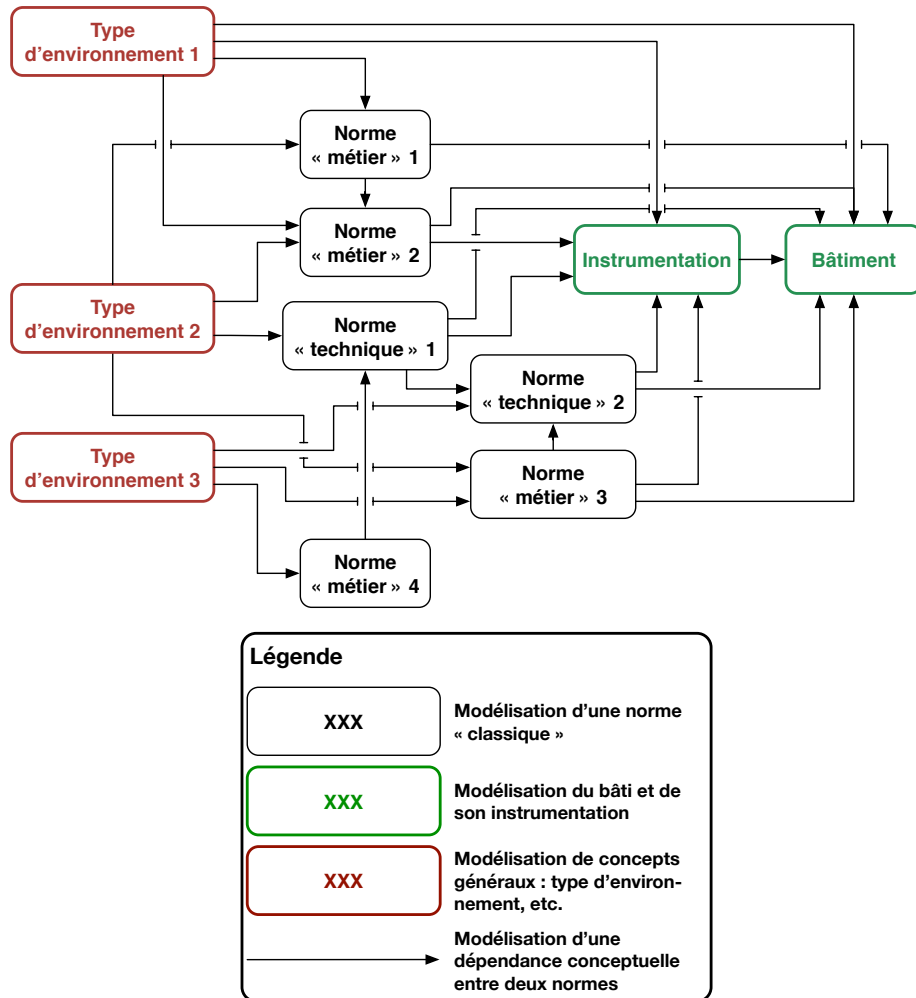


FIGURE 6.5 – Illustration de la modélisation réticulaire de normes

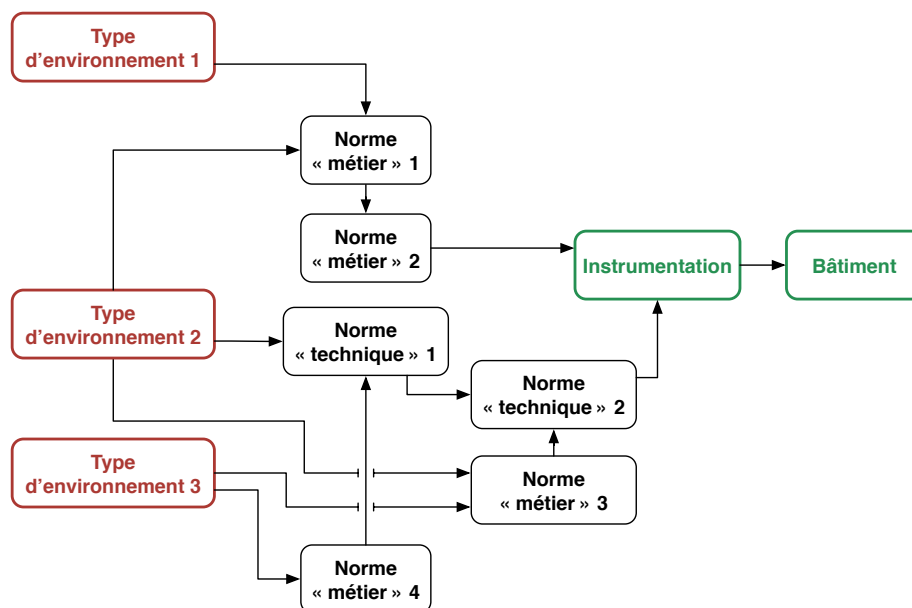


FIGURE 6.6 – Illustration d’une modélisation « filtrée » de ses redondances

Si nous appliquons ce filtrage à la Figure 6.5, nous obtenons la Figure 6.6.

Nous constatons une simplification extrême de notre graphe. Les diamants se sont réduits et, si nous avons toujours plusieurs racines correspondant aux différents types d’environnements connus, il n’y a plus qu’une seule feuille « Bâtiment » précédée d’« Instrumentation ».

Si le bâtiment est certainement toujours une « feuille » du graphe, celui-ci étant la « brique de base » de l’environnement normé⁷, il pourrait théoriquement être possible que l’instrumentation ne soit pas le seul nœud dépendant du nœud « bâtiment ». Ce serait le cas de normes encadrant exclusivement le bâti (et absolument pas son instrumentation).

6.2.4 Inversion des dépendances

Cette étape-ci est très simple, elle inverse seulement les relations présentes dans la Figure 6.6 afin d’en faciliter la lecture. Cette opération permet de remettre les relations dans le sens de leur traitement, ainsi on débute par une représentation de l’environnement « physique » (le nœud « Bâtiment »), puis de l’instrumentation (le nœud « Instrumentation ») et ensuite de chaque norme par ordre de dépendance. Comme nous l’avons exprimé précédemment, cela permet de travailler par étapes successives, chacune d’elle apportant son lot de concepts et de contraintes supplémentaires, convergeant ainsi vers un environnement conforme.

7. Il ne peut par définition exister un concept plus simple que le bâtiment. On pourra à la rigueur subdiviser le bâtiment en sous-concepts (murs, mobiliers, etc.), mais cela revient au même.

Cette étape nous permet d'obtenir la Figure 6.7, dernière étape avant la mise en forme finale. Sans aucune surprise, nous obtenons un DAG ayant pour seule racine la représentation du bâtiment et pour feuilles les différents types d'environnements normés étudiés.

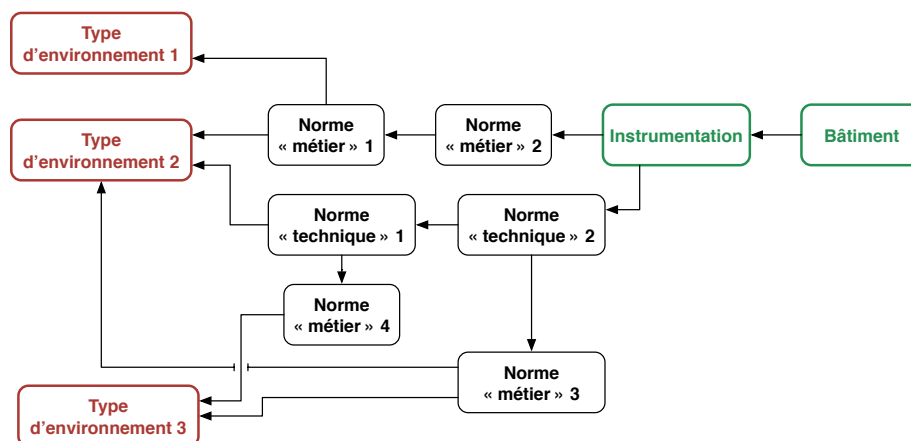


FIGURE 6.7 – Illustration d'une modélisation « inversée »

Pour cette dernière étape, nous considérons séparément chaque type d'environnement afin d'aboutir à l'étape de sélection présentée dans la Figure 5.8.

Techniquement, le passage d'une représentation « globale » incluant tous les types d'environnements à une représentation « spécifique » contenant seulement le type étudié est plutôt simple. C'est simplement l'extraire du sous-graphe composé de l'ensemble des « chemins » partant de la racine « bâtiment » et aboutissant au type d'environnement étudié.

Dans la section suivante, nous considérons seulement des sous-graphes présentant la modélisation de l'arrangement normatif de types particuliers d'environnements normés.

6.2.5 « Linéarisation » des contraintes

La dernière étape est la « linéarisation » des contraintes, c'est-à-dire « l'aplatissement » du graphe de norme afin d'obtenir une liste d'étapes sans ambiguïté à appliquer successivement pour passer d'un bâtiment « brut » à un environnement normé valide.

Traitement des types d'environnement « simples »

Pour comprendre cette opération, prenons un exemple simple, celui de la modélisation du « Type d'environnement 1 » dans la Figure 6.6.

Ce type factice est relativement simple, il requiert l'application de la « Norme métier 1 » ; celle-ci impose des contraintes sur des concepts de la « Norme métier 2 », qui à son tour contraint l'instrumentation.

Une fois les relations remises dans le « bon sens » (Figure 6.7), nous obtenons logiquement un environnement validable en 4 étapes : modélisation du bâti, ajout de l'instrumentation, validation de contraintes ou expression de nouveaux concepts par la « Norme métier 1 » puis par la « Norme métier 2 ». Enfin, nous arrivons à la feuille « Type d'environnement 1 » n'exprimant certainement pas de contraintes, mais indiquant la fin de la représentation et de la validation du projet. Tout cela est résumé dans la Figure 6.8



FIGURE 6.8 – Exemple : Type d'environnement 1 — graphe normatif

Dans cette dernière figure, nous remarquons l'inutilité de la feuille « Type d'environnement normé 1 ». En effet, une fois le graphe inversé et le sous-graphe pertinent extrait, maintenir cette feuille est inutile, nous pouvons nous concentrer sur les contraintes et les concepts. Cela nous permet d'obtenir le schéma de la Figure 6.9.



FIGURE 6.9 – Exemple : Type d'environnement 1 — graphe normatif filtré

Ce cas (trop) simple, s'il nous a permis d'illustrer la suppression de la feuille représentant le type d'environnement normé, ne permet pas d'appréhender l'ensemble de la « linéarisation » des contraintes. Pour ce faire, le « Type d'environnement 2 » sera bien plus intéressant.

Traitement des types d'environnements complexes

Dans la Figure 6.10, nous rappelons le sous-graphe composant le deuxième type d'environnement modélisé.

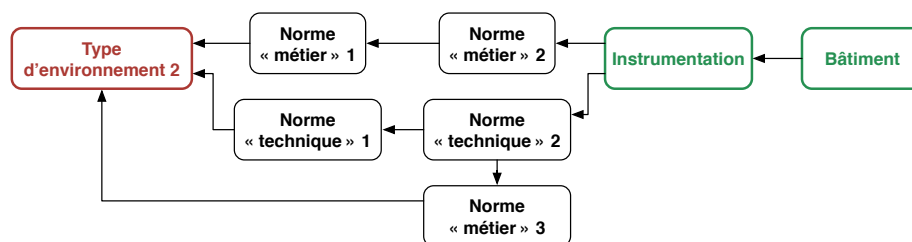


FIGURE 6.10 – Exemple : Type d'environnement 2 — graphe normatif

Nous constatons un accroissement de la complexité. Cette fois, le type d'environnement requiert le respect de plusieurs normes, créant ainsi un véritable

« arbre » normatif entre la racine « Bâtiment » et la feuille « Type d'environnement 2 ». Cette structure arborescente est d'autant plus visible lorsque nous supprimons le type d'environnement comme précédemment (Figure 6.11).

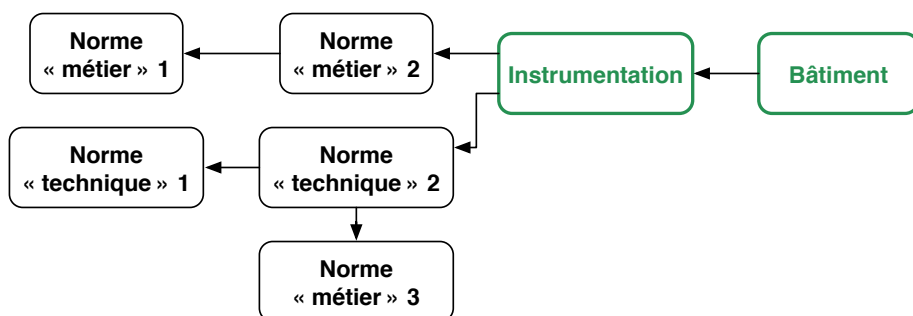


FIGURE 6.11 – Exemple : Type d'environnement 2 — graphe normatif filtré

À cet instant, nous constatons l'émergence d'un DAG avec une relation d'ordre partiel (représenté par les relations) sur laquelle nous souhaitons effectuer un tri (dit « topologique ») pour nous retrouver avec un graphe comme dans l'exemple précédent.

Tri topologique

Dans l'importante littérature composant le sujet [BJG09, HBF12], l'une des définitions les plus précises et accessible nous semble celle-ci :

Définition 2 *En théorie des graphes, et plus spécialement en algorithmique des graphes, un tri topologique d'un graphe acyclique orienté est une extension linéaire de l'ordre partiel sur les sommets déterminés par les arêtes, c'est-à-dire un ordre total compatible avec ce dernier. En d'autres termes, il s'agit d'un ordre de visite des sommets, tel qu'un sommet soit toujours visité avant ses successeurs.*

Nous attirons le lecteur sur la dernière phrase, parfaite métaphore de notre approche de la vérification de contraintes, appliquée aux environnements normés.

Enfin, nous rappelons que le passage d'un ordre partiel en un ordre total ne peut, mis à part les cas simplistes, donner lieu à une unique solution⁸. Ainsi, il existe naturellement plusieurs ordres totaux pour un ordre partiel initial, la seule contrainte étant le respect des ordres de succession.

En terme algorithmique, il existe de très nombreuses solutions pour réaliser des tris topologiques, c'est un sujet largement abordé par la communauté scientifique depuis les premiers travaux sur les graphes dans les années 60 [Kah62].

L'algorithme 1 est issu de ces premiers travaux et, appliqué à notre structure particulière, constitue une approche simple, mais compréhensible à la réalisation d'un tri topologique.

8. NdR : de par la définition même des ordres partiels

Algorithme 1 : Tri topologique utilisant un parcours en largeur

Entrées : Le DAG de normes avec un ordre partiel sur les sommets
Sorties : Un ordre total compatible avec l'ordre partiel en entrée

- 1 ListeTriée \leftarrow []
- 2 ProchainsNœuds \leftarrow [Bâtiment]
- 3 **tant que** *ProchainsNœuds n'est pas vide* **faire**
- 4 n, ProchainsNœuds \leftarrow ProchainsNœuds[0], ProchainsNœuds[1 :]
- 5 ListeTriée \leftarrow ListeTriée \cup n
- 6 **pour tout** *m avec une relation e de* $n \rightarrow m$ **faire**
- 7 Enlever la relation e du graphe
- 8 **si** *m n'a plus d'autre relation entrante* **alors**
- 9 ProchainsNœuds \leftarrow S \cup m

10 **retourner** ListeTriée

Cet algorithme correspond à ce que l'on nomme un parcours en « largeur » d'un graphe (Figure 6.12, les numéros des nœuds symbolisant leur ordre de traitement).

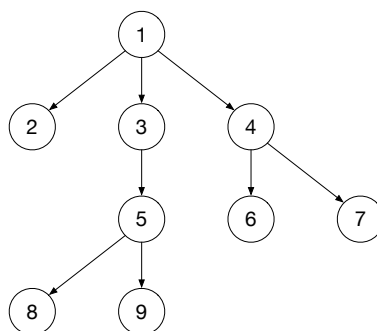


FIGURE 6.12 – Illustration d'un parcours de graph en largeur

Cela consiste à traiter l'ensemble des nœuds d'un « niveau »⁹ avant d'aborder ceux du niveau suivant.

Dans l'algorithme 1, ce comportement est matérialisé par la ligne 9 qui insère les nouveaux nœuds à traiter en fin de liste plutôt qu'en début, entraînant mécaniquement le traitement de tous les nœuds d'un niveau avant de pouvoir passer au suivant.

Si nous appliquons ce principe de tri en largeur au « Type d'environnement 2 » (Figure 6.11), nous obtenons la liste de nœuds illustrée dans la Figure 6.13.

Si ce tri est tout à fait valide d'un point de vue logique (l'ensemble des contraintes est bien respecté), il en va tout autrement de son intérêt « métier ». En effet, celui-ci est loin d'être évident. . . chaque nœud représente une norme

9. NdR : un « niveau » de DAG correspond à l'ensemble de nœuds de même profondeur. Par exemple, les nœuds de niveau « 1 » sont ceux directement rattachés à la racine.

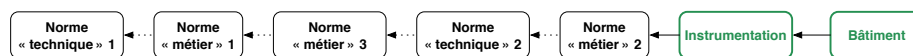


FIGURE 6.13 – Exemple : Type d’environnement 2 — graphe normatif trié en largeur

(c.-à-d., un ensemble de concepts « métiers » précis) et la présence d’un nœud indique que le type d’environnement étudié impose le respect de cette norme et (éventuellement) y applique certaines contraintes. Ainsi, « sauter » de norme en norme soit de concept en concept, pour revenir ensuite aux premières contraintes, nous semble des plus fastidieux.

Face à ce parcours « en largeur », il existe son opposé, le parcours « en profondeur ». Si dans le premier, l’objectif est de traiter l’ensemble des nœuds d’un niveau avant de passer au suivant ; dans le second, c’est de traiter les nœuds d’une branche — quel que soit leur niveau — avant de passer à la suivante, en veillant, naturellement, à respecter les dépendances.

Nous obtenons alors l’algorithme 2, très similaire au premier, il diffère seulement par la ligne 9. Dans celui-ci, plutôt qu’ajouter les nouveaux nœuds disponibles en fin de liste (après ceux du niveau courant), nous les ajoutons en début de liste pour les traiter en priorité.

Algorithme 2 : Tri topologique utilisant un parcours en profondeur

Entrées : Le DAG de normes avec un ordre partiel sur les sommets

Sorties : Un ordre total compatible avec l’ordre partiel en entrée

```

1 ListeTriée ← [ ]
2 ProchainsNœuds ← [ Bâtiment ]
3 tant que ProchainsNœuds n'est pas vide faire
4   n, ProchainsNœuds ← ProchainsNœuds[0], ProchainsNœuds[1 :]
5   ListeTriée ← ListeTriée ∪ n
6   pour tout m avec une relation e de n → m faire
7     Enlever la relation e du graphe
8     si m n'a plus d'autre relation entrante alors
9       ProchainsNœuds ← m ∪ S
10 retourner ListeTriée
```

En comparaison avec l’algorithme « en largeur » (Figure 6.12), nous obtenons maintenant le parcours présenté dans la Figure 6.14. Le nœud 5 y est devenu le nœud 4 et son sous-arbre est, par conséquent, traité en priorité.

Si nous appliquons ce principe de tri en profondeur au « Type d’environnement 2 » (Figure 6.11), nous obtenons la liste de nœuds illustrée dans la Figure 6.15.

Objectivement, les tris « naïfs » sont très loin d’être optimaux du point de vue du « métier ». En effet, prendre simplement l’ordre des branches tel qu’il est bâti

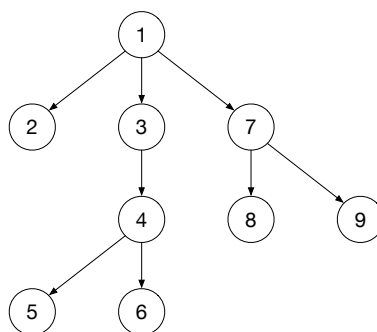


FIGURE 6.14 – Illustration d'un parcours de graphe en profondeur

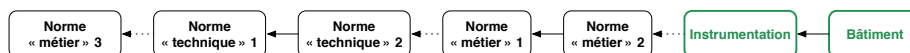


FIGURE 6.15 – Exemple : Type d'environnement 2 — graphe normatif trié en profondeur

par la modélisation réticulaire (donc, simplement lié à des ordres d'insertion ou des modalités de gestion) nous semble pour le moins « simpliste ». Naturellement, cela permettra de valider un environnement, nous ne le nions pas, mais les étapes de validation manqueront de « cohérence » et de « perspective » d'un point de vue d'expert.

Il conviendrait plutôt de piloter le parcours des branches en prenant en compte des spécificités « métiers » : la criticité [Fra03] des sujets traités dans les normes, l'importance relative des différentes « branches » d'un type d'environnement, la valeur ajoutée « conceptuelle » des normes ou encore des statistiques de respects des contraintes afin de travailler prioritairement sur les sujets présentant le plus souvent des erreurs. Ces choix auront traditionnellement peu d'impact sur l'algorithme de tri, mais énormément sur la pertinence « industrielle » de la solution. Ces compromis ne correspondent pas vraiment à des thématiques de recherche, mais plutôt à des travaux d'ingénierie.

6.3 Conclusion

Nous voici arrivés au terme de la présentation de notre contribution sur la modélisation des corpus normatifs et nous en présenterons la réification dans la partie de cette thèse dédiée à nos expérimentations (Partie III).

Nous disposons maintenant pour chaque type d'environnement analysé d'une suite d'étapes permettant de passer d'un bâti « brut » à un environnement normé valide (tout au moins pour l'ensemble des contraintes vérifiable « statiquement »). Le questionnaire évoqué dans la Figure 5.8 permettra de sélectionner la bonne liste d'étapes en fonction du besoin « client » et s'assurer ainsi d'un vocabulaire commun.

En situation réelle, notre approche de la modélisation de corpus normatif s'emploie de deux manières différentes : comme « base de données » des normes (la piste empruntée lors de nos expérimentations) et comme une liste de scénarios pour la réalisation d'environnements normés.

Aide à la qualification

Le chapitre précédent s'est concentré sur la modélisation réticulaire des arrangements normatifs encadrant la conception et la réalisation d'environnements normés (phase '0' de notre processus global).

Cette proposition présentait une modélisation sous forme de graphe, où chaque « norme » (à prendre dans le sens le plus large possible du terme) était reliée à l'ensemble des autres textes (ou concepts) requis pour sa compréhension et son application. Une certaine souplesse sur la nature de ces relations permettait de traiter le cas des normes seulement recommandées (mais non applicables).

In fine, nous obtenions par l'application d'un tri topologique sur les ordres partiels contenus dans le graphe, une suite de nœuds (et donc de normes) à traiter successivement pour passer d'un environnement « brut » à un environnement normé valide.

L'objectif est, bien évidemment, de stopper la modélisation d'un projet dès les premiers signes de non-conformité et ainsi pouvoir indiquer à l'utilisateur le point (et plus spécifiquement la contrainte) précis de « décrochage ». L'utilisateur doit pouvoir outrepasser cette alerte, mais ce choix est en lui-même une donnée « métier » critique, car il doit figurer dans les rapports rédigés lors de la réalisation d'un environnement normé.

Nous nous attelons maintenant à la description de notre proposition de mise en œuvre de l'outil d'aide à la qualification d'un projet précis d'environnement normé.

7.1 Principe

Cette présentation suivra le déroulement de la phase '1' du schéma 5.8, un questionnaire de sélection puis l'outil d'aide à la qualification.

L'objectif du questionnaire est double ; il s'agit dans un premier temps de

s'assurer de la concordance des vocabulaires¹ et, ensuite, de la présence d'une suite connue d'étapes pour le type d'environnement visé. En d'autres termes, « trouvons un discours commun et assurons-nous de disposer d'une méthodologie de vérification pour cet objectif ».

L'objectif de l'outil d'aide à la qualification est évident, c'est à lui de dérouler les étapes permettant à l'utilisateur de représenter son projet et de le vérifier. En revanche, sa mise en œuvre est nettement moins triviale, nous verrons ainsi que cet outil est composé de plusieurs parties ; c'est à la fois un « chef d'orchestre », qui doit suivre les étapes issues de la représentation de l'arrangement normatif, mais également l'outillage permettant à l'utilisateur de modéliser son projet lors de chacune des étapes et de vérifier le respect des contraintes règlementaires.

Nous débutons par le questionnaire.

7.2 Questionnaire

La suite de questions vise à converger vers un type précis d'environnement normé, potentiellement assorti de spécificités complémentaires ajoutant des contraintes au projet « général ». L'absence de convergence indiquant l'absence de connaissance préalable du type d'environnement visé et par conséquent l'impossibilité d'appliquer notre méthode.

7.2.1 Création

Tout d'abord, il est important de le rappeler, le questionnaire est issu de l'analyse préalable (Chapitre 6), il est bâti en fonction des contraintes observées et des textes connus. Son objectif est d'associer à des questions (ou des suites de questions), un type d'environnement ou une spécificité complémentaire. Ces spécificités dépendront du domaine « métier » général concerné : médical, nucléaire, nanotechnologies, etc.

7.2.2 Forme

Dans l'absolu, nous n'imposons pas de forme au questionnaire. Nous avons cependant privilégié dans nos travaux la forme de carte euristique où chaque nœud est une question, les différentes branches représentant les différentes réponses possibles. L'absence d'une réponse adéquate est le signe d'un problème de communication (vocabulaire incompatible?) ou de l'absence de traitement en amont du problème concerné.

Détaillons maintenant la structure générale d'une carte euristique (Figure 7.1).

La première question sert à trier parmi les « grandes thématiques » du domaine « métier » concerné. Par exemple, dans le domaine « médical », on commencera par trier les activités de chirurgie, des activités de pharmacologie, etc.

1. Une étape à rapprocher de « l'élicitation » en Ingénierie des Exigences. Pour mémoire, l'élimination est la phase d'expression des besoins à la fois fonctionnels et non fonctionnels du système à concevoir.

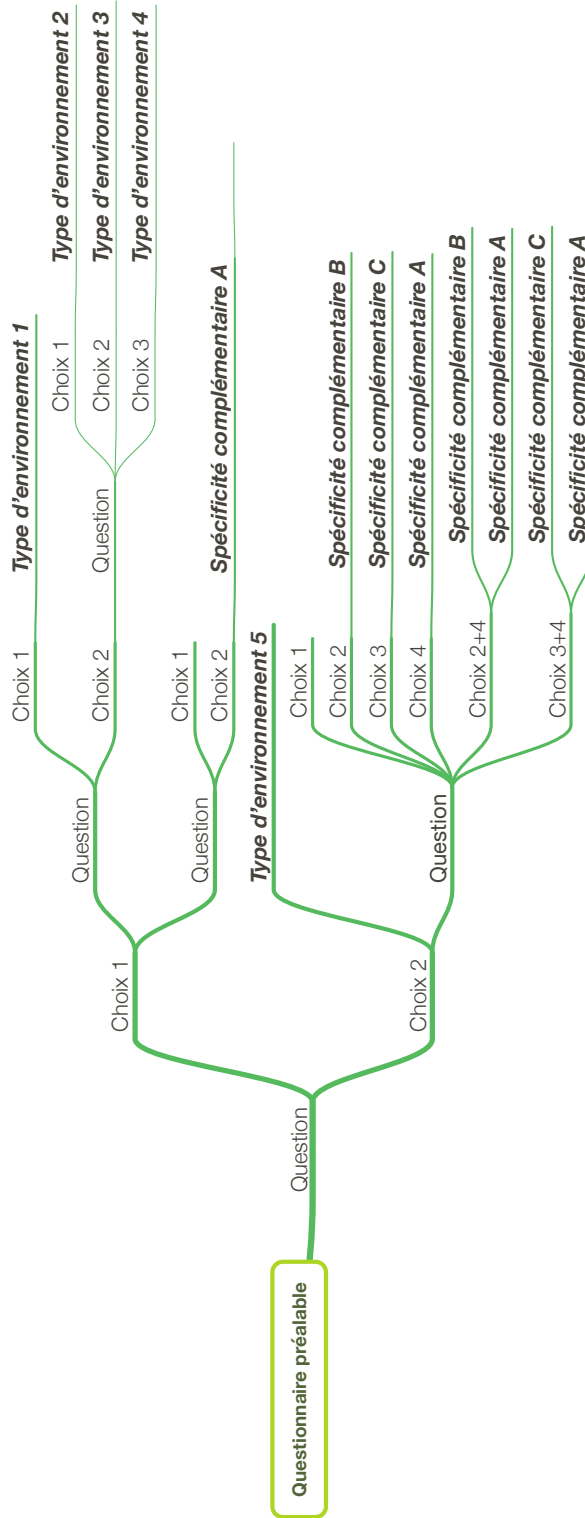


FIGURE 7.1 – Structure générale d'un « questionnaire »

Toutes les questions suivantes serviront ensuite à identifier le type d'environnement précis visé par l'utilisateur ; par essence, on débutera par des questions « très générales » pour aboutir à des questions extrêmement précises (d'où l'image de l'entonnoir employé précédemment).

Le Chapitre 10 présente un exemple complet de questionnaire dans le domaine médical.

7.2.3 Usage

Étudions maintenant l'usage d'un questionnaire à l'aide de celui fourni en Figure 7.1. Cet exemple, relativement simple, est représentatif d'un questionnaire tel que nous le proposons.

Dans le cas d'un « choix 1 » à la première question, deux nouvelles questions sont à traiter :

1. La première permettra, selon les réponses suivantes, de distinguer 4 types d'environnements différents. On remarquera la nécessité d'une question supplémentaire pour distinguer parmi les trois derniers types, et, par la même occasion, la possibilité de traiter des questions « ouvertes » pour lesquelles il existe plus de deux réponses autorisées.
2. La seconde conduira à — si l'utilisateur répond à la seconde question par « choix 2 » — une « spécificité complémentaire A » pour l'environnement à réaliser.

En revanche, avec le choix 2, le type d'environnement est déjà connu et la dernière question sert à identifier certaines « spécificités complémentaires » possibles. En fonction de la réponse effectuée, des « spécificités complémentaires » pourront être ajoutées au projet.

En résumé, le questionnaire fonctionne de la manière suivante :

- Toutes les questions issues d'un choix doivent être traitées séquentiellement (comme dans le cas du « choix 1 »).
- Les choix sont mutuellement exclusifs, c'est-à-dire qu'une seule branche peut-être empruntée pour répondre aux questions.

Dans les cas où certaines des possibilités pourraient être cumulées (comme dans le cas du « choix 2 »), il faut prévoir des branches supplémentaires incluant les différentes combinaisons autorisées.

Cela n'aboutit pas nécessairement à l'ensemble des combinaisons possibles pour le nombre de choix initiaux autorisés (soit 2^n où « n » est le nombre de choix initiaux), certaines combinaisons étant interdites (pour des raisons physiques, normatives ou industrielles).

- De manière classique, plusieurs spécificités peuvent être associées à un même projet et peuvent être impliquées par un seul choix (comme dans le cas du « choix 2 »).

En revanche, s'il est tout à fait possible de vouloir réaliser plusieurs types d'environnements possibles dans un seul projet (lors de la construction d'infrastructures importantes), cela reste — à l'échelle de ce doctorat — hors d'atteinte en raison de la complexité des modèles résultant de tels choix. Pour ce genre de situation, nous invitons l'utilisateur à fracturer son projet en sous-projets spécifiques.

- Il est tout à fait possible qu'une « spécificité complémentaire » soit impliquée par plusieurs branches différentes. Cela illustre simplement la possibilité qu'une même spécificité puisse être réalisée (ou, simplement, exister) dans plusieurs types d'environnements. Par exemple, dans le domaine médical et plus précisément en pharmacopée, la préparation de médicaments stériles peut être réalisée dans différents « types » de pharmacies.

7.2.4 Du graphe des normes à l'outil d'aide à la qualification

Une fois tous les choix effectués, l'utilisateur dispose d'un (ou plusieurs) type(s) précis d'environnement(s) normé(s) et d'un ensemble (potentiellement vide) de « spécificités complémentaires ». Cela permet d'interroger le graphe de normes pour obtenir la suite d'étapes permettant de réaliser l'environnement normé ciblé.

Cette interrogation est totalement dépendante de l'implémentation informatique retenue pour le questionnaire et le graphe de normes. Dans notre cas (cf. Partie III), nous interrogeons littéralement (au sens des bases de données informatiques) le graphe de normes pour obtenir la suite d'étapes utilisée dans l'outil que nous allons maintenant présenter.

7.3 Outil de qualification

Le questionnaire a permis à l'utilisateur de disposer de la formulation exacte de son objectif, ceci incluant également des spécificités complémentaires lorsque cette cible est « non triviale » et nécessite des ajustements ou des précautions additionnelles.

Nous allons maintenant procéder à la qualification d'un projet de cette nature, en commençant par récupérer la liste d'étapes nécessaire à la représentation de cet objectif puis en utilisant cette liste comme guide pour la représentation du projet de l'utilisateur et sa validation.

7.3.1 Extraction des étapes

La première étape, pour la validation d'un projet concret d'environnement normé, consiste à extraire la liste d'étapes nécessaire à la représentation du type de projet visé. Naturellement, cette extraction s'effectuera sur le graphe de norme obtenu dans le chapitre précédent.

Selon la forme retenue pour la représentation des arrangements normatifs, cela peut être une opération plus ou moins complexe. Dans notre cas, il s'agit de la réalisation des étapes décrites dans la Section 6.2.5 et illustrées tout au long des Figures 6.7, 6.10, 6.11 et 6.15. Pour rappel, il s'agit de l'extraction du sous-graphe correspondant au problème posé, puis au traitement de ce graphe pour y réaliser un tri topologique. Cela nous permettra d'obtenir un ordre total sur les normes à appliquer, correspondant à la liste d'étapes à suivre pour passer d'un environnement « brut » à un environnement normé du type souhaité, validé concept par concept. Ainsi, cette liste est un ordre total sur l'ordre

partiel présent dans le graphe de normes et correspond, par définition, à un choix particulier de priorité de traitement des différents chemins du graphe. Ces priorités sont des éléments éminemment « opérationnels » et ils sont primordiaux pour l'optimisation de la solution. L'avantage de notre approche est de déléguer ces choix à l'élément logiciel se chargeant du tri topologique et de permettre ainsi à chacun d'adopter son propre choix de priorité².

7.3.2 Orchestration des étapes

Nous allons maintenant présenter le traitement de cette liste d'étapes par l'outil d'aide à la qualification. Nous débuterons par une présentation du fonctionnement général (« schématique ») de notre proposition puis nous en viendrons aux détails techniques, et plus précisément à notre emploi de la modélisation « orientée aspects » (*Aspect-Oriented Modeling* en anglais).

Principe

La Figure 7.2 illustre synthétiquement ce que nous détaillerons précisément par la suite. Cette approche « naïve » permettra au lecteur intéressé par le principe de notre contribution de l'appréhender sans en aborder les détails techniques. Dans la suite de cette section, nous utiliserons les chiffres et les lettres positionnés dans la colonne de gauche de cette illustration pour nous repérer ; les chiffres correspondent aux différents « états » et les lettres aux opérations à réaliser pour passer à « l'état » suivant.

Dans un premier temps (1), nous obtenons la liste de nœuds correspondant à l'objectif identifié dans le questionnaire (cf. section précédente). Cette liste ordonnée contient tous les éléments du sous-graphe concerné, dans cette forme si caractéristique que nous avons décrite dans les chapitres précédents et sur laquelle nous allons revenir plus précisément maintenant, notamment pour ses implications dans notre mise en œuvre de la qualification.

La première opération à réaliser (A) consiste à extraire la « charge utile » des nœuds. Cette extraction revient simplement à remplacer chaque nœud par le(s) métamodèle(s) ou aspect(s) qu'il contient. Naturellement, les nœuds sans contenu pertinent seront silencieusement ignorés et l'ensemble de métamodèles et d'aspects restant correspond à la représentation et la validation de l'environnement normé ciblé.

En raison de la structure très particulière de nos sous-graphes, cette liste (2) aura toujours la même forme générale. Ainsi, on trouvera toujours, en premier lieu, le métamodèle des bâtiments permettant de représenter le bâti « brut » et se concentrant seulement sur les aspects indépendants du domaine « métier » : bâtiments, étages, pièces, portes, bancs.

On trouvera ensuite le ou les aspects liés à l'instrumentation, leur nombre dépendant des différents types d'instrumentations et d'équipement requis pour le bon fonctionnement de l'environnement ciblé et donc, par définition, du type de cet environnement.

2. Cela constitue pour les industriels une forme d'avantage concurrentiel issu de leur maîtrise plus ou moins prononcée des environnements ; certains choix de traitement étant, d'après nos expérimentations, clairement plus intéressants que d'autres.

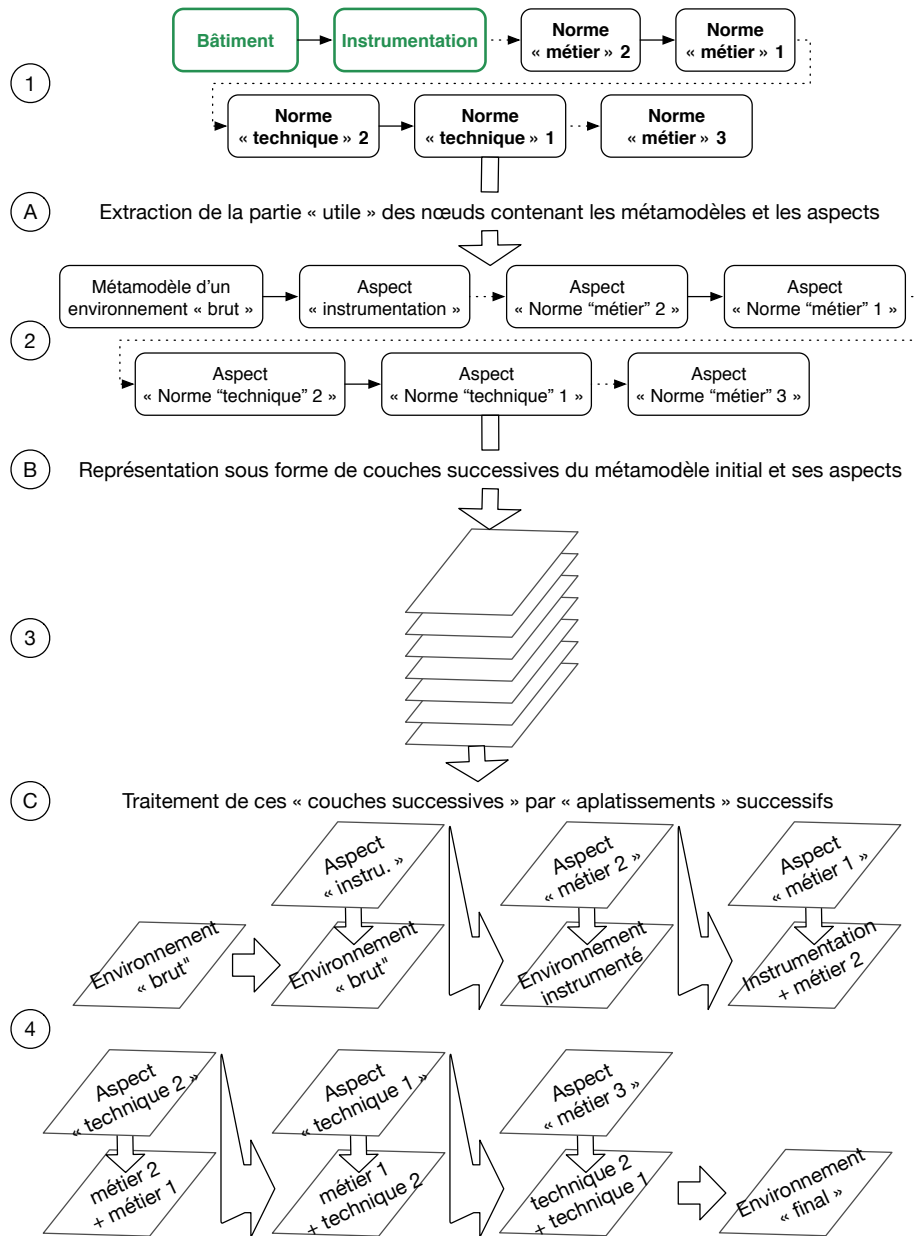


FIGURE 7.2 – Fonctionnement « schématique » de l’outil d’aide à la qualification

Enfin, on trouvera l'ensemble des éléments liés au type précis d'environnement ciblé, c'est-à-dire un nombre plus ou moins important d'aspects complémentaires liés aux différents types de normes (« techniques » ou « métiers ») imposées.

Dans le point (B), nous proposons de considérer ces artéfacts de modélisation non plus comme des éléments « horizontaux » (une liste d'étapes [2]), mais comme un millefeuille « vertical » (une superposition de couches [3]). Nous considérons cela comme la représentation la plus efficace et intuitive de notre contribution.

Nous considérons alors le métamodèle initial comme le point de départ d'un métamodèle final complet que nous allons bâtir itérativement en « incorporant successivement » les aspects. Chacun de ces aspects est « visuellement » une « couche » supplémentaire du millefeuille, tandis qu'il est techniquement une mise à jour des concepts existants (ajout de concepts, de contraintes, etc...).

Une fois cette représentation posée (3), il faut la traiter convenablement, c'est-à-dire commencer par le métamodèle « bâtiment », puis y intégrer l'aspect « instrumentation » afin d'obtenir un nouveau métamodèle (« environnement instrumenté » dans le schéma). À cet environnement instrumenté, nous allons intégrer un premier aspect « métier » qui va construire un nouveau métamodèle (« instrumentation + métier 2 » sur le schéma). Une nouvelle fois, nous prenons ce nouveau métamodèle, nous intégrons un second aspect « métier » qui va donner encore un nouveau métamodèle (« métier 2 + métier 1 » dans le schéma). Nous réitérons ensuite cette mécanique avec tous les aspects identifiés afin de terminer sur un environnement « final », métamodèle complet intégrant tous les concepts et toutes les contraintes.

Maintenant le principe général de cette contribution présenté, nous allons détailler plus précisément le point (4). Cette opération constitue le véritable travail « d'orchestration » de la modélisation des environnements normés, c'est cette étape qui va « encadrer » la modélisation et permettre à un utilisateur de représenter peu à peu l'ensemble de son projet et de le faire vérifier étape par étape, selon le prisme de chacun des textes ou des concepts intégrés.

Déroulement précis

La Figure 7.3 détaille le point 4 de la Figure 7.2, cette section explicite les points clés de cette étape. Pour simplifier les explications, nous avons une nouvelle fois utilisé des indices, dans le cas présent de « A » à « I ».

Cette suite d'étapes est conforme à l'esprit AOM, c'est-à-dire à l'utilisation d'aspects pour transformer et enrichir des modèles et métamodèles. Dans la suite de cette section, nous considérons qu'un utilisateur souhaitant représenter son projet dispose de l'outillage basique de modélisation informatique lui permettant de produire un modèle et de vérifier sa conformité à un métamodèle et des contraintes exprimées dans les langages classiques (OCL, etc.)³. Dans la suite de cette section, nous différencions clairement l'atelier de modélisation, permettant à l'utilisateur de représenter son projet, de l'outil d'orchestration s'occupant d'appliquer les aspects et de vérifier la conformité des modèles produits au métamodèle ainsi qu'aux règles de chaque étape.

3. Nous adoptons cette posture pour simplifier les explications, mais c'est un point largement discuté dans le Chapitre 9 et la Section 13.2.3.

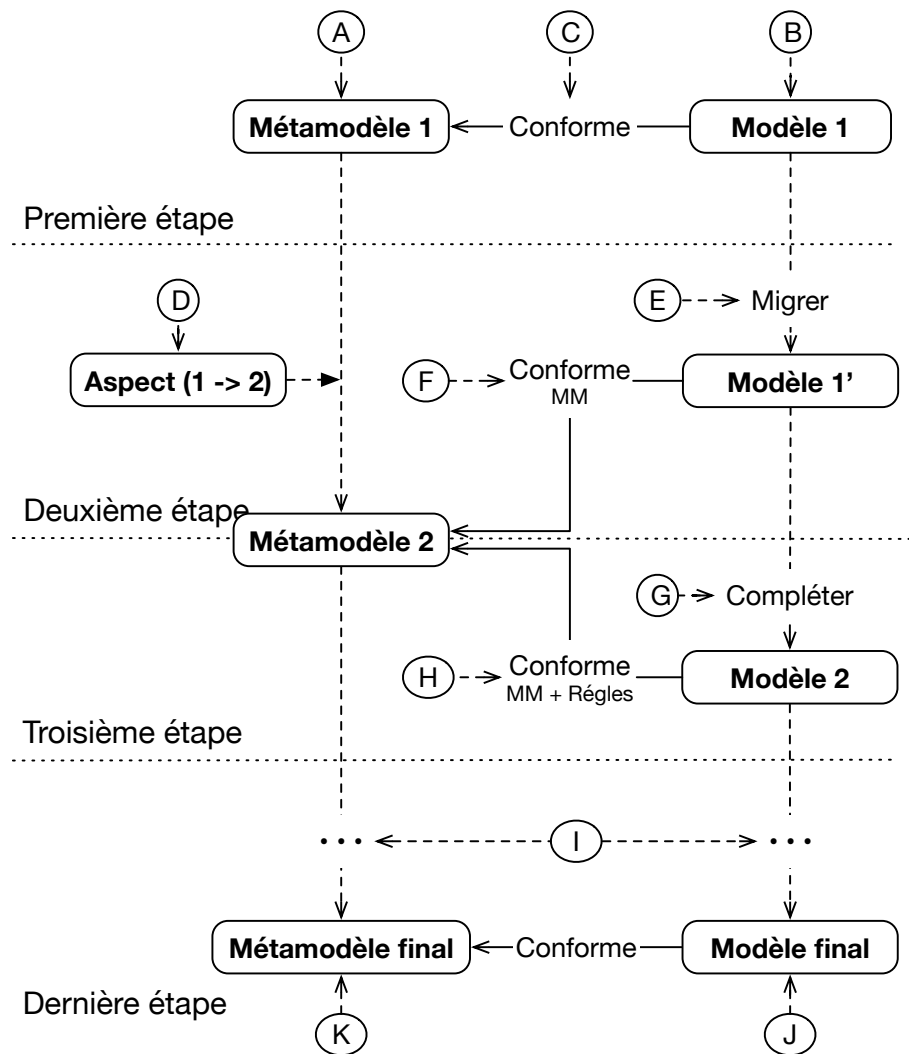


FIGURE 7.3 – Fonctionnement précis et technique de l’outil d’aide à la qualification

Première étape (points A, B et C)

Comme explicité, nous considérons le premier métamodèle (point A), celui du bâtiment, comme le point de départ de chaque représentation.

Ce métamodèle inclut l'ensemble des concepts traditionnellement présents dans un bâti (site, bâtiment, étage, pièce, porte, banc, etc.), avec éventuellement quelques variantes selon le type d'environnement ciblé (comme des « réacteurs » pour les environnements « nucléaires »). Cependant, dans de telles situations, nous inviterions plutôt les concepteurs à ajouter des couches complémentaires de modélisation après le bâti « brut », correspondant aux concepts totalement liés au type d'environnement ciblé, plutôt qu'à modifier le métamodèle initial, cela afin de protéger au maximum la généricité et la « portabilité » de ce métamodèle initial. Ces compléments prendraient la forme d'aspects complémentaires, donc de nœuds supplémentaires dans le graphe de normes, entre l'instrumentation et le bâti initial.

Pour l'utilisateur, le premier objectif est de modéliser une première version de son environnement représentant la structure « physique » de son projet. Cela va donner naissance au « Modèle 1 » (point B) qu'il fera évoluer tout au long de l'utilisation de l'outil d'aide à la qualification, par enrichissements successifs.

En raison de la nature même de notre approche, il est important de traiter avec la plus grande attention cette première représentation. En théorie, toutes les étapes suivantes ne seront que des précisions, des corrections de cette modélisation initiale pour la rapprocher d'un environnement complet et valide. Nous allons seulement reprendre inlassablement étape après étape, l'environnement initialement défini pour l'enrichir de nouveaux équipements, de nouveaux concepts, mais également de nouvelles contraintes.

Ainsi, dans cette première étape il ne faut pas seulement que l'environnement produit par l'utilisateur soit formellement valide (c'est-à-dire, conforme au Méta-Modèle 1 ou MM1), ce que nous pouvons vérifier, mais plutôt qu'il corresponde à ses ambitions réelles, ce que nous ne pouvons malheureusement pas totalement certifier.

L'outil d'orchestration va ensuite s'assurer de la conformité entre ce « Modèle 1 (M1) » et le « MétaModèle 1 (MM1) » (opération C), l'objectif est de s'assurer que le modèle représente un environnement initial valide, c'est-à-dire une base saine pour poursuivre le travail. En cas de non-conformité de l'environnement représenté, les contraintes non respectées sont indiquées et l'utilisateur peut alors corriger son projet pour répondre à ces manquements. Autant d'itérations que nécessaire sont réalisées entre l'atelier de modélisation et le sous-système réalisant la vérification de conformité entre le modèle et le métamodèle.

Nous disposons maintenant d'un premier modèle M1 conforme au MM1.

Deuxième étape (points D, E et F)

L'outil d'orchestration réalise alors l'opération D, c'est-à-dire l'application de l'Aspect 1 (A1) sur le MM1. Cela va conduire à la production d'un métamodèle 2 (MM2) qui nous servira de cadre formel pour les modèles 1' (M1') et 2 (M2).

Dans l'immense majorité des cas, cela correspondra à l'introduction des concepts liés à l'instrumentation. Comme nous l'avons proposé pour le bâti, rien

n'empêche (au contraire) de représenter l'instrumentation sous forme de plusieurs nœuds dans le graphe de normes, afin de séparer clairement les différentes thématiques : traitement de l'air, réseaux de capteurs, etc. Cela aboutira à la constitution de plusieurs aspects liés à l'instrumentation, en totale cohérence avec notre approche.

L'application de cet aspect suit le fonctionnement classique de l'AOM : modification de concepts existants, ajout de nouveaux concepts, ajout de nouvelles contraintes, etc. Il est important de noter que ce premier aspect est le seul dont le déroulement peut-être totalement prévu au moment de la réalisation du graphe de normes. En effet, c'est le seul aspect où le métamodèle initial est connu par avance⁴.

Pour l'utilisateur, la prochaine opération (E) concerne la transformation de M1 en M1', c'est-à-dire la mise en conformité (F) du premier modèle produit par l'utilisateur (respectant MM1) avec le nouveau métamodèle MM2. Dans un premier temps, nous invitons l'utilisateur à se concentrer seulement sur la mise en conformité de son premier modèle, et non sur l'intégration de nouveaux concepts, opération qu'il réalisera dans l'étape suivante.

Selon la mise en œuvre choisie pour les aspects, il est tout à fait envisageable d'intégrer les éléments requis dans l'aspect pour qu'une grande partie du modèle soit elle aussi transformée, en parallèle au métamodèle. L'opération E serait ainsi largement simplifiée, même si — et c'est un point à ne pas négliger — dans notre type d'activité, il faut se méfier des automatisations excessives. Bien au contraire, il faut veiller à préserver la sémantique des objets manipulés lors des mises à jour, se préserver de toute « sur-interprétation » des objectifs de l'utilisateur. Une nouvelle fois, nous sommes confrontés à l'obligation de disposer d'une forte expertise « métier » pour offrir des automatismes pertinents.

Nous disposons maintenant d'un modèle M1' conforme au MM2, mais où peu de nouveaux concepts ont été introduits. Les modifications se sont bornées à assurer la conformité du M1' avec ce nouveau cadre conceptuel.

Troisième étape (G et H)

Cette troisième étape débute par un travail de l'utilisateur sur l'atelier de modélisation pour réaliser l'opération G, c'est-à-dire l'intégration des nouveaux concepts issus de A1 dans le modèle M1'. Nous obtenons ainsi un nouveau modèle (M2), toujours conforme avec le MM2.

Le lecteur vigilant pourra faire remarquer les similarités évidentes entre les opérations E et G, consistant toutes deux à prendre le modèle M1 (respectivement M1') pour réaliser un nouveau modèle M1' (respectivement M2) conforme avec le MM2. Nous avons fait le choix de séparer en deux opérations disjointes ce travail pour deux raisons :

- Séparer le travail de mise en conformité du modèle existant avec celui d'intégrer de nouveaux concepts. Bien que dans de nombreuses situations, cela soit la même chose (l'intégration des nouveaux concepts étant obligatoire pour respecter les nouvelles contraintes), il y a de très nombreux cas (notamment lors de nos expérimentations) où ce n'est pas le cas et une telle présentation nous semble plus évidente.

4. Les implications de ce constat seront discutées dans les étapes suivantes.

- Isoler la partie pouvant profiter des automatisations fournies par les aspects présents dans le graphe de normes. En effet, comme nous l'avons exprimé dans l'étape précédente, cette automatisation est tout à fait possible et différencier ces deux étapes permet de mettre en avant la phase relevant de l'automatisation (et de sa correction) de celle requérant absolument un travail de l'utilisateur.

Dernière opération de cette étape (G), l'inévitable vérification par l'outil d'orchestration de la conformité entre le M2 produit par l'utilisateur et le MM2 servant de référence conceptuelle. En l'occurrence cela correspond à vérifier la bonne intégration de l'instrumentation dans le projet ciblé. Comme lors de l'opération C, autant d'itérations que nécessaire sont réalisées entre l'atelier de modélisation et l'outil d'orchestration pour assurer la conformité parfaite entre le modèle et le métamodèle.

Nous disposons maintenant d'un troisième modèle M2 totalement conforme au MM2 et dont tous les nouveaux concepts utiles apportés par le premier aspect ont été introduits.

À partir de la quatrième étape (point I, J et K)

Les étapes suivantes vont consister à reproduire les opérations D à H pour tous les aspects obtenus lors de l'interrogation du graphe de normes.

Nous allons donc, pour chaque aspect :

- l'appliquer sur le métamodèle courant afin d'obtenir une nouvelle version de celui-ci (via l'outil d'orchestration) ;
- demander à l'utilisateur de modifier son modèle actuel pour respecter le nouveau métamodèle et intégrer ses nouveaux concepts (via l'atelier de modélisation) ;
- assurer la conformité du nouveau modèle avec le métamodèle courant (via l'outil d'orchestration).

La mise en œuvre de ces étapes est la phase la plus critique de notre approche de la qualification de projet.

En effet, contrairement à l'aspect « instrumentation » — reposant sur un métamodèle connu et permettant, par nature, de prévoir la forme finale du métamodèle — ce n'est pas le cas des aspects suivant reposant, par définition, sur des métamodèles « temporaires », générés « dynamiquement » lors des phases précédentes.

Notre approche reposant sur un graphe de normes, ce fonctionnement est inévitable et les aspects seront naturellement employés dans un très grand nombre de contextes, dépendant de la proximité entre les différents types d'environnements normés étudiés. C'est son originalité et sa principale force.

Ceci n'est pas rédhibitoire, mais requiert en revanche une extrême rigueur dans la gestion et la construction du graphe de normes. Il faut s'assurer que l'ensemble des « concepts requis » indiqué pour le nœud (cf. Figure 6.2) couvre bien tous les éléments de modélisations modifiés par l'aspect contenu dans ce même nœud⁵.

5. Nous n'avons pas poussé les expérimentations sur ce point, mais c'est une opération tout à fait automatisable. Cela consiste à parcourir tous les chemins du graphe issus de la racine « Bâtiment » pour y chercher un aspect manipulant un concept n'ayant pas été précédemment défini. La correction « automatique » du nœud devient alors tout à fait possible.

Finalement, une fois les dernières itérations entre l'outil d'orchestration et l'atelier de modélisation effectuées, nous obtenons un modèle « final » (point J) conforme au métamodèle « final » (point K), et devant, si la phase de préanalyse du type d'environnement normé a bien été menée, représenter un projet valide et offrant toutes les garanties nécessaires pour une qualification de son fonctionnement.

Nous disposons maintenant du modèle final, totalement conforme avec l'ensemble des contraintes exprimé par les experts « métiers » lors de l'analyse de l'arrangement normatif et prêt à l'emploi pour la plateforme de traçabilité.

Conclusion

Nous avons désormais présenté l'ensemble de notre contribution pour la qualification d'un projet d'environnement normé, tous les aspects en ont été présentés et le lecteur intéressé pourra sans problème s'attacher à en réaliser une implémentation.

Pour mémoire, notre contribution à la qualification d'un projet d'environnement normé peut se concevoir de deux manières équivalentes.

D'un point de vue plutôt « imagé » (Figure 7.2), cela peut se concevoir sous la forme d'un millefeuille normatif dont les premières couches sont l'environnement physique et son instrumentation, tandis que toutes les couches supérieures sont les différentes normes à respecter.

D'un point de vue plus « technique » (Figure 7.3), cela consiste à exploiter au maximum l'AOM pour avancer en quinconce sur le projet, réaliser un premier modèle conforme au métamodèle initial, modifier ce métamodèle pour qu'il corresponde à un nouveau degré d'analyse, puis transformer à son tour le modèle afin de le rendre compatible avec ce nouveau métamodèle et plus proche de l'objectif final réel. Ce fonctionnement est ensuite reproduit sur le nombre total d'aspects à appliquer pour passer du métamodèle du bâti brut au métamodèle final.

7.4 Conclusion

Ce chapitre nous a permis de présenter notre mise en œuvre de l'outil d'aide à la qualification dans son principe, son fonctionnement, mais également ses limitations.

Le principe général de notre approche peut se concevoir comme un « millefeuille » de normes, les premières feuilles étant l'environnement « physique » et son « instrumentation », tandis que les « couches » supérieures sont les normes à appliquer. Le passage d'une feuille à l'autre est réalisé par des aspects, ceux-ci représentant les modifications à appliquer sur l'ensemble de modèles existants (« des couches précédentes ») pour proposer de nouveaux concepts, transformer les concepts existants et rajouter des contraintes supplémentaires.

Nous avons suivi la piste originale en Ingénierie des Exigences d'utiliser une approche orientée « modèle » couplée à une approche orientée « aspect ». Cette voie nous apparaît plutôt innovante et inhabituelle dans le panorama actuel de ces disciplines et nous en proposons des éléments d'implémentation et d'illustration dans les chapitres suivants.

Troisième partie
Mise en œuvre

Graphes de normes

Nous abordons maintenant les expérimentations concrètes menées sur nos contributions théoriques.

Nous allons pour cela débiter par un chapitre dédié à la mise en œuvre de notre représentation réticulaire des arrangements normatifs. Cette mise en œuvre repose sur l'emploi d'une base de données orientée « graphe » (en l'occurrence NEO4J) pour représenter et interroger le graphe de normes. Au terme de cette mise en œuvre, nous disposerons d'une implémentation fonctionnelle et opérationnelle de notre contribution.

Le chapitre suivant sera quant à lui dédié à la réification de notre principe de qualification d'un projet d'environnement normé, en liaison directe avec les travaux présentés dans ce chapitre.

Dans les sections suivantes, nous commencerons par une introduction générale visant à expliciter nos choix technologiques et effectuer un bref rappel théorique sur les concepts et outils manipulés.

Ensuite, nous aborderons notre mise en œuvre en elle-même et ses résultats.

Enfin, nous conclurons par une synthèse de nos résultats ainsi qu'un aperçu des perspectives offertes par ces travaux.

8.1 Rappel de l'objectif

Afin de faciliter la compréhension de ce chapitre, nous nous permettons un bref rappel de la contribution concernant la représentation réticulaire des arrangements normatifs, présentée au Chapitre 6.

La représentation proposée transforme un arrangement normatif en « graphe de normes ». Dans celui-ci, chaque norme constitue un « nœud » contenant les informations suivantes :

Titre : Titre complet du texte normatif, correspondant à celui fourni par l'organisme éditant le document.

Version : Version du document, correspondant à celle fourni par l'organisme éditant le document.

Description : Description du document, champ documentaire permettant aux lecteurs d'appréhender plus facilement l'objet de la norme représentée.

Concepts fournis : Liste des concepts fournis par la norme.

Concepts requis : Liste des concepts requis pour la norme.

Révision : Indice de révision de la modélisation de la norme.

Contenu/Charge utile : Partie « utile » du nœud contenant la représentation de la norme pour la technologie employée pour la qualification de projet ou de fonctionnement. Il s'agit de métamodèles ou d'aspects dans le cadre du Chapitre 9 et de contraintes dans le cadre de la Section 11.3¹.

Les champs « concepts fournis » et « concepts requis » permettent d'exprimer les liens existants entre les textes normatifs. Ainsi, la dépendance à un concept particulier d'une norme sera représentée de part et d'autre de la relation à l'aide des champs idoines, la représentation du caractère « optionnel » d'un texte normatif étant quant à elle laissée à la discrétion des implémentations.

Enfin, outre la simple représentation complète d'un graphe de norme, les différentes opérations présentées dans l'analyse du cadre normatif (Section 6.2) doivent pouvoir être automatisées :

Nettoyage des dépendances : suppression des doublons par une détection des concepts accessibles par transitivité (cf. Section 6.2.3).

Inversion des dépendances : afin de pouvoir identifier les « chemins » menant de la représentation du bâtiment à celle du type d'environnement visé (cf. Section 6.2.4).

Linéarisation des contraintes : application d'un tri topologique sur les chemins identifiés pour transformer un ordre partiel de traitement des normes en un ordre total (cf. Section 6.2.5).

8.2 Sélection de la technologie

Nous allons maintenant pouvoir extraire les contraintes liées à notre contribution, puis discuter des technologies possibles pour y répondre.

8.2.1 Contraintes

D'après le contexte précédemment défini, la technologie à retenir doit satisfaire les contraintes suivantes :

- Les normes doivent pouvoir être représentées sous la forme « d'objets » informatiques aux caractéristiques figées et clairement définies.
- La représentation générale doit pouvoir être « interrogée » afin d'en extraire un sous-ensemble d'objets possédant des caractéristiques particulières.

1. En raison de la nature même de notre approche de la représentation des arrangements normatifs, ce champ n'a aucun intérêt pour la compréhension de ce chapitre.

manières ; un très grand nombre de SGBD ont émergé en fonction du type de données manipulées, de leur mise en forme, de leurs modalités de conservation, etc. (Figure 8.1) et, depuis les années 90, les SGBD Relationnels (SGBDR) sont devenus la norme [GMUW09].

Rappels historiques

Pour mémoire, le concept de SGBDR a émergé dans les années 70 après la célèbre publication de Ted Codd [Cod70], proposant de représenter l'organisation des données fournies par l'utilisateur sous la forme de tables, nommées « relations », plus simples à manipuler. En revanche, au niveau du système, cela demandait des structures de données complexes afin de garantir l'efficacité et la rapidité de traitements des interrogations effectuées. Mais la vraie nouveauté, le véritable apport, était l'idée d'exprimer les requêtes dans un langage dédié, de haut niveau, indépendant de cette structuration interne complexe des données, découplant ainsi la représentation du stockage des données.

Cela donna rapidement naissance à plusieurs langages d'interrogation, dont l'un des pionniers est Structured English QUery Language (SEQUEL) [CB74] plus tard renommé, pour des raisons de copyright, Structured Query Language (SQL) [Opp04, Int11]. Ce langage est actuellement le plus répandu au monde pour l'interrogation de bases de données [GMUW09].

Conclusion

Devant le cahier des charges précédemment décrit, l'utilisation d'une approche relationnelle ne nous a pas semblé pertinente. D'autres représentations, plus proches de notre objectif, existent et devraient permettre d'améliorer grandement la simplicité et l'efficacité de notre réponse.

Néanmoins, afin de conserver la rigueur scientifique nécessaire à des travaux doctoraux, nous avons tout de même procédé à une mise en œuvre dans une technologie relationnelle. Cette mise en œuvre est présentée dans l'Annexe A.

Nous allons maintenant présenter les différentes technologies dites « NoSQL » puis, enfin, décrire précisément la technologie retenue dans le cadre de nos travaux.

8.2.3 Seconde hypothèse : NoSQL

Si les années 90 ont vu l'hégémonie des SGBDR, les années 2010 ont connu l'avènement de bases de données dites « NoSQL »² visant à répondre aux nouveaux enjeux posés par le « Web 2.0 », l'Internet des Objets (IoT), le « Big Data », etc.³

2. La communauté s'accorde à dire que le terme NoSQL est ambigu et doit être compris comme « Not Only SQL » plutôt que « Not SQL ».

3. Comme pour de nombreux concepts et technologies informatiques, ce n'est pas réellement « l'avènement » de nouvelles technologies, mais le « rafraîchissement » de concepts existants pour répondre à de nouveaux besoins et de nouveaux usages. Ici, le parallèle entre l'approche NoSQL et les « bases de données orientées objet » est évident... Pour être polémique, on fera remarquer que les annuaires LDAP [Zei06] étaient déjà, à leur façon, des « bases de données NoSQL ».

Ces cas d'utilisation ont une manière différente d'approcher la manipulation des données. Pour ceux-ci, deux objectifs principaux :

- Offrir un langage de manipulation « réellement » adapté aux types de données manipulées. Par exemple, le langage SQL semble relativement peu adapté à la manipulation de bases de données ne contenant que des clés et des valeurs (cf. 8.2.3).
- Optimiser certaines caractéristiques « secondaires » : nombres de requêtes par seconde, type de stockage employé, distribution des données, etc.

Nous allons maintenant détailler rapidement chacun de ces types de bases de données (illustrés dans la Figure 8.2), à l'exception des « BDD Orientées graphe » qui feront l'objet d'une section à part entière (pour des raisons évidentes)⁴.

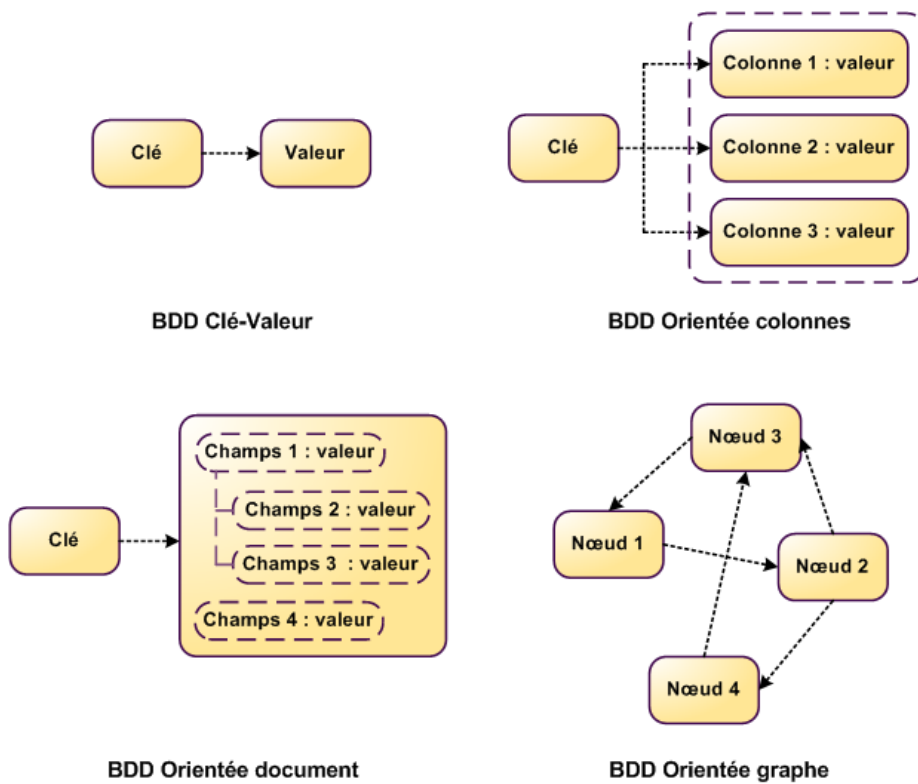


FIGURE 8.2 – Panorama des SGBD « NoSQL » [Fig10]

Les bases de données « clé-valeur »

Ce premier type de base de données répond à un cas d'utilisation très simple, celui d'un client souhaitant seulement obtenir la valeur associée à une clé en sa possession. Ainsi, cela s'apparente — en forçant le trait — à une table de hachage persistante accessible par le réseau.

4. Nous avons utilisé plusieurs articles pour nous aider dans la rédaction de cette section [Fig10, Dum14].

Par conséquent, le gestionnaire de données n'a aucune connaissance sur le type de données manipulé, il manipule seulement des chaînes de caractères sans aucune sémantique particulière. L'ensemble des traitements doit être réalisé par l'utilisateur interrogeant la base de données : mise en forme de la donnée reçue, filtrage des données, etc. Par corollaire, il est souvent impossible de modifier ou lire une partie seulement de la donnée associée à une clé.

Avec un modèle aussi simple, ce type de base de données est extrêmement véloce, fonctionne même à très grande échelle et peut manipuler des milliards de données⁵.

Il y a naturellement de nombreuses fonctionnalités supplémentaires associées à ces bases de données pour traiter les opérations traditionnellement associées à ce type d'utilisation : supports de stockage distribués ou volatiles, mécanisme de publication/souscription, synchronisation et réplication distante, techniques permettant le passage à l'échelle (partitionnement, tables de hachage distribuées, etc.).

Les implémentations les plus courantes sont Redis [Red15], Amazon DynamoDB [Ama15] et MemCached [Mem15].

Les bases de données « orientées document »

Fournissant un modèle de données plus sophistiquées, les bases de données « orientées document » consistent en une collection de paires (clé, document) où le document contient des données organisées de manière hiérarchique (comme le XML ou le JSON). Le document comme unité d'information, dans une forme semi-structurée, rend ces bases de données particulièrement adaptées aux usages du « Web 2.0 ».

À la différence des bases de données clé-valeur ne possédant aucune idée du contenu associé à la clé, celles-ci sont conscientes du contenu et par conséquent peuvent effectuer des indexations de différents champs et offrent des requêtes plus élaborées, permettant ainsi aux utilisateurs de déléguer une partie des traitements à la base de données. Des documents organisés de manière hiérarchique et semi-structurée permettent une grande souplesse et une grande versatilité des objets représentés, permettant ainsi aux développeurs de maximiser la pérennité des données.

L'unité de partitionnement de la base de données étant la clé (comme pour les bases de données « clé-valeur ») cela permet de disposer des mêmes fonctionnalités : distribution, réplication, etc.

Les implémentations les plus courantes sont MongoDB [Mon15], RethinkDB [Ret15] et CouchDB [Apa15c].

Les bases de données « orientées colonnes »

Nous quittons maintenant l'espace des bases de données réalisant des associations clés-valeurs (quelle que soit la complexité des valeurs) pour en revenir

5. Pour la petite histoire [Red12], un des plus importants sites de vidéo pour adultes fonctionne intégralement sur une base de données Redis [Red15], assumant en 2012 une charge d'environ 100 millions de pages vues par jour et un cluster d'esclaves Redis pouvant supporter 300.000 requêtes par seconde.

à une modélisation plus proche du relationnel. Plus proche, certes, mais à la logique diamétralement opposée. . .

Dans les bases de données orientées colonnes, nous n'allons plus manipuler les lignes de tables dont le nombre de colonnes est statique, mais les colonnes elles-mêmes. Dans cette structuration, chaque ligne peut avoir un nombre différent de colonnes, les colonnes étant indexées par une même clé.

Ce type de système « passe à l'échelle » en fragmentant les lignes sur plusieurs machines distinctes, chacune d'entre elles s'occupant de la gestion de colonnes précises. Seuls les attributs réellement manipulés sont verrouillés lors des cycles d'écritures permettant ainsi de largement paralléliser les traitements s'effectuant sur différents attributs d'un même enregistrement.

Les implémentations les plus courantes sont Apache HBase [Apa15e] et Apache Cassandra [Apa15b].

Les bases de données « orientées graphe »

Une base de données « orientée graphe » est une base de données orientée objet utilisant la théorie des graphes, donc avec des nœuds et des arcs, permettant de représenter et stocker les données. Cette représentation étant celle nous intéressant dans le cadre de ces travaux, nous allons débiter par quelques rappels théoriques avant de décrire leur application dans une base de données.

Rappels conceptuels

Pour une définition formelle d'un graphe orienté, on pourra consulter le premier chapitre « Basic Terminology, Notation and Results » de l'ouvrage de Bang-Jensen [BJG09]. Ce chapitre couvre tous les aspects théoriques nécessaires à une bonne compréhension du sujet ; en voici un extrait sur les principes généraux :

Définition 3 *A directed graph (or just digraph) D consists of a non-empty finite set $V(D)$ of elements called vertices and a finite set $A(D)$ of ordered pairs of distinct vertices called arcs. We call $V(D)$ the vertex set and $A(D)$ the arc set of D . We will often write $D = (V, A)$ which means that V and A are the vertex set and arc set of D , respectively. The order (size) of D is the number of vertices (arcs) in D ; the order of D will sometimes be denoted by $|D|$.*

Application dans une base de données NoSQL

Une base de données « orientée graphe » est un système de stockage suivant la théorie des graphes et dont la persistance des données respecte les relations présentes entre les éléments voisins d'un graphe.

Ainsi, un tel type de base de données s'avère pertinent dans tous les cas d'utilisation relevant du parcours ou du traitement de graphes. Ceci est donc idéal pour des requêtes s'apparentant à des recherches de chemin, des parcours de graphes de données (en largeur, en profondeur), des calculs d'arbres « recouvrants » et pour tous les algorithmes classiques de la théorie des graphes [BJG09]. Nous disposons ainsi d'un formalisme et d'une plateforme d'exécution bien plus adaptés à notre problème qu'avec les SGBDR classiques précédemment étudiés.

En revanche, les cas de recherche « classiques » portant sur la valeur d'attributs ou sur le type de données représentées ne sont pas adaptés à ce type de

représentation. Dans de tels cas d'utilisation, on adjoint à la base de données « orientées graphe » des index, simples ou plus complexes comme Apache Lucene [Apa15f] (et son dérivé Apache SolR [Apa15g] ou Elasticsearch [Ela15]), afin d'éviter l'interrogation du graphe pour répondre à ces requêtes.

Par définition, ces bases de données ne sont pas adaptées à un passage à l'échelle par simple ajout de nouveaux serveurs. Celles-ci requièrent des mécanismes de synchronisation complexes pour réaliser des écritures sur le graphe, et, par corolaire, ces ajouts vont complexifier et ralentir ces opérations. De manière identique, modifier le partitionnement « physique » des données reste relativement complexe et doit être évité. Ici, il faudra donc nécessairement compenser l'augmentation de la complexité issue de l'augmentation du nombre de serveurs par une augmentation de leur puissance individuelle.

L'implémentation la plus courante est NEO4J [Neo15b]; c'est cette dernière que nous allons maintenant détailler.

8.3 NEO4J

NEO4J [Gup15], est une base de données orientée « graphe », open source, fournissant une structure de données flexible, où l'accent est mis sur les relations entre les objets plutôt que sur les objets eux-mêmes.

8.3.1 Un peu d'histoire [Jor15]

Le projet NEO4J a débuté en 2000 quand Emil Eifrem, Johan Svensson et Peter Naubauer ont constaté l'impact énorme de l'utilisation d'un modèle relationnel pour représenter les données de leur application. Cela concernait autant les performances d'exécution de l'application finale que la gymnastique mentale employée lors des développements. Après investigation, la source de leur problème était l'inadéquation entre leur modèle de données et le SGBD employé (relationnel); ils travaillaient alors sur une application dont la particularité était l'importance des relations entre les données face aux données elles-mêmes (des graphes).

Comme nous l'avons mis en évidence pour notre propre objectif dans l'Annexe A, bien que la représentation d'une telle structure de données dans une base relationnelle soit tout à fait possible, les performances obtenues sont fortement impactées par les traitements supplémentaires requis. De plus, outre les traitements nécessaires pour la représentation et l'utilisation correcte de ces relations, il faut également ajouter ceux nécessaires à leur gestion et leur modification afin de s'assurer que les relations « logiques » (donc sans gestion native par le SGBDR) soient respectées. Naturellement, la complexité de l'application et du processus de développement lié évolue avec l'augmentation des relations, que ce soit par leur nombre ou leur sémantique.

Après avoir cherché de nouvelles solutions à leurs problèmes et fait quelques essais d'approches alternatives, ils ont commencé le « Projet Neo ». Neo devait créer une base de données spécifique, fournissant une modélisation, un stockage et une recherche de données adaptées à ces cas d'utilisations spécifiques. Malgré tout, ce modèle tenait à conserver certains concepts « clés » comme les propriétés

ACID ou la notion de transaction. . . L'objet de ce projet était ainsi de trouver le meilleur compromis entre une modélisation adaptée à des problèmes de type « graphe » et les concepts traditionnels des bases de données relationnelles.

Nous allons maintenant décrire les seuls points saillants de la technologie NEO4J dans le cadre de nos travaux doctoraux. Ainsi, tous les éléments liés au stockage des données et au passage à l'échelle de celui-ci seront ignorés, le lecteur intéressé pourra se reporter à la documentation officielle du projet.

Nous allons plutôt nous concentrer sur les modalités d'interaction avec NEO4J et sur le langage d'interrogation proposé, « CYPHER ».

8.3.2 Modalités d'interaction avec NEO4J

Il existe deux principaux modes d'interrogation avec une base de données NEO4J : un mode « embarqué » et un accès « distant ».

Base de données embarquée

Ce premier type d'interaction — limité au langage Java — est dédié à la manipulation d'une version « embarquée » de la base de données, fonctionnant dans le processus même de la JVM (comme peuvent l'être les solutions HSQL [The15b] et Derby [The15a]).

Pour certains scénarios spécifiques, cela offre des avantages indéniables : simplification et amélioration des tests unitaires, excellentes performances pour des cas d'utilisation sans accès réseau ou à faible parallélisme, etc. . . .

En outre, l'utilisation d'une base de données interne s'effectue au travers d'une API Java particulière offrant des fonctionnalités exclusives. Celle-ci permet à l'utilisateur de manipuler directement le graphe et lui fournit une vision « objet » du graphe de données. Cela autorise notamment des algorithmes de parcours de graphe entièrement personnalisables par le développeur qui peut interagir directement avec le moteur de NEO4J.

Accès distant à la base de données

À l'opposé, cette solution utilise NEO4J comme un serveur REST classique, offrant plusieurs points d'entrée en fonction de la requête à réaliser.

Il existe deux principaux types de points d'entrée :

- des URL spécifiques, dédiées à l'opération concernée. Ainsi, le point d'entrée « /node » permet, lorsqu'il est interrogé, de manipuler des nœuds (création, destruction, interrogation). Le serveur répond alors avec les informations pertinentes en fonction de la requête en rappelant systématiquement l'API générale disponible pour l'utilisateur, permettant ainsi à celui-ci de « découvrir » le serveur.
- une URL générique, dédiée à l'exécution de requêtes CYPHER quelconques. Le client transmet la requête à exécuter avec les paramètres de celle-ci et le serveur lui répond avec les différents éléments mentionnés dans la clause RETURN.

C'est cette dernière solution que nous avons retenue pour nos expérimentations afin de pouvoir manipuler de manière indépendante le serveur hébergeant

les données et le client effectuant les requêtes. Cela nous permet de disposer d'une interface de visualisation (cf. 8.6) distincte du programme réalisant les insertions, les parcours, etc...

8.3.3 Le langage CYPHER

Cette section n'a pas vocation à décrire entièrement le langage d'interrogation CYPHER mais de fournir les notions nécessaires au lecteur pour en appréhender le fonctionnement, le potentiel expressif et l'utilité dans le cadre de nos recherches⁶.

Introduction

CYPHER est un langage de requête déclaratif autorisant une interrogation et une mise à jour du graphe à la fois compréhensive et efficace. C'est un langage dont la grammaire est relativement simple tout en restant très puissante à l'usage (dû à sa parfaite adéquation avec sa cible). Ainsi, des requêtes de base de données très complexes peuvent facilement être exprimées, en se concentrant sur les aspects « métiers » plus que sur les aspects « techniques ».

CYPHER est conçu pour être un langage naturel de requête, adapté à la fois aux développeurs et aux experts métiers devant travailler sur des graphes. Par choix des développeurs, ce langage est plus adapté à la lecture qu'à l'écriture.

CYPHER est inspiré par un certain nombre d'approches différentes et se fonde sur des pratiques d'interrogations établies. La plupart des mots-clés comme WHERE et ORDER BY sont inspirés par SQL tandis que les requêtes orientées « recherche de motifs » sont empruntées à SPARQL. Enfin, certains autres éléments de langage ont été empruntés à Haskell et Python.

Structure

La structure des requêtes CYPHER est calquée sur celle du langage SQL, l'utilisation conjointe de plusieurs clauses.

Les clauses s'enchaînent et, telles des chaînes de médiation, se nourrissent successivement ; chaque clause consommant les éléments produits par la clause précédente pour produire ses propres résultats. Par exemple, les nœuds correspondant à une clause MATCH serviront de contexte à la clause suivante.

Plutôt qu'une liste exhaustive des clauses du langage (disponible dans sa documentation [Neo15a]), nous allons seulement présenter ici celles nécessaires à la lecture d'un graphique et quelques opérations mineures d'écritures.

MATCH : La structure réticulaire recherchée (un nœud, deux nœuds liés par une ou plusieurs transitions, etc.). Il s'agit de la méthode traditionnelle d'obtention de données à partir du graphe.

WHERE : La partie de la clause MATCH permettant d'ajouter des contraintes sur la structure recherchée.

CREATE : Création de nœuds et de relations dans le graphe. Une ou plusieurs clauses MATCH précèdent souvent cette clause pour positionner correctement la nouvelle relation dans le graphe existant.

6. Cette section utilise abondamment la documentation officielle du langage [Neo15a].

RETURN : les éléments de la clause MATCH à retourner à l'utilisateur (correspond à ce qui suit la clause SELECT des requêtes SQL).

Ces quelques mots-clés, s'ils peuvent paraître simplistes, permettent tout de même de réaliser des interrogations extrêmement complexes sur la base de données.

Digression : comparaison avec SQL pour notre type de problème

Le code source 8.1 illustre parfaitement cette affirmation. Cette requête identifie les cycles de longueur arbitraire dans un graphe quelconque.

```
|| MATCH (a)-[*]->(b) WHERE a = b RETURN *;
```

Listing 8.1 – Requête CYPHER identifiant des cycles de longueur arbitraire dans un graphe quelconque

Pour mémoire, les codes A.5 et A.6 illustraient des opérations similaires en utilisant seulement le langage SQL. Le premier permettait de chercher des chemins de longueur 2 et le second permettait de lister tous les chemins partant d'un nœud particulier (détectant de manière indirecte les cycles).

Cela conforte de manière indéniable les positions exprimées dans les chapitres précédents. Réaliser nos objectifs à l'aide d'un langage relationnel est tout à fait possible. Un langage dédié à cet usage reposant sur des structures de données adaptées permet néanmoins d'obtenir un formalisme bien plus facilement compréhensible et accessible à des non-initiés.

Structure : détails

Si la digression précédente nous a permis de constater rapidement la puissance de ce langage, attardons-nous quelques instants sur sa syntaxe avant de passer à notre mise en œuvre du graphe de normes⁷.

Dans les exemples suivants, nous utiliserons le graphe de la Figure 8.3.

Dans celui-ci :

- « John »⁸ est l'ami⁹ de « Sara »¹⁰ et « Joe ».
- « Sara » est l'ami de « Maria ».
- « Joe » est l'ami de « Steve ».
- « John » est donc l'ami « au second degré » de « Maria » et « Steve ».

Débutons par une requête devant trouver les amis des amis de la personne nommée « John » (Figure 8.4).

Décrivons maintenant la requête CYPHER effectuée :

- Le pattern recherché est « nœud → nœud → nœud », avec :
 - « john » désignant le nœud de départ ;
 - « fof » désignant le nœud d'arrivée.

7. L'exemple est emprunté à la documentation officielle de NEO4J.

8. Abus de langage pour un nœud dont la valeur de l'attribut « name » est « John ».

9. Dans cet exemple, le terme « ami » est celui des réseaux sociaux. Il peut donc être interprété de multiples façons dont celle de « suivre » (cf. Twitter).

10. Abus de langage pour l'existence d'une relation de type « friend » entre deux nœuds

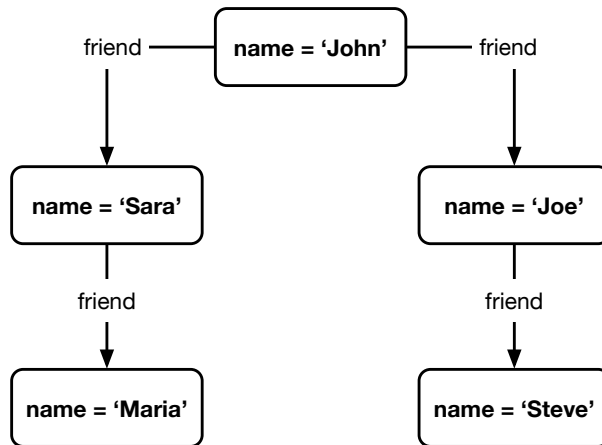


FIGURE 8.3 – CYPHER : exemple de graphe

```

MATCH (john {name: 'John'})-[:friend]->()-[:friend]->(fof)
RETURN john.name, fof.name

```

john.name	fof.name
"John"	"Maria"
"John"	"Steve"

2 rows

FIGURE 8.4 – CYPHER : exemple de requête sur un graphe (1/2)

- Les contraintes appliquées au pattern sont :
 - Le point de départ (« John ») est un nœud dont l'attribut « name » vaut « John ».
 - Les relations sont de type « friend ».
- Nous récupérons le nom du nœud initial (« john.name ») et le nom de l'ami de l'ami (« fof.name »).

Remarque auxiliaire, pour le lecteur plus familier avec les requêtes « SQL », la syntaxe « {name : 'John'} » est strictement équivalente à une clause WHERE, où l'on filtrerait la valeur de la colonne « name » de la table contenant les personnes.

Seconde illustration, nous cherchons maintenant toutes les personnes possédant un ami dont le prénom commence par « S » (Figure 8.5).

```
MATCH (user)-[:friend]->(follower)
WHERE user.name IN ['Joe', 'John', 'Sara', 'Maria', 'Steve'] AND follower.name =~ 'S.*'
RETURN user.name, follower.name
```

user.name	follower.name
"John"	"Sara"
"Joe"	"Steve"

2 rows

FIGURE 8.5 – CYPHER : exemple de requête sur un graphe (2/2)

Décrivons maintenant la requête CYPHER effectuée :

- Le pattern recherché est « nœud → nœud », avec :
 - « user » désignant le nœud de départ ;
 - « follower » désignant le nœud d'arrivée.
- Les contraintes appliquées au pattern sont :
 - Le point de départ est un nœud dont l'attribut « name » vaut « Joe », « John », « Sara », « Maria » ou « Steve ».
 - Les relations sont de type « friend ».
 - Le point d'arrivée est un nœud dont l'attribut « name » commence par « S ».
- Nous récupérons le nom du nœud initial (« user.name ») et le nom de l'ami (« follower.name »).

Ces deux requêtes sont suffisantes pour permettre aux lecteurs d'appréhender l'implémentation décrite dans ce chapitre, dont la mise en œuvre fait l'objet de la section suivante.

8.4 Mise en œuvre : API et implémentation

L'objectif principal d'un travail doctoral n'est pas la définition d'une API de programmation, nous aborderons donc relativement rapidement cette thématique,

nous contentant de donner les précisions nécessaires au lecteur pour comprendre la stratégie mise en place et l'adéquation entre nos ambitions et nos travaux de validation.

8.4.1 La notion de norme

La première interface pertinente est celle décrivant la notion de « norme » (code source 8.2).

```

package com.noc.phd.norms.model;

import java.util.Set;

/**
 * Norm contract.
 *
 * @author Anthony GELIBERT
 * @version 1.3.0
 */
public interface Norm {
    /** Full name of the norm. */
    String getFullName();
    /** Version of the norm. */
    String getVersion();
    /** Aliases of the norm. */
    Set<String> getAliases();
    /** Description of the norm. */
    String getDescription();
    /** Concepts provided by the norm. */
    Set<String> getProvidedConcepts();
    /** Concepts required by the norm. */
    Set<String> getRequiredConcepts();
    /** Payload corresponding to this norm. */
    String getPayload();
    /** Norm to JSON String. */
    String toJSONString();
}

```

Listing 8.2 – API pour la représentation d'une norme

On y retrouve tous les éléments décrits dans le schéma 6.2 et la section afférente. Une norme est un objet immuable fournissant l'ensemble des caractéristiques attendues. On notera cependant l'ajout d'une méthode permettant d'obtenir une liste d'alias pour la norme représentée. Si cet ajout peut paraître relativement mineur, il n'en est rien. Cet ajout à l'implémentation nous a permis de nous adapter lorsque différents publics utilisent différents termes pour désigner le même texte, cela arrive régulièrement pour les documents possédant des acronymes, des identifiants, des codes, des noms usuels, etc.¹¹

Conformément aux spécificités de NEO4J (Section 8.3) et nos choix de mise en œuvre (Section 8.4), les implémentations doivent fournir une méthode `toJSONString` représentant la norme au format JSON afin d'être transmise à NEO4J par son API REST¹².

11. Il conviendra donc de ne surtout pas négliger cet attribut lors de l'analyse des textes normatifs et du travail avec les experts « métier ».

12. L'implémentation de cette interface est totalement classique et n'offre aucun intérêt dans

8.4.2 L'interaction avec le serveur

C'est cette interaction que nous allons maintenant décrire par l'interface NEO4J (code source 8.3).

```

package com.noc.phd.norms.db;

import com.noc.phd.norms.model.Norm;

import java.util.List;

/**
 * Neo4J Simple API.
 *
 * @author Anthony GELIBERT
 * @version 4.0.0
 */
public interface Neo4J {
    /** Check if the database is running. */
    boolean isRunning();
    /**
     * Create a node for this norm.
     *
     * @return node number.
     */
    int create(final Norm norm);
    /** Remove all nodes. */
    boolean clear();
    /** Get nodes required to model the given concept. */
    List<Integer> getNodesToModel(final String concept);
}

```

Listing 8.3 – API pour la supervision générale d'un serveur NEO4J

On y retrouve les méthodes correspondant à nos besoins :

- test de fonctionnement du serveur ;
- création d'une norme (par un objet de l'interface précédente) ;
- effacement du serveur ;
- et interrogation de la base de données pour obtenir la liste ordonnée des nœuds permettant de modéliser un concept particulier.

Les détails concrets de ces fonctionnalités sont classiquement laissés à la discrétion des implémentations.

Dans notre cas, nous avons fait le choix d'une communication distante (cf. Section 8.3) avec des échanges entre notre plateforme d'essai et le serveur NEO4J par l'API REST de ce dernier.

Cela nous permet de réaliser les trois premières opérations, la dernière — l'extraction du chemin de modélisation — étant relativement différente.

8.4.3 Stratégie de parcours de la base de données

Il existe de nombreuses façons d'interroger la base de données pour en extraire la liste ordonnée des nœuds permettant de modéliser un concept particulier. On peut reposer sur les algorithmes classiques de la théorie des graphes, sur un

le cadre de cette thèse, elle permet simplement de créer des normes et implémente rapidement chacune de ces méthodes d'interrogation

parcours interactif avec un expert « métier » effectuant les sélections, sur les outils fournis par NEO4J, etc.

Cette multiplicité nous a conduit à définir une interface fonctionnelle (code source 8.4) nous permettant de faire varier les méthodologies et de confirmer ou informer nos hypothèses.

```

package com.noc.phd.norms.db;

import java.util.List;

/**
 * Contract for a travel strategy.
 *
 * @author Anthony GELIBERT
 * @version 1.0.0
 */
@FunctionalInterface
public interface TravelStrategy {
    /**
     * Given a start node and a target node,
     * give the path between these two points.
     */
    List<Integer> getPathForConcept(final int start,
                                   final int target);
}

```

Listing 8.4 – API pour les stratégies de parcours du graphe de norme

Fort de l'implémentation de ces différentes interfaces, nous avons pu bâtir un gestionnaire de scénarios pour nos validations.

8.4.4 Gestionnaire de scénarios

Nous cherchions un moyen d'automatiser facilement les opérations routinières que sont la vérification de présence du serveur, sa remise à zéro, l'insertion d'un certain nombre de nœuds et la recherche de chemin parmi ceux-ci.

Pour ce faire, nous avons mis en forme nos objectifs dans une interface (code source 8.5) et réalisé une implémentation partielle de celle-ci factorisant l'ensemble des comportements « communs » à nos différentes expérimentations (code source 8.6).

Si l'interface mérite peu de commentaires, attardons-nous quelque peu sur la classe abstraite. Cette dernière suit le protocole précédemment décrit.

On commence par valider la connexion avec le serveur NEO4J, puis on efface complètement le graphe actuel, enfin on ajoute l'ensemble des nœuds formant le scénario visé. Il devient alors possible d'interroger le serveur pour obtenir la suite de nœuds permettant de modéliser un type d'environnement particulier. Cette interrogation est transmise à l'implémentation sous-jacente de l'API NEO4J (code source 8.3) qui, une fois le type d'environnement localisé, emploiera la stratégie définie (code source 8.4) pour extraire la liste de nœuds correspondante.

La génération de cette liste de nœuds reste à la discrétion des implémentations concrètes de notre classe abstraite, nous permettant ainsi de générer rapidement des scénarios représentant des environnements extrêmement riches.

```

package com.noc.phd.norms.scenario;

import java.util.List;

/**
 * Contract for a modeling scenario.
 *
 * @author Anthony GELIBERT
 * @version 1.0.0
 */
public interface ModelingScenario {
    /** Prepare the scenario. */
    void prepare();
    /**
     * Ask how to model a given <i>concept</i>.
     *
     * @param concept Searched concept.
     *
     * @return List of nodes (by their ID).
     */
    List<Integer> askConcept(final String concept);
}

```

Listing 8.5 – API pour un scénario de modélisation

8.4.5 Conclusion

Dans cette section, nous avons décrit les interfaces et les fonctionnalités bâties pour assurer nos validations.

Lors de nos expérimentations, nous avons associé l'utilisation d'implémentation de la classe `AbstractModelingScenario` (afin de « préparer » la base de données) avec le moteur Web décrit dans la Section 8.5.1 pour consulter et manipuler le graphe de normes.

De cette manière, nous avons obtenu une plateforme de validation complète, en mesure de déployer un scénario arbitraire et disposant des outils adéquats pour procéder à son analyse. De plus, le langage CYPHER a confirmé sa totale adéquation avec nos besoins et nous avons pu sans aucun problème bâtir les requêtes nécessaires à la vérification de nos développements (recherche de cycles, extraction de sous-graphes, recherche de chemins, etc.).

Naturellement, l'implémentation et, plus généralement, l'approche décrites dans cette section représentent seulement l'infrastructure employée dans le cadre de nos travaux doctoraux. Cela ne constitue en aucun cas la finalité industrielle de notre contribution ni son éventuelle exploitation commerciale.

8.5 Résultats

Nous allons dans ce chapitre décrire nos résultats d'un point de vue général et sans détailler le cas d'utilisation sous-jacent ; l'objectif est d'illustrer l'implémentation décrite dans la section précédente pour ensuite pouvoir l'appliquer sur un cas réel (Chapitre 10).

```

package com.noc.phd.norms.scenario;

import com.noc.phd.norms.db.*;
import com.noc.phd.norms.model.Norm;

import com.sun.jersey.api.client.ClientHandlerException;

import org.apache.logging.log4j.*;

import java.util.List;

/**
 * Abstract implementation of a modeling scenario.
 *
 * @author Anthony GELIBERT
 * @version 2.0.0
 */
public abstract class AbstractModelingScenario
    implements ModelingScenario {
    /** Neo4J client. */
    private final Neo4J m_client;

    /**
     * Create an abstract modeling scenario by providing
     * the server information.
     */
    protected AbstractModelingScenario(final String host,
                                        final int port) {
        m_client = new RESTNeo4J(host, port);
    }

    @Override
    public final void prepare() {
        try {
            if (!m_client.isRunning()) {
                throw new UnavailableServerException();
            }
        }
        catch (final ClientHandlerException e) {
            throw new UnavailableServerException(e);
        }
        m_client.clear();
        listOfNorms().forEach(m_client::create);
    }

    @Override
    public final List<Integer> askConcept(final String concept) {
        return m_client.getNodesToModel(concept);
    }

    /**
     * Return the list of norms which should be inserted.
     *
     * @return List of norms
     */
    protected abstract List<Norm> listOfNorms();
}

```

Listing 8.6 – Classe abstraite pour un scénario de modélisation

8.5.1 Représentation visuelle

Avant de décrire le déroulement d'une modélisation d'arrangement normatif, commençons par détailler la forme employée pour représenter les différents concepts en présence. Ne requérant pas, pour ces travaux universitaires, une mise en forme particulièrement détaillée, nous avons préféré nous reposer sur les possibilités offertes nativement par NEO4J en les adaptant à notre objectif¹³.

```
node {
  diameter: 80px;
  color: #30B6AF;
  border-color: #46A39E;
  border-width: 2px;
  text-color-internal: #FFFFFF;
  font-size: 11px;
  caption: '{name}';
}

relationship.NEXT_STEP {
  color: #FCC940;
  shaft-width: 8px;
  font-size: 8px;
  padding: 3px;
  text-color-external: #000000;
  text-color-internal: #000000;
  caption: '{type}';
  border-color: #F3BA25;
}

relationship.REQUIRES {
  color: #4356C0;
  shaft-width: 3px;
  font-size: 8px;
  padding: 3px;
  text-color-external: #000000;
  text-color-internal: #FFFFFF;
  caption: '{type}';
  border-color: #3445A2;
}
```

Listing 8.7 – Fichier CSS employé pour notre représentation du graphe de normes

Cela correspond à l'application du fichier CSS du code source 8.7. Pour l'illustrer, nous allons étudier la Figure 8.6, dans laquelle nous avons déclaré les nœuds « Bâtiment » et « Instrumentation » (le second dépendant du premier).

Une fenêtre typique de l'interface Web de NEO4J se décompose de la manière suivante :

Le cadre :

Le bandeau supérieur contient la requête CYPHER envoyée à NEO4J.

Dans ce chapitre, nous allons faire très souvent appel à la même requête :

¹³. Pour information, NEO4J fournit nativement une interface web d'interrogation et de consultation du graphe de données. La représentation graphique des données est adaptable par l'emploi d'un fichier CSS fourni par l'utilisateur.

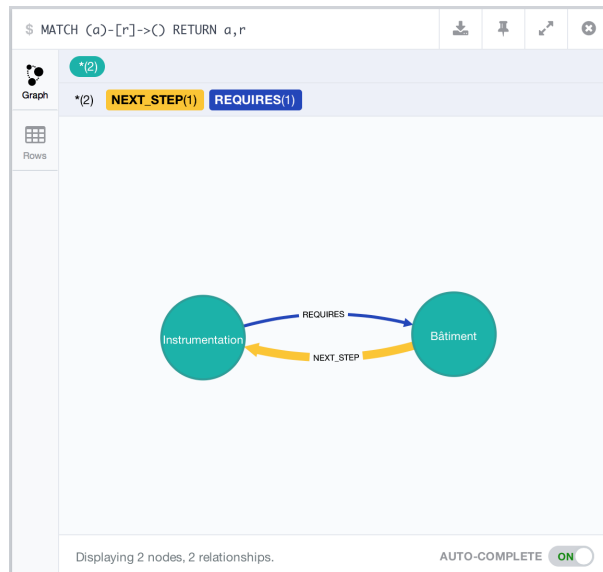


FIGURE 8.6 – Graphe de normes : représentation visuelle

```
MATCH (a)-[r]->(c) RETURN a,r
```

Au regard de la Section 8.3, cette requête est simple à appréhender. On demande à NEO4J d'isoler l'ensemble des structures « nœud → nœud », on nomme « a » le nœud « source » et « r » la relation existante entre les deux nœuds. Une fois les éléments isolés, NEO4J nous retourne l'ensemble des éléments « a » et « r », avec, dans le mode de représentation sélectionné, une remise en forme des données pour transformer cette suite de nœuds et de relation en un graphe structuré.

La colonne de gauche permet de changer de mode entre un affichage « graphe » (celui de la Figure 8.6) et un affichage tabulaire (semblable à celui des SGBDR).

Le bandeau intermédiaire contient des statistiques sur le résultat de la requête (nombre de nœuds, nombre de relations [global et par type]).

Le bandeau inférieur contient des informations contextuelles et les détails du nœud actuellement sélectionné (le cas échéant).

La zone de représentation au centre de la fenêtre contient la représentation des données selon le mode d'affichage sélectionné dans la colonne de gauche.

Représentation « orientée graphe » : les aspects visuels sont totalement modifiables. Nous présentons ici les choix effectués pour les illustrations de cette section.

Les normes : elles sont représentées sous forme de cercle bleu, le titre de la norme — affiché en blanc — permet de distinguer les différents nœuds. L'ensemble des métadonnées de la norme est enregistré dans le nœud, et s'affiche dans le bandeau inférieur.

Les relations de dépendances : les lignes bleues comportant le texte `REQUIRES` représentent les liaisons de dépendances, leur source est le nœud requérant le concept et leur destination le nœud le fournissant.

Les relations de modélisation : les lignes oranges, plus épaisses, avec le texte `NEXT_STEP` représentent les différentes étapes de modélisation.

Maintenant les aspects purement esthétiques abordés, nous allons illustrer les premières étapes de la création d'un graphe (Section 8.5.2) puis nous présenterons un graphe de normes complet (Section 8.5.3).

8.5.2 Construction d'un graphe

Dans cette section, nous allons commencer par représenter les premières étapes de construction d'un graphe typique. Pour ce faire, nous allons ajouter les nœuds listés dans le tableau en Figure 8.7.

Nom	Version	Concepts	
		Requis	Fournis
Recommandations Cytotoxiques	1	Instrumentation Bâtiment ISO 14644-1	Cytotoxiques
NF S 90-351	2013	ISO 14644-1 Instrumentation Bâtiment	NF S 90-351
ISO 14644-1	1999	Instrumentation Local	ISO 14644-1
Instrumentation	1	Bâtiment	Instrumentation
Bâtiment	1		Bâtiment

FIGURE 8.7 – Graphe de normes : liste des nœuds utilisés pour l'illustration de la construction d'un graphe

Tout d'abord, commençons par l'ajout d'un premier nœud « Bâtiment », celui-ci n'a aucune dépendance et ne requiert aucun concept supplémentaire (Figure 8.8).

Ensuite (Figure 8.9), procédons à l'ajout d'un second nœud « Instrumentation », dépendant de « Bâtiment ». Nous constatons l'ajout automatique de deux dépendances ; dans cette implémentation, nous avons fait le choix de calculer les liaisons de modélisation (`NEXT_STEP`) au moment de l'insertion dans le graphe. Ici, nous avons seulement deux dépendances de direction opposée.

L'insertion de la norme « ISO 14644-1 » (Figure 8.10) est la première étape réellement intéressante, on ajoute une norme dépendant de deux autres nœuds déjà liés. On observe le comportement décrit dans les chapitres précédents, les liens « `REQUIRES` » sont insérés tels que décrits dans les dépendances de la norme « ISO 14644-1 » tandis que la relation « `NEXT_STEP` » est ajoutée seulement pour le lien avec l'instrumentation, le lien direct avec le bâtiment étant redondant.

L'insertion des « Recommandations Cytotoxiques » (Figure 8.11), dépendante de l'ensemble des nœuds précédemment insérés, suit la même logique. Les liens « `REQUIRES` » respectent les spécifications issues de l'analyse normative réalisée par les experts du domaine. Ensuite, les dépendances sont analysées afin d'isoler

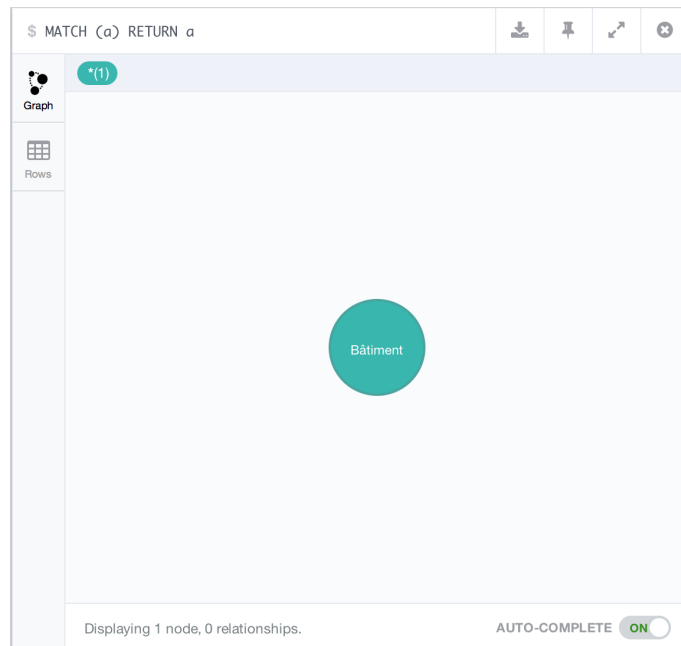


FIGURE 8.8 – Graphe de normes : construction (étape 1)

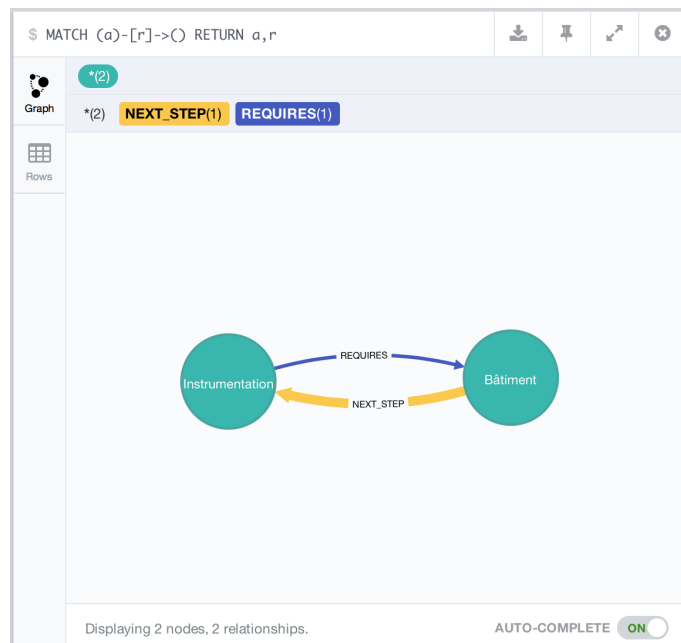


FIGURE 8.9 – Graphe de normes : construction (étape 2)

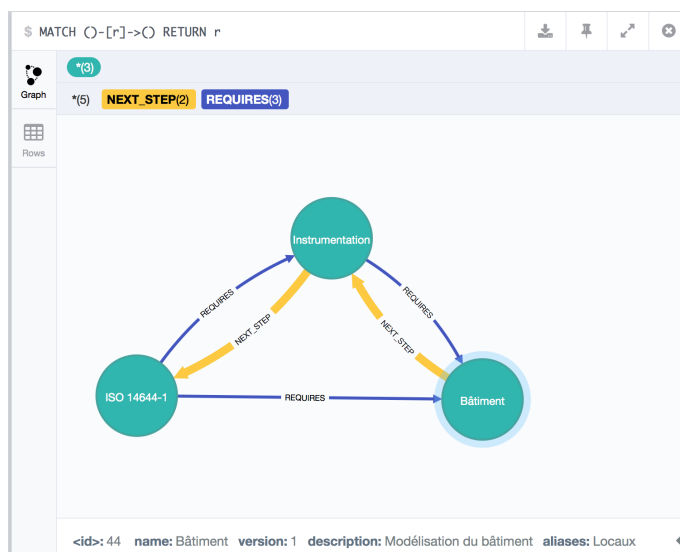


FIGURE 8.10 – Graphe de normes : construction (étape 3)

les liens réellement utiles pour la phase de modélisation informatique et les redondances sont alors filtrées.

Enfin, l'insertion de la norme « NF S 90-351 » (Figure 8.12), dépendant des mêmes normes que les « Recommandations cytotoxiques » abouties à la même représentation et donc la création des mêmes relations. Par conséquent, on aboutit à la création d'une alternative entre le traitement de la branche « NF S 90-351 » et la branche « Recommandations cytotoxiques » lors de la modélisation.

En terme « métier », cela correspondra à « choisir » entre le traitement d'un caractère spécifique de l'environnement considéré (ici, la manipulation de produits cytologiques) et celui d'un caractère plus générique (ici, les notions de salles propres et d'environnements maîtrisés). Ces deux normes impliquant des contraintes sur les mêmes éléments, sans interdépendance, seule l'expérience « métier » permettra de définir un ordre de traitement pertinent.

Maintenant que nous avons illustré quelques insertions de normes, attachons-nous à la description d'un graphe complet.

8.5.3 Graphe complet

Un exemple de graphe complet est présenté dans la Figure 8.13. On constate le fonctionnement caractéristique de notre approche avec toutes les caractéristiques décrites dans nos différentes contributions.

Ainsi, le graphe a une forme générale de « losange » avec — dans le sens des exigences (relations REQUIRES) — le type d'environnement normé analysé (une URCC) comme racine et le bâtiment comme seule feuille. On constate l'augmentation du nombre de relations sortantes des nœuds lorsque l'on avance dans le graphe puis la convergence de ces relations vers le nœud « Bâtiment », concept « primordial » sans aucune dépendance. L'ordre d'apparition des nœuds

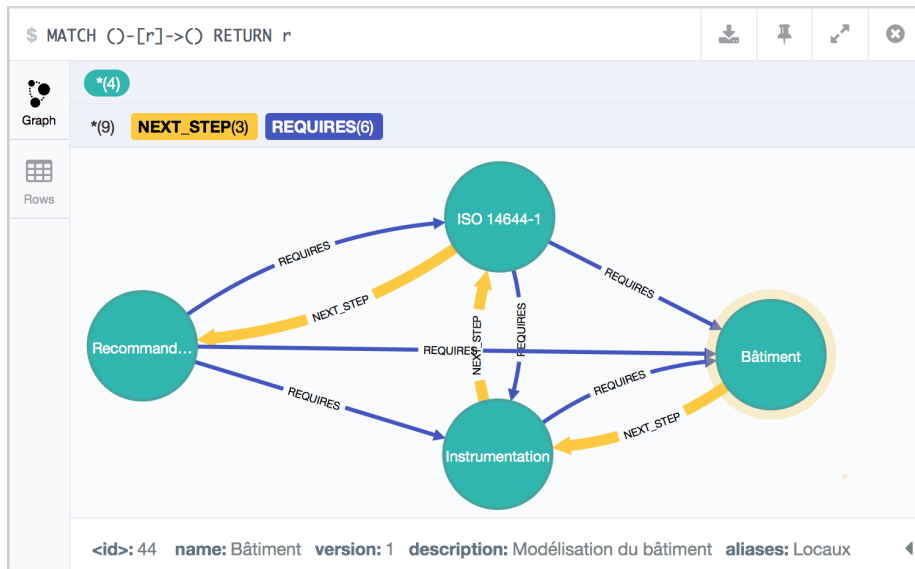


FIGURE 8.11 – Graphe de normes : construction (étape 4)

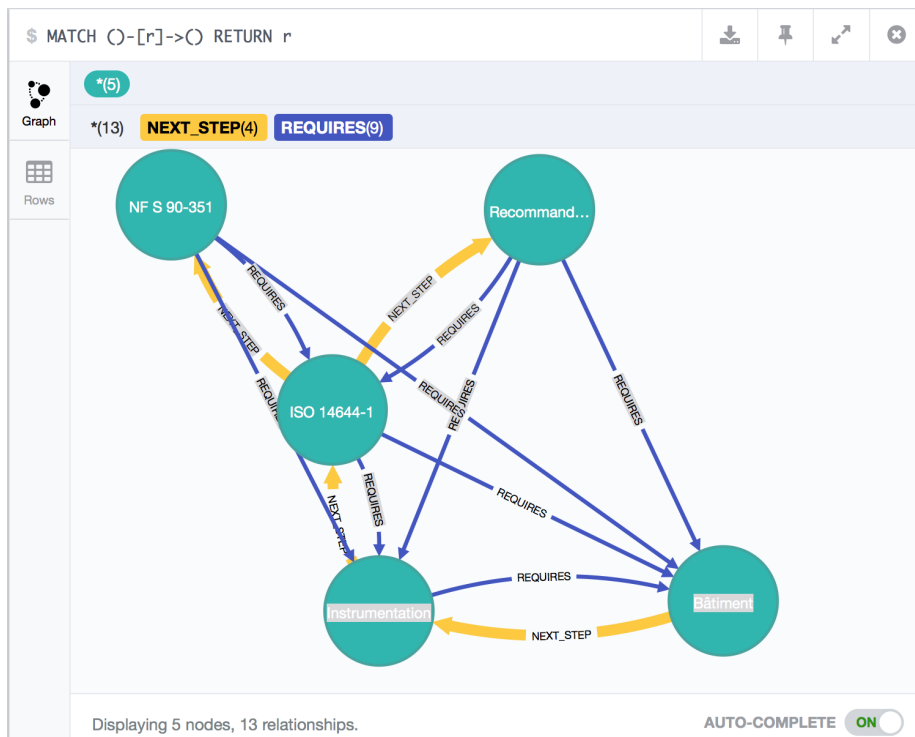


FIGURE 8.12 – Graphe de normes : construction (étape 5)

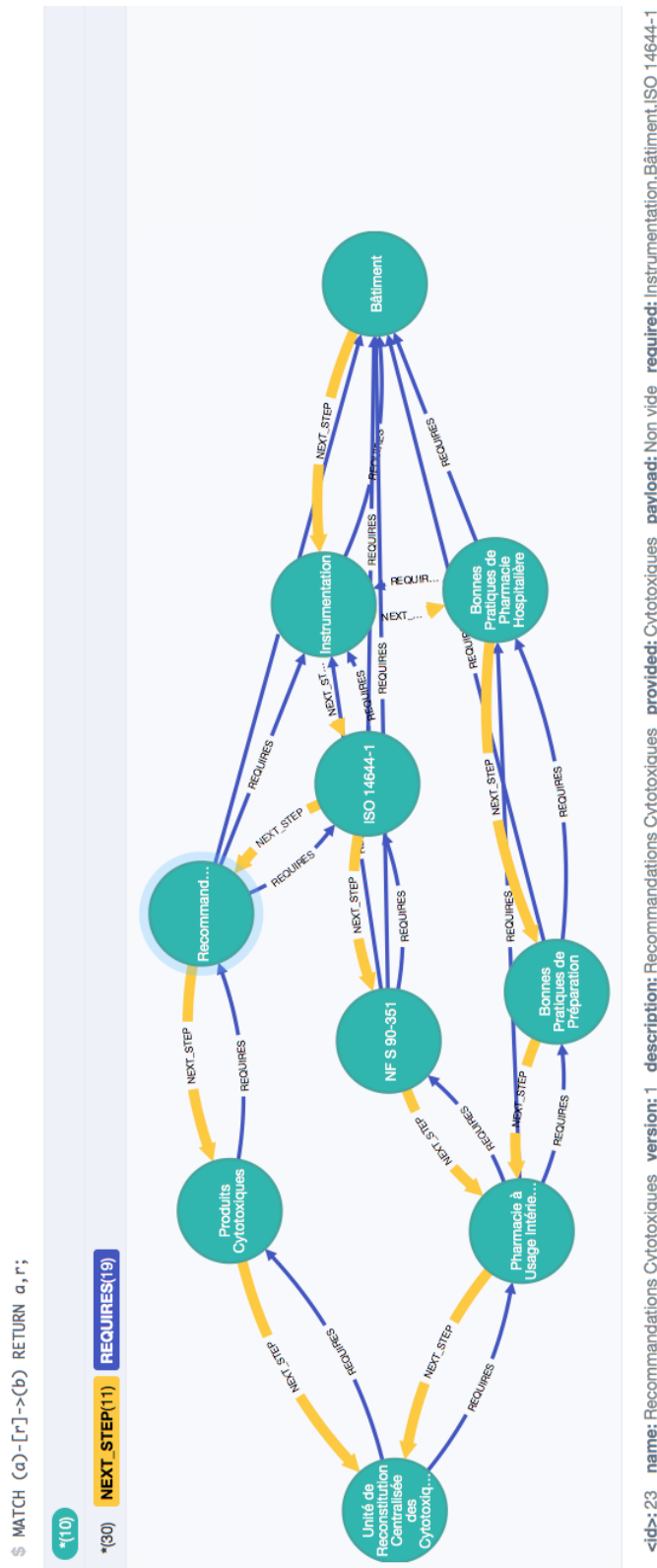


FIGURE 8.13 – Graphe de normes : exemple de graphe complet

suit l'enchaînement que nous avons décrit : type d'environnement, concepts généraux, normes métiers, normes techniques, instrumentation et enfin bâtiment.

Dans le sens contraire, celui des étapes de modélisations (NEXT_STEP), deux caractéristiques intéressantes (et conforme à nos analyses théoriques) :

- la forme générale d'un « losange » dont la source est le nœud « Bâtiment » et l'arrivée le type d'environnement modélisé (ici, une URCC) ;
- si l'on se place d'un futur point de vue de « modélisation » (Chapitre 9), l'URCC étant seulement un nœud symbolique, elle peut-être ignorée, et nous avons alors une structure arborescente « classique ».

Cette représentation nous permet d'illustrer facilement la nécessité d'application d'un tri topologique pour passer d'une structure arborescente (donc, par nature, partiellement ordonnée) à une structure linéaire (totalement ordonnée) que l'outil de modélisation devra suivre.

8.6 Conclusion

En conclusion, nous avons montré par l'expérience la faisabilité et l'intérêt de notre approche. Nous avons pu élaborer une base de données « orientée graphe », reposant sur la technologie NEO4J, et conformes à nos exigences.

Avec cette implémentation, nous arrivons à obtenir des solutions pour l'ensemble de nos utilisateurs : utilisateurs « finaux » comme développeurs et intégrateurs.

Les premiers, maîtrisant parfaitement les aspects « métiers » mais ne souhaitant pas s'investir dans les aspects « informatiques », trouveront (au travers des fonctionnalités « orientées web » proposées par cette technologie) une interface de travail adaptée à leurs usages. Nous n'avons pas souhaité investir outre mesure dans cette première interface au cours de cette thèse, les résultats de l'implémentation pour développeurs se sont avérés suffisamment probants pour n'avoir aucun doute sur la faisabilité et la mise en œuvre de cette version ¹⁴.

Les seconds — principales cibles des travaux de cette thèse — trouveront, au travers des API REST et JAVA de NEO4J relativement complètes, les moyens d'implémenter toutes les fonctionnalités requises. L'objectif principal était de réaliser une librairie réalisant toutes les opérations décrites dans le Chapitre 6, les résultats sont conformes à nos attentes. La librairie produite permet également de manipuler le graphe et les normes à « haut niveau » afin d'adapter la solution au contexte industriel initial de ces travaux (cf. Section 11.3).

Naturellement, il reste des améliorations à apporter à cette implémentation pour atteindre le degré de sophistication attendu par les utilisateurs. Dans ce chapitre, nous avons décrit les principaux travaux complémentaires sur cette thématique, que ce soit pour des travaux universitaires ou industriels.

Pour les aspects industriels, nous pouvons citer l'adjonction du texte original au nœud le représentant, la réalisation d'outils simplifiant les opérations traditionnelles d'ajout/modification de nœuds ou encore la réalisation d'une interface Web de gestion globale du graphe de norme.

¹⁴. D'autant plus que nous ne voyons que peu d'intérêt, en termes de recherche, à un tel travail.

Pour les aspects universitaires, nous pouvons citer la conception d'outils d'audit du graphe pour en identifier les incohérences ou pour simplifier l'ajout de nœuds en étant capables de pratiquer des consolidations partielles de graphes. De manière similaire, des reprises sur nos algorithmes seront sans doute nécessaires lorsque le volume de données deviendra important. Nous avons pointé plusieurs faiblesses contraignant l'ajout de nœuds et la recherche dans le graphe qui pourront facilement trouver réponse en appliquant les résultats issus des travaux en théorie des graphes [BJG09].

Maintenant que nous disposons d'une implémentation pour l'analyse d'arrangement normatif, nous pouvons nous intéresser à l'implémentation des outils requis pour l'analyse d'un projet d'environnement normé.

Outil d'aide à la qualification

Après avoir proposé une mise en œuvre complète de la représentation réticulaire d'un arrangement normatif, nous allons aborder celle de l'outil d'aide à la qualification.

Cette mise en œuvre repose sur l'emploi d'un framework d'Aspect-Oriented Modeling (AOM) KerMeta [Tri15, Div15, JBF11] pour représenter le projet d'environnement normé (sous la forme d'un métamodèle initial) et les différentes normes (sous la forme d'aspects). Au terme de ce chapitre, nous disposerons d'un embryon d'implémentation de notre contribution pour lequel il restera néanmoins d'importants travaux d'ingénierie à effectuer pour poursuivre la démarche.

Dans les sections suivantes, nous commencerons par une introduction générale visant à expliciter nos choix technologiques et effectuer un bref rappel théorique sur les concepts et outils manipulés.

Ensuite, nous aborderons la mise en œuvre en elle-même et ses résultats.

Enfin, nous conclurons par une synthèse de nos résultats ainsi qu'un aperçu des perspectives offertes par ces travaux.

9.1 Rappel de l'objectif

Afin de faciliter la compréhension de ce chapitre, nous nous permettons un bref rappel de la contribution sur l'outil d'aide à la qualification, présentée au Chapitre 7 (p. 99).

Pour éclairer la compréhension de cette mise en œuvre, nous allons tout d'abord rappeler l'architecture devant être mise en place (Figure 5.8 p. 75) en l'adaptant avec notre mise en œuvre de la représentation de l'arrangement normatif (Figure 9.1).

Nous allons maintenant détailler rapidement ce schéma.

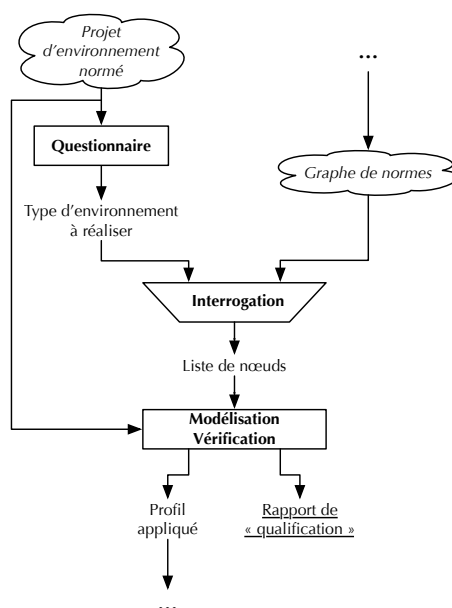


FIGURE 9.1 – Architecture à mettre en œuvre pour l'outil d'aide à la qualification

Nous disposons (par la phase précédente) d'un graphe de normes contenant (entre autres) les étapes de modélisation du type d'environnement considéré.

D'un autre côté, nous demandons à l'utilisateur (muni de son projet d'environnement normé) de répondre à un questionnaire qui permet de déterminer le type « précis » d'environnement normé à analyser. Muni de cette information, il devient possible d'interroger la base de données pour en extraire la liste de nœuds triés par ordre de traitement.

Nous en venons alors à l'outil de « Modélisation/Vérification » (rappelé dans les paragraphes suivants) qui utilisera les « charges utiles » contenues dans les nœuds (cf. Section 7.3.2 p. 104) pour aider à la qualification du projet.

Les livrables attendus après cette étape sont le « rapport de qualification » (pour l'opérateur humain) et le « profil appliqué » devant faire le lien avec la dernière phase de notre méthodologie (traçabilité et audit sur site de l'environnement durant l'usage).

Nous pouvons maintenant rappeler le fonctionnement attendu de l'outil de « Modélisation/Vérification », la Figure 9.2 (réplique de la Figure 7.3 p. 107) fournit les précisions nécessaires.

Nous extrayons du nœud « Bâtiment » le métamodèle initial (A), celui du bâti, pour permettre à l'utilisateur de représenter la structure de son environnement (sites, bâtiments, étages, pièce, portes, bancs, équipements non médicaux, etc.).

Une fois ce premier modèle (B) réalisé, l'outil d'orchestration doit extraire la « charge utile » du second nœud, appliquer un ou plusieurs aspects sur le métamodèle initial (D) et proposer à l'utilisateur de mettre en conformité son modèle initial avec ce nouveau métamodèle (E). Quand le second modèle sera réalisé, les règles contenues dans l'aspect pourront être appliquées pour vérifier la conformité du projet (F).

On procédera ensuite de la même manière avec tous les nœuds suivants (G, H et I) pour bâtir étape par étape le modèle final représentant l'environnement visé complet (J et K).

Une dernière phase devra extraire de ce modèle final les éléments du profil « appliqué ».

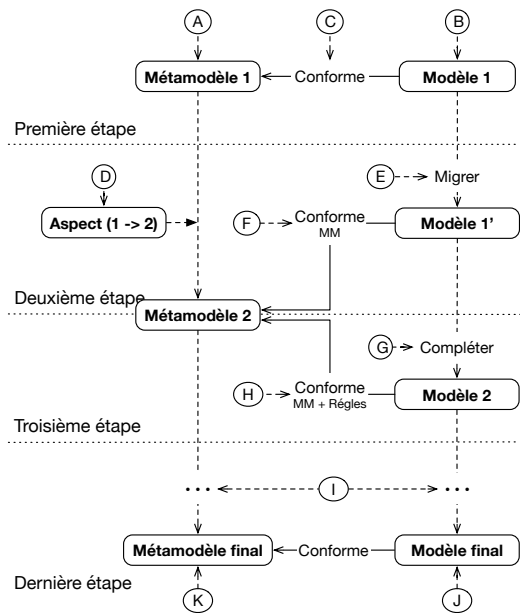


FIGURE 9.2 – Rappel du fonctionnement attendu de l’outil d’aide à la qualification

9.2 Sélection de la technologie

Nous allons maintenant pouvoir extraire les contraintes liées à notre contribution, puis discuter des technologies possibles pour y répondre.

La complexité et les contraintes posées par cette mise en œuvre concernent principalement l’outil de « modélisation/vérification », la partie « questionnaire » étant bien plus classique et ne présentant pas d’intérêt majeur (d’un point de vue de recherches doctorales). Tout au plus, discuterons-nous de son intégration dans notre démarche globale.

Dans le cas présent, la première étape d’analyse réglementaire prenant la forme d’un graphe de normes, le questionnaire (quelle qu’en soit sa forme) doit simplement aboutir à un type d’environnement présent dans le graphe.

Pour notre part, nous avons simplement utilisé un questionnaire écrit (n’ayant que peu d’intérêt d’être discuté ici), dont nous donnerons une illustration dans le Chapitre 10.

Dans la suite de ce chapitre, nous emploierons régulièrement des termes issus de l’IDM que nous allons maintenant rappeler rapidement.

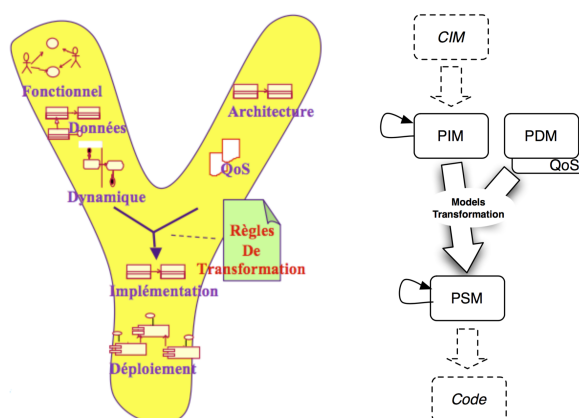


FIGURE 9.3 – MDA : Un processus en Y dirigé par les modèles [Com08]

La Figure 9.3 présente le processus classique « en Y » des architectures dirigées par les modèles (*Model Driven Architecture* en anglais) :

Modèle CIM¹ Ce modèle est totalement « abstrait », il fournit une description fonctionnelle de l'application, aussi bien en termes de services fournis que de dépendances. À ce stade, il n'y a aucun détail d'implémentation dans le modèle.

Modèles PIM² Ces modèles sont ceux d'analyse et de conception de l'application. À ce stade, on procède au découpage de l'application en modules, sous-modules, etc. Le rôle des modèles PIM est de fournir une vision structurelle et dynamique de l'application, mais indépendamment des questions techniques liées à une implémentation dans une technologie particulière.

Modèle PDM³ Ce modèle (lorsqu'il est employé) permet de décrire la structure et les fonctions techniques relatives à une plateforme d'exécution particulière (une modélisation des spécifications de la JVM serait une forme de PDM). L'association d'un PIM et d'un PDM permettra de générer un PSM.

Modèles PSM⁴ Ces derniers modèles, se rapprochant le plus du code final de l'application, décrivent l'implémentation d'une application sur une plateforme d'exécution particulière.

D'après le contexte précédemment défini, notre objectif s'intègre, en IDM, dans le sous-domaine de la « transformation de modèles » (*model weaving* en anglais), et plus particulièrement, dans la catégorie des « transformations exogènes » (cf. Figure 9.4). Nous devons donc sélectionner, parmi toutes les alternatives disponibles [MVG06, Val10, WMJC12], celles respectant les contraintes

1. Computational Independent Model.
 2. Platform Independent Model.
 3. Platform Dependant Model.
 4. Platform Specific Model.

décrites ci-après ⁵.

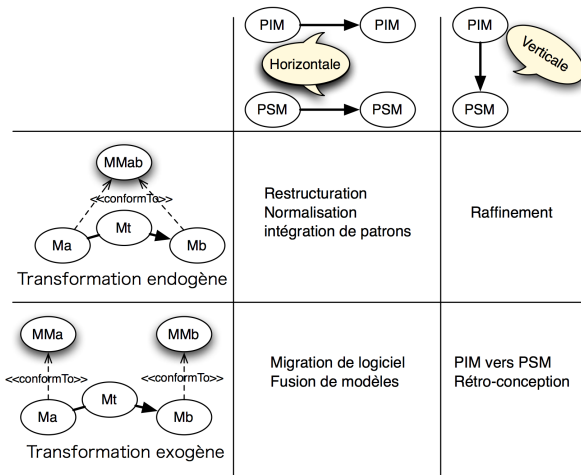


FIGURE 9.4 – Classes de transformation en IDM [CCP10]

La principale contrainte est liée à notre façon d’appliquer les transformations exogènes et, plus particulièrement, dans le cadre d’une application complète de la modélisation/vérification, à notre faible maîtrise des métamodèles intermédiaires (étape « I » de la Figure 9.2).

En effet, dans ce domaine, un certain nombre de solutions fournissent les outils permettant de décrire les transformations nécessaires pour passer d’un métamodèle A à un métamodèle B tous deux « connus » préalablement. Ces transformations peuvent prendre différentes formes, plus ou moins formelles, et ainsi s’exprimer, par exemple, sous la forme de modèles avec leur propre métamodèle (de ce qu’est une transformation de modèles), ou encore des règles de transformations pour exprimer le passage d’un métamodèle à un autre. Ces technologies existent également pour l’union de métamodèles (plus proche de notre objectif), la fusion de modèles de deux métamodèles différents A et B pour les exprimer dans un troisième métamodèle C.

Dans ce cas de figure, l’objectif est de « traduire » des concepts. Comment le concept X exprimé dans le métamodèle A s’exprime-t-il dans le métamodèle B ?

Le standard Query/View/Transformation (QVT) de l’Object Management Group (OMG) [Obj15a] vise précisément à normaliser l’expression des transformations entre les langages définis avec MetaObject Facility (MOF) : filtrage (Query), création d’une vue temporaire (View) et expression des transformations (Transformations).

Malheureusement, dans notre cas, nous ne disposons pas à l’avance de tous les métamodèles.

En effet, si pour la première et la seconde étape (lettres A à F de la Figure 9.2) nous disposons bien des métamodèles initiaux, finaux et des transformations à

⁵. Nous avons fait le choix de présenter seulement les contraintes majeures et réellement pertinentes pour le choix de la technologie de mise en œuvre.

effectuer, ce ne sera plus le cas à partir de l'étape suivante (lettre G), lorsque nous commencerons à intégrer les différentes normes et les différents concepts réglementaires.

Ceci est une contrainte inhérente à notre proposition. L'objet même de notre contribution sur la représentation réticulaire d'arrangements normatifs est précisément d'abandonner une maîtrise « ferme » de l'enchaînement des normes afin de reposer exclusivement sur la notion de « dépendances réglementaires ». Cette dernière permet de garantir qu'une norme est applicable sur un environnement puisque tous les concepts nécessaires à son expression et à sa vérification ont été préalablement introduits.

Ainsi, cela nous a semblé un cas d'utilisation intéressant pour les techniques orientées « aspects » (en opposition à QVT, mais toujours liée au MOF pour des raisons de compatibilité). Les « dépendances réglementaires » nous assurent de disposer des éléments attendus lors de l'application de chaque aspect⁶ et nous avancerons donc de norme en norme par des tissages successifs (Figure 9.5).

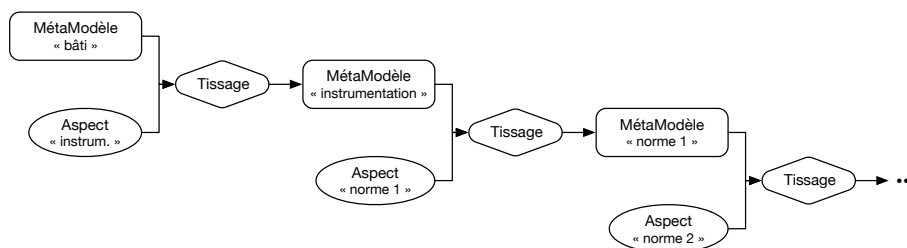


FIGURE 9.5 – Modélisation d'un environnement normé par tissages successifs

Pour le type d'application souhaitée, nous devons disposer d'aspects « structurels », permettant d'introduire les nouveaux concepts proposés par une norme, et d'aspects « comportementaux », pour le travail d'analyse et la mise en œuvre de certaines contraintes complexes.

Enfin, disposer d'une approche « programmatique » nous semblait nécessaire afin de pouvoir manipuler les métamodèles, les aspects (quels qu'ils soient), les vérifications, etc.

N'oublions pas que pour implémenter cette phase de notre méthodologie, il faut réaliser une sorte de « chef d'orchestre » qui enchainera les différentes étapes fournies par le graphe de normes : intégration du nouvel aspect, modification du modèle actuel pour se conformer à ce nouveau métamodèle, application des contraintes contenues dans l'aspect et passage à la suite.

Parmi les technologies permettant l'AOM [SSK⁺06] et répondant à ces critères, l'une d'entre elles a particulièrement retenu notre attention : Ker-Meta [Tri15, Div15, JBF11, JCB⁺15].

6. Une opération nommée le « tissage » (*aspect weaving* en anglais) dans le vocabulaire de la Programmation Orientée Aspects (POA) (*Aspect-Oriented Programming* en anglais).

9.3 KerMeta 2

KerMeta est un atelier de métamodélisation (langage et outils associés), fruit du travail de l'équipe Triskell [Tri15] (devenue DiverSE [Div15]) de l'IRISA.

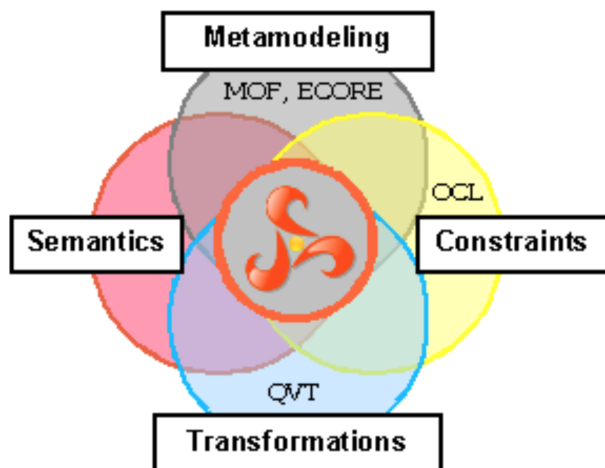


FIGURE 9.6 – Positionnement de KerMeta [DFF⁺10]

9.3.1 Introduction

Initialement, KerMeta résulte du constat de la relativement faible maîtrise par les ingénieurs en informatique « classiques » des outils, langages et concepts du paradigme IDM : modèles, métamodèles, métamétamodèles, Domain-Specific Language (DSL), Domain-Specific Modeling Language (DSML), etc.

En revanche, la Programmation Orientée Objet (POO), les diagrammes de classes UML et la notion de « conception par contrat » (pré-conditions, post-conditions et invariants) sont très largement répandus et maîtrisés, ne nécessitant aucune formation complémentaire même chez de jeunes ingénieurs.

Par conséquent, la mise au point d'un langage de métamodélisation et son outillage, à la croisée des chemins entre le paradigme IDM et ces différents concepts bien connus, présentait un certain intérêt⁷.

Dans cet esprit, le langage KerMeta présente les caractéristiques suivantes :

Fortement typé Le langage prend en charge les génériques, les expressions lambda et permet de considérer les modèles comme des « types », autorisant ainsi la découverte d'erreurs courantes dès le début du processus de développement.

Orienté « objets » Il dispose des caractéristiques « classiques » des langages objets : héritage multiple, liaison dynamique et réflexivité.

7. Attention, dans la littérature et la suite de cette section, on appellera KerMeta aussi bien le langage de métamodélisation mis en œuvre par Triskell que l'outillage qui l'accompagne. Dans la mesure du possible, nous avons essayé d'être suffisamment explicites pour permettre au lecteur de différencier les passages traitant du « langage » KerMeta de ceux traitant de l'atelier de métamodélisation intégré à Eclipse.

Orienté « modèles » Les modèles sont des « concepts primordiaux » du langage (*first-class citizen* en anglais) permettant ainsi de les manipuler aussi facilement que des objets en POO (cf. point précédent).

Orienté « aspects » Il permet le tissage d'éléments hétérogènes provenant de diverses sources (ECore⁸ [Ecl16a], KerMeta, OCL. . .) afin de construire de nouveaux outils à partir de métamodèles existants.

Conception par contrat Cela permet d'augmenter la fiabilité des outils produits (pré-conditions, post-conditions et invariants).

D'un point de vue plus technique, KerMeta est construit comme une extension d'Essential MOF (EMOF) [Obj15b] (un sous-ensemble de la spécification MOF 2.0 [Obj15b]). De cette manière, tout modèle EMOF constitue un programme KerMeta valide, et réciproquement.

L'atelier KerMeta est un environnement complet, basé sur un DSL orienté objet et optimisé pour l'ingénierie des métamodèles, disponible sous la forme d'un plug-in Eclipse, permettant entre autres :

- la spécification de syntaxes abstraites, avec une double sémantique statique (via OCL) et dynamique, en connexion avec la syntaxe concrète ;
- la définition de modèles et de métamodèles de prototypage et de simulation, via la définition de modèles exécutables ;
- la transformation de modèles ;
- le tissage d'aspects.

En raison de sa mise en œuvre (un plug-in Eclipse), KerMeta dispose immédiatement d'un environnement de modélisation complet.

À celui-ci, il ajoute (notamment) un compilateur, permettant d'exécuter et de déployer un programme KerMeta dans un environnement Java classique, les éléments nécessaires à la coloration syntaxique et l'auto-complétion de code, divers outils de transformations (import/export) comme `ecore2kermeta`, `kermat2ecore`, `kermeta2xmi`, `xmi2kermeta` ou `xmi2ecore`.

9.3.2 Syntaxe

Dans la mesure où l'objectif est d'illustrer la mise en œuvre de notre contribution, nous détaillons seulement les éléments pertinents de la syntaxe KerMeta dans ses orientations « objets », « modèles » et « aspects »⁹ sans entrer trop dans les détails.

Langage orienté « objets »

Pour illustrer les aspects « objets » de KerMeta, nous utiliserons un exemple simpliste sur les livres avec un premier code source déclarant des livres (9.1), permettant d'illustrer les concepts objets « classiques », et un second déclarant des lecteurs (9.2), permettant d'illustrer la notion de vérification avec une exception.

8. ECore est une variante (à quelques nuances près) d'EMOF définie dans l'Eclipse Modeling Framework (EMF).

9. La suite de cette section est largement inspirée du manuel de référence KerMeta [DFF⁺10].

```

// Un ecrivain
class Ecrivain {
  attribute nom : kermeta::standard::String
}

// Le concept abstrait d'un document
abstract class Document {
  reference auteur : Ecrivain
  attribute texte : kermeta::standard::String
}

// Un "vrai" document
class Livre inherits Document {}

// Un type "particulier" de document
class LivrePourEnfant inherits Livre {
  attribute ageMinimal : kermeta::standard::Integer
}

```

Listing 9.1 – Exemple de l'utilisation des concepts « objet » basiques dans KerMeta

```

// Une exception particuliere pour la verification de l'age
class AgeException inherits kermeta::exceptions::Exception {}

// Le concept abstrait d'un lecteur de livres pour enfants.
abstract class Lecteur {
  operation lire(livre : LivrePourEnfant) : Void is abstract
}

// Un lecteur "enfant"
class Enfant inherits Lecteur {
  attribute age : Integer

  operation initialize(age : Integer) : Enfant is
  do
    self.age := age
    result := self
  end

  // Leve une exception si l'enfant n'a pas l'age requis
  operation lire(livre : LivrePourEnfant) : Void is
  do
    if age < livre.ageMinimal then
      raise AgeException.new
    end
  end
}

```

Listing 9.2 – Exemple de vérifications basiques dans KerMeta en utilisant des exceptions

Langage orienté « modèles »

Comme nous l'avons indiqué, KerMeta est une extension d'EMOF et fournit les outils nécessaires pour manipuler simplement des modèles.

Considérons le diagramme de classes UML de la Figure 9.7.

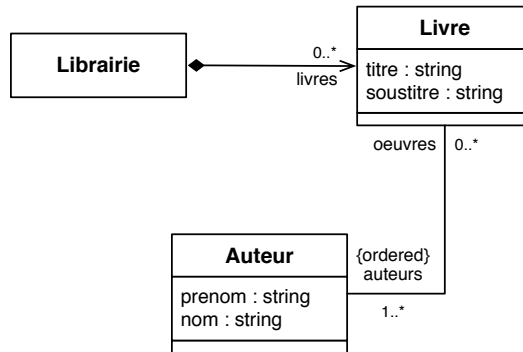


FIGURE 9.7 – Modèle UML simpliste d'une librairie [DFF+10]

Celui-ci peut simplement être représenté sous la forme des classes KerMeta du code source 9.3.

```

class Librairie {
  attribute livres : set Livre[0..*]
}

class Livre {
  attribute titre : kermeta::standard::String
  attribute soustitre : kermeta::standard::String

  reference auteurs : oset Author[1..*]#oeuvres
}

class Auteur {
  attribute prenom : kermeta::standard::String
  attribute nom : kermeta::standard::String

  reference oeuvres : set Livre[0..*]#auteurs
}
  
```

Listing 9.3 – Représentation du modèle UML d'une librairie

L'association est l'un des concepts clés dans l'utilisation et la définition des modèles, KerMeta fournit donc les outils pour les manipuler et les représenter facilement.

Le MOF définit le concept de « propriété » qui généralise les notions d'attributs et d'associations (composite ou non) que l'on peut trouver dans UML. KerMeta fournit les moyens de distinguer tous les cas possibles, le mot-clé **attribute** définissant un lien de « composition », alors que le mot-clé **reference** permet de définir simplement une association.

Le formalisme proposé par KerMeta permet de représenter tout type d'association ou de composition [DFF⁺10].

Langage orienté « aspects »

KerMeta fournit les éléments de langage « aspects » afin de pouvoir tisser des métamodèles avec des aspects structurels ou comportementaux.

Pour réaliser cela, KerMeta dispose d'un opérateur de composition statique `require` permettant d'étendre un métamodèle existant avec de nouveaux éléments comme des propriétés, des opérations, des contraintes ou des classes. Cet opérateur permet de définir ces différents aspects séparément, et de les intégrer ensuite automatiquement dans le métamodèle, permettant ainsi de réutiliser facilement des métamodèles existants ou de les scinder en plusieurs briques à assembler selon la situation.

De plus, cette façon de procéder se rapproche de la bonne pratique de l'Open/Closed Principle (OCP) dans les langages orientés objet. Dans le cas présent, on « étend » un métamodèle plutôt que de modifier directement son propre métamodèle. Cela permet d'organiser les aspects orthogonaux ou métiers dans des unités différentes intégrées lors des différentes phases de tissage selon les besoins.

Avec ce mécanisme, KerMeta peut prendre en charge l'ajout de nouvelles méta-classes, de nouvelles sous-classes, de nouvelles méthodes, de nouvelles propriétés, de nouveaux contrats à des métamodèles existants.

Grâce à cet opérateur de composition, KerMeta reste un « noyau » que l'on peut facilement intégrer dans ses propres développements de métamodélisation. Cette approche est en symbiose avec les objectifs développés dans ce mémoire.

9.4 Mise en œuvre et résultat

KerMeta fournit le cœur de notre traitement par aspect, mais il reste insuffisant à lui seul.

Il faut l'entourer d'une logistique complémentaire : réalisation des différents modèles, enchaînement des différentes phases de tissage, etc. la supervision générale de ces outils est le rôle du « chef d'orchestre » décrit précédemment.

9.4.1 Outillage

Outre « KerMeta 2 Studio » [Tri16], nous avons fait appel à Eclipse dans sa version orientée « modélisation » (EMF¹⁰ [Ecl16b, SBMP08]). Dans cette version, nous disposons des outils nécessaires pour la création de modèles conformes à un métamodèle donné.

De plus, KerMeta étant fourni sous la forme d'un plug-in Eclipse, nous pouvons basculer facilement entre les différentes activités à mener : modélisation, tissage, etc.

10. EMF est une version « standard » d'Eclipse dans laquelle tous les plug-ins pertinents de l'Eclipse Modeling Project (EMP) ont été préinstallés.

9.4.2 Architecture générale

L'architecture mise en place est décrite dans la Figure 9.8.

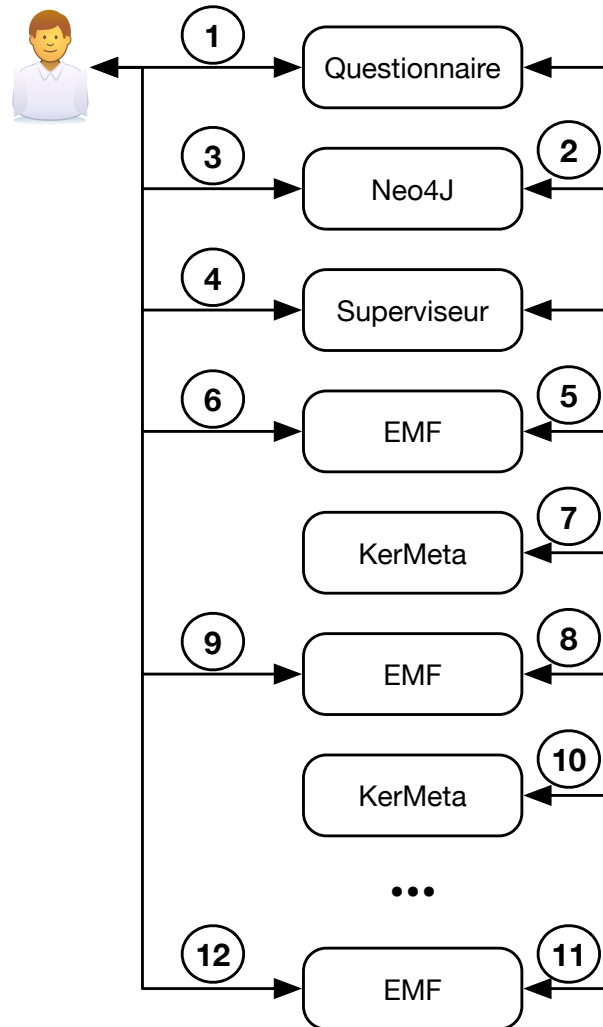


FIGURE 9.8 – Architecture de notre mise en œuvre de l'outil d'aide à la qualification d'un projet d'environnement normé

Les éléments composant cette architecture ont été précédemment décrits. On reconnaît :

- le questionnaire servant à identifier le type d'environnement parmi ceux précédemment analysés ;
- Neo4J pour contenir le graphe de normes ;
- le superviseur¹¹ assurant le lien entre les outils liés à la modélisation et l'utilisateur ;

11. Il a été précédemment évoqué sous le vocable de « chef d'orchestre » dans ce mémoire.

- EMF et KerMeta pour la modélisation et l'analyse du projet d'environnement normé.

Examinons maintenant les interactions entre l'utilisateur et ces composants (et entre les composants eux-mêmes).

9.4.3 Fonctionnement général

D'un point de vue général, l'algorithme est le suivant (les numéros de la liste correspondent à ceux de la Figure 9.8) :

1. L'utilisateur répond au questionnaire pour obtenir le type d'environnement normé concerné.
2. Le questionnaire interroge Neo4J avec le type d'environnement identifié.
3. L'utilisateur obtient de Neo4J la liste des étapes de modélisation correspondantes.
4. L'utilisateur transmet cette liste d'étapes au chef d'orchestre qui se charge d'enchaîner les étapes suivantes en rendant la main à l'utilisateur dans les phases de modélisation.
5. Le chef d'orchestre commence par proposer EMF avec le métamodèle initial (Bâtiment) comme référence.
6. L'utilisateur peut alors modéliser la topologie de son projet. Une fois terminé :
 - il enregistre son modèle ;
 - quitte EMF ;
 - « rend la main » au superviseur.
7. Le chef d'orchestre procède au tissage de l'aspect lié à l'instrumentation sur le métamodèle du bâtiment.
8. Le chef d'orchestre propose EMF avec l'ancien modèle et le nouveau métamodèle.
9. L'utilisateur peut alors mettre à jour son modèle pour le remettre en conformité avec le nouveau métamodèle et ajouter des éléments correspondant aux nouveaux concepts introduits. Une fois terminé :
 - il enregistre son modèle ;
 - quitte EMF ;
 - « rend la main » au superviseur.
10. Les étapes 7, 8 et 9 sont reprises pour chaque aspect contenu dans les différentes étapes. Cela nous amène aux étapes 11 et 12 correspondant à l'aspect et au modèle final de l'environnement normé à bâtir.

Dans cette mise en œuvre, nous avons fait le choix d'enregistrer dans chaque nœud la liste d'aspects permettant de représenter la cible du nœud. Ainsi à l'étape 3, c'est cette liste de fichiers que nous devrions transmettre au superviseur.

9.4.4 Résultat

Si nous avons pu valider la notion de modélisation « itérative » d'un environnement normé, nous n'avons malheureusement pas pu, dans le cadre de ce doctorat, atteindre le degré d'automatisation souhaité. Nous avons manuellement

déroulé les différentes étapes, jouant nous-mêmes le rôle du « chef d'orchestre » entre les différents outils.

Nous considérons toujours cela comme possible et notre mise en œuvre évoluera dans ce sens (cela fait partie de nos pistes de réflexion actuelles, cf. Section 14.3)...

9.5 Conclusion

En conclusion, nous avons montré qu'il existe une approche technique viable pour nos objectifs, reposant sur un sous-domaine particulier de l'IDM avec ses outils et ses axes de recherches propres. Depuis la réalisation de notre mise en œuvre, la solution KerMeta a très largement évolué, nous confortant dans son adéquation avec nos objectifs.

Avec cet embryon d'implémentation, nous sommes arrivés à obtenir de premiers résultats encourageants quant à la viabilité de notre contribution et son implémentation complète.

Dans cette première mise en œuvre, nous avons fait le choix de nous concentrer sur les aspects « conceptuels » (représentation du métamodèle initial de l'environnement, représentation de normes sous la forme d'aspects, etc.) ainsi que sur l'« AOM » afin de valider l'application des aspects pour bâtir un métamodèle de manière incrémentale et ainsi construire un environnement normé « couche par couche ».

De plus, nous avons constaté que l'AOM (et plus généralement les approches liées à l'IDM) sont très bien adaptées à la diversité des contraintes rencontrées dans la réglementation encadrant les environnements confinés médicaux : mélange de contraintes documentaires et techniques avec des conséquences tant dans les justificatifs à produire que dans les vérifications « concrètes » à effectuer (à la conception et à l'utilisation).

En contrepartie, nous avons dû éluder « l'outillage » qu'il reste à réaliser pour permettre à l'utilisateur de manipuler facilement les métamodèles et les aspects (« l'enrobage » de notre contribution). En raison du contexte très particulier de réalisation de cette partie des travaux, nous avons préféré nous concentrer sur le cœur de la contribution plutôt que sur sa périphérie.

Au terme de ce chapitre, les pistes de recherche ont été clairement identifiées.

Nous devons poursuivre notre exploration de la représentation des normes « par aspects » pour traiter des corpus plus importants et des arrangements normatifs plus complexes, plus élaborés. L'observation des travaux connexes aux nôtres nous a montré que c'est principalement dans la volumétrie que l'on rencontre des difficultés dans ce type de travaux.

L'objectif final devant certainement représenter l'aboutissement de nos efforts, d'un point de vue industriel, est l'intégration de cette étape de modélisation et de vérification dans l'outillage existant (c'est-à-dire sans ajouter de logiciel supplémentaire venant modifier le flot de conception d'un environnement normé), par conséquent l'intégration à la MNB discutée dans la Section 14.3 (p. 14.3).

Maintenant que nous avons détaillé nos contributions et leurs possibles mises en œuvre, nous pouvons les illustrer sur un cas concret (tout en protégeant la propriété intellectuelle de Nocosium) abordant les deux premières étapes de notre méthodologie.

Illustration

Pour illustrer la méthodologie décrite dans cette thèse, nous avons fait le choix d'un cas simple, mais représentatif.

Ainsi, nous allons maintenant décrire cet environnement « fil rouge » afin de pouvoir y projeter chacune de nos contributions.

Dans la suite de ce chapitre, nous commencerons par présenter nos motivations et l'environnement étudié (Section 10.1).

Ensuite, nous avons fait le choix de procéder en plusieurs étapes successives :

- La Section 10.2 présentera un arrangement normatif partiel pour le type d'environnement considéré, en détaillant simplement l'objectif « général » de chaque texte.
- La Section 10.3 analysera ces mêmes textes pour illustrer la construction itérative du graphe de normes.
- La Section 10.4 présentera un questionnaire correspondant à cette analyse.
- La Section 10.5 détaillera le traitement du graphe effectué suite à son interrogation pour extraire les étapes.
- Les Sections 10.6, 10.7 illustreront l'interrogation du graphe pour préparer la phase de modélisation.
- La Section 10.8 illustrera rapidement le déroulement de la modélisation de notre cas d'illustration.

Enfin, nous dresserons une rapide conclusion dans la Section 10.9 avant d'aborder, au prochain chapitre, les transferts technologiques.

10.1 Présentation

Une Unité de Reconstitution Centralisée des Cytotoxiques (URCC) assure la réalisation des préparations de chimiothérapie anticancéreuse et la gestion des traitements en essai clinique. L'équipe de ce type d'unité est généralement constituée de pharmaciens et de préparateurs en pharmacie, spécialisés dans

le traitement par chimiothérapie anticancéreuse, obligatoirement placés sous la responsabilité d'un pharmacien désigné.

Cette unité est ce que l'on nomme une Pharmacie à Usage Intérieur (PUI), un terme que nous allons définir dans les sections suivantes.

10.1.1 Cas réel étudié

Pour étayer notre analyse d'URCC, nous nous appuyerons sur un projet réalisé par Nocosium pour lequel nous disposons de l'ensemble des éléments techniques

Pour d'évidentes raisons de confidentialité, tous les détails techniques de cet environnement ont été modifiés afin de protéger le client tout en conservant la pertinence de l'analyse.

10.1.2 Motivations

Nous avons choisi d'utiliser ce type d'environnement, et ce cas précis, parce qu'il constitue un exemple « efficace » d'application de notre approche.

Les URCC font intervenir l'ensemble des éléments techniques présentés dans notre approche et d'une manière tout à fait abordable pour les non-spécialistes, pour peu que l'on sache se concentrer sur les aspects fondamentaux.

Nous avons fait le choix de simplifier largement le problème pour ne pas surcharger inutilement les explications. Notre objectif est de fournir au lecteur une situation représentative de la réalité, tout en restant suffisamment concise pour ne pas nécessiter, à elle seule, plusieurs chapitres d'explications.

10.1.3 Structure de l'environnement étudié

La Figure 10.1 présente la structure de l'environnement « fil rouge ».

C'est une représentation du bâti avec les différentes pièces (incluant leur volume, usage et pression relative), les portes et quelques équipements (un réfrigérateur médical et deux hottes à flux laminaire).

Comme exposé, l'objectif est de préparer des médicaments pour les traitements oncologiques sous des hottes à flux laminaire. Ici, le principal risque de contamination est celui de l'environnement extérieur (« sale ») sur les produits manipulés.

Par conséquent, il faut — entre autres mesures — empêcher l'air « sale » d'atteindre les préparations et bien différencier le circuit « propre » du circuit « sale »¹. L'environnement sera donc constitué (conformément aux explications de la Section 2.3, à la p. 23) d'un ensemble de paliers de protections.

Nous allons maintenant fournir quelques détails complémentaires afin d'assurer une bonne compréhension du cas considéré :

Exterieur désigne le lieu dans lequel se situe (ou se situera) l'URCC.

— La porte « A » est donc la porte d'entrée.

1. Dans le cas réel, ce n'est pas le seul vecteur de contamination à étudier, mais nous avons fait le choix de celui-ci dans l'exemple « fil rouge ». Il a le mérite d'être particulièrement évocateur et d'être facilement compréhensible.

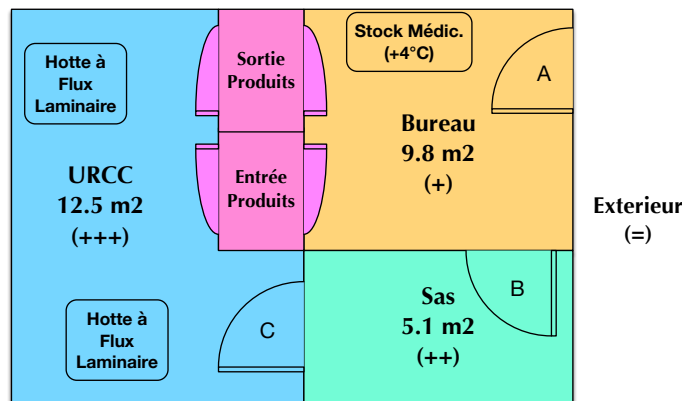


FIGURE 10.1 – Structure de l'environnement normé

- La pression atmosphérique extérieure sert de référence pour la mesure de la pression relative du bureau.

Bureau désigne un dégagement permettant au personnel de saisir les informations pertinentes sur un ordinateur (absent de cette représentation), de stocker des produits dans un réfrigérateur à 4 °C et de faire circuler les produits (cf. les passe-plats).

- Ce bureau est en surpression par rapport à l'extérieur afin d'empêcher l'air d'entrer. L'écart de pression recommandé [Age13] est de 15 à 45 Pa.
 - La porte « A » se ferme donc naturellement par l'effet de « surpression »².
 - Sa pression atmosphérique sert de référence pour la mesure de la pression relative du sas.
- Le stockage médical posera des contraintes particulières pour s'assurer du maintien d'une température « conforme ».

Sas sert de palier entre le bureau et la salle de préparation.

- Les manipulateurs y changent de tenue.
- Il possède deux zones : une zone d'entrée et une zone « propre », séparée de la première par un banc fixe néanmoins démontable pour les entrées exceptionnelles de gros matériel (hottes, servantes, tabouret), qui permet l'entrée du personnel et du matériel en salle de préparation³.
- Le sas est en surpression par rapport au bureau afin d'empêcher l'air d'entrer dans la salle de préparation. L'écart de pression minimal est de 15 Pa (donc, +30 par rapport à l'environnement).
 - La porte « B » se ferme donc naturellement par l'effet de « surpression ».
 - Sa pression atmosphérique sert de référence pour la mesure de la pression relative de l'URCC.

2. En effet, le différentiel de pression « pousse » la porte du bureau vers l'extérieur, entraînant donc sa fermeture « automatique ». Par corollaire, si la porte était installée dans le sens inverse, elle aurait tendance à s'ouvrir si elle n'est pas maintenue par le montant.

3. Le banc n'est pas représenté sur la Figure 10.1 pour ne pas nuire à sa lisibilité.

URCC est la salle de préparation.

- Elle contient deux hottes à flux laminaire sous lesquelles seront effectuées les préparations.
- Les outils et produits empruntent les passe-plats (cf. ci-dessous).
- L'URCC est en surpression par rapport au sas afin d'empêcher l'air d'entrer dans la salle de préparation. L'écart de pression minimal est de 15 Pa (donc, +45 par rapport à l'environnement).
- La porte « C » se ferme donc naturellement par l'effet de « surpression ».

Afin de faciliter le contrôle des surpressions et de limiter les contaminations aéroportées, un contrôle d'accès est mis en place :

- Les portes « A » et « B » ne peuvent s'ouvrir simultanément.
- Les portes « B » et « C » ne peuvent s'ouvrir simultanément.

Pour éviter les contaminations par les outils et les produits, et séparer les produits « propres » des produits « sales », deux passe-plats sont disposés :

- Le premier « entrée produits » permet de faire entrer des produits « propres » dans la salle de préparation.
- Le second « sortie produits » permet de faire sortir les produits « sales » de la salle de préparation.
- Ces passe-plats sont « décontaminant », c'est-à-dire qu'ils permettent de faire une première décontamination des outils salis par la préparation avant de les faire passer par le bureau.
- Naturellement, ces passe-plats fonctionnent comme des écluses et leurs deux portes ne peuvent s'ouvrir simultanément.

En conclusion, nous avons un environnement dédié à la préparation de médicaments cytotoxiques, particulièrement fragiles par rapport à l'environnement externe.

Par conséquent, plusieurs méthodes d'asepsie progressive sont mises en place :

- Des paliers de surpression afin d'empêcher l'air de l'extérieur d'atteindre la salle de préparation (chaque pièce agissant comme un « mur » d'air supplémentaire).
- Deux chemins différents, celui du matériel (passant par le bureau puis les passe-plats) et celui du personnel (passant par le bureau puis le sas).
- Une séparation claire du « propre » et du « sale » avec des équipements locaux permettant une première décontamination (passe-plats pour le matériel et sas de changement pour le personnel).

10.2 Arrangement normatif considéré

L'environnement visé est une URCC, la première étape consistera donc à bâtir l'arrangement normatif encadrant ce type d'environnement. Comme nous l'avons indiqué, cette première étape était le travail de Nocosium et nous nous bornons ici à fournir une forme simplifiée au minimum nécessaire à la compréhension de notre méthodologie. Ainsi, certains aspects à régler pour une conformité réelle de l'environnement ont été volontairement éludés.

L'arrangement considéré est illustré dans la carte euristique de la Figure 10.2.

Dans cette section, nous allons seulement lister et décrire brièvement ces documents. La section suivante sera quant à elle dédiée à leur analyse.

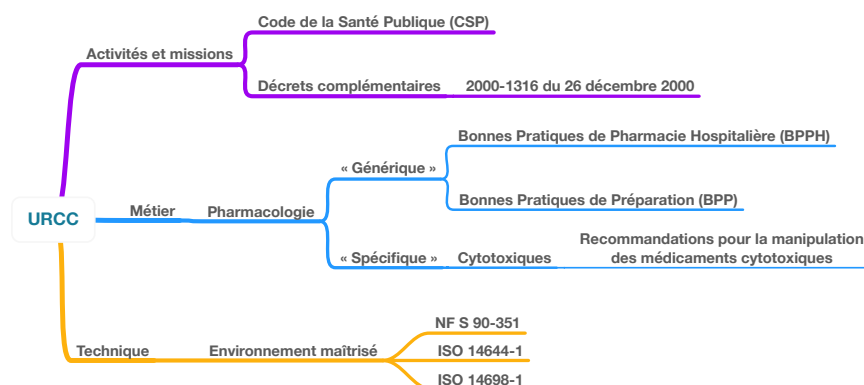


FIGURE 10.2 – Hiérarchie normative

10.2.1 Code de la Santé Publique

Le Code de la Santé Publique (CSP) [Min] détermine le champ du droit de la santé publique.

Ses six parties (elles-mêmes divisées en livres, titres, chapitres et articles) couvrent l'ensemble des domaines de la santé : droit des personnes en matière de santé, droit particulier propre à certaines populations ou maladies et dépendances, droit des professions de santé, droit des produits de santé et droit des établissements et services de santé.

Le CSP est régulièrement complété par des décrets modifiant son contenu.

10.2.2 Décret n° 2000-1316 du 26 décembre 2000

Ce décret insère plus d'une centaine de nouveaux articles dans le CSP statuant sur la définition, les activités autorisées, etc. d'une PUI. Parmi ces articles, l'Art. R. 5104-20 prévoit que les PUI fonctionnent conformément aux Bonnes Pratiques de Pharmacie Hospitalière.

Il a depuis été mis à jour par plusieurs autres décrets (Décret n° 2002-796 du 3 mai 2002, Décret n° 2004-451 du 21 mai 2004, Décret n° 2004-802 du 29 juillet 2004 et Décret n° 2005-840 du 20 juillet 2005) peu pertinents pour l'illustration de notre approche.

10.2.3 Bonnes Pratiques de Pharmacie Hospitalière

Les « Bonnes Pratiques de Pharmacie Hospitalière » (dites BPPH) sont un recueil français de bonnes pratiques [Dir01], créé en 2001, qui encadre le domaine « métier » des pharmacies hospitalières.

Il constitue une particularité française, la Communauté Economique Européenne (CEE) fournissant alors seulement un « guide de Bonnes Pratiques de Fabrication » définissant seulement des objectifs à atteindre.

10.2.4 Bonnes Pratiques de Préparation

L'article L. 5121-5 du CSP impose que la préparation de médicaments doive être réalisée en conformité avec les bonnes pratiques et que ces bonnes pratiques prévoient notamment les modalités de suivi permettant d'assurer la traçabilité des médicaments.

Les BPP [Age07] exposent les principes des bonnes pratiques de préparation, qui s'appliquent à l'ensemble des préparations, notamment magistrales, officinales et hospitalières, réalisées dans les établissements disposant d'une Pharmacie à Usage Intérieur dument autorisée ou dans les officines de pharmacie conformément aux textes en vigueur.

10.2.5 Recommandations pour la manipulation des médicaments cytotoxiques

Les recommandations « métier » figurant dans ce document [Cen02] concernent la gestion, la préparation, l'administration, le stockage et l'élimination des médicaments cytotoxiques.

En raison des risques encourus pour le personnel et du respect de la stérilité des préparations, le groupe de travail a envisagé deux solutions dans les établissements de santé : soit une unité de reconstitution centralisée, soit des unités de soins ayant un équipement de protection pour la manipulation de ces médicaments.

10.2.6 NF S 90-351

Cette norme technique [Age13] intitulée « Etablissements de santé – Zones à environnement maîtrisé – Exigences relatives à la maîtrise de la contamination aéroportée » précise les exigences de sécurité sanitaire pour la conception, la construction, l'exploitation, la maintenance, le contrôle et l'utilisation des installations de traitement et de maîtrise de la qualité de l'air dans les établissements de santé.

10.2.7 ISO 14644-1

Cette norme technique [Int99a] intitulée « Salles propres et environnements maîtrisés apparentés – Partie 1 : Classification de la propreté de l'air » définit la classification de la propriété particulière des salles propres et des espaces à empoussièrement maîtrisé pour les trois états d'occupation : « après construction », « au repos » ou « opérationnel ». Il mentionne des recommandations quant aux conditions de conformité.

10.2.8 ISO 14698-1

Cette norme technique [Int03] intitulée « Salles propres et environnements maîtrisés apparentés – Maîtrise de la biocontamination – Partie 1 : Principes

généraux et méthodes » établit les principes et la méthodologie fondamentale d'un Système formalisé de maîtrise de la biocontamination, destiné à évaluer et à maîtriser la biocontamination, dans le cadre de l'application des technologies des salles propres. Elle spécifie les méthodes requises pour assurer une surveillance cohérente des zones à risque et pour appliquer les mesures de maîtrise adaptées au degré de risque concerné.

10.3 Analyse normative

Maintenant les différents textes présentés, nous allons pouvoir les reprendre pour les analyser. Afin de garantir la propriété intellectuelle de Nocosium et ne pas complexifier inutilement la lecture et la compréhension de cette étude de cas, nous avons une nouvelle fois cherché le bon compromis entre réalisme de l'analyse et complexité de la présentation⁴.

Ainsi, volontairement, nous avons fait le choix de ne pas être (trop) générique, et traiter toutes les URCC, mais de nous concentrer sur les éléments nécessaires pour le cas d'illustration.

10.3.1 Code de la Santé Publique (CSP)

Les Pharmacies à Usage Intérieur sont définies (fonctions et missions) dans la partie législative du Code de la Santé Publique (5^e partie, 1^{er} livre, 2^e titre, 6^e Chapitre).

L'article L. 5126-1 spécifie « les établissements de santé et les établissements médicosociaux dans lesquels sont traités des malades [...] peuvent disposer d'une ou plusieurs pharmacies à usage intérieur dans les conditions prévues au présent chapitre. L'activité des Pharmacies à Usage Intérieur est limitée à l'usage particulier des malades dans les établissements de santé ou médicosociaux où elles ont été constituées ou qui appartiennent au syndicat interhospitalier. »

Ainsi, le CSP n'est pas une norme à transformer en contrainte, mais plutôt un support pour nourrir le questionnaire, celui qui nous servira dans la phase d'élicitation (cf. Chapitre 7, p. 99).

- Dans un premier temps, nous obtenons le graphe en Figure 10.3. Ce sont :
- trois nœuds symboliques représentant les concepts généraux⁵ (URCC, PUI, cytotoxiques) que nous pouvons d'ores et déjà inférer à la lecture du CSP ;
 - deux nœuds représentant le bâti et l'instrumentation. Ceux-ci constituent les premières étapes pour la modélisation d'un environnement normé et ne sont donc pas issus de l'arrangement normatif, mais fournis dans le cadre de la mise en œuvre de notre méthodologie⁶.

Pour l'instant, ces deux groupes de nœuds ne sont pas reliés, aucune contrainte n'a été exprimée.

4. Dans cette section, nous restons au niveau du graphe de normes, le contenu des nœuds sera abordé ultérieurement.

5. Dans la mesure où ces concepts n'ont pas un numéro de version propre et que nous en sommes à la première révision de notre analyse, nous mettons « 1_1 »

6. Par conséquent, nous imposons nous-mêmes les numéros de versions de ces nœuds.

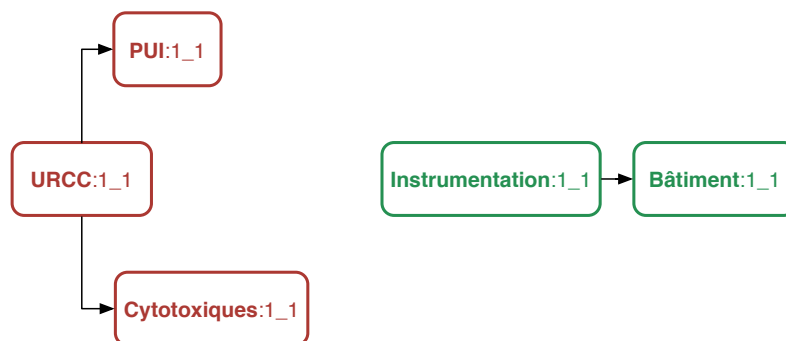


FIGURE 10.3 – Illustration : graphe après le CSP

Ensuite, nous avons, parmi les articles notables du CSP sur les PUI concernant les réglementations tierces, ceux imposant le respect des BPP (cf. 10.2.4) ainsi que des BPPH (cf. 10.2.3).

Cela nous permet d'obtenir le graphe présenté à la Figure 10.4 où nous avons simplement inséré les nœuds correspondants, qu'il restera encore à traiter.

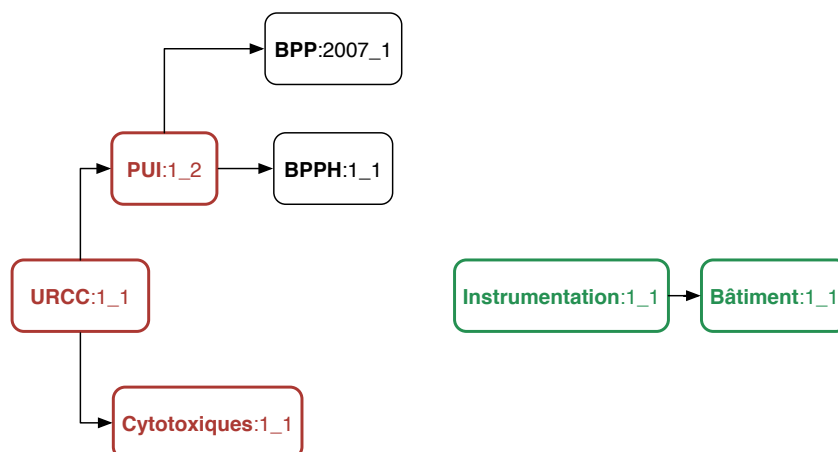


FIGURE 10.4 – Illustration : graphe après le Décret n° 2000-1316

10.3.2 Bonnes Pratiques de Pharmacie Hospitalière

Ce sont principalement des contraintes structurelles et documentaires. Les contraintes structurelles sont liées à l'imposition des différentes zones constituant une Pharmacie Hospitalière (réception, stockage, préparation, distribution et dispensation) ; tandis que les aspects documentaires sont imposés par les contraintes organisationnelles (supervision, hiérarchie, etc.) et de documentation des process employés.

À l'issue de cette étape, nous avons le graphe en Figure 10.5. Nous avons

complété les dépendances du nœud BPPH.

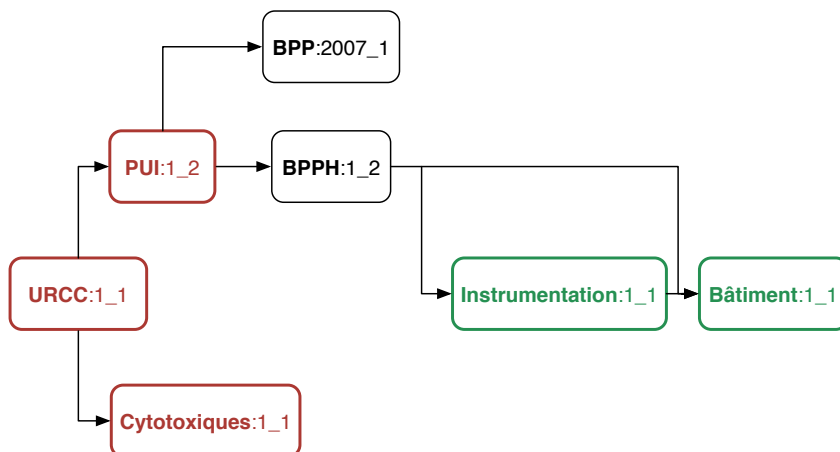


FIGURE 10.5 – Illustration : graphe après les BPPH

10.3.3 Bonnes Pratiques de Préparation (BPP)

Cette norme encadre l'activité métier de la préparation pharmacologique ; avec des spécificités en fonction du type d'environnement (PUI, officine, etc.) et du type de préparation (stérile, etc.).

À l'issue de cette étape, nous avons le graphe en Figure 10.6. Nous avons complété les dépendances du nœud BPP.

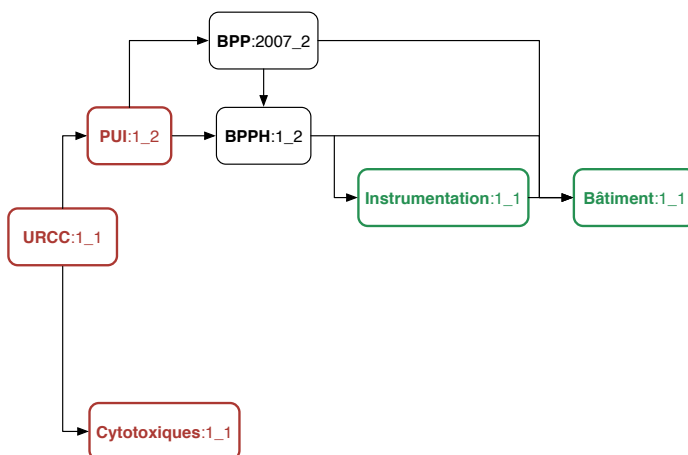


FIGURE 10.6 – Illustration : graphe après les BPP

10.3.4 Recommandations pour la manipulation des médicaments cytotoxiques

Cette norme métier spécifie les différentes configurations de locaux autorisées en fonction du matériel de préparation installé, du nombre de pièces disponibles et de leur classement ISO. Il spécifie également les différentes opérations de maintenance et de contrôle à réaliser.

À l'issue de cette étape, nous avons le graphe en Figure 10.7. Nous avons introduit le nœud « Recommandations Cytotoxiques » ainsi que ses dépendances.

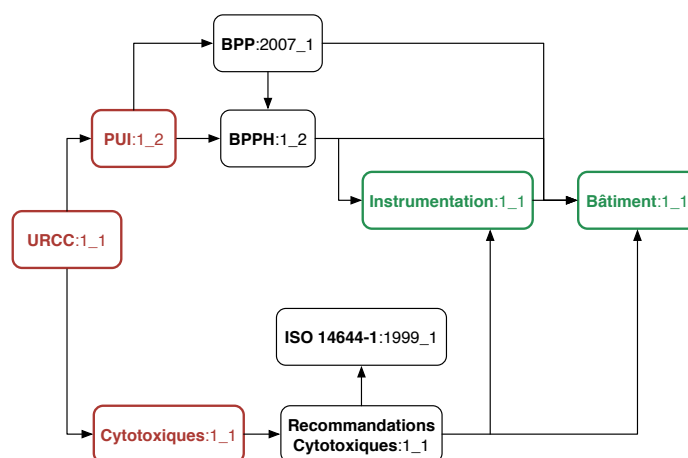


FIGURE 10.7 – Illustration : graphe après les recommandations cytotoxiques

10.3.5 ISO 14644-1

Cette norme est primordiale dans le domaine de la propreté de l'air dans la mesure où elle définit les catégories de propreté de l'air : ISO 1 à ISO 9. Chaque catégorie contraint le nombre maximal de particules admises par taille de particule ; l'ISO 1 étant la classe la plus contraignante.

À l'issue de cette étape, nous obtenons le graphe en Figure 10.8, où nous avons complété le nœud « ISO 14644-1 ».

10.3.6 NF S 90-351

Cette norme est incontournable pour les zones à environnement maîtrisé, elle s'applique pour tous les projets. Outre les contraintes documentaires, elle définit la notion de classe de risque (1 à 4) dépendant du type d'environnement et imposant des classes ISO, des températures, des vitesses d'air, etc.

À l'issue de cette étape, nous obtenons le graphe en Figure 10.9, où nous avons introduit le nœud « NF S 90-351 ».

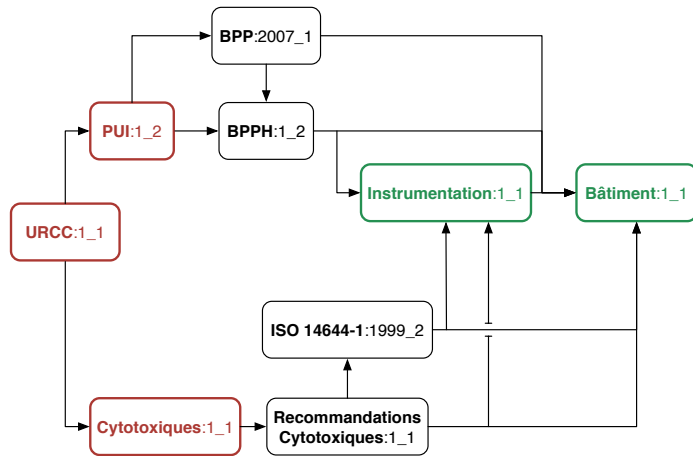


FIGURE 10.8 – Illustration : graphe après l’ISO 14644-1

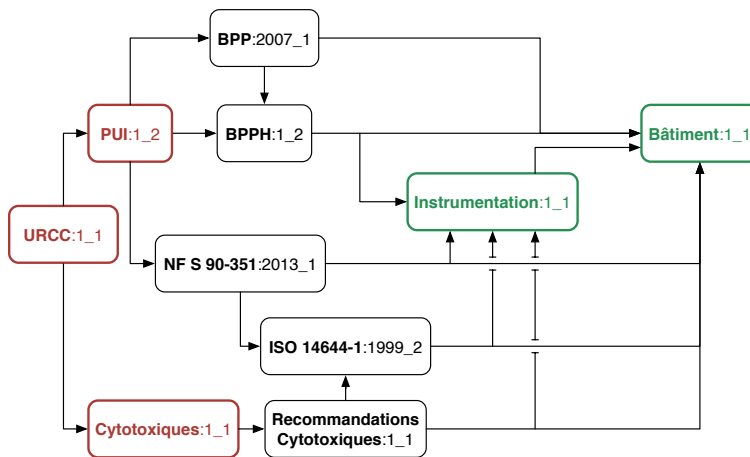


FIGURE 10.9 – Illustration : graphe après la NF S 90-351

10.3.7 ISO 14698-1

Cette norme a principalement des conséquences documentaires. Elle impose la définition d'un Système formalisé (analyse de risque, niveaux cibles, d'alertes et d'action) et son application.

Dans la mesure où cette norme est documentaire (et par conséquent, présente peu d'intérêt pour ce mémoire), nous avons fait le choix de ne pas l'intégrer dans le graphe de norme. Néanmoins, nous estimions important d'illustrer concrètement ce type de normes, qui n'impose pas des contraintes directement sur l'environnement, mais plutôt sur le procédé qui s'y déroule, en obligeant son analyse, sa documentation et son encadrement.

10.3.8 Conclusion

Nous avons maintenant parcouru l'ensemble des documents présents dans l'arrangement normatif. Nous avons volontairement occulté le contenu des nœuds (aspects et métamodèles) afin de nous concentrer sur les aspects purement « réticulaires » (la notion de « dépendances » normatives).

Dans notre mise en œuvre avec Neo4J, cela se traduit par la Figure 10.10.

Nous allons maintenant présenter le questionnaire issu de cette même analyse.

10.4 Questionnaire

Pour rappel, le questionnaire est le document utilisé dans la première étape de notre méthodologie afin d'assurer un langage commun entre l'utilisateur et notre outil, de même qu'à guider l'utilisateur dans ses différents choix⁷. Ce questionnaire est bâti parallèlement à l'analyse normative, en complément de celle-ci, afin de servir d'introduction à la sélection du type d'environnement correspondant à l'objectif de l'utilisateur (cf. Figure 5.8 p. 75).

Dans le cas présent, l'étude normative présentée précédemment nous permet d'aboutir à la carte euristique en Figure 10.11⁸. Les « notes internes » n'ont pas vocation à apparaître « réellement » et sont plus des guides de lecture pour la compréhension du lecteur dans le cadre de ce mémoire.

En raison de notre mise en œuvre particulière de la représentation d'un arrangement normatif, nous devons aboutir à la fin du questionnaire à un nœud connu du graphe de normes. Lors de nos expérimentations, le questionnaire était une simple suite de questions (tel qu'illustré ci-dessous), mais rien n'empêcherait d'en faire une version interactive, directement liée au graphe de normes.

Pour le cas présent :

1. Question : quel est le type de l'activité (au sens le plus large) qui sera menée dans l'environnement ciblé ?

La réponse attendue dans le cadre de cette illustration est « *pharmacologie* ». D'autres réponses possibles auraient été, par exemple, « *animalerie* » ou « *chirurgie* ».

7. C'est une forme d'élicitation du besoin utilisateur (dans une forme vraiment très légère).

8. Attention, nous présentons ici seulement les choix liés à une URCC, en aucun cas une carte euristique généraliste pour les environnements confinés médicaux.

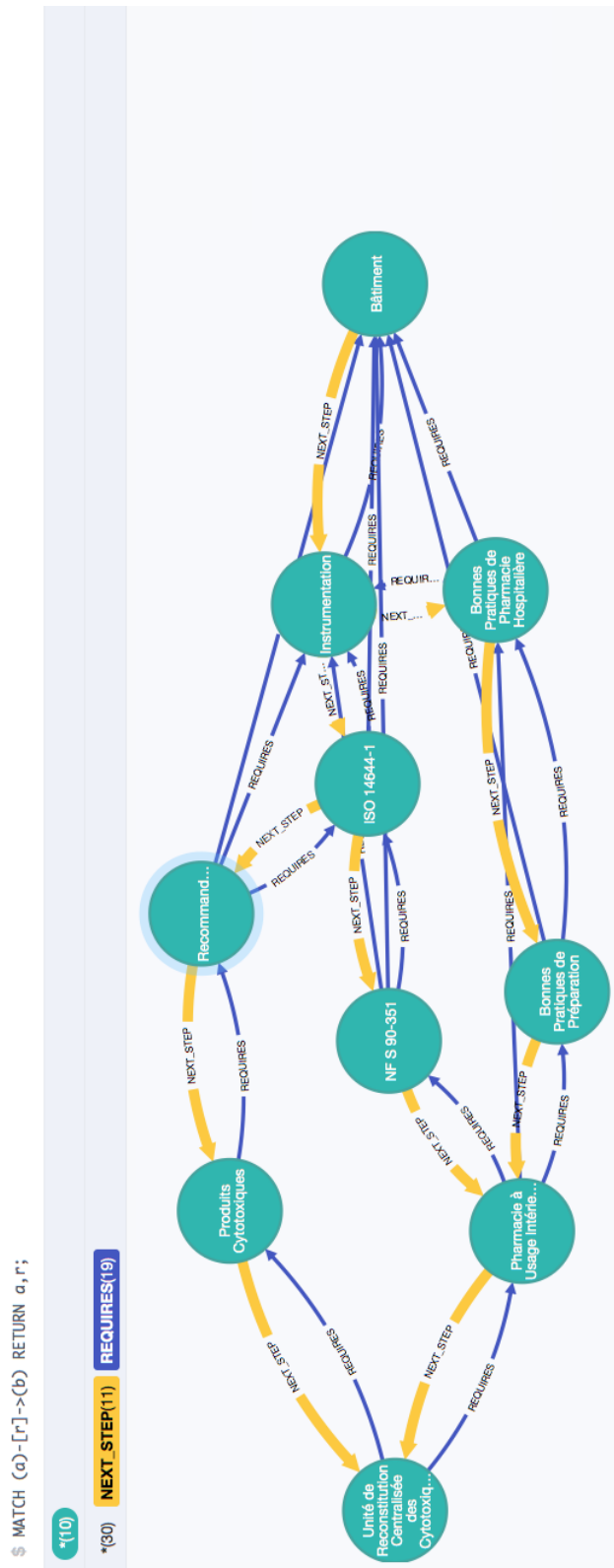


FIGURE 10.10 – Illustration : graphe complet dans Neo4J

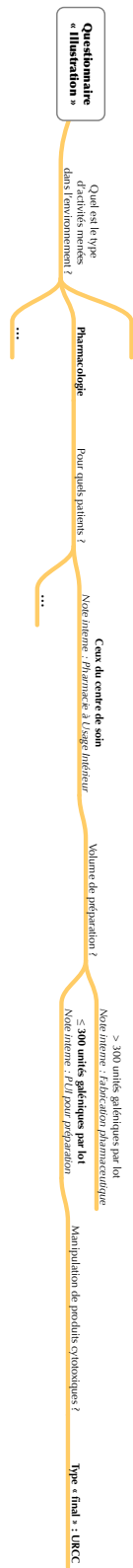


FIGURE 10.11 – Illustration : carte euristique simplifiée d'une URCC

2. Question : quels seront les patients concernés par les préparations effectuées ?

La réponse attendue dans le cadre de cette illustration est « *les patients du centre de soin* ». Cela nous oriente immédiatement vers une Pharmacie à Usage Intérieur (PUI).

3. Question : quel est le volume de préparations à réaliser ?

- (a) Réponse 1 : plus de 300 unités galéniques par lot.

D'un point de vue réglementaire, on mène dans cet environnement une activité de « *fabrication pharmaceutique* » (éludée dans ce chapitre).

- (b) Réponse 2 : jusqu'à 300 unités galéniques par lot.

D'un point de vue réglementaire, on mène dans cet environnement une activité de « *préparation pharmaceutique* » (présentée dans ce chapitre).

Selon l'activité ciblée par l'utilisateur, il pourrait s'arrêter à cette question. En effet, si l'on examine le graphe de la Figure 10.9, cela revient à faire une recherche sur le nœud « PUI » (dans la mesure où nous n'avons pas discriminé les activités de préparation et de fabrication pharmaceutique⁹). On aboutirait alors au sous-graphe de la Figure 10.12.

Par conséquent, les questions suivantes sont « fermées » (oui/non) et servent seulement à affiner le type d'environnement normé médical en fonction de contraintes spécifiques prévues dans l'environnement. Nous fournissons ci-dessous quelques questions à titre d'exemple, la dernière est la seule pertinente dans le cas présent.

4. Question : y a-t-il des préparations de médicaments à risque ou particulièrement dangereux ?

D'un point de vue réglementaire, cela ajoute des contraintes supplémentaires liées à la « *production dangereuse* ».

5. Question : y a-t-il des préparations de médicaments stériles ?

D'un point de vue réglementaire, cela ajoute des contraintes supplémentaires liées à la « *production stérile* ».

6. Question : y a-t-il une manipulation de produits cytotoxiques ?

D'un point de vue réglementaire, cela ajoute des contraintes supplémentaires liées à la « *manipulation de produits cytotoxiques* ».

Dans le cas d'une PUI, cela nous permet d'affirmer que le type d'environnement correspondant aux activités décrites est une *Unité de Reconstitution Centralisée des Cytotoxiques (URCC)*.

Dans le cas de l'environnement illustré dans ce chapitre, c'est une PUI produisant jusqu'à 300 unités galéniques par lot et manipulant des produits cytotoxiques.

Ces réponses conduisent à interroger le graphe de normes à la recherche du nœud URCC et, ainsi, d'extraire le graphe de la Figure 10.13, étudiée dans le prochain chapitre.

9. Dans le cas contraire, nous aurions deux nœuds « PUI » différents, l'un lié aux Bonnes Pratiques de Préparation (BPP) et l'autre aux Bonnes Pratiques de Fabrication (PBF).

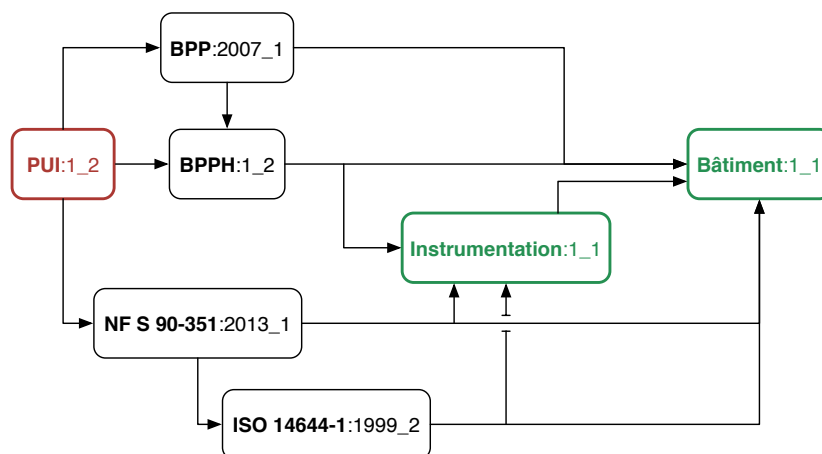


FIGURE 10.12 – Illustration : sous-graphe concernant les Pharmacies à Usage Intérieur

10.5 Traitement du graphe

La Figure 10.13 présente le graphe finalement obtenu au terme de l'analyse de ces normes.

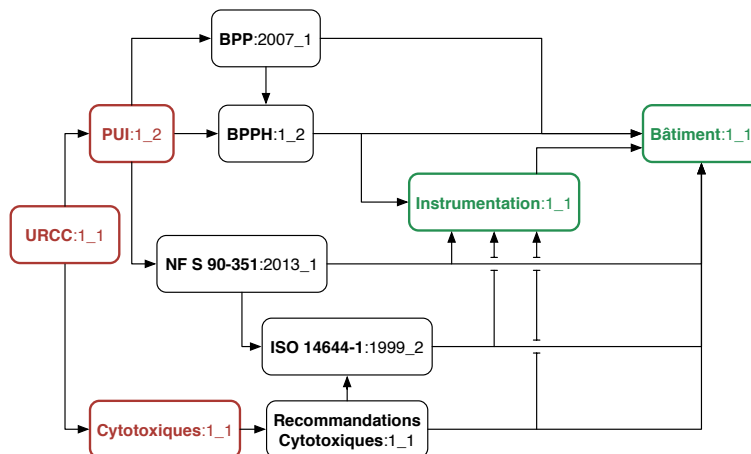


FIGURE 10.13 – Illustration : graphe concernant les URCC

Dans notre mise en œuvre avec Neo4J, toutes ces étapes de traitement ont été internalisées afin d'être réalisées par le SGBD lui-même (avec plus ou moins d'aide de notre part en fonction de nos différentes implémentations). La réponse avec la liste des étapes est donc immédiate. Mais dans le cas présent, afin d'illustrer totalement notre approche, nous avons fait le choix de détailler et de dérouler « manuellement » les différents traitements effectués sur le graphe.

Nous allons maintenant procéder aux étapes de traitement : nettoyage, inversion et tri topologique du graphe.

10.5.1 Nettoyage

Maintenant ce graphe obtenu, nous allons pouvoir le « nettoyer » (cf. Section 6.2.3 p. 88). Pour rappel, nous allons simplement élaguer les liens redondants en privilégiant les liaisons par transitivité (Figure 10.14).

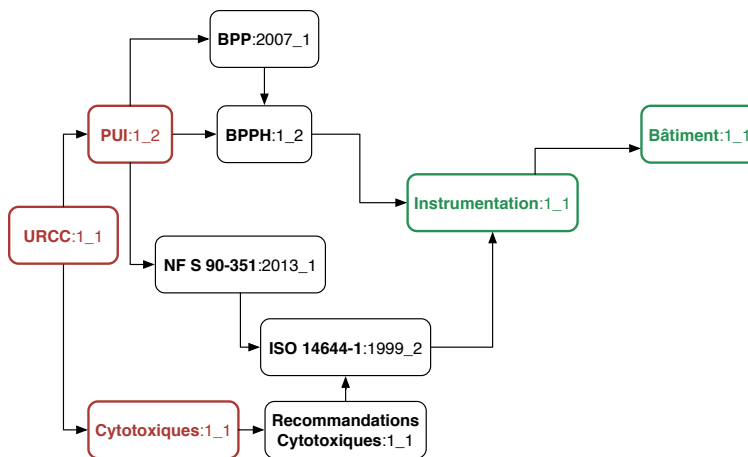


FIGURE 10.14 – Illustration : nettoyage du graphe

10.5.2 Inversion

Maintenant ce graphe obtenu, nous allons pouvoir inverser les dépendances pour préparer la structure finale, sur laquelle appliquer le tri topologique final (cf. Section 6.2.4 à la p. 90) dans la Figure 10.15.

10.5.3 Tri topologique

Dernière étape, supprimer les nœuds « factices » (URCC, PUI et cytotoxiques) (Figure 10.16) puis appliquer un tri topologique sur les nœuds restants (cf. Section 6.2.5 p. 91).

Dans la mesure où le graphe étudié ne fournit qu'un ordre partiel entre les différents nœuds, il existe plusieurs tris topologiques possibles (un point que nous avons discuté précédemment).

Dans les Figures 10.17 et 10.18, nous fournissons deux tris possibles pour l'arbre de la Figure 10.16. Le choix entre les différents arbres possibles devra être laissé à des experts « métiers » ou faire l'œuvre d'un travail statistique pour évaluer la pertinence de l'un par rapport à l'autre en fonction du taux d'erreur identifié.

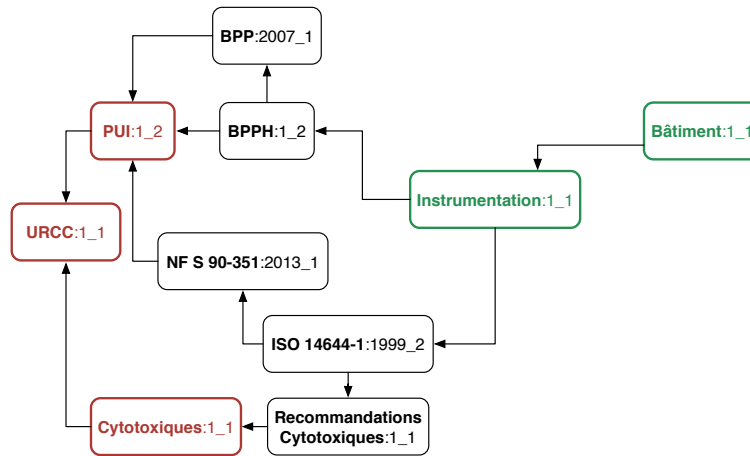


FIGURE 10.15 – Illustration : inversion du graphe

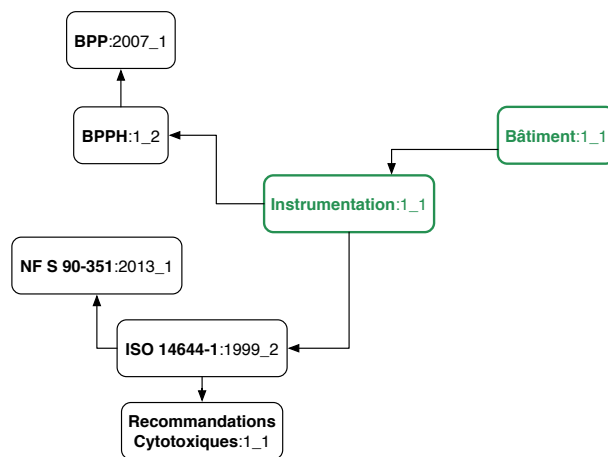


FIGURE 10.16 – Illustration : préparation au tri topologique du graphe

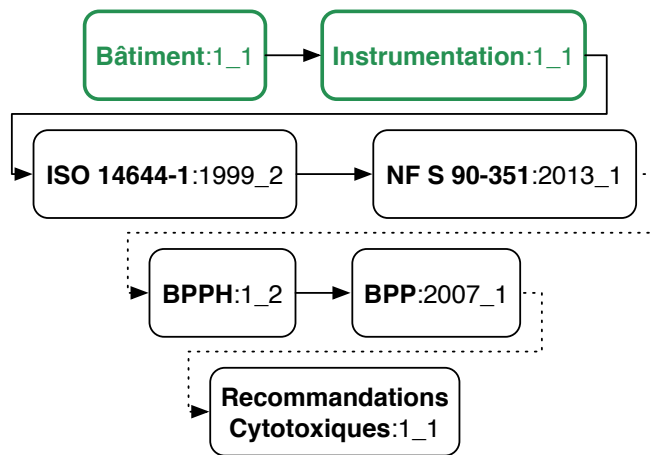


FIGURE 10.17 – Illustration : tri topologique du graphe

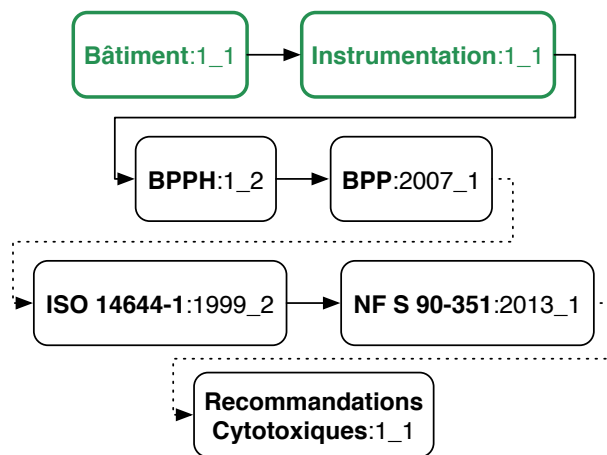


FIGURE 10.18 – Illustration : tri topologique alternatif du graphe

10.6 Architecture informatique

Dans le cadre de nos tests, nous avons mis en place l'architecture de la Figure 10.19.

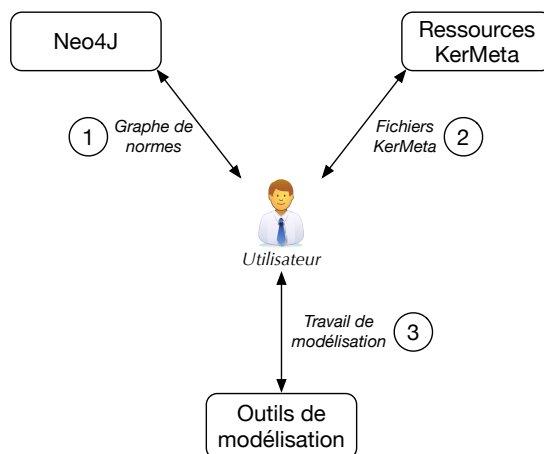


FIGURE 10.19 – Illustration : architecture informatique employée

Les différents éléments utilisent les implémentations décrites dans les Chapitres 8 et 9. On retrouve donc :

L'utilisateur est le pivot central de cette architecture ; il manipulera l'ensemble des systèmes et des serveurs¹⁰.

Neo4J sert à stocker le graphe de normes (cf. Chapitre 8). Ses interactions avec l'utilisateur sont l'insertion de normes et la recherche des étapes de modélisation nécessaires pour représenter et vérifier un projet d'environnement.

Ressources KerMeta contenant les différents métamodèles et aspects nécessaires à la représentation et l'analyse d'un projet d'environnement normé (cf. Chapitre 9). C'est l'interrogation du bloc « Neo4J » qui permettra de connaître les fichiers à récupérer parmi tous ceux disponibles.

Outils de modélisation contenant principalement les différents outils évoqués au Chapitre 9 servant à la modélisation (KerMeta et EMF) et quelques scripts d'assistance.

Dans une perspective plus « dynamique », l'utilisateur insère des normes dans le graphe lors de la phase d'analyse d'un type d'environnement normé (ce que nous avons nous même fait pour l'URCC et décrit dans les sections précédentes).

Ensuite, lorsqu'il veut vérifier un projet précis d'URCC (tel que celui décrit dans la Section 10.1), il interroge Neo4J afin d'extraire : la liste ordonnée des nœuds, la liste ordonnée des normes et enfin celle des métamodèles et aspects. La première liste est seulement utilisée pour le déverminage de notre implémentation,

10. Lorsque nous aurons approfondi les travaux d'automatisation de nos contributions, il pourra adopter une position moins prédominante.

la seconde permet d'avoir facilement un moyen de comparaison avec nos études « théoriques » et enfin la dernière est celle devant être réellement utilisée pour la modélisation.

Enfin, une fois obtenue la liste ordonnée des fichiers KerMeta pertinents, l'usage alterné de KerMeta et EMF permet de réaliser les différentes modélisations et vérifications.

10.7 Interrogation du graphe

Dans le cadre de notre URCC, nous avons produit le code source B.1 (p. 238) effectuant l'insertion des nœuds et l'interrogation du graphe. Cette classe affiche sur la console les trois informations mentionnées dans la section précédente (liste des nœuds, des normes et des fichiers KerMeta).

Nous avons également ajouté la production d'un fichier texte¹¹ contenant seulement la dernière liste (les fichiers KerMeta) afin de faciliter la réalisation des étapes suivantes (et notamment la récupération des fichiers concernés).

Lors de son exécution, le programme produit la sortie suivante :

```
-----
Travaux doctoraux d'A. GELIBERT
  Application sur une URCC
-----
Préparation base de donnée (URCC)
Recherche des étapes (URCC)
Liste des noeuds : [0, 1, 5, 13, 14, 15, 16]
Liste des normes :
  - Bâtiment
  - Instrumentation
  - ISO 14644-1
  - Recommandations Cytotoxiques
  - NF S 90-351
  - Bonnes Pratiques de Pharmacie Hospitalière
  - Bonnes Pratiques de Préparation
Etapes de modélisation :
  - ${MODELING_FILES_ROOT}/building.kmt
  - ${MODELING_FILES_ROOT}/instrumentation.kmt
  - ${MODELING_FILES_ROOT}/iso14644-1.kmt
  - ${MODELING_FILES_ROOT}/recommandations-cytotoxiques.kmt
  - ${MODELING_FILES_ROOT}/NFS90-351.kmt
  - ${MODELING_FILES_ROOT}/bpph.kmt
  - ${MODELING_FILES_ROOT}/bpp.kmt
```

Dans notre implémentation, afin de pouvoir facilement déplacer les ressources KerMeta, nous avons fait le choix d'introduire une constante¹² servant à représenter la racine de l'espace contenant les fichiers.

11. model-steps.txt

12. MODELING_FILES_ROOT

De la sorte, cet espace a pu être : un dossier local, un serveur web, un serveur FTP ou des serveurs plus complexes. La seule contrainte est qu'une fois la macro étendue, la ressource soit représentable sous la forme d'une URL bien formée.

10.8 Modélisation

Un script se charge alors de parcourir le fichier `model-steps.txt` pour préparer les ressources nécessaires à la modélisation (code source B.2 dans la Section B.2 à la p. 239).

Nous disposons alors d'un espace contenant tous les fichiers prêts au travail avec les outils de modélisation :

```
1_building.kmt
2_instrumentation.kmt
3_iso14644-1.kmt
4_recommandations-cytotoxiques.kmt
5_NFS90-351.kmt
6_bpph.kmt
7_bpp.kmt
```

On observe que nous avons extrait du graphe un ordre différent (mais valide) de ceux présentés dans les Figures 10.17 et 10.18.

Pour exploiter ces fichiers, il faudra maintenant suivre l'algorithme suivant (illustré dans la Figure 10.20) :

1. Utiliser `1_building.kmt` pour générer un fichier ECore compatible EMF.
2. Modéliser un premier environnement respectant `1_building.ecore` (format XMI).
3. Utiliser `1_building.kmt` et `2_instrumentation.kmt` pour générer un nouveau fichier ECore.
4. Reprendre l'environnement précédent pour le rendre compatible avec `2_instrumentation.ecore`.
5. Utiliser `1_building.kmt`, `2_instrumentation.kmt` et `3_iso14644-1.kmt` pour générer un nouveau fichier ECore.
6. Reprendre l'environnement précédent pour le rendre compatible avec `3_iso14644-1.ecore`.
7. Répéter ces opérations pour tous les fichiers présents dans le dossier `model-steps`.

Le lecteur intéressé pourra trouver en Annexe B, dans la Section B.3 (p. 237), le métamodèle du bâtiment au format UML (Figure B.1) et KerMeta 2 (code source B.3).

Dans le cadre de nos expérimentations, nous avons réalisé manuellement les différentes opérations de modélisation de l'environnement de la Figure 10.1, en utilisant EMF¹³.

13. Dans l'absolu, cette opération pourrait être réalisée de manière « automatique » par importation d'un fichier de CAO en ne retenant que les éléments conformes au métamodèle, c'est une piste discutée dans la Section 14.3.

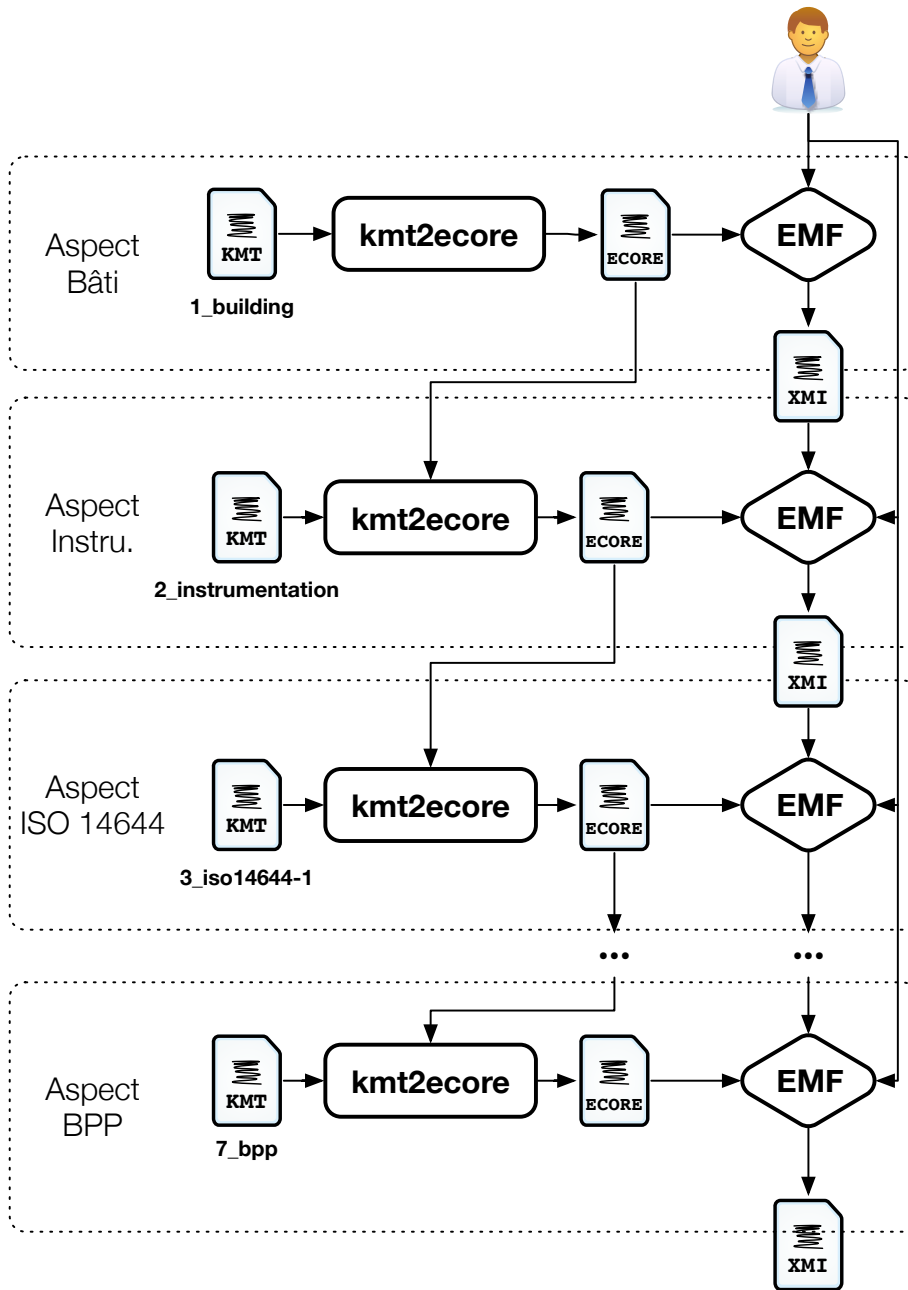


FIGURE 10.20 – Illustration : algorithme de modélisation

Ainsi pour produire le modèle initial du bâti, nous utilisons EMF pour générer à partir de notre fichier `1_building.ecore` un atelier de modélisation dédié (une procédure « classique » d'utilisation du framework). Cela nous permet d'obtenir la vue présentée à la Figure 10.21.

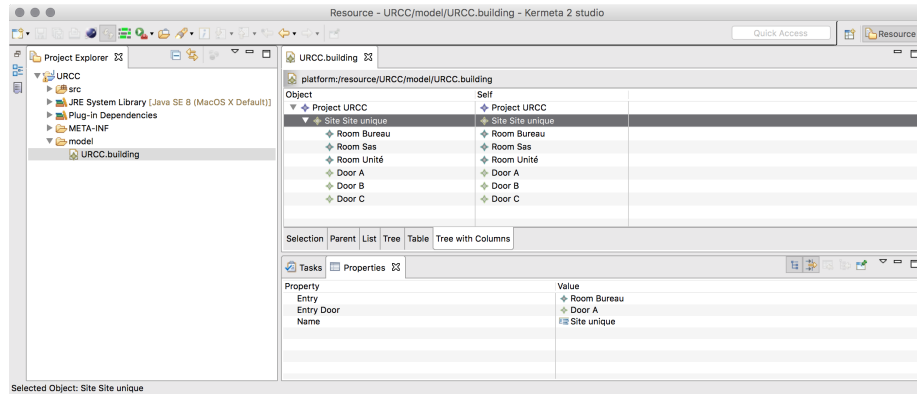


FIGURE 10.21 – Illustration : modèle initial du bâti

Ensuite, nous pouvons utiliser le fichier `2_instrumentation` (code source B.4 dans la Section B.4, p. 239) pour modifier l'ECORE du bâti et ainsi obtenir un nouveau métamodèle introduisant les différents équipements possibles et modifiant la classe `Room` en conséquence. Une nouvelle fois, nous utilisons EMF pour générer un nouvel atelier de modélisation dédié et modifier le modèle précédent en conséquence (Figure 10.22).

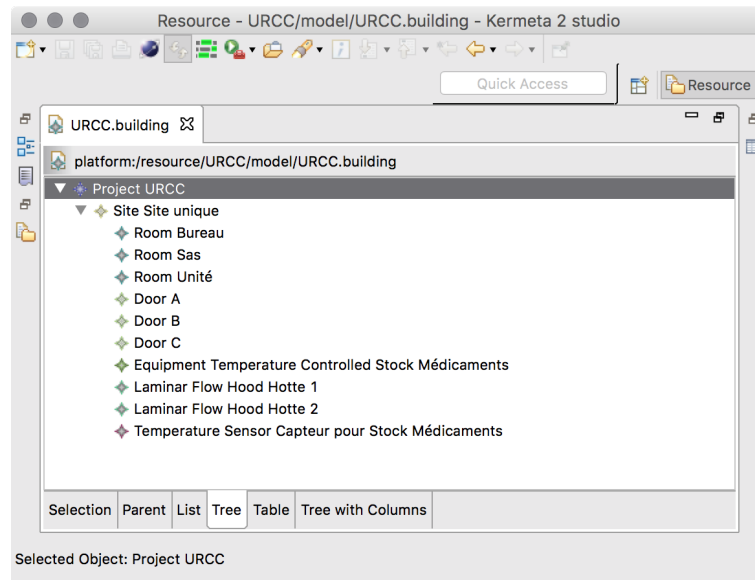


FIGURE 10.22 – Illustration : modèle du bâti enrichi de l'instrumentation

On procède de la même manière pour chacun des aspects.

Jusqu'à cette deuxième étape, il y a peu de contraintes à appliquer. À ce stade, les principales contraintes sont des cardinalités particulières dans les métamodèles afin de s'assurer, par exemple, que les équipements dont la température est régulée disposent bien de capteurs associés ¹⁴.

Par la suite, des contraintes plus importantes seront à analyser, parmi lesquelles :

ISO 14644-1 voit l'introduction de la notion de « classe ISO » pour une pièce (cf. Section 10.3.5).

Par conséquent, cet aspect crée l'attribut `ISOclass` (dont la valeur peut varier entre 1 et 10, cette dernière représentant les pièces non classées) dans la classe `Room` afin de pouvoir spécifier le niveau attendu. Cela nous conduira à des contraintes opérationnelles liées à la vérification de conformité du taux particulier avec les indications formulées.

Nous profitons également de cet aspect pour spécifier toute la mécanique de ventilation que nous avons seulement évoquée dans l'aspect « Instrumentation ».

Recommandations Cytotoxiques dont le principal intérêt dans le cadre de ces travaux est synthétisé dans la Figure 10.23.

Ce sont des contraintes sur la structuration de l'environnement en fonction du type d'équipement choisi pour réaliser les préparations : classe ISO, équipements, circulation des produits, sas d'habillage, bureau, etc.

NF S 90-351 est primordiale lors d'un travail sur les environnements maîtrisés. Elle introduit de très nombreuses contraintes techniques, parmi celles-ci :

- L'ajout d'un attribut `riskClass` dans la classe `Project` afin de représenter la classe de risque d'un projet. Celle-ci est totalement dépendante de l'activité concernée (Figure 10.24 et 10.25) ¹⁵ et conditionne de nombreuses valeurs (Figure 10.26).
- L'ajout de contraintes (et de capteurs associés) afin de mesurer la vitesse d'écoulement de l'air dans les pièces concernées.
- L'ajout de contraintes concernant la maîtrise de la contamination : gradients de pression et structuration de ces gradients (cf. ci-dessous).
- L'ajout de contraintes concernant l'adéquation entre les classes ISO déclarées et le niveau de la classe de risque (Figure 10.25).
- L'ajout de contraintes concernant la cinétique d'élimination des particules et la classe de propreté microbiologie (d'autres attributs introduits par cet aspect).
- L'ajout de contraintes pour rendre obligatoire la présence de capteurs de température dans la pièce et de vérifications à l'usage de valeurs dans la plage préconisée.
- L'ajout de contraintes pour vérifier le taux de brassage de la pièce (cf. l'exemple de calcul dans la Figure 10.25).

14. Pour rappel, ces capteurs devront être régulièrement étalonnés lorsque l'environnement sera en fonctionnement

15. La définition de la valeur « réelle » attendue sera donc une contrainte dans le nœud URCC.

HOTTE A FLUX LAMINAIRE	ISOLATEUR
<p align="center">Pièce de stockage réservée aux opérations suivantes Ouverture des cartons, stockage primaire des solutions de grand volume, des spécialités et des consommables.</p>	
<p align="center">Bureau avec une zone d'archivage Il sert à l'acte pharmaceutique avec entre autres la documentation, le dossier patient, le système informatique et les protocoles de préparation.</p>	
<p align="center">Sas d'habillage</p> <p>Divisé en 2 pièces</p> <ul style="list-style-type: none"> ▪ une dite "<i>sale</i>" pour le déshabillage et le lavage des mains ▪ l'autre dite "<i>propre</i>" pour l'habillage spécifique avec un traitement de l'air de classe B = ISO 5 <p>Autres éléments</p> <ul style="list-style-type: none"> ▪ Equipement du poste de lavage des mains dans la zone dite "<i>sale</i>" ▪ Pas de porte coulissante <p>Pas d'ouverture simultanée des différentes portes</p>	<p align="center">Sas d'habillage</p> <ul style="list-style-type: none"> ▪ 1 seul pièce permettant le déshabillage, le lavage des mains et l'habillage spécifique
<p align="center">Sas d'entrée des dispositifs médicaux et des médicaments</p>	
<p align="center">Pièce de préparation</p> <p><i>Climatisation pour la conservation des médicaments stockés</i></p> <p>Deux zones de paillasses</p> <ul style="list-style-type: none"> ▪ une pour la préparation du matériel et des médicaments introduits sous hotte ou isolateur ▪ la seconde pour le contrôle des préparations et du conditionnement. <p>Zone de stockage avec étagères pour les produits consommables et un réfrigérateur pour les spécialités à conserver à +4°C. Visible de la zone du bureau</p>	
<p align="center">Qualité de l'air exigée par les bonnes pratiques de fabrication</p>	
<p align="center">Classe B = ISO 5</p> <p>Système de surpression et de filtration d'air Moyen de communication avec le bureau (interphone, téléphone) Système d'évacuation extérieure de l'air de la hotte</p>	<p align="center">Classe D = ISO 8</p> <p>Système d'évacuation et d'extraction des gaz vers l'extérieur Dispositif d'extraction de l'acide peracétique Installation d'air comprimé et d'un évier pour le test d'étanchéité des gants.</p>

FIGURE 10.23 – Recommandations Cytotoxiques : descriptif des locaux en fonction du type d'équipement pour la reconstitution des cytotoxiques [Cen02]

Classe de risque	1*	2	3	4
Régime d'écoulement d'air	non unidirectionnel	non unidirectionnel	non unidirectionnel unidirectionnel	unidirectionnel
Temps de repos de la salle, hors toute présence humaine	/	≥ cinétique d'élimination : 20 min	≥ cinétique d'élimination : 10 min	≥ cinétique d'élimination : 5 min
Contrainte d'accès	/	/	accès obligatoire depuis zone à environnement maîtrisé	accès obligatoire depuis zone à environnement maîtrisé
Maîtrise des flux	procédure standard	procédure de gestion des risques de contaminations croisées matériel entrant/sortant ...	procédure de gestion des risques de contaminations croisées matériel entrant/sortant ...	procédure de gestion des risques de contaminations croisées matériel entrant/sortant ...
Matériaux à mettre en œuvre	/	/	désinfectable et étanche	désinfectable et étanche
Organisation du travail	/	spécifique	formation du personnel aux gestes et postures	formation du personnel aux gestes et postures
Organisation des contrôles particuliers de l'air	/	annuel	annuel	annuel (sauf exigence particulière)
Organisation des contrôles microbiologiques de l'air	/	/	annuel	annuel (sauf exigence particulière)
Organisation des contrôles microbiologiques des surfaces	/	/	annuel	annuel (sauf exigence particulière)
Fréquence de nettoyage	quotidienne	quotidienne	à chaque intervention ou cycle d'utilisation	à chaque intervention ou cycle d'utilisation
Requalification	/	annuelle	annuelle et à chaque remplacement de filtres terminaux	annuelle et à chaque remplacement de filtres terminaux
CAS DES SALLES D'OPERATIONS				
Taille de la salle d'opération	/	/	≥ 40 m ²	> 40 m ²
Taille de la zone d'environnement du patient	/	/	/	la taille du plafond unidirectionnel définit la taille de l'environnement du patient
* Non concerné par le document.				

FIGURE 10.24 – NF S 90-351 : points de vigilance [Age13]

Hospitalisation Type de zone et activités associées	Classe de risque	Observation sur les activités
Chambre d'hospitalisation standard	1	
Chambre d'hospitalisation pour patients infectieux	1	Selon analyse de risque et appréciation médicale il peut être utile que la chambre soit conçue pour être maintenue en dépression par rapport à son environnement
Chambre de soins intensifs	2	Cas général et unité de cardiologie
Chambre de surveillance continue	1 ou 2	Se référer à la réglementation en vigueur
Chambre de brûlés	4 ^a	Selon analyse de risque et appréciation médicale
Chambre unités protégées (hématologie)	4 ^a	
Chambre d'hématologie (hors unités protégées)	3	
Chambre Unité de transplantation d'organes	3	Selon analyse de risque et appréciation médicale
Chambre unité de post greffe	3	Selon analyse de risque et appréciation médicale
Pharmacie et Pharmacotechnie	Classe de risque	Observation
Radio pharmacie	2	Domaine réglementé (voir réglementation sur les radio-nucléides et BPP ch9)
Reconstitution cytotatique	4 ^a	Domaine réglementé (voir BPP)
Fabrication solution parentérale	4 ^a	Domaine réglementé (voir BPP)
Stérilisation	Classe de risque	Observation
Zone de conditionnement et de stockage	2	Se référer à la réglementation et aux BPPH en vigueur
Anatomo-Pathologie	Classe de risque	Observation
Salle de macroscopie	1	Locaux à maintenir en dépression pour éliminer les conservateurs (vapeurs, etc.)
Radiologie	Classe de risque	Observation sur les activités
Salle d'imagerie interventionnelle	3	Une étude spécifique est à mener en fonction des caractéristiques spatiales et volumétriques de l'imageur et de l'activité
Laboratoires	Classe de risque	Observation sur les activités
Laboratoire de sécurité biologique	À définir par l'établissement	Domaine réglementé (voir norme NF EN 12128, Code du Travail Arrêté du 18 juillet 1994, Arrêté du 16 juillet 2007, Arrêté du 23 janvier 2013, Guide de la commission de génie génétique février 2001, etc.)
Fécondation in vitro (FIV)	3	Sont concernés les locaux dédiés aux ponctions folliculaires
^a Pour les locaux de classe de risque 4, seule la zone située sous le flux unidirectionnel est considérée comme conforme aux exigences relative à cette classe. Chaque établissement doit affecter au reste du local une classe de risque inférieure ou égale.		

FIGURE 10.25 – NF S 90-351 : classes de risque en fonction du type d'activité [Age13]

Classe de risque	Classe de propreté particulaire	Cinétique d'élimination des particules	Classe de propreté micro-biologique	Pression différentielle (positive ou négative)	Plage de températures	Régime d'écoulement de l'air de la zone à protéger	Autres spécifications, valeur minimale
4 ^a	ISO 5	CP 5	M1	15 Pa ± 5 Pa	19 °C à 26 °C	Flux unidirectionnel	Zone sous le flux Vitesse d'air de 0,25 m/s à 0,35 m/s taux d'air neuf du local ≥ 6 volumes/heure
3	ISO 7	CP 10	M10	15 Pa ± 5 Pa	19 °C à 26 °C	Flux unidirectionnel ou non unidirectionnel	taux de brassage ≥ 15 volumes/heure
2	ISO 8	CP 20	M100	15 Pa ± 5 Pa	19 °C à 26 °C	Flux non unidirectionnel	taux de brassage ≥ 10 volumes/heure

^a Le taux de brassage, dans le cas particulier d'un flux unidirectionnel, doit être fixé indépendamment pour la zone située sous le flux et pour l'ensemble du local considéré.

Exemple de calcul : pour une salle d'opération de 200 m³ équipée d'un flux unidirectionnel recycleur de 3 m × 4 m.

Un plafond de 3 m × 4 m qui souffle à 0,3 m/s produit **12 960 m³/h**.

Le volume de la zone sous flux est de 40 m³ ce qui donne un taux de brassage de **324 vol/h**.

Si l'on considère que **6 vol/h** d'air neuf sont suffisants pour assurer la surpression de la salle et l'élimination des polluants, le débit nécessaire sera de **1 200 m³/h** d'air neuf.

Si l'air neuf est introduit dans le flux unidirectionnel, la zone sous flux sera balayée par 11 760 m³/h d'air recyclé et 1 200 m³/h d'air neuf.

Il faut donc pour les zones à risque 4 (ou à risque 3 si un flux unidirectionnel est mis en place) :

- choisir un flux unidirectionnel de taille suffisante pour protéger toute la zone à risque pour le patient ;
- fixer une vitesse d'air suffisante pour assurer la propreté de l'air sur l'ensemble du volume sous le flux ;
- choisir un taux d'air neuf suffisant pour évacuer les polluants présents dans la salle et assurer une surpression par rapport à son environnement.

FIGURE 10.26 – NF S 90-351 : valeurs guide de performance au repos [Age13]

Dernier point que nous souhaiterions approfondir, celui de la structuration des gradients de pression dans le cadre de la norme NF S 90-351.

C'est typiquement le type de contrainte difficilement représentable avec OCL et pour lequel une approche programmatique comme KerMeta prend tout son sens.

En effet, pour mettre en place l'asepsie progressive et les paliers de surpression (ou dépression), plusieurs scénarios sont possibles du plus simple (les cascades de pression de la Figure 10.27 ou celles de notre propre exemple) au plus complexe avec l'introduction de pièces tampons servant à recueillir l'air contaminé ou au contraire servir de barrière pour un processus fragile (Figure 10.28).

Il nous a semblé plus simple de vérifier ce type de contraintes en programmant le parcours du modèle (nous avons représenté la porte et la pièce d'entrée) pour vérifier la présence des cascades de pression par KerMeta.

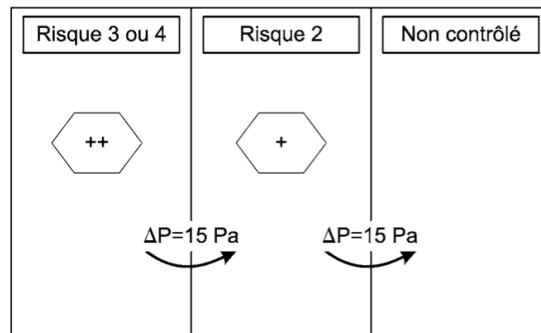


FIGURE 10.27 – NF S 90-351 : cascades de pression [Age13]

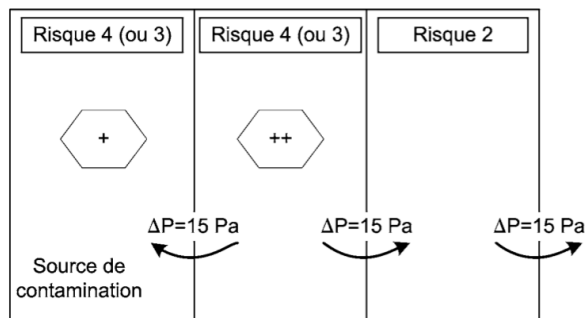


FIGURE 10.28 – NF S 90-351 : schéma de principe cascade de pressions avec un local à risque recevant une activité à risque [Age13]

10.9 Conclusion

Nous arrivons au terme de cette illustration permettant de comprendre et d'apprécier la philosophie et le déroulement de notre méthodologie. Le reste des réflexions et expérimentations ne nous appartient pas et demeure la propriété intellectuelle de Nocosium.

La difficulté pour la définition de cette illustration fut de trouver le bon compromis entre complexité/réalisme de l'environnement étudié et clarté/facilité de compréhension pour le lecteur de ce mémoire. Nous avons donc dû nous astreindre à un certain nombre de simplifications pour conserver l'esprit d'une URCC tout en ne complexifiant pas trop les graphes et analyses.

Après avoir illustré nos mises en œuvre sur un cas d'exemple, nous pouvons maintenant présenter les transferts technologiques effectués dans le cadre de ces travaux doctoraux.

Transfert des contributions

Ce chapitre s'attache à la description des transferts industriels opérés sur les contributions exposées dans les Chapitres 5, 6 et 7 ainsi que sur les expérimentations des Chapitres 8 et 9.

Comme dernier préalable à ce chapitre, nous mettons l'accent sur l'importance de l'aspect « opérationnel » (industriel). Nous quittons ici l'aspect universitaire des travaux pour rentrer dans celui plus industriel de son contexte d'application et d'expérimentation.

11.1 Contexte

Les travaux présentés dans cette thèse se nourrissent des besoins exprimés par les entreprises spécialisées dans la conception d'environnements normés, pour proposer une approche globale à la qualification des environnements réalisés, une problématique typique d'Ingénierie des Exigences.

Plus précisément, ces travaux ont débuté dans le cadre d'une convention CIFRE de l'ANRT au sein d'une TPE spécialisée dans la réalisation d'environnements confinés médicaux.

C'est dans ce cadre d'activité très particulier que les expérimentations ont été menées, avec trois conséquences majeures :

1. la proximité et la disponibilité d'un cadre réel pour la mise en application de l'approche proposée dans cette thèse ;
2. l'application des travaux au domaine médical français plutôt qu'à d'autres secteurs d'activités et d'autres types de contraintes¹ ;

1. En parallèle du domaine « médical », nous avons également appliqué nos travaux aux Bâtiments Basse-Consommation (BBC) [Gel14].

3. l'obligation de proposer, lors des transferts technologiques, des solutions viables dans ce type d'entreprise : jeune, de petite taille et dont le principal domaine de compétence n'est pas l'informatique.

Ainsi, les travaux réalisés et présentés dans ce chapitre ont, dans un premier temps, donné lieu à l'élaboration de prototypes au sein du laboratoire puis à un transfert vers l'entreprise Nocosium.

La Section 11.2 décrit une mise en œuvre « complète » de l'approche proposée, incluant également la plateforme de traçabilité, non traitée dans cette thèse.

Dans la mesure où — comme exposé dans les chapitres précédents — il n'existe pas d'outil « industriel » ou même « industrialisable » permettant de réaliser nos objectifs de modélisation, il convenait de proposer également une version alternative de cette mise en œuvre, pouvant être produite par une entreprise sans requérir une importante phase de recherche. C'est ce que nous ferons dans le Chapitre 11.3, avec une approche opérationnelle de l'architecture, atteignable par une petite structure, et offrant tout de même une réelle innovation pour la réalisation et l'utilisation d'environnements normés.

11.2 Mise en œuvre « complète »

Nous allons maintenant présenter la mise en œuvre complète de nos travaux proposée dans le cadre de cette convention CIFRE. Comme pour les chapitres précédents, nous débuterons par un schéma général que nous détaillerons dans un second temps.

11.2.1 Schéma

La Figure 11.1 met en perspective l'ensemble des concepts issus des chapitres précédents dans un cadre d'utilisation réel (avec les opérateurs humains impliqués) et fournit une réponse complète à la question de la plateforme de traçabilité. La réalisation des audits sur l'environnement sera débattue dans la conclusion de ce chapitre.

Nous allons maintenant détailler ce schéma en déroulant la méthodologie chronologiquement. Nous avons cependant fait le choix de ne pas nous attarder outre mesure sur les points similaires à ceux présentés dans les chapitres précédents².

11.2.2 Détails

Rappel des deux premières phases

Les premières étapes du processus, les phases « 0 » et « 1 » sont strictement identiques à celles présentées dans les Chapitres 6/8 et 7/9. Cette section s'attache plutôt à une description détaillée de la phase « 2 », c'est-à-dire la plateforme de traçabilité.

Dans les phases « 0 » et « 1 », nous retrouvons naturellement les étapes classiques de notre approche :

2. Afin de simplifier la lecture de ces schémas, la numérotation employée est identique à celle du Chapitre 5.

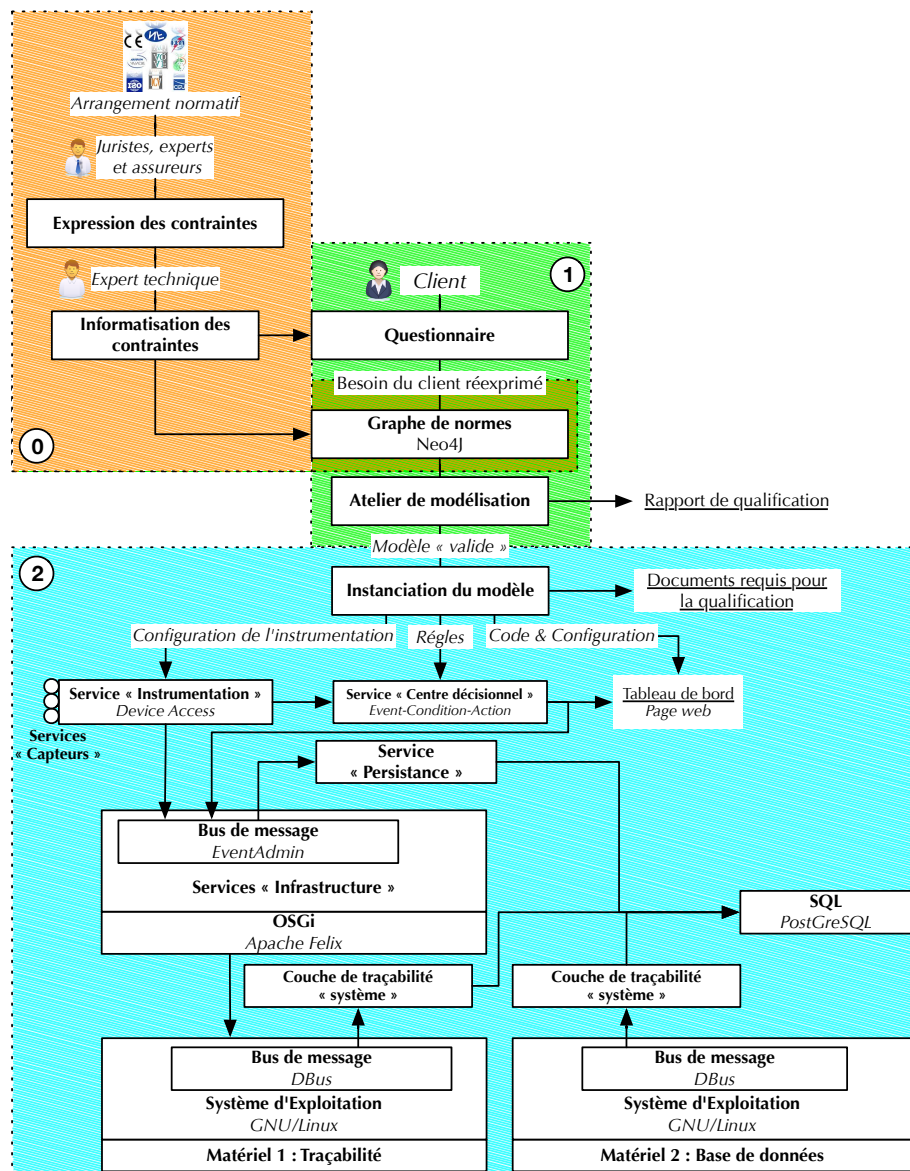


FIGURE 11.1 – Mise en œuvre « complète » de notre approche

1. Constitution de l'arrangement normatif³.
2. Extraction et expression des contraintes par un groupe d'experts « métiers » (juristes, experts et assureurs. . .).
3. Informatisation de ces contraintes par des experts techniques (ingénieurs informatiques et assimilés).
4. Utilisation d'un questionnaire par le client pour déterminer précisément la requête à effectuer sur le graphe de normes.
5. Interrogation du graphe de normes pour en extraire la liste d'étapes permettant de modéliser un projet d'environnement médical aux caractéristiques souhaitées.

Pour rappel, une fois ces étapes réalisées, nous disposons du métamodèle d'un environnement « brut » incluant seulement les concepts de bâtiments, d'étages, de pièces, de murs, de portes, de mobiliers « classiques », etc. ; ainsi que d'un ensemble d'aspects permettant d'enrichir ce métamodèle pour pouvoir représenter l'ensemble des concepts métiers requis, et de vérifier que la modélisation effectuée est conforme à la réglementation correspondante.

6. Utilisation de l'atelier de modélisation pour représenter le projet souhaité, en prenant soin de le valider au fur et à mesure de sa définition.

Cette dernière étape nous a donc fourni un modèle d'environnement médical « valide », comportant toutes les garanties pour sa traçabilité durant l'utilisation. La troisième phase pouvant donc débiter, nous abordons maintenant la plateforme de traçabilité.

Troisième phase : instanciation du modèle et plateforme de traçabilité

Cette dernière phase est particulière car elle est à la charnière entre la qualification d'un projet (« abstrait » par nature) et la qualification d'un fonctionnement (donc par nature, une surveillance « réelle », « sur le terrain »).

Elle se découpera donc en deux étapes, une passe de traitement sur le modèle théorique de l'environnement et un travail « réel », en continu, dans l'environnement de production. Naturellement, plusieurs mois (voir années) pourront séparer ces deux étapes⁴.

Instanciation du modèle : le profil appliqué

Notre première action consiste à poursuivre la logique de l'Ingénierie Dirigée par les Modèles (IDM) dans une étape (classique) « d'instanciation du modèle ». Cependant, dans notre cas, cette opération ne visera pas à générer du code logiciel (ou seulement marginalement), mais plutôt à transformer le modèle « validé » en un ensemble d'éléments permettant de nourrir les éléments physiques placés de l'environnement réel. Cet ensemble d'artefacts constitue ce que nous avons appelé précédemment le « profil appliqué » de l'environnement normé.

3. Une étape masquée dans notre schéma, car indépendante des travaux présentés dans cette thèse (cf. introduction du Chapitre 6).

4. Un point peu évoqué dans ce document, mais sur lequel nous reviendrons dans la conclusion générale de ce chapitre. Il ne faut pas voir ce délai comme une contrainte, dans le type de projets concernés par cette thèse (médical, nucléaire, nanotechnologies. . .), c'est inhérent à la nature du sujet traité.

Nous considérons quatre types d'artefacts :

Configuration de l'instrumentation. Il s'agit d'un premier jeu d'artefact composé des éléments nécessaires à la configuration de l'instrumentation pour obtenir les données nécessaires à la qualification du fonctionnement. Cela concerne principalement la configuration des capteurs, actionneurs et autres systèmes tiers nécessaires à la prise de décision de la plateforme de traçabilité. Parmi ces données, on trouvera, par exemple, la fréquence de relevé des valeurs environnementales.

Selon le degré de maîtrise de la plateforme environnementale et la souplesse des systèmes tiers déployés dans l'environnement, il peut être possible d'aller relativement loin dans l'automatisation issue de ces artefacts. Avec des produits parfaitement maîtrisés, on peut sereinement envisager que seules les adresses et la nature du bus de communication soient figées, le reste de la configuration étant inféré par la plateforme de traçabilité après consultation des artefacts.

Règles de fonctionnement. Le deuxième jeu d'artefact est évident, il s'agit des règles de fonctionnement à vérifier en permanence sur les données issues de l'instrumentation. Ce jeu de règles est composé de toutes les contraintes dynamiques relevées dans l'arrangement normatif et c'est la configuration de l'instrumentation qui le rend possible ; il n'a d'ailleurs aucun sens sans celle-ci. La forme de ces règles de fonctionnement dépend de la nature du module devant traiter et appliquer ces règles (cf. sous-section suivante).

Ces règles de fonctionnement doivent naturellement s'exécuter de manière permanente et autonome dans la plateforme de traçabilité pour garantir un suivi « en temps réel » de l'environnement ciblé. Lorsqu'une de ces règles est transgressée, une alerte doit être affichée dans le tableau de bord et notifiée au personnel présent dans les lieux.

Configuration du tableau de bord. Le troisième jeu d'artefact décrit la configuration du tableau de bord affiché aux utilisateurs. C'est naturellement la conséquence logique des règles décrites précédemment. C'est la manière d'afficher les alertes et de les mettre en valeur pour le personnel médical.

Il peut paraître surprenant pour le néophyte de proposer une « configuration » du tableau de bord issue du modèle d'environnement. En effet, on pourrait estimer cette configuration « unique », déterminée au préalable en fonction du type général d'activité (médical, nucléaire, industrie...) et se contentant d'afficher les alertes. Malheureusement, ce n'est pas aussi simple. L'expérience « métier » montre clairement qu'au sein d'un même domaine, en fonction du type précis d'activité menée, certaines notifications ont bien plus d'importance que d'autres, même si cela n'est pas clairement indiqué dans l'arrangement normatif.

Documents requis pour la qualification. Le dernier jeu de contraintes comprend les contraintes dites « documentaires », c'est-à-dire la liste de l'ensemble des documents à produire pour faire qualifier l'environnement concerné. Cette liste présente peu d'intérêt dans ces travaux, car elle est relativement simple à obtenir et doit simplement donner lieu à une « check-list » pour la direction de projet afin de s'assurer de n'oublier aucune pièce documentaire.

C'est évidemment un point très fortement lié aux types précis d'activités menées dans l'environnement. Pour rappel, nous avons préféré éluder ce type de contrainte dans cette thèse, certains travaux de recherche s'étant déjà attelés à la façon d'automatiser la vérification de présence de ces documents [PW12].

Une fois l'instanciation effectuée, nous disposons de tous les éléments pour configurer notre plateforme de traçabilité, dont nous allons maintenant décrire l'architecture.

Plateforme de traçabilité

Pour réaliser la plateforme de traçabilité (rappelée dans la Figure 11.2), nous nous sommes tournés vers une architecture orientée « services » à base de services OSGi; nous offrant ainsi la souplesse et le dynamisme requis pour notre objectif.

Nous rappelons en effet notre objectif de disposer d'un outil de vérification permanent, suffisamment souple pour s'adapter à l'ensemble des environnements normés (et notamment à leur évolution permanente tant normative que technique) tout en offrant les garanties de robustesse incontournables dans de telles situations.

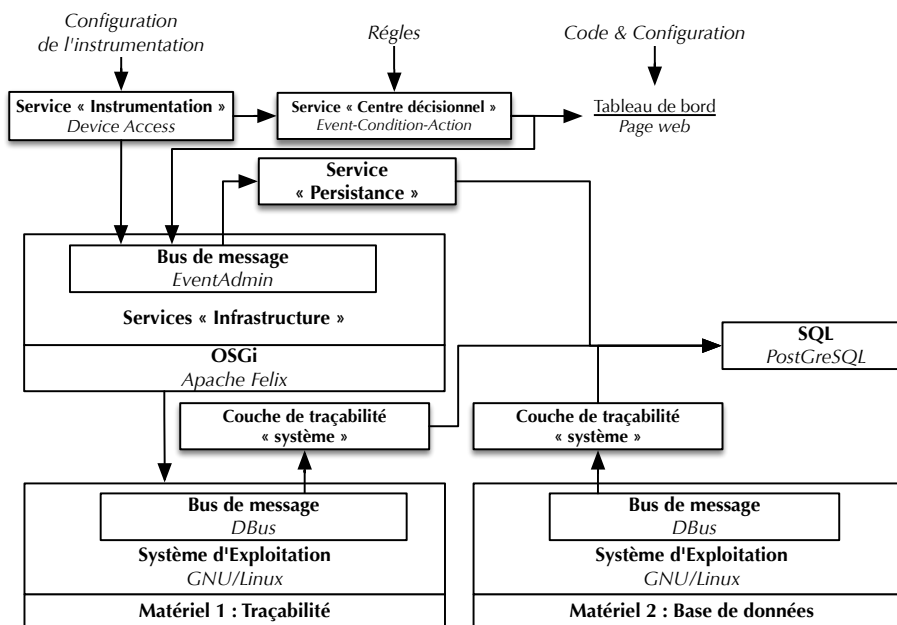


FIGURE 11.2 – Plateforme de traçabilité : une approche orientée services

L'architecture globale de notre plateforme de traçabilité est relativement simple, plusieurs plateformes matérielles indépendantes sont utilisées conjointement, pour sécuriser l'isolation des différentes fonctions recherchées.

Une première plateforme — nommée « Traçabilité » sur le schéma — réalise l'acquisition des données, l'application des règles et l'affichage du tableau de bord. C'est la partie « opérationnelle » de notre architecture. Afin d'isoler les

traitements et de maximiser la sécurité et l'intégrité des données, cette plateforme ne conserve pas les événements, elle les transmet à une seconde plateforme matérielle.

Une seconde — nommée « Base de données » sur le schéma — réalise le stockage de l'ensemble des événements reçus et des alertes émises lors de la vérification des règles « métiers ». Naturellement, dans un contexte opérationnel, cette plateforme n'a de sens que si elle est physiquement isolée de la première.

Nous pourrions enfin trouver (absent de la Figure 11.2) plusieurs copies des événements enregistrés pour augmenter la sécurité des données acquises et permettre de procéder à des audits indépendamment de la production. Ces doublons seraient entièrement calqués sur le modèle de la seconde plateforme.

Chacune de ces plateformes est bâtie sur le même principe, des systèmes d'exploitation identiques et surtout le même agent logiciel se chargeant d'enregistrer tous les événements « systèmes » pouvant avoir un quelconque intérêt pour la traçabilité du fonctionnement des plateformes : démarrage, arrêt, connexion de périphériques, connexion d'utilisateurs, etc.

Pour la plateforme nommée « Traçabilité », nous utilisons le *framework* OSGi FELIX [Apa15d], sur lequel nous déployons — outre les services systèmes évidents — le bus de message « EventAdmin ». Ce bus de message constitue une « courroie de transmission » pour l'échange d'informations entre les différents services « fonctionnels ». Un service d'enregistrement « écoute » l'ensemble des messages échangés sur le bus et archive tous les événements pertinents en les transmettant à la plateforme « Base de données ».

Cela constitue le bloc « infrastructure » de notre plateforme de traçabilité, sur lequel nous déployons les blocs « fonctionnels ». Nous obtenons une suite de service, articulée de la façon suivante : les services d'instrumentation (communiquant avec les capteurs, actionneurs et services tiers), transmettent les données au « centre décisionnel » (appliquant les règles « métiers ») pour ensuite, lorsqu'une action est nécessaire, mettre à jour le tableau de bord et enregistrer cet événement.

Le centre décisionnel est un service visant à vérifier le bon fonctionnement de l'environnement normé. Pour le réaliser, nous avons étudié deux pistes :

Moteur de règles : cette approche, dont DROOLS [JBo15, Bro09] est une des incarnations dans le monde Java, consiste à définir les contraintes comme des règles que doivent appliquer ou vérifier les différents éléments. Le profil « appliqué » serait donc un ensemble de règles à respecter. Dans le schéma, nous avons préféré le terme d'Event-Condition-Action (ECA) plus générique et accessible.

Chaine de médiation : cette approche, dont CAMEL [Apa15a, IA10] est une des incarnations dans le monde Java, consiste à définir un enchaînement de blocs logiques. Chacun de ces blocs effectue une opération logique simple (sélection, agrégat...) et c'est l'enchaînement de ces éléments simples qui crée la fonction de haut niveau recherchée. Le profil serait donc dans ce cas une chaîne de médiateurs, directement déployable dans la plateforme et qui réaliserait les opérations requises.

Notre choix s'est arrêté sur le moteur de règles, plus accessible et pour lequel la collaboration avec les experts « métiers » nous a semblé plus simple.

11.2.3 Conclusion

Cette mise en œuvre complète de notre architecture détaille une réponse complète aux questions posées dans l'introduction de cette thèse ; la proposition d'un flot de conception complet en ingénierie des exigences permettant de qualifier un environnement normé de sa conception à son usage.

Malheureusement, le cœur de cette solution, la vérification d'un projet d'environnement normé, tel que nous l'avons abordé dans ces travaux, repose sur des technologies dont la maturité n'est pas adaptée au contexte très particulier d'une convention CIFRE avec une TPE. Ces technologies sont trop jeunes et demandent trop d'efforts de développement pour être accessibles à d'aussi petites structures.

Afin de pouvoir tout de même disposer d'une solution concrète dans le cadre de ces travaux, nous avons proposé une approche alternative, détaillée dans la prochaine section.

11.3 Adaptation aux contraintes industrielles

L'objet de cette version est d'éviter les éléments techniquement instables (plus précisément inadaptés au contexte de ces travaux) pour se concentrer sur les éléments accessibles, tout en proposant toujours une amélioration notable de l'usage des environnements normés médicaux. Cela peut être vu comme une forme de contribution « industrielle » en opposition à une contribution « universitaire » se devant d'être innovante au niveau de l'état de l'art, même si cela doit faire appel à des technologies et des concepts en devenir.

Nonobstant les aspects industriels, cette contrainte supplémentaire est intéressante d'une perspective « recherche », elle nous permet d'aborder une autre forme d'expérimentation sur nos travaux.

En effet, dans ce document, nous avons présenté une approche « flexible » ouvrant la voie à plusieurs mises en œuvre possibles pour la même problématique. C'est cette flexibilité que nous allons maintenant mettre à l'épreuve. Pouvons-nous conserver le « traitement des arrangements normatifs » et le « suivi d'un environnement normé en permanence » sans passer par la phase de « qualification d'un projet » ?

11.3.1 Schéma

La Figure 11.3 reprend l'architecture de la Figure 11.1 dans une variante plus adaptée à notre cible industrielle.

Nous allons maintenant détailler ce schéma en comparant les points de divergence avec la structure fournie précédemment.

11.3.2 Détails

Si nous conservons le fonctionnement « général » des différentes phases de notre approche, nous en adaptons en revanche l'« usage ».

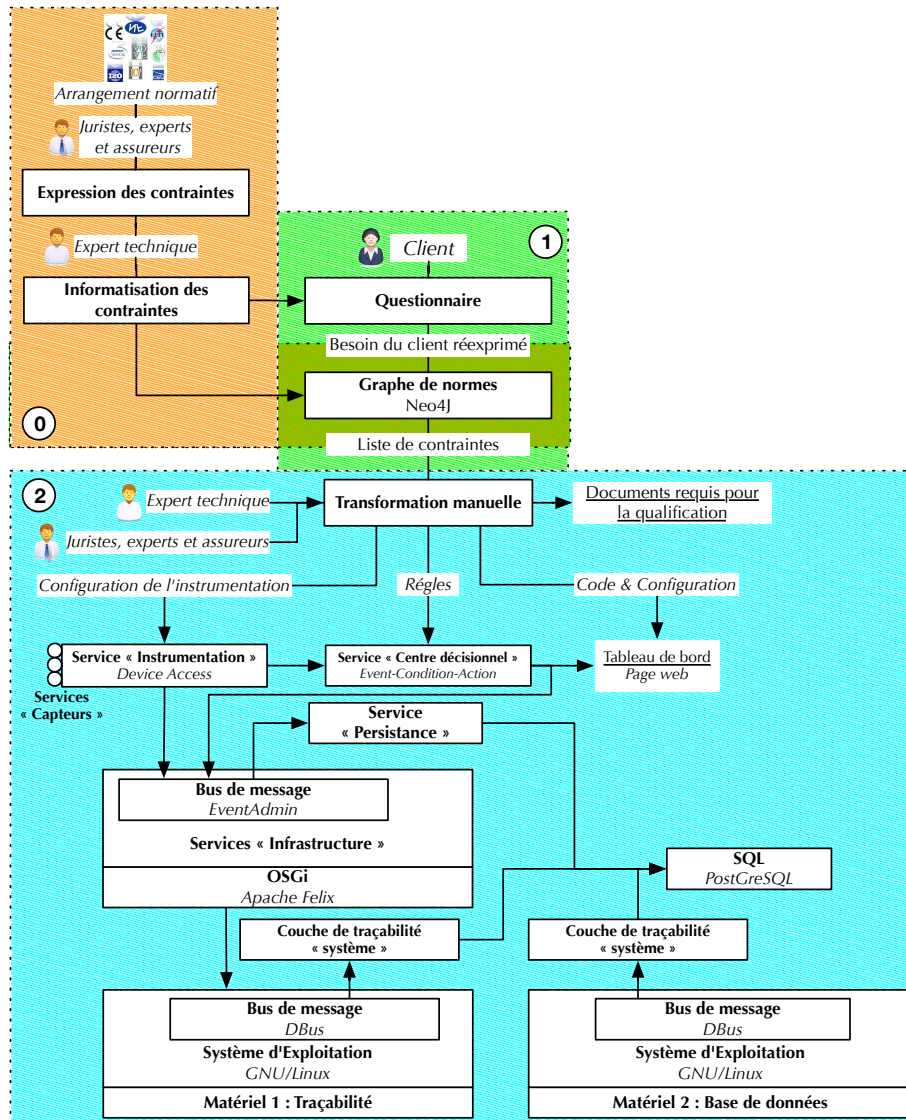


FIGURE 11.3 – Mise en œuvre « simplifiée » de notre approche, sans la phase de qualification d'un projet

Ainsi, si l'analyse préalable du type d'environnement est préservée — et produit un graphe de normes aux caractéristiques similaires à celles présentées précédemment — nous n'accompagnons pas les normes de leur métamodèle, mais simplement des contraintes opérationnelles qu'elles imposent. Ces contraintes sont exposées dans un langage « métier » suffisamment simplifié pour permettre aux experts techniques de travailler. Cela reste tout de même très proche de la version précédente, les mêmes contraintes devant alors être intégrées aux modèles. On assiste ici simplement à un allègement des capacités de l'approche.

Tout cela est rendu possible par notre choix d'isoler la représentation des normes (la forme) des contraintes « concrètes » qu'elles imposent (le fond) afin de pouvoir faire évoluer l'un sans nécessairement impacter l'autre⁵. Dans le cas présent, nous avons simplifié le fond tout en préservant la puissance de la forme.

L'interrogation du graphe de normes est toujours réalisée, mais plutôt qu'une suite d'étapes de modélisation, c'est une liste de contraintes qui sera retournée.

Les experts techniques et métiers pourront alors travailler conjointement à la production des artefacts présentés précédemment (cf. 11.2.2). Certaines opérations automatisées seront alors réalisées manuellement et l'ensemble des règles à vérifier sera repris pour configurer la plateforme de traçabilité.

La vérification de conformité de fonctionnement de l'environnement ciblé sera similaire à celle présentée dans la Section 11.2.2.

11.3.3 Conclusion

Cette section nous a permis d'illustrer la versatilité de notre proposition et notamment son importante capacité d'adaptation aux contraintes de mise en œuvre.

Nous avons ainsi montré qu'il était possible d'éviter la qualification d'un projet tout en conservant l'opportunité de « capitaliser » les connaissances acquises lors du traitement des différents arrangements normatifs ainsi qu'une approche souple et robuste pour la vérification en temps réel du respect des règles de fonctionnement par une plateforme de traçabilité évolutive et adaptable.

Cette nouvelle version de l'architecture plus adaptée à l'entreprise initialement cible de ces travaux est en cours de réalisation ; des prototypes embarquant les « briques de base » ont été réalisés. Certains détails présentés ici ont ainsi été modifiés pour s'adapter aux compétences et aux fonctionnements internes de la société, ils ont été volontairement éludés pour se concentrer sur les aspects en lien avec ces travaux.

11.4 Conclusion générale

Ce chapitre nous a permis de présenter une mise en œuvre complète de notre proposition (Figure 11.1), reprenant pour cela les travaux expérimentaux initiés dans les chapitres précédents, y apportant la conclusion finale par une implémentation orientée « services » de la plateforme de traçabilité.

Les chapitres précédents ayant montré la complexité de la phase de qualification d'un projet et les efforts requis pour atteindre un niveau « industriel »,

5. Cette approche de la représentation des normes est l'une des contributions de ce doctorat.

il fallait fournir une version alternative (ou plutôt simplifiée) de l'architecture, reposant sur des humains pour la qualification, mais profitant de toutes les autres opportunités offertes par notre contribution. D'un point de vue « universitaire », cette version « simplifiée » visait à expérimenter et mettre à l'épreuve la souplesse de notre contribution et sa faculté (telle qu'attendue dans la problématique) à s'adapter aux contraintes spécifiques des environnements normés et de leurs acteurs. Nous avons proposé la simplification présentée dans la Figure 11.3.

Avant sa liquidation, Nocosium abordait ces travaux sous deux angles, correspondant aux deux grands composants de l'approche simplifiée.

Une partie de son département R&D œuvrait à la réalisation d'une première version « industrielle » de la plateforme de traçabilité, reposant sur les concepts décrits dans ce chapitre. *In fine*, un déploiement sur site était prévu ; le « terrain de jeu » ciblé étant un projet très innovant — faisant déjà l'objet de plusieurs dépôts de brevets —, une salle d'essais cliniques aux caractéristiques uniques au monde.

L'objectif — conforme aux attentes industrielles exprimées en introduction de ce document — reste de déployer un équipement capable de valider les règles de fonctionnement correspondant à ce type d'environnement, puis d'en faire un retour « pertinent » aux utilisateurs. Cela requiert le respect de nombreuses contraintes supplémentaires, passées sous silence dans ce document⁶, parmi lesquelles on pourra citer la qualification comme « Dispositif Médical » [Eur07], le respect de la norme 21 CFR PART 11 [U.S97] ou la manipulation de données cliniques [CDI15, Tea13, Int05].

Les experts biomédicaux et juridiques de Nocosium travaillaient quant à eux sur la constitution de la base de données des normes telle que présentée dans le Chapitre 8 pour un usage dans la version simplifiée de l'architecture. Ceci sous-entend la constitution de différents arrangements normatifs, où les normes ne sont pas représentées par des métamodèles et des aspects, mais plutôt par des jeux de contraintes à vérifier par la plateforme de traçabilité.

6. Un grand nombre de ces contraintes avait été abordé dans les premières publications réalisées sur ces travaux [GJGC12]

Quatrième partie

**Conclusion, limites et
perspectives**

Conclusion

Les sociétés modernes — et la société française ne fait pas exception — sont devenues extrêmement friandes de normes, de conventions, de spécifications, etc. Les activités et productions techniques humaines se sont ainsi vues largement encadrées et nécessitent toujours plus de formation et de compétences afin d'être réalisées dans les « règles de l'art » ; ce respect conditionnant dans de nombreux cas le droit d'exercer une activité ou de commercialiser un produit (marquages divers, professions réglementées, etc.). Les diplômes, normes, standards, recueils de bonnes pratiques, etc. visent ainsi à fournir des garanties et à poser des garde-fous sur les méthodes de conception et la qualité des produits finis.

Appliqué aux environnements (au sens « bâti » du terme), cela se traduit par des contraintes toujours plus poussées sur leur conception, construction et utilisation. C'est d'autant plus vrai pour les environnements très fortement encadrés comme les centrales nucléaires ou les environnements médicaux auxquels nous nous sommes intéressés dans ces travaux.

Malheureusement, à l'heure actuelle, il n'existe pas une solution générale et générique permettant de suivre un environnement de sa conception jusqu'à son utilisation afin de garantir sa conformité à l'arrangement normatif auquel il est soumis. Ainsi, les travaux et produits actuels s'intéressent soit à l'amont, soit à l'aval, mais sans fil conducteur général et sans aucun lien entre ces différentes analyses.

Dans cette thèse, nous nous sommes attachés à la définition d'une approche globale et générique d'Ingénierie des Exigences permettant de suivre un environnement normé de sa conception jusqu'à son utilisation (Figure 12.1 issue du Chapitre 5).

Cette approche globale est composée de trois phases différentes, calquées sur le cycle de vie de l'environnement ciblé :

Phase « 0 » Analyser l'arrangement normatif d'un type particulier d'environnement normé à l'aide d'une typologie de contraintes normatives,

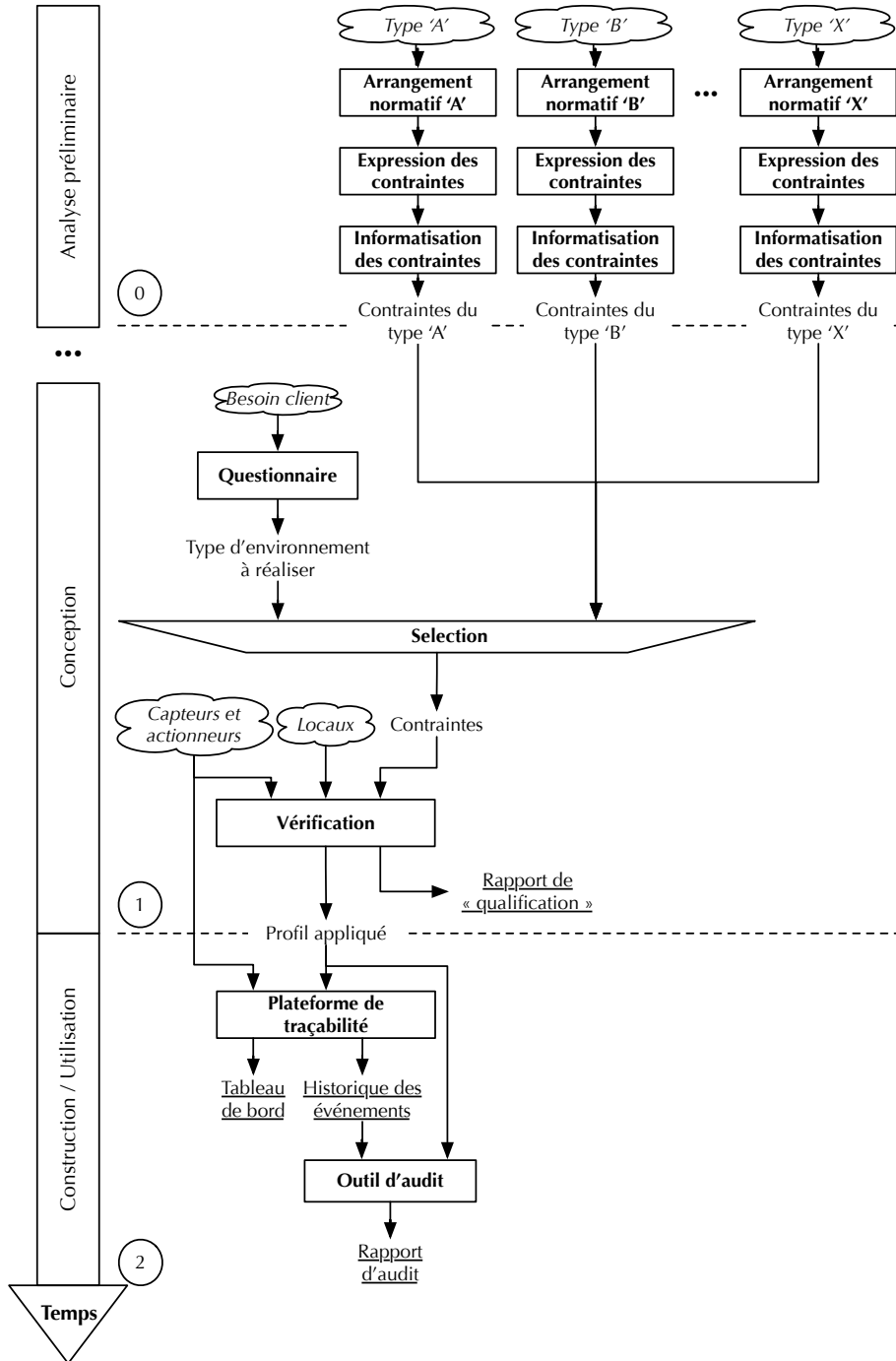


FIGURE 12.1 – Méthodologie proposée

élaborée durant cette thèse.

phase « 1 » Confronter une modélisation de l'environnement prévu avec les contraintes listées dans la phase précédente. À l'issue de cette confrontation, une forme de « rapport » doit contenir l'ensemble des contraintes à vérifier directement lors de l'usage de l'environnement.

phase « 2 » Vérifier en temps réel, à l'aide d'une plateforme de traçabilité déployée « sur site », le respect des contraintes issues de la phase précédente. Toutes les données relevées sur site sont enregistrées afin d'être mises à disposition d'un outil d'audit visant à réaliser des analyses « macroscopiques » sur le fonctionnement de l'environnement.

Durant ce doctorat, nous nous sommes concentrés sur les deux premières phases de ce cadre intellectuel général, c'est-à-dire l'analyse préliminaire d'un type d'arrangement normatif (phase « 0 ») et la validation de la conception de l'environnement projeté (phase « 1 »).

Notre deuxième contribution (Chapitre 6) est une manière originale de représenter les arrangements normatifs, sous la forme d'une structure réticulaire. Ainsi, notre graphe de normes est constitué de nœuds représentant les textes normatifs et de relations représentant les liens conceptuels existant entre ceux-ci.

Plus précisément, chaque nœud est identifié par son triplet « nom, version et révision », il contient l'ensemble des métadonnées exprimées par les experts et ses relations sont issues des exigences conceptuelles identifiées. Ces relations désignent tant les liens conceptuels existant entre des normes que l'application de contraintes par la source sur la destination de la relation.

L'implémentation réalisée pour nos expérimentations (Chapitre 8) propose un framework agnostique de l'implémentation de l'outil d'aide à la qualification. Les nœuds du graphe n'étant que les véhicules de la représentation de la norme concernée, ils n'y posent aucune contrainte.

Notre troisième contribution (Chapitre 7) est une manière originale d'analyser la conformité d'un projet de bâtiment en utilisant pour cela la (méta)modélisation orientée aspect.

Plus précisément, nous proposons d'employer :

- un métamodèle initial permettant de réaliser un modèle du bâtiment « physique » (pièces, murs, portes, etc.) ;
- un ou plusieurs aspects permettant d'ajouter au métamodèle initial l'instrumentation prévue des locaux (capteurs, actionneurs, systèmes tiers, etc.) ;
- un aspect par texte normatif à respecter pour atteindre la conformité finale. Chacun de ces aspects introduit de nouveaux concepts et pose de nouvelles contraintes sur le travail précédemment réalisé.

Chaque métamodèle ou aspect est contenu dans le nœud représentant la norme concernée. La suite d'aspects à appliquer sur le métamodèle initial dépendra des nœuds présents dans le graphe de normes et liés au type d'environnement à valider.

Ainsi, une forme de « chef d'orchestre » doit interroger le graphe de norme pour en extraire la liste des étapes (c.-à-d. des métamodèles et aspects) à traiter pour arriver à représenter totalement un projet et en vérifier les contraintes. À chaque étape, celui-ci applique l'aspect sur le métamodèle précédent, propose à l'utilisateur de modifier son modèle afin d'utiliser les nouveaux concepts

disponibles puis applique les contraintes pour vérifier la conformité actuelle du projet.

Nos expérimentations, qu'elles soient doctorales (pour la réalisation des implémentations et des cas illustratifs de cette thèse) ou industrielles (dans le cadre de feu la société Nocosium), nous ont convaincus de la pertinence et de la faisabilité de la phase « 0 ». Il est tout à fait possible d'arriver à des bases de données normatives de taille importante et largement réutilisables entre les analyses.

En revanche — et à notre grand regret —, il reste encore beaucoup de travail sur la phase « 1 ». En effet, l'outillage logiciel disponible pour le type exact d'approche que nous souhaitons adopter n'est pas suffisamment mature pour une utilisation industrielle et requiert encore un important travail d'ingénierie pour trouver son public (cf. Section 14.3).

Enfin, nos premières expérimentations dans le cadre des transferts technologiques de ces travaux, nous rendent confiants quant à la réalisation de la phase « 2 ». Nous avons par ailleurs défini une version alternative de notre approche générique (Figure 12.1) éludant la phase « 1 » afin de passer directement de l'analyse d'un type d'environnement normé à sa vérification *in situ*.

Ce type d'ajustement peut permettre de réaliser les premières mises en exploitation de nos contributions, sans attendre de disposer de solutions matures pour la phase « 1 ».

Limites

Dans cette thèse de doctorat, nous avons présenté plusieurs contributions originales, illustrées sur un cas simplifié d’environnement normé, nous ayant servi de « fil rouge » tout au long de ce document.

Lors de la réalisation de cette illustration et des expérimentations internes à l’entreprise Nocosium, plusieurs limitations ou points de discussions ont été identifiés.

L’objectif de ce chapitre est de présenter ces différentes limitations et surtout d’y apporter un regard critique.

13.1 Limites à notre analyse du cadre normatif

Lors de la conception de cette méthodologie, nous avons fait de nombreuses itérations entre la théorie et l’expérimentation, entre le concept « sur feuille » et les cas réels de normes et de bâtiments. Ces aller-retour nous ont permis d’identifier un certain nombre de limites « conceptuelles » à notre approche.

Ce sont ces limites que nous allons maintenant présenter.

13.1.1 Représentation réticulaire des normes

Un premier ensemble de limitations concerne la représentation réticulaire des normes. Ce choix si particulier, bien qu’il ouvre des pistes de recherches et de développements intéressantes, doit être manipulé avec précaution.

Structure cyclique

L’ensemble de notre méthode de travail repose sur la transformation d’un arrangement normatif en un graphe de normes avec une intégration des liens conceptuels sous forme de relations entre les nœuds.

Ce graphe a une structure atypique, liée au contexte très particulier de son élaboration. Tout n'est pas possible. Ainsi, une certaine structure, qualifiée de « diamant » dans cette thèse, se fait jour et c'est cette forme si particulière que nous exploitons. Cette exploitation passe par l'extraction de DAG représentant l'ensemble des normes à respecter pour réaliser un environnement valide, sous forme d'étapes.

Par définition, les DAG sont des graphes acycliques et c'est une caractéristique primordiale pour la réalisation de notre tri topologique, tel qu'indiqué dans les algorithmes 1 et 2 (pages 94 et 95). L'application de ces algorithmes sur des graphes cycliques aura une conséquence extrêmement simple : le graphe final obtenu après la suppression de toutes les relations contiendra un ensemble de nœuds et de relations formant un cycle. Ceux-ci n'auront jamais été inclus dans la liste de nœuds triés, car le prédicat « ***m n'a plus d'autre relation entrante*** » (ligne 8, page 94) ne sera jamais validé dans le cycle. Par conséquent, nous ne pourrons jamais aboutir à un ordre total sur les nœuds de ce DAG et appliquer notre méthode (Figure 13.1).

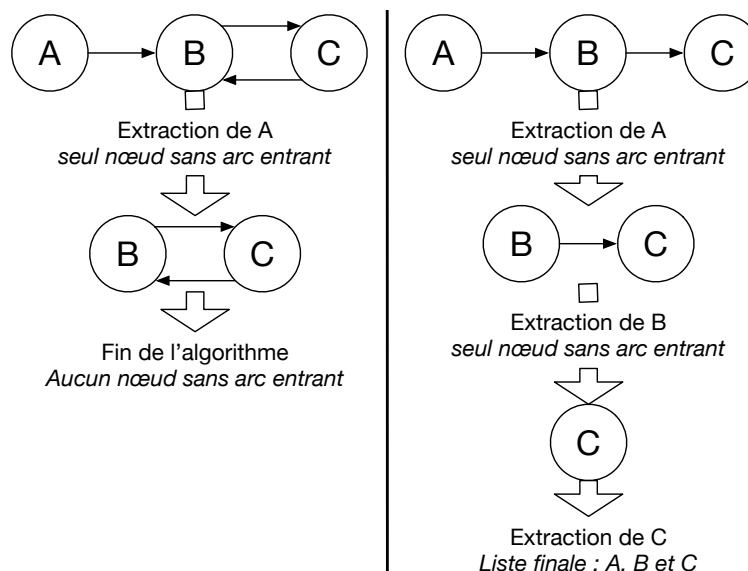


FIGURE 13.1 – Illustration de l'impossibilité d'appliquer un tri topologique sur un graphe cyclique

Pour traiter cette limitation, plusieurs solutions existent, nous allons commencer par traiter les solutions les plus évidentes et « naïves » afin de pouvoir proposer une synthèse finale « pertinente ».

Nous ne pouvons « casser » arbitrairement les relations. Celles-ci représentent des dépendances conceptuelles, notre façon d'exprimer qu'une norme requiert les concepts apportés par une autre pour pouvoir être vérifiée. Par conséquent, briser un cycle revient éventuellement à traiter un nœud sans avoir la certitude d'avoir défini tous les concepts requis. Cela peut fonctionner dans certaines circonstances, mais cela ne peut être appliqué « aveuglément ».

Dans un registre similaire, nous pourrions introduire seulement les concepts et vérifier les contraintes dans un dernier temps. Néanmoins, il nous paraît « dangereux » de laisser un utilisateur représenter l'intégralité de son environnement (y compris ses aspects les plus pointus et précis) pour terminer par une vérification qui pourrait mettre en défaut des paramètres initiaux.

Maintenant les solutions naïves traitées, prenons le temps d'analyser correctement le problème. Qu'est-ce qu'un « cycle » dans la modélisation de l'environnement, dans une perspective « métier » ?

C'est tout simplement une norme qui impose des contraintes sur les concepts exprimés par une autre norme, cette dernière faisant de même sur la première (avec potentiellement plusieurs intermédiaires). Or cette situation est tout à fait impossible, cela empêcherait l'application d'un groupe de normes !

Il y a donc deux origines possibles à une telle situation :

1. Le législateur (ou tout autre éditeur de textes normatifs) s'est fourvoyé et a introduit un cycle dans l'applicabilité de ses textes. Ce type de situation est plutôt rare et par conséquent devrait entraîner une suspicion sur le travail de modélisation. Toutefois, cela reste possible et notre contribution est un outil de représentation et de détection de ce genre d'erreur ¹.
2. Les experts « métiers » et « informatiques » ont commis des erreurs dans l'expression des contraintes induites ou dans la représentation réticulaire des textes. Ce type de situation est la plus fréquente ².

En résumé, dans la plupart des cas, ce type de problème est le symptôme d'une erreur de modélisation et une vérification de la modélisation devant être le premier réflexe à adopter. Cependant, dans les rares cas où de vraies erreurs normatives sont constatées, nous conseillons de modifier la modélisation pour « rompre » le cycle.

Contradiction des contraintes

Cette limitation est très proche de la précédente, mais concerne une autre partie de l'analyse et le problème surviendra ultérieurement.

Précédemment, nous traitons l'interdépendance des normes, un problème que l'on pourrait qualifier de « syntaxique », c'est-à-dire dans l'expression de l'arrangement normatif, et non dans son contenu (bien qu'ici la forme et le fond soient fortement liés).

Nous traitons ici l'appréhension d'arrangement normatif dont l'une des normes impose la présence, l'absence ou la configuration particulière d'un équipement (donc une contrainte statique d'instrumentation) et une autre impose une contrainte opposée ; c'est-à-dire l'absence, la présence ou une configuration antagoniste des mêmes équipements. La question est d'ordre « sémantique », elle n'empêchera donc pas la « résolution » du graphe et la découverte de la

1. Naturellement, cela peut sembler facile à éviter sur deux textes, c'est beaucoup plus complexe dans un cadre « réel » avec un très grand nombre de textes différents et plusieurs éditeurs. Les cycles pourraient apparaître seulement sur des dizaines de textes et dans des cas particuliers d'application d'une norme. Éviter ce type de situation fait partie du travail des organismes de normalisation (qu'ils soient étatiques ou non).

2. Par conséquent, nous conseillons aux lecteurs souhaitant reproduire nos travaux de développer des outils pour la détection de cycles dans leur modélisation.

suite d'étapes à réaliser pour représenter puis vérifier un environnement normé, mais plutôt de dérouler ces étapes. Donc, ce n'est pas dans cette étape-ci que se posera le problème, mais dans la suivante, lorsque l'on cherchera à qualifier ce projet.

Notre approche de la représentation des arrangements normatifs est totalement agnostique à la représentation des normes et des contraintes elles-mêmes. Si c'est, par ailleurs, l'un des intérêts majeurs de notre contribution, cela retarde l'identification des contradictions normatives. Elles apparaîtront seulement lors de la qualification d'un projet, soit relativement éloignées de leur introduction.

Pour ce type de limitation, nous pouvons facilement fournir au lecteur curieux un exemple de contradiction (tiré du domaine médical), présenté dans la Figure 13.2.

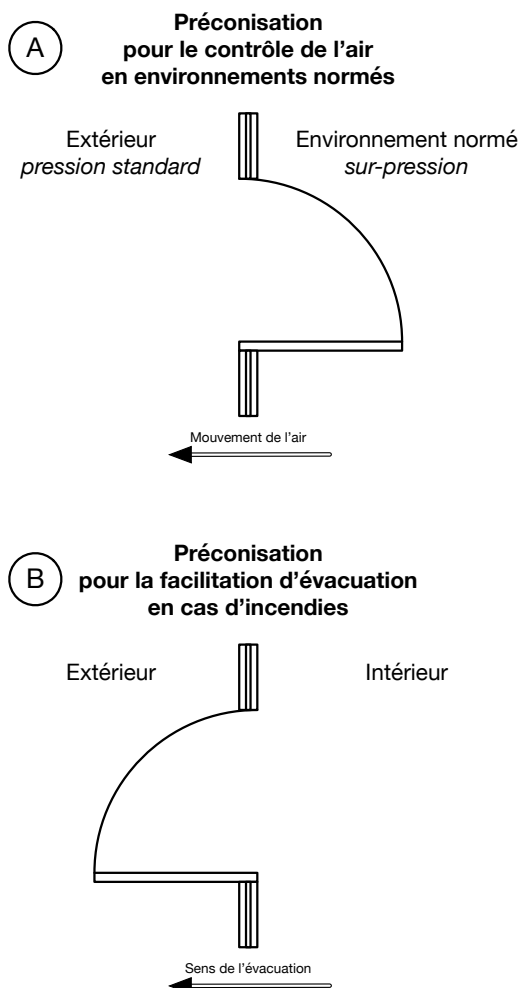


FIGURE 13.2 – Illustration d'un conflit de contraintes normatives

Les paliers de surpression des pièces dans les environnements normés médicaux

(cf. Chapitre 2.3.2, p. 25) imposent — pour plus de robustesse et d'efficacité — des portes dont le sens d'ouverture permet une fermeture « automatique » par le jeu de pression. C'est le cas A dans la Figure 13.2, le mouvement d'air induit par la surpression va mécaniquement fermer la porte, si la porte avait eu un sens d'ouverture inverse (cas B) alors la porte se serait naturellement « ouverte » et les moteurs, les joints, etc. auraient été sollicités de manière bien plus importante (sans même aborder la difficulté de fermeture pour les utilisateurs).

Malheureusement, si cette porte est sur le passage d'évacuation « incendie » (ce qui peut être fort probable selon l'activité menée), la contrainte inverse est imposée, afin de faciliter la sortie des personnes. C'est le cas B dans la Figure 13.2, lors d'un incendie, les personnes présentes vont tenter de sortir à tout prix et créer une cohue. Dans cette situation, l'obligation de « tirer » la porte pour pouvoir sortir est réhabilitaire, on souhaitera plutôt la « pousser ».

Ainsi, dans de telles situations, quelle contrainte privilégier et à quel texte se référer ? Doit-on prioriser les normes « métiers » ou celles « génériques » ? C'est d'autant plus complexe que dans l'exemple proposé, les deux textes posent des contraintes de « sécurité », si l'on facilite la fermeture de la zone, c'est également pour en améliorer la sécurité et les performances. Donc, de deux règles de sécurité antagonistes, quelle est celle que l'on doit appliquer ?

Pour ce type de limitation, après les vérifications d'usage sur la correction de l'expression des contraintes dans les normes concernées, il faut s'en référer aux experts « métiers » pour comprendre comment traiter le problème.

Dans de nombreux cas, nous estimons plus pertinent de suivre la procédure actuellement en vigueur dans les environnements normés : constater l'écart entre les préconisations et la réalisation, puis le justifier. Ici, l'outil d'aide à la qualification pourra automatiquement générer la constatation de ce genre de choix entre deux contraintes irréconciliables.

13.1.2 Bilan

Dans cette section, nous avons détaillé les limitations conceptuelles rencontrées lors de la mise en œuvre de cette méthodologie de représentation et d'exploitation des arrangements normatifs. Ces limitations ne sont en aucun cas réhabilitaires et, au contraire, la souplesse de la méthode et son isolation de la manière de modéliser les contraintes et les projets réels d'environnements se sont avérées extrêmement utiles.

Après analyse, les quelques contraintes liées à la modélisation réticulaire peuvent facilement être résolues. Ces contraintes traduisent principalement des défauts d'analyse ou des défauts d'implémentations des algorithmes de tris, cela n'est absolument pas réhabilitaire.

13.2 Aide à la qualification : approche théorique

Lors de la définition de cette approche, lors de sa conception puis de son expérimentation, plusieurs limitations (ou point de discussion) ont été identifiées, nous allons maintenant les présenter.

N'ayant pas identifié de limites « majeures » dans notre questionnaire, nous nous concentrerons sur l'outil d'aide à la qualification.

13.2.1 « Aspect-Oriented Modeling » ou « Métamodèle fixe » ?

Le premier sujet abordé n'est pas à proprement parler une « limitation », mais plutôt une approche alternative à notre représentation en « millefeuille » des arrangements normatifs sur laquelle il nous semble intéressant de revenir.

Ainsi, il aurait été tout à fait envisageable de représenter chaque arrangement normatif sous la forme d'un métamodèle « global » (potentiellement issu de la composition de plusieurs métamodèles) couvrant tous les aspects du problème et incluant toutes les règles et contraintes liées au type d'environnement considéré. Cette approche, alternative à celle présentée tout au long de cette thèse, aurait été tout à fait possible ; mais nous l'avons cependant écartée lors de nos expérimentations. Si cette méthodologie peut paraître intéressante dans un premier temps, cet avantage s'estompe rapidement lorsque l'on commence à expérimenter plus précisément l'implémentation.

L'avantage d'une modélisation « globale » réside dans la simplicité de son exploitation par l'utilisateur. Ainsi, le métamodèle général peut facilement être utilisé par l'outillage classique d'IDM et permettre ainsi d'expérimenter très rapidement la constitution et la validation d'un projet d'environnement normé.

En terme d'« outillage informatique » c'est donc très nettement la solution la plus simple, mais en prenant un peu de recul, le constat est bien plus nuancé.

Tout d'abord, la vérification n'est plus réalisable « étape par étape » mais globalement. Ainsi, on perd la notion d'itération (et donc de « focus ») offerte par les aspects. Ce choix permet à l'utilisateur de se concentrer sur un point particulier (celui de l'aspect actuel) avant de passer au sujet suivant ; les vérifications étaient réalisées successivement, isolant facilement les raisons d'une non-conformité. Nous perdons cet avantage pour revenir à une vérification globale, moins pertinente et nettement moins lisible pour l'utilisateur.

Ensuite, lors de la conception des métamodèles, l'intérêt est également nettement moins évident. Durant la phase d'analyse préalable, il faudra bâtir des métamodèles complexes et extrêmement redondants entre les différentes analyses³. Par la force des choses, cela conduira *in fine* les concepteurs à diviser les études en plusieurs métamodèles « dédiés », intriqués selon les thématiques requises par le type d'environnement normé considéré ; une approche s'inscrivant totalement dans la thématique de la « mégamodélisation » [BJV04].

Notre contribution est donc simplement le constat d'une évolution naturelle et prévisible de la conception dû à la nature même des travaux menés. Nous avons seulement fait le choix de l'AOM à d'autres solutions similaires basées sur de la transformation de (méta)modèles [WMJC12, MVG06].

Ainsi, en résumé, si nous pouvons trouver un intérêt lié à l'outillage informatique d'un métamodèle fixe (« complet ») pour chaque type d'environnement

3. Comme nous l'avons établi précédemment, les concepts liés aux bâtiments seront toujours identiques, l'instrumentation sera très fortement semblable, les mêmes normes techniques seront toujours imposées, etc.

normé, cet avantage est très largement remis en cause pour l'utilisateur final et le concepteur des métamodèles.

Par conséquent, notre approche « millefeuille » réalisée par AOM nous a semblé la plus intéressante, sans pour autant fermer la porte à d'autres initiatives comme l'intégration de nos travaux dans une superstructure de type « mégamodélisation » [BJV04].

13.2.2 Complexité et passage à l'échelle

Cette deuxième contrainte prolonge la section précédente ; elle est inhérente à la problématique abordée dans ce doctorat. Comment se comporte notre contribution lorsque les types de projets manipulés se complexifient ou que l'on commence à intégrer de grands nombres de normes optionnelles ou d'aspects complémentaires ?

Nous l'avons abordé précédemment, seul le premier aspect est totalement décidable et repose sur un métamodèle connu. Toutes les étapes suivantes reposent sur des compositions de métamodèles (issus de l'application d'un ou plusieurs aspects sur ce dernier métamodèle connu). Par conséquent, lorsque l'on est face à des projets complexes, les derniers aspects sont appliqués sur des métamodèles particulièrement complexes et pour lesquels une extrême attention aura dû être portée lors de la conception. Est-ce envisageable ?

Pour traiter cette limitation, nous préférons prendre cette problématique à contre-pied. La complexification des analyses est inhérente à notre problème. Quoique nous puissions faire, le nombre de facteurs à prendre en compte et à traiter s'accroîtra avec la taille de l'arrangement normatif. La vraie question est plutôt de savoir si notre approche AOM est plus pertinente qu'une autre dans une telle situation.

Face à des approches « figées » comme les métamodèles uniques (ou les combinaisons fixes de métamodèles), le couplage du graphe de normes avec un traitement à base d'aspects nous a semblé faciliter le passage à l'échelle, en comparaison aux expérimentations sur métamodèle fixe.

En effet, grâce aux aspects, les principales difficultés liées à la reprise des analyses précédentes, à la mise à jour des normes, au traitement de variantes d'arrangements normatifs, etc. s'estompent. Il n'y a plus un métamodèle unique à conserver puis à découper ou reprendre en fonction des nouveaux types d'environnements analysés, seulement de nouvelles normes à traiter et des aspects à rédiger en se basant sur tous les travaux précédents. Nous obtenons ainsi une forme « séparation des responsabilités » entre les différentes normes.

Enfin, pour nuancer la perte de maîtrise sur les métamodèles durant la conception, en réalisant des outils d'aide à la décision pour le graphe de normes, on peut retrouver la simplicité d'un métamodèle unique pour la conception des nouveaux aspects en générant le métamodèle précédent, en fonction des dépendances conceptuelles exprimées.

13.2.3 Industrialisation

Dernière contrainte, celle de l'intégration de notre méthodologie dans le flot de conception classique d'un environnement normé, décrit dans la Section 3.1 (p. 32).

Nos prospections industrielles ont montré l'extrême diversité des logiciels employés et la « fidélité » des entreprises. Lorsque le savoir-faire a été acquis par les salariés, les entreprises sont très réticentes au changement (pour d'évidentes questions de retour sur investissement). Cela rend obligatoire le support d'un très grand nombre de solutions pour être pertinent à grande échelle. Comment parvenir à une intégration efficace entre notre approche et les différents outils du marché ? Il faudrait pouvoir s'interfacer proprement avec ces logiciels pour récupérer les informations fournies par l'utilisateur au logiciel de CAO et travailler conjointement avec celui-ci pour enrichir directement son interface des informations normatives. L'atelier de modélisation deviendrait alors totalement masqué et intégré au logiciel de CAO, l'outil d'orchestration opérant alors en arrière-plan.

Actuellement, nous n'avons su apporter une réponse définitive à cette question, les développements mentionnés précédemment dépassant largement le cadre d'un doctorat. Nous déléguons donc à des travaux post-doctoraux l'intégration industrielle de notre proposition.

13.3 Limites à notre mise en œuvre avec Neo4J

Lors de nos expérimentations et durant l'implémentation de cette plateforme de validation, nous avons rencontré relativement peu de problèmes pertinents à discuter ici. Et, à tout bien considérer, un seul point mérite réellement notre attention, il s'agit de l'impact global d'une modification locale.

La méthode la plus pratique lors de la préparation de nos scénarios (et du développement de la plateforme d'expérimentation) était de procéder à l'insertion des nœuds « dans l'ordre », c'est-à-dire en commençant par le bâtiment et en terminant par le ou les types d'environnement ciblés.

Lorsque des modifications ponctuelles « de test » étaient nécessaires, nous pouvions employer l'interface Web pour exécuter des requêtes CYPHER sur le graphe.

De la sorte, l'ajout se greffait « sur » ou « à côté » des couches actuelles et jamais comme un nouvel « intermédiaire », ce qui simplifiait grandement les algorithmes et restait suffisant pour nos expérimentations.

L'insertion de nœuds « terminaux » — c'est-à-dire, dépendant des nœuds présents dans le graphe mais dont aucun des nœuds actuels ne dépend — suit en effet en tout point les algorithmes présentés dans les chapitres précédents de cette thèse. Cela ne requiert aucun traitement supplémentaire.

En revanche, lorsque l'on insère un nouveau nœud intermédiaire — ce qui correspond, en terme « métier », à l'enrichissement des contraintes normatives pesant sur un type d'environnement ou un concept déjà étudié précédemment — il nous faut refaire une analyse du graphe pour l'ensemble des nœuds impactés par cette opération. En effet, cette insertion peut avoir changé des règles de transitivité existantes : ajout de nouvelles étapes de modélisation, suppression d'analyses redondantes, etc.

Dans le cas où les relations d'un nœud voisin viendraient à être modifiées, il faudrait alors procéder de manière identique à l'analyse de son propre voisinage et reproduire cette opération jusqu'à atteindre le point de stabilité du graphe⁴.

4. On entend par « point de stabilité », l'itération où aucune des relations du graphe n'est

Si cette situation était tolérable pour une plateforme d'expérimentation doctorale, ce ne sera pas le cas lors d'une exploitation industrielle. Dans ce cas, un ordre arbitraire d'insertion devra être implémenté.

13.4 Conclusion

Nous arrivons au terme de ce chapitre sur les limitations identifiées dans nos contributions.

Si nous avons pu apporter des réponses et nuancer certains facteurs que ce soit par leur faible impact dans des cas réels ou par l'existence de méthodes de contournement ou d'ajustements à nos plans initiaux ; il subsiste tout de même des contraintes auxquelles il conviendrait de répondre dans des travaux postdoctoraux.

Ainsi, la question de l'industrialisation de ces travaux et du passage à l'échelle sur des environnements « très » complexes mérite que nous menions des expérimentations plus poussées sur des environnements de taille plus importante. C'est ce à quoi s'attachait Nocosium avant sa liquidation, nous espérons pouvoir poursuivre cela d'une autre manière.

De plus, la problématique de l'outillage pour l'atelier de modélisation est encore bien présente. Notre proposition basée sur KerMeta nous a permis d'illustrer nos propos, de tester notre contribution sur des exemples simples, d'observer la généralité de la proposition, mais c'est encore insuffisant pour une véritable exploitation en situation réelle.

Nous allons présenter certains de ces travaux dans le prochain chapitre.

Perspectives

Sur la base des contributions et limitations présentées dans cette thèse dans la section précédente, nous envisageons plusieurs pistes de travail pouvant enrichir de manière significative ces travaux.

Ce sont ces différentes pistes que nous allons maintenant explorer dans les sections suivantes. Afin de ne pas distraire le lecteur, et de rester dans un contexte de recherche universitaire, nous avons éludé les évidentes poursuites en ingénierie informatique pour améliorer l’outillage existant. Nous avons donc fait le choix de nous concentrer sur la présentation de trois pistes de réflexion, parmi les plus pertinentes et porteuses d’une cohérence « globale ».

Ainsi, dans la Section 14.1, nous débiterons par une discussion des recherches à mener sur la phase d’audit des informations techniques relevées lors de l’usage de l’environnement (soit, le bloc « Outil d’audit » de la Figure 12.1 [p. 204]), c’est-à-dire sur les pistes d’automatisation et d’exploration de données lors de l’audit.

Ensuite, dans la Section 14.2, nous discuterons de l’intégration de nos travaux dans la mouvance dite « Model@run.time » [BFCA14] et sur les potentiels bénéfiques offerts par cette approche. Il s’agit là de traiter du renforcement des travaux pour atteindre l’objectif ultime de pouvoir propager (et utiliser) nos modèles, de la conception de l’environnement jusqu’à l’usage, sans aucune perte d’information.

Enfin, dans la Section 14.3, nous aborderons l’une des pistes industrielles, à savoir l’intégration de nos travaux dans le contexte de la MNB (cf. 3.1.1 [p. 33]) et les apports qu’une approche telle que la notre pourrait apporter à l’outillage existant.

14.1 Auditer un environnement normé

Dans l’ensemble de cette thèse, nous sommes restés relativement silencieux sur « l’outil d’audit ». Si celui-ci a bien été spécifié dans la Section 5.4.2 (puis

rappelé dans les principaux schémas « généraux »), aucune information n'a été donnée quant à sa réalisation.

Pour mémoire, cet outil doit réaliser un audit de l'environnement. C'est une forme bien plus poussée d'analyse utilisant l'ensemble des informations disponibles pour identifier l'origine d'une anomalie ou, à minima, le contexte exact de sa première occurrence.

En opposition au travail mené par la « plateforme de traçabilité », l'outil d'audit conduit des analyses « macroscopiques »¹. Ces dernières visent à établir des liens entre des éléments bien plus disparates, inaccessibles en temps réel, sur des échelles de temps bien différentes.

L'intérêt de cette analyse est de prendre à contre-pied la plateforme de traçabilité, ouvrant ainsi la voie à une nouvelle dimension d'analyse.

Si, durant l'utilisation de l'environnement, les contraintes opérationnelles nous bornent à réaliser un cycle permanent « acquisition des informations → vérification des règles → émission d'alertes », il en va autrement pour l'audit.

Dans celui-ci, les événements ont déjà été analysés et nous savons que les règles de bases ont été validées. Nous cherchons plutôt à détecter l'apparition d'un phénomène qui, s'il n'est pas strictement anormal (auquel cas, la plateforme de traçabilité l'aurait identifié), pourrait avoir conduit à un écart (pour lequel l'audit a été mené), ceci est donc totalement différent.

Cet audit ne peut pas seulement utiliser les modèles (ou les règles) mis aux point dans la phase d'analyse préalable, il doit — au sens le plus littéral du terme — « penser hors du cadre ».

Pour cela, il faut savoir dépasser les modélisations existantes et essayer d'établir des corrélations entre différentes situations identifiées comme incorrectes, dangereuses, etc. C'est la détection de ces liens qui rendra l'audit réellement pertinent, capable de produire de nouvelles règles conduisant, non plus à des alarmes strictes, mais à des suspicions, des conseils, etc.

Comme nous l'avions déjà évoqué dans le Chapitre 5, une telle démarche s'inscrit totalement dans les approches classiques de l'Intelligence Artificielle [Nil14], plus particulièrement de l'apprentissage automatique (*machine learning* en anglais) [MRT12] et notamment, dans notre cas, du sous-domaine de l'« apprentissage supervisé » (*supervised learning* en anglais).

Sur cette thématique, nous nous rapprocherons des travaux menés actuellement dans plusieurs équipes de recherche [TA12, PHG⁺12, BMP⁺14].

Nous proposons d'utiliser des statistiques pour identifier des corrélations entre des situations identifiées comme « à risque » ou « interdites » et aboutir à la définition de nouvelles règles, l'isolation de « signaux faibles » permettant d'enrichir l'analyse en temps réel. Ensuite, nous proposons d'utiliser des outils d'apprentissage supervisé pour arriver à détecter de nouvelles situations risquées en se basant sur les précédentes analyses.

En synthèse, nous complétons le déterminisme adopté dans ces travaux doctoraux afin de tirer parti des analyses statistiques, de l'exploration de données

1. En opposition aux analyses « microscopiques » réalisées en temps réel, si l'on nous pardonne cet abus de langage.

et du *supervised learning* pour identifier des situations « à risque » pour lesquelles il n'existe pas de certitude absolue.

De plus, si l'on applique la même démarche à un degré supérieur d'abstraction, en considérant maintenant non plus des projets « particuliers » mais directement des types d'environnements normés, nous atteignons un niveau d'analyse et de réflexion bien supérieur et fort prometteur.

En effet, cela nous ouvre un champ complet d'exploration, plus orienté « métier », analysant le respect et l'applicabilité des corpus normatifs globaux, du point de vue du législateur ou d'importants cabinets de maîtrise d'œuvre et d'ouvrage².

Comparer les résultats d'analyse de multiples environnements d'un même type permettrait de déterminer la prévalence des infractions et d'isoler des comportements globaux similaires. Cela peut aboutir à d'intéressantes réflexions ; ainsi, une infraction dont la prévalence est de 1 % sur l'ensemble des environnements observés n'aura que peu d'intérêt, mais qu'en est-il d'anomalies observées dans plus de 10 % des situations ?

Pour le législateur, cela signale un problème normatif, qu'il soit dans le texte lui-même (manquant de clarté ou étant trop contraignant) ou dans la vérification de sa mise en application. Pour le maître d'œuvre, c'est un levier pour l'amélioration de ses environnements de façon notable face à la concurrence.

14.2 Model@run.time

Dans ce travail de doctorat, nous avons de nombreuses fois souligné l'importance de préserver l'information, de ne surtout pas l'altérer. Nous avons toujours privilégié l'enrichissement des modèles à leur transformation afin d'éviter toute perversion qui nuirait à la pertinence des analyses futures.

Dans cet optique, nous débutons par une modélisation des arrangements normatifs qui sera reprise au moment de la modélisation d'un projet d'environnement pour enfin être instanciée lors de la mise en exploitation du projet. De cette manière, toutes les informations obtenues lors de l'étude des arrangements normatifs restent disponibles lorsque l'on traite un projet particulier. Ces informations sont toujours disponibles lors de l'instanciation du modèle final, permettant ainsi de maximiser la pertinence des analyses menées en temps-réel.

Malheureusement, cette dernière opération, aussi contextualisée soit-elle, marque la fin de la préservation des modèles. En effet, l'instanciation est une forme d'« applanissement », nous allons perdre la profondeur et la complexité des (méta-)modèles précédents pour en tirer de « simples » fichiers de configurations. Nous ne pourrons plus revenir au (méta-)modèles initiaux et nous serons alors « prisonniers » des éléments extraits lors de l'instanciation. Ce sera notre unique cadre de pensée et toute mise à jour nécessitera une nouvelle instanciation. . .

Notre objectif est de poursuivre la conservation des données jusqu'à l'exploitation de l'environnement, de sorte que la plateforme de traçabilité dispose d'un cadre de pensée bien plus structuré et étoffé, qu'elle pourra (si nécessaire)

2. Nous estimons que seuls des cabinets importants souhaitant atteindre les plus hauts degrés d'exigence et disposant des moyens humains et techniques adéquats trouveront un intérêt à de telles analyses.

questionner et — pourquoi pas — faire évoluer. En d’autres termes, il s’agit de rendre la plateforme de traçabilité capable d’une forme « d’introspection », une capacité à comprendre, lorsqu’une règle est enfreinte, l’origine de cette contrainte et, par corollaire, sa justification.

Les retours d’expériences complèteront idéalement ceux de nos précédentes pistes de réflexion (s’interrogeant sur la prévalence et la répartition des écarts normatifs pour un type d’environnement particulier), permettant ainsi au législateur d’avoir un regard critique sur la réalité de l’application des corpus normatifs.

D’un point de vue thématique, la conservation du cadre de modélisation et la définition d’un modèle « exécutable » s’inscrit parfaitement dans la mouvance dite « *model@run.time* » [BFCA14, Fou13].

Historiquement [BBF06], l’objectif était de garder la couche de modélisation qui résolvait les problèmes de construction logicielle complexe au moment de la conception, et de l’inclure directement dans la plateforme finale. Ainsi, présent également à l’exécution (*runtime* en anglais), ce modèle pouvait alors servir l’ensemble des outils qui manipulent le système pendant son fonctionnement afin d’en faire un système dynamiquement adaptable.

Par la suite [BBF09], le modèle est devenu une abstraction devant piloter les systèmes adaptatifs (et non être le système lui-même). Deux visions s’affrontent alors : les modèles comportementaux et structurels. Les premiers sont les plus souvent représentés sous forme d’automates (une abstraction du comportement attendu du système général) et les seconds se sont assez naturellement rapprochés des paradigmes composants [MBJ⁺09] (une abstraction de l’environnement dans lequel fonctionne le système).

Les débouchés possibles d’application de ce paradigme sur notre cas d’application sont multiples. Tout d’abord, la réalisation d’une version « exécutable » du modèle ouvre la porte à d’intéressantes perspectives de simulation du comportement de l’environnement (le modèle) selon l’utilisation qui en est faite³. Ensuite, disposer du modèle à l’exécution nous permettrait de l’utiliser comme un « super ADL » [Fou13] et tout le travail de recherche serait alors de définir le niveau d’abstraction pertinent pour ne pas « noyer » les outils de traçabilité et d’audit tout en conservant le maximum d’information.

14.3 Intégration à la MNB

Dans la Section 3.1.1 (p. 33), nous avons présenté le concept de « Maquette Numérique du Bâtiment » [Val16].

Pour mémoire, la MNB est une méthode de travail, un processus, utilisant une maquette numérique 3D intelligente comme élément central des échanges entre les différents intervenants à l’acte de construire. Initiée par l’architecte, cette maquette est ensuite rendue accessible aux différents bureaux d’étude dans le but d’être complétée, voire modifiée techniquement. Chaque acteur enrichit alors le projet de son propre savoir-faire et de sa propre expérience.

3. Cela demandera de disposer d’une représentation fidèle d’environnement en fonctionnement, un corollaire de la piste de réflexion précédente.

« Techniquement », la MNB associe à une représentation en 3D du bâtiment une base de données de l'ensemble des éléments constituant l'ouvrage.

L'informatisation massive des procédés dans les divers secteurs industriels a permis à la thématique de la MNB de disposer d'une attention particulière et fait ainsi l'objet de projets nationaux [Pro16b], de conférences nationales et internationales [BIM16a,BIM16b,Pro16a], d'ouvrages [Ken15,LS16,HM15,Hol16], etc...

Il ne faut cependant pas se méprendre sur les ambitions poursuivies par cette informatisation.

Actuellement, la MNB est un outil de collaboration, de communication et de consultation pour les différents intervenants lors de différentes phases de vie d'un projet de bâtiment (conception, construction et gestion patrimoniale).

En revanche, les solutions actuelles fournissent peu de retour sur le projet réalisé. Ceux-ci se limitent principalement à des calculs d'ingénierie « technique » (fluides, thermique, etc.), d'ingénierie « financière » (coûts, rentabilité, etc.) ou des vérifications « métier » (cohérence entre espace disponible et matériel prévu, absence de conflit entre les différents passages de câbles ou de tuyau et les cloisons, etc.).

Avec le bâtiment complet en 3D (intégrant ses courants forts, ses courants faibles, son aéraulique, etc.) et la base de données complète des équipements prévus, il devrait néanmoins être possible de réaliser de nombreuses analyses et aides à la décision. Il s'agit là d'une thématique qui s'inscrirait totalement dans celle de l'exploration de données (*data mining* en anglais).

Les travaux présentés dans cette thèse de doctorat peuvent par conséquent s'inscrire totalement dans ces axes de recherche.

L'objectif serait d'utiliser les informations disponibles comme les éléments de la phase de modélisation⁴ puis d'appliquer nos vérifications pour analyser la conformité du projet. C'est le premier axe d'interaction (de la MNB vers nos outils).

Ensuite, il faudra reporter dans le modèle 3D et la base de données les résultats de notre analyse afin de permettre au manipulateur de disposer d'un retour visuel et contextualisé de son travail. C'est le deuxième axe d'interaction (de nos outils vers la MNB).

Cette perspective est un important travail d'ingénierie dans la mesure où cela nécessiterait une intrication entre les outils de la MNB et notre propre outillage. Pour mener à bien nos objectifs, il faudrait alors pouvoir « lire et écrire » de manière interactive dans la maquette, une opération complexe totalement dépendante des logiciels de modélisation⁵, etc.

4. Cela consisterait donc à transformer les informations 3D et la base de données en modèles informatiques tels que nous les avons présentés précédemment.

5. Qui sont, comme nous l'avons évoqué dans la Section 3.1.1, très nombreux.

Cinquième partie

Annexes



Modélisation réticulaire avec un SGBDR

A.1 Adaptation à notre objectif

La Figure A.1 présente une modélisation UML de notre objectif, totalement applicable dans un SGBDR. La représentation peut paraître succincte (deux classes et une annotation sur une relation), mais c'est très largement suffisant pour répondre à notre problématique.

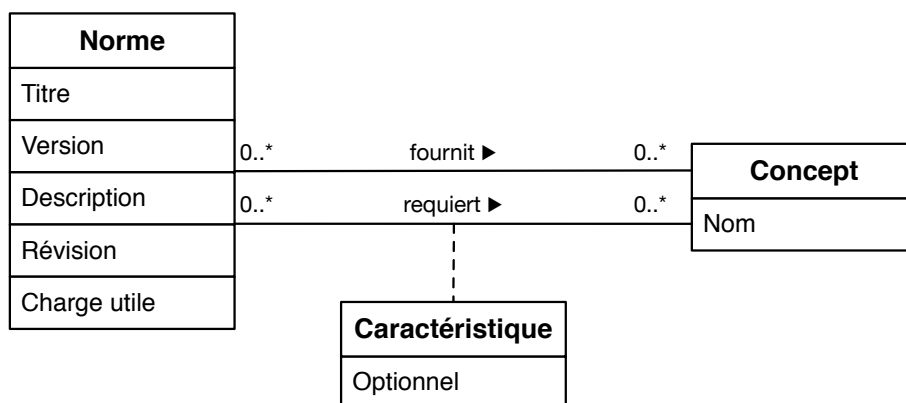


FIGURE A.1 – Modélisation réticulaire des normes : représentation UML

La compréhension de ce schéma est relativement aisée :

- Chaque norme serait représentée par une nouvelle instance de la classe « Norme », on indique dans celui-ci son titre, sa version, sa description, sa révision et sa charge utile.
- Les liens entre les normes sont représentés par deux relations distinctes vers une classe « Concept » permettant, comme son nom l'indique, de représenter les concepts requis et fournis par chaque norme.

- La relation « requiert » est annotée par une classe « Caractéristique » permettant d'identifier les relations optionnelles.

Pour illustrer cette représentation (et les suivantes), nous utiliserons le graphe de normes (correspondant à l'expression des contraintes de trois types d'environnements normés) présenté dans la Figure A.2¹.

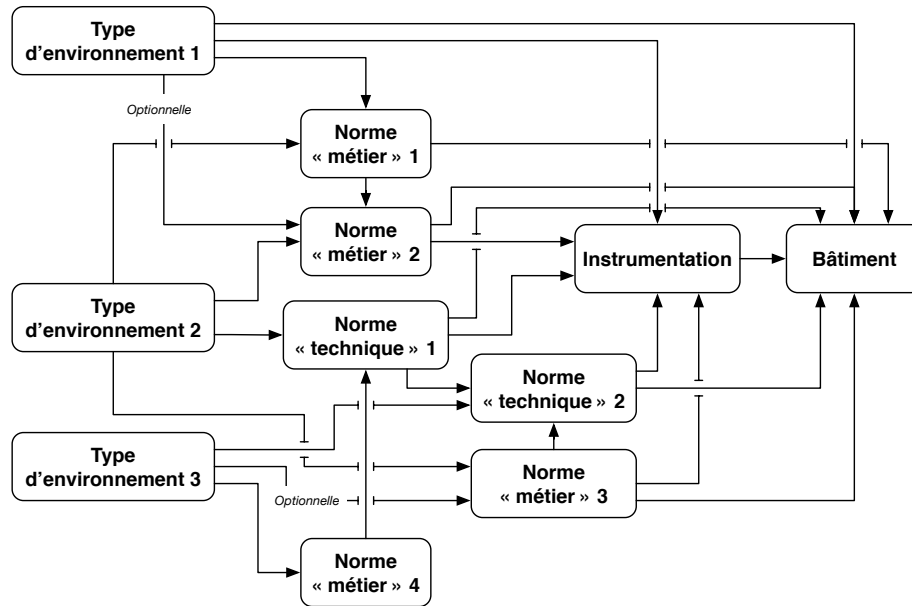


FIGURE A.2 – Graphe de normes pour l'illustration des différents types de représentation

L'une des mises en œuvre possibles de cette modélisation UML dans une base relationnelle² est présentée dans les schémas A.3 et A.4.

On trouve dans cette représentation deux tables « principales », « Norme » et « Concept », chacune décrivant le concept apparenté. Afin de faciliter la compréhension et la représentation de cette représentation, nous avons volontairement masqué la « description » et la « charge utile » des normes (une information non pertinente dans le contexte présent).

D'un point de vue technique, nous utilisons les champs « titre », « version » et « révision » comme la clé primaire composite de la table « Norme ». Cela induit l'unicité de ce triplet dans l'intégralité de la table, une contrainte conforme à notre objectif. La table « Concept » contient un seul champ « nom », déclaré comme une clé primaire et nous garantissant ainsi son unicité dans l'ensemble de la représentation.

Nous trouvons trois tables supplémentaires, « Concepts fournis », « Concepts

1. Le lecteur attentif n'aura pas manqué de reconnaître le graphe employé pour présenter notre contribution sur l'analyse d'un arrangement normatif dans le Chapitre 6, Figure 6.5

2. Plusieurs représentations relationnelles sont possibles pour la même modélisation UML, nous présentons ici celle employée pour prendre nos décisions de mise en œuvre.

Norme		
<i>Titre (pk)</i>	<i>Version (pk)</i>	<i>Révision (pk)</i>
Bâtiment	1	1
Instrumentation	1	1
Norme « technique » 1	1	1
Norme « technique » 2	1	1
Norme « métier » 1	1	1
Norme « métier » 2	1	1
Norme « métier » 3	1	1
Norme « métier » 4	1	1
Type d'environnement 1	1	1
Type d'environnement 2	1	1
Type d'environnement 3	1	1

Concept
<i>Nom (pk)</i>
Bâtiment
Instrumentation
NT 1
NT 2
NM 1
NM 2
NM 3
NM 4

Concepts fournis				
<i>ID (pk)</i>	<i>Nom (fk)</i>	<i>Titre (fk)</i>	<i>Version (fk)</i>	<i>Révision (fk)</i>
1	Bâtiment	Bâtiment	1	1
2	Instrumentation	Instrumentation	1	1
3	NT 1	Norme « technique » 1	1	1
4	NT 2	Norme « technique » 2	1	1
5	NM 1	Norme « métier » 1	1	1
6	NM 2	Norme « métier » 2	1	1
7	NM 3	Norme « métier » 3	1	1
8	NM 4	Norme « métier » 4	1	1

FIGURE A.3 – Graphe de normes : représentation relationnelle (1/2)

Concepts requis				
<i>ID (pk)</i>	<i>Nom (fk)</i>	<i>Titre (fk)</i>	<i>Version (fk)</i>	<i>Révision (fk)</i>
1	Bâtiment	Instrumentation	1	1
2	Bâtiment	Norme « métier » 1	1	1
3	Bâtiment	Norme « métier » 2	1	1
4	Bâtiment	Norme « technique » 1	1	1
5	Bâtiment	Norme « technique » 2	1	1
6	Bâtiment	Norme « métier » 3	1	1
7	Bâtiment	Type d'environnement 1	1	1
8	Instrumentation	Type d'environnement 1	1	1
9	Instrumentation	Norme « métier » 2	1	1
10	Instrumentation	Norme « technique » 1	1	1
11	Instrumentation	Norme « technique » 2	1	1
12	Instrumentation	Norme « métier » 3	1	1
13	NM 1	Type d'environnement 1	1	1
14	NM 1	Type d'environnement 2	1	1
15	NM 2	Type d'environnement 1	1	1
16	NM 2	Type d'environnement 2	1	1
17	NM 2	Norme « métier » 2	1	1
18	NT 1	Type d'environnement 2	1	1
19	NT 1	Norme « métier » 4	1	1
20	NT 2	Norme « technique » 1	1	1
21	NT 2	Type d'environnement 3	1	1
22	NT 2	Norme « métier » 3	1	1
23	NM 3	Type d'environnement 2	1	1
24	NM 3	Type d'environnement 3	1	1
25	NM 4	Type d'environnement 3	1	1

Caractéristique		
<i>ID (pk)</i>	<i>ID Relation requise(fk)</i>	<i>Optionnel</i>
1	15	vrai
2	23	vrai

FIGURE A.4 – Graphe de normes : représentation relationnelle (2/2)

requis » et « Caractéristique ». Ce sont de simples tables de liaison, établissant des liens entre les clés primaires des deux premières tables.

Ainsi, « Concepts fournis » et « Concepts requis » établissent des liens entre des normes — identifiées par le triplet \langle titre, version, révision \rangle — et des concepts — identifiés par leur nom — pour les concepts fournis (respectivement requis). Afin de faciliter la lecture de ces tables, nous avons réduit les liaisons conceptuelles à leur plus simple expression ; chaque norme se déclare comme « concept fourni » et toutes les autres normes utilisent ce concept pour exprimer leurs exigences. Bien que « simpliste », cette approche permet d’alléger les tables sans nuire à notre réflexion.

La dernière table « Caractéristique » permet de représenter les exigences optionnelles, elle inclut simplement une référence vers une déclaration d’exigence.

Dernier point, dans les sections suivantes nous commettrons sciemment l’abus de langage consistant à nommer « nœud » une norme dans le contexte d’une base de données relationnelle. C’est effectivement une impropriété dans la mesure où dans une telle représentation, les normes sont des « enregistrements » de la table « Norme » et non des « nœuds » d’un graphe. Cependant, nous n’estimons pas cet abus de langage préjudiciable dans la mesure où il permet de simplifier et d’alléger la réflexion autour de l’implémentation relationnelle.

De manière identique, nous nommerons indifféremment « arc » ou « relation » les liens conceptuels existant entre deux normes, bien que, dans notre représentation, il n’existe pas de formulation « simple » ou « directe » des relations entre normes.

A.2 Pertinence de cette représentation

Bien que possible, travailler sur cette représentation n’est pas chose aisée... Nous allons maintenant détailler les raisons de cette complexité en présentant la réalisation de chaque opération.

A.2.1 Filtrer les redondances

Avec une telle structure, la détection des redondances est loin d’être facile à exprimer. Pour le comprendre, interrogeons-nous sur ce qu’est une redondance afin de pouvoir formaliser cela dans un langage relationnel.

Pour rappel, dans notre approche, une redondance entre les dépendances d’un nœud est un enchainement de dépendances « en diamant ». C’est-à-dire un nœud pour lequel les arcs sortants (correspondant aux concepts requis) convergent (avec potentiellement plusieurs niveaux intermédiaires) vers un même nœud.

La Figure A.5 illustre ces diamants et la nécessaire filtration à effectuer ; les chemins rouges sont les liens redondants et les chemins verts les liens à conserver.

La motivation de cette sélection est évidente lorsque l’on se penche sur la prochaine étape (l’inversion des dépendances).

Les nœuds étant examinés dans le sens inverse de leur déclaration d’exigence, les liens « directs » sautant plusieurs étapes (comme ceux de la Figure A.5) sont strictement inutiles. Nous ne pourrions les utiliser, sauf à traiter une norme avant que l’ensemble de ses concepts requis soit disponible. Par conséquent, traiter la

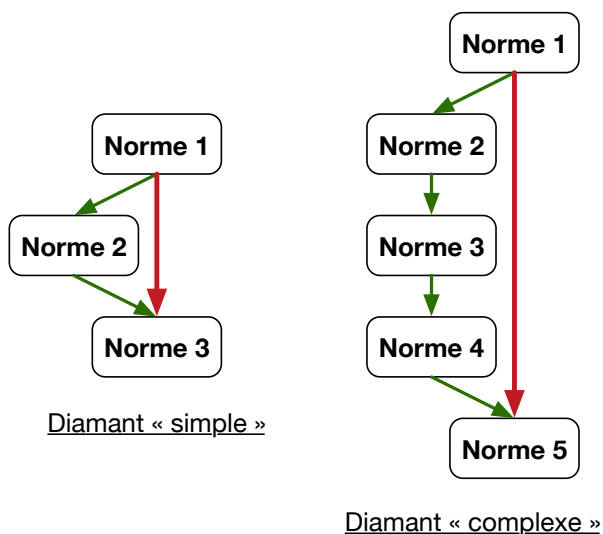


FIGURE A.5 – Graphe de normes : dépendances en diamant

norme 1 avant d’avoir traité la norme 2 pour le cas « simple » et les normes 2, 3 et 4 pour le cas « complexe » est inutile.

Ces relations peuvent donc aisément être supprimées pour simplifier les étapes suivantes.

La réalisation de ce travail avec un langage relationnel est cependant loin d’être évidente (et performante).

Algorithmiquement, cela consiste à prendre pour une norme donnée tous ses arcs deux à deux. Ensuite, pour chaque paire d’arcs, vérifier s’il existe un chemin entre la destination du premier arc (« A ») et la destination du second (« B »), puis dans le sens inverse (de « B » vers « A »). Dans le cas où un chemin existerait, le lien entre la norme et la destination du chemin doit être supprimé. Cet algorithme est illustré dans la Figure A.6.

Malheureusement, comme nous le verrons dans le paragraphe sur « l’extraction de chemins de longueur variable », c’est une opération complexe à réaliser et dont les performances sont relativement discutables.

Nous concluons donc ce paragraphe ainsi, en reportant l’appréciation définitive de la complexité de cette opération à celle sur l’extraction de chemins.

A.2.2 Inversion des dépendances

L’inversion des dépendances est une opération « théoriquement » simple consistant à simplement « retourner » les relations de notre graphe orienté, pour passer d’une vision « dépendances conceptuelles » à une formalisation « étapes conceptuelles ». C’est un point que nous avons largement abordé dans le Chapitre 6.

Dans le cas présent, cela consiste à trouver toutes les normes dépendantes de l’enregistrement « Bâtiment » (c.-à-d. toutes les normes requérant un concept

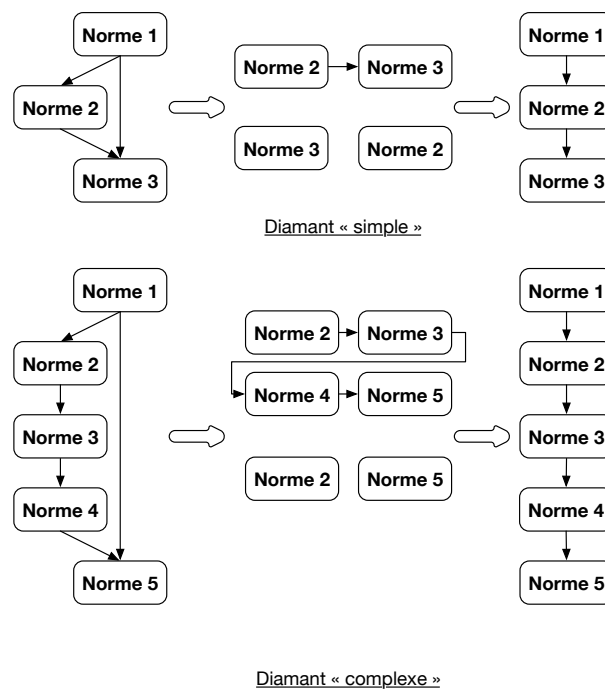


FIGURE A.6 – Graphe de normes : filtrage des diamants

fourni par « Bâtiment »). Techniquement, cela consiste à faire pour chaque norme une jointure entre la table « Concepts requis » et la table « Concepts fournis ».

Nous fournissons ici un exemple pour l'inversion des dépendances du nœud « Bâtiment » (version 1, révision 1) afin de déterminer toutes les normes exigeant le concept de « Bâtiment » :

```

SELECT Titre, Version, Révision
FROM "Concepts requis" CR
INNER JOIN
  (SELECT Nom
   FROM "Concepts fournis" CF
   WHERE CF.Titre == "Bâtiment"
         AND CF.Révision == 1
         AND CF.Version == 1) F
ON CR.Nom == F.Nom;

```

Listing A.1 – Exemple d'inversion de dépendance pour le nœud « Bâtiment »

Cette opération devra être répétée pour chaque nœud du graphe (dans le cas présent, chaque enregistrement de la table « Norme ») afin d'obtenir une version « inversée » du graphe original. Pour éviter de trop lourds traitements, nous pourrions bâtir une forme « d'index » synthétisant les résultats de ces requêtes.

En conclusion, l'opération est réalisable même si la jointure effectuée adopte un formalisme relativement éloigné du problème posé et si de nombreuses questions se posent sur une application à grande échelle.

A.2.3 Extraction de chemins de longueur variable

Nous allons maintenant traiter l'opération d'extraction de chemins de longueur variable dans le graphe de normes.

Nous avons deux principaux usages de cet « extracteur » :

- le premier est de pouvoir rapidement déterminer s'il existe un chemin entre deux nœuds afin de pouvoir procéder à l'opération de filtrage des redondances (cf. paragraphes précédents).
- le second est d'isoler le sous-graphe correspondant à un type particulier d'environnement normé.

Pour rappel, ce sous-graphe est composé de l'ensemble des nœuds et des arcs présents sur l'un des chemins entre le nœud « Bâtiment » et le nœud représentant le type d'environnement ciblé.

Pour le premier usage, nous allons utiliser le sous-graphe Figure A.7, sous-ensemble du graphe de la Figure A.2.

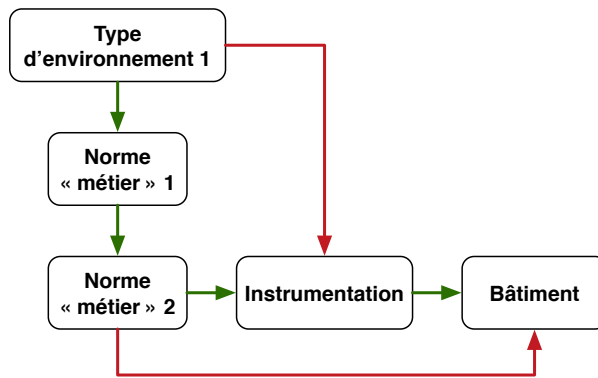


FIGURE A.7 – Graphe de normes : sous-graphe pour illustrer le calcul de chemins

Ce graphe comporte quelques redondances (en rouge dans la Figure A.7) qui nous permettront d'illustrer la détection des deux types de diamants évoqués précédemment (simple et complexe).

Pour le diamant « simple », celui issu de la « Norme Métier 2 », l'objectif est de vérifier s'il existe un chemin entre ces deux destinations (« Instrumentation » et « Bâtiment »). Les nœuds étant adjacents, les requêtes SQL restent relativement simples, bien que l'on commence à deviner la rapide progression en complexité des interrogations lorsque la longueur des chemins va grandir.

La requête SQL A.2 énumère les concepts requis par « Instrumentation », puis vérifie si « Bâtiment » fait partie des nœuds fournissant l'un de ces concepts. Lors de l'exécution de cette requête nous obtenons une entrée, « Bâtiment », prouvant ainsi l'existence d'un chemin dans le sens « I → B ».

La requête SQL A.3 réalise la même opération, mais dans le sens inverse. Lors de l'exécution de cette requête, nous obtenons une réponse vide, prouvant ainsi l'absence de chemin dans le sens « B → I ».

En conséquence, nous pouvons supprimer en toute sécurité la relation entre la « Norme Métier 2 » et « Bâtiment ».

```
SELECT Titre, Révision, Version
FROM (
  SELECT Titre, Révision, Version
  FROM "Concepts fournis" CF
  INNER JOIN
    (SELECT Nom
     FROM "Concepts requis" CR
     WHERE CR.Titre == "Instrumentation"
           AND CR.Révision == 1
           AND CR.Révision == 1) R
  ON CF.Nom == R.Nom)
WHERE Titre == "Bâtiment";
```

Listing A.2 – Détection d'un chemin direct entre « Instrumentation » et « Bâtiment »

```
SELECT Titre, Révision, Version
FROM (
  SELECT Titre, Révision, Version
  FROM "Concepts fournis" CF
  INNER JOIN
    (SELECT Nom
     FROM "Concepts requis" CR
     WHERE CR.Titre == "Bâtiment"
           AND CR.Révision == 1
           AND CR.Révision == 1) R
  ON CF.Nom == R.Nom)
WHERE Titre == "Instrumentation";
```

Listing A.3 – Détection d'un chemin direct entre « Bâtiment » et « Instrumentation »

Pour le diamant « complexe », celui issu du type d'environnement 1, nous devons identifier un chemin de longueur 2, les requêtes sont plus complexes...

```

-- On vérifie si Instrumentation fait parti des
-- fournisseurs de fournisseurs de Norme "Métier" 1
SELECT Titre, Révision, Version
FROM
  -- Jointure pour trouver les fournisseurs des concepts
  -- requis par les fournisseurs des concepts requis par
  -- Norme "Métier" 1
  (SELECT Titre, Révision, Version
   FROM "Concepts fournis" CF2
   INNER JOIN
     (SELECT Nom
      FROM "Concepts requis" CR2
      INNER JOIN
        -- Jointure pour trouver les fournisseurs des
        -- concepts requis par Norme "Métier" 1.
        (SELECT Titre, Révision, Version
         FROM "Concepts fournis" CF
         INNER JOIN
           (SELECT Nom
            FROM "Concepts requis" CR
            WHERE CR.Titre == 'Norme "Métier" 1'
              AND CR.Révision == 1
              AND CR.Révision == 1) R
          ON CF.Nom == R.Nom) R2
        ON CR2.Titre == R2.Titre
        AND CR2.Révision == R2.Révision
        AND CR2.Version == R2.Version) R3
     ON CF2.Nom == R3.Nom) F
WHERE Titre == "Instrumentation"
      AND Révision == "1"
      AND Version == "1"

```

Listing A.4 – Détection d'un chemin de longueur 2 entre « Norme Métier 1 » et « Instrumentation »

La requête SQL A.4 énumère les concepts requis par « Norme Métier 1 », puis en cherche les fournisseurs. Ensuite, elle répète cette opération avec ceux-ci : elle énumère leurs exigences et en cherche les fournisseurs. Enfin, elle vérifie si « Instrumentation » fait partie de cette dernière liste. *Dans le cas présent, c'est bien le cas, la requête renvoie bien une entrée et prouve ainsi l'existence d'un chemin dans le sens « NM 1 → I ».*

La requête SQL A.5 réalise la même opération, mais dans le sens inverse. *Lors de l'exécution de cette requête, nous obtenons une réponse vide, prouvant ainsi l'absence de chemin dans le sens « I → NM 1 ».*

En conclusion, la formulation de l'extraction de chemin de longueur fixe est relativement fastidieuse avec des jointures SQL classiques.

Mais qu'en est-il des chemins de longueur variable (inconnue lors de l'interrogation de la base de données), obligatoire pour traiter les cas non triviaux du premier cas d'utilisation (où la longueur du chemin est initialement inconnue) et répondre à notre deuxième cas d'utilisation ?

Une première réponse « naïve » est d'observer la régularité du formalisme

```

-- On vérifie si Instrumentation fait parti des
-- fournisseurs de fournisseurs de Instrumentation
SELECT Titre, Révision, Version
FROM
  -- Jointure pour trouver les fournisseurs des concepts
  -- requis par les fournisseurs des concepts requis par
  -- Instrumentation
  (SELECT Titre, Révision, Version
   FROM "Concepts fournis" CF2
   INNER JOIN
     (SELECT Nom
      FROM "Concepts requis" CR2
      INNER JOIN
        -- Jointure pour trouver les fournisseurs des
        -- concepts requis par Instrumentation.
        (SELECT Titre, Révision, Version
         FROM "Concepts fournis" CF
         INNER JOIN
           (SELECT Nom
            FROM "Concepts requis" CR
            WHERE CR.Titre == 'Instrumentation'
                  AND CR.Révision == 1
                  AND CR.Révision == 1) R
           ON CF.Nom == R.Nom) R2
         ON CR2.Titre == R2.Titre
          AND CR2.Révision == R2.Révision
          AND CR2.Version == R2.Version) R3
        ON CF2.Nom == R3.Nom) F
 WHERE Titre == 'Norme "Métier" 1'
       AND Révision == "1"
       AND Version == "1"

```

Listing A.5 – Détection d'un chemin de longueur 2 entre « Instrumentation » et « Norme Métier 1 »

des requêtes pour la solution de longueur fixe. Nous pourrions automatiser la génération de la requête et ainsi tester l'existence de chemins en énumérant les différentes longueurs possibles (ce qui requiert tout de même de connaître la longueur maximale autorisée).

Une meilleure solution serait de passer par des requêtes récursives, les seules qui permettraient de pouvoir traiter des longueurs arbitraires de manière efficace sans énumérer valeurs possibles ou créer des requêtes de taille rédhibitoire. Cela a longtemps été impossible à réaliser (car contraire même aux principes initiaux du langage SQL), jusqu'à la spécification « SQL 1999 » dite « SQL 3 » [Int99b] qui a enfin pallié à cette absence par l'introduction de requêtes dites Recursive Common Table Expressions (RECURSIVE CTE). Le calcul de chemin de longueur arbitraire devient alors possible et nous obtenons une réponse conforme.

Un exemple d'une telle requête est fourni dans le code source A.6, permettant de lister tous les nœuds issus du « Type d'environnement 1 » quelle que soit la longueur du chemin.

```

WITH RECURSIVE
  path_from_te1(Titre, Version, Révision) AS
  -- Valeur initiale (point de départ)
  (VALUES("Type d'environnement 1", "1", "1")
  -- Requête de récurrence
  UNION
  -- Identifie les normes fournissant les concepts requis
  -- par l'étape précédente
  SELECT CF.Titre, CF.Version, CF.Révision
  FROM "Concepts fournis" CF,
       "Concepts requis" CR,
       path_from_te1
  WHERE CF.Nom == CR.Nom
        AND CR.Titre == path_from_te1.Titre)
SELECT * FROM path_from_te1;

```

Listing A.6 – Détermination de l'ensemble des nœuds présent sur les chemins partant de « Type d'environnement 1 »

Bien que nous arrivions à un premier résultat, il reste néanmoins encore de nombreuses requêtes complémentaires à effectuer sur A.6 pour obtenir exactement l'objectif, mais le principe est posé. C'est tout de même suffisant pour se faire une idée définitive des avantages et des inconvénients du langage SQL pour notre usage.

A.3 Conclusion

En conclusion, l'approche purement relationnelle ne nous apparaît pas satisfaisante. Ce fonctionnement par table est loin de la réalité « intellectuelle » du problème, demandant ainsi une gymnastique mentale ainsi que des traitements importants pour arriver à traduire les concepts « orientés graphe » dans ce formalisme « relationnel ». Néanmoins, nous ne nions pas la possibilité de réaliser notre approche avec ce type de technologie, nous estimons seulement qu'il doit exister des solutions plus adaptées et dont tant le formalisme que le fonctionnement sont plus proches de notre problématique.



URCC : codes sources complémentaires

Dans ce chapitre, nous fournissons à titre d'exemple quelques éléments issus du travail sur l'URCC.

B.1 Application Java

Dans le cadre de notre illustration sur une URCC (Section 10.7 à la p. 177), nous proposons une application effectuant l'insertion des nœuds et l'interrogation du graphe (code source B.1). Cette classe affiche sur la console trois types d'informations : la liste ordonnée des nœuds, la liste ordonnée des normes et enfin celle des métamodèles et aspects.

La première liste est seulement utilisée pour le déverminage de notre implémentation, la seconde permet d'avoir facilement un moyen de comparaison avec nos études « théoriques » et enfin la dernière est celle devant être réellement utilisée pour la modélisation.

B.2 Script de traitement de la liste des fichiers

- Dans cette section, nous fournissons le script utilisé pour :
- récupérer les fichiers listés par le programme précédent ;
 - renommer les fichiers pour faciliter le travail de modélisation ;
 - les regrouper dans un même dossier de travail.

B.3 Métamodèle du bâti

Dans cette section, nous présentons notre modèle de bâti utilisé pour le travail sur l'URCC. Pour rappel, ce métamodèle comporte exclusivement les éléments liés au « bâtiment », il n'y a donc aucun appareillage, aucun équipement, etc.


```

package com.noc.phd.norms;

import com.noc.phd.norms.scenario.ModelingScenario;
import com.noc.phd.norms.scenario.URCCModeling;

import java.io.FileWriter;
import java.io.IOException;

/**
 * Simple application to test the REST API.
 *
 * @author Anthony GELIBERT
 * @version 2.0.0
 */
public final class ModelingStarter {
    /** Host. */
    public static final String SERVER_HOST = "localhost";
    /** Port. */
    public static final int SERVER_PORT = 7474;

    /** Main method without parameters. */
    public static void main(final String... args) {
        System.out.println("-----\n"+
            " Travaux doctoraux d'A. GELIBERT\n" +
            "   Application sur une URCC  \n" +
            "-----");
        ModelingScenario urcc = new URCCModeling(SERVER_HOST,
            SERVER_PORT);
        System.out.println("Préparation base de donnée (URCC)");
        urcc.prepare();
        System.out.println("Recherche des étapes (URCC)");
        System.out.printf("Liste des noeuds : %s\n",
            urcc.askConceptForIDs("URCC"));
        System.out.println("Liste des normes :");
        urcc.askConceptForNames("URCC")
            .forEach(s -> System.out.printf("\t- %s\n", s));

        try (FileWriter fw = new FileWriter("model-steps.txt")) {
            System.out.println("Etapes de modélisation :");
            urcc.askConceptForPayloads("URCC")
                .forEach(s -> {
                    System.out.printf("\t- %s\n", s);
                    try { fw.write(s + '\n'); }
                    catch (final IOException e) {
                        e.printStackTrace();
                    }
                });
        }
        catch (final IOException e) {e.printStackTrace();}
    }
}

```

Listing B.1 – Illustration : application d'insertion et d'interrogation du graphe

```
#!/bin/sh

# SET THIS MACRO TO CHANGE THE FILES LOCATION
export MODELING_FILES_ROOT="${HOME}/PhD/KerMeta"
# SET THIS MACRO TO CHANGE THE PRODUCED FOLDER
export MOD_STEP_FOLD="modeling-steps"
# Create a folder for the modeling files
mkdir -p ${MOD_STEP_FOLD}

# A simple counter to record the aspects order
number=0
# For each KM file
for step in `cat model-steps.txt`; do
  # Get the file
  cp `echo ${step} | envsubst` ${MOD_STEP_FOLD}
  # Rename the file
  ## Increment the counter
  number=$((number + 1))
  ## Get the old filename (without MODELING_FILES_ROOT macro)
  old=`basename ${step}`
  ## Create the real filename (#_file.kmt)
  filename="${number}_${old}"
  ## Rename the file
  mv ${MOD_STEP_FOLD}/${old},${filename}
done
```

Listing B.2 – Illustration : récupération des fichiers KerMeta

La Figure B.1 présente le métamodèle du bâtiment sous la forme d'un diagramme UML (plus visuel) tandis que le code source B.3 en présente la version KerMeta 2.

B.4 Aspect pour l'instrumentation

Dans cette section, nous présentons l'aspect « instrumentation » (code source B.4) permettant d'ajouter les différents capteurs et systèmes tiers présents dans l'environnement. Il modifie également la notion de Room pour la lier aux équipements.

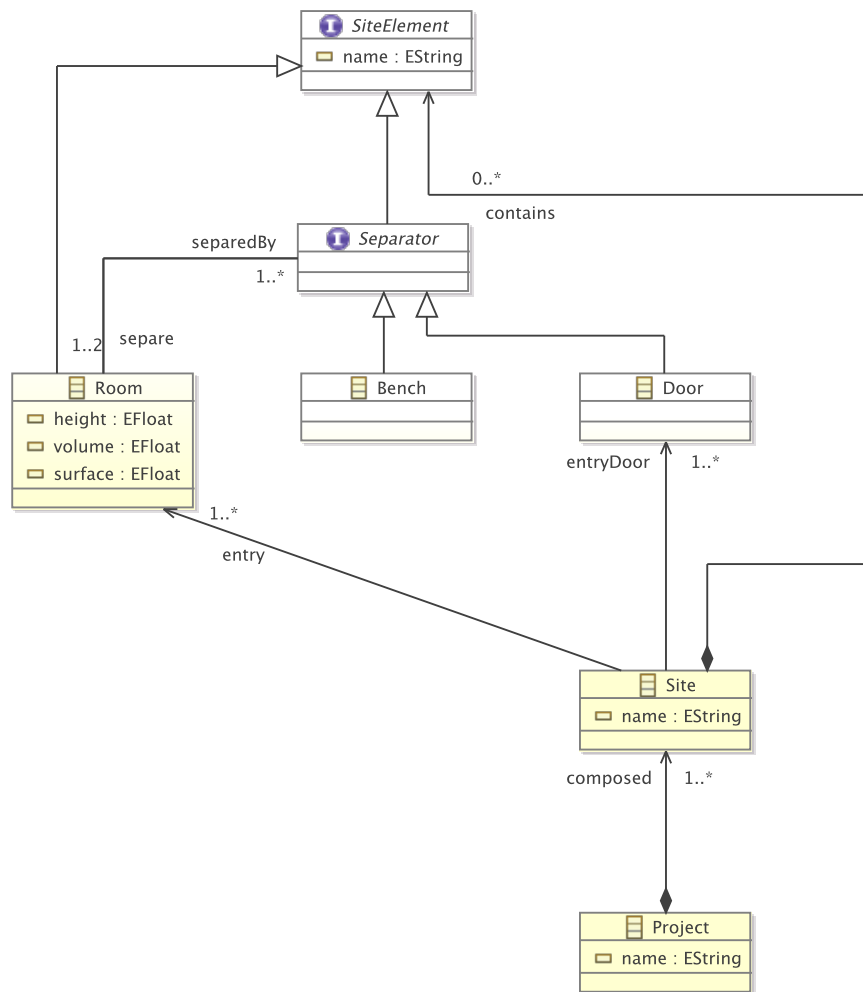


FIGURE B.1 – Illustration : métamodèle du bâti (diagramme de classe UML)

```

using kermeta::standard::*

package fr_ag_phd_kermeta {
  /** Le projet désigne l'environnement normé à représenter. */
  class Project {
    /** Le nom du projet. */
    attribute name : String
    /** Les différents sites composant le projet. */
    attribute composed : Site[1..*]
  }

  /** Un site est un groupe de pièce communicantes dans un
  environnement normé. */
  class Site {
    /** Le nom du site. */
    attribute name : String
    /** Les différents éléments composants le site. */
    attribute contains : SiteElement[0..*]
    /** Un site dispose d'une ou plusieurs pièces "d'entrée". */
    reference entry : Room[1..*]
    /** Un site dispose d'une ou plusieurs portes "d'entrée". */
    reference entryDoor : Door[1..*]
  }

  /** Type abstrait regroupant tous les éléments
  pouvant être présent dans un site. */
  abstract class SiteElement {
    /** Tous les éléments sont nommés. */
    attribute name : String
  }

  /** Les pièces d'un site. */
  class Room inherits SiteElement {
    /** La "hauteur sous plafond" de la pièce. */
    attribute height : Float
    /** Le volume de la pièce. */
    attribute volume : Float
    /** La surface de la pièce. */
    attribute surface : Float
    /** Chaque pièce est séparée des autres par
    un ensemble de séparateurs. */
    reference separedBy : Separator[1..*]#separe
  }

  /** Les éléments séparant les pièces. */
  abstract class Separator inherits SiteElement {
    /** Les deux pièces séparées par l'élément. */
    reference separe : Room[1..2]#separedBy
  }

  /** Les portes sont des séparateurs. */
  class Door inherits Separator {
  }

  /** Les bancs sont des séparateurs. */
  class Bench inherits Separator {
  }
}

```

Listing B.3 – Illustration : métamodèle du bâti

```
using kermeta::standard::*
using building::*

package fr_ag_phd_kermeta {
  abstract class Instrumentation inherits SiteElement {}
  abstract class Sensor inherits Instrumentation {}
  abstract class ExternalSystem inherits Instrumentation {}
  class TemperatureSensor inherits Sensor {}
  class PressureSensor inherits Sensor {}
  class HVAC inherits ExternalSystem {}
  class AccessControl inherits ExternalSystem {}

  aspect class Room {
    reference equipment : Equipment[0..*]#room
  }
  class RoomPressureControlled inherits Room {
    attribute pressure : Integer
    reference sensors : PressureSensor[1..*]
  }
  class Equipment inherits SiteElement {
    attribute surface : Integer
    reference room : Room[0..*]#equipment
  }
  class LaminarFlowHood inherits Equipment {
    attribute type : Integer
  }
  class EquipmentTemperatureControlled inherits Equipment {
    attribute temperature : Integer
    reference sensors : TemperatureSensor[1..*]
  }
}
```

Listing B.4 – Illustration : aspect « instrumentation »

Bibliographie

- [ACR⁺15] José Aguilar, Héctor Cancela, L. Rincón, G. Giraldo, R. Mazo, C. Salinesi, and D. Diaz. Method to identify corrections of defects on product line models. *Electronic Notes in Theoretical Computer Science*, 314 :61 – 81, 2015.
- [Age07] Agence Française de Sécurité Sanitaire des Produits de Santé. Bonnes Pratiques de Préparation. Technical report, Ministère du travail, de l’emploi et de la santé, Dec 2007.
- [Age13] Agence Française de NORmalisation. Établissements de santé - Zones à environnement maîtrisé - Exigences relatives à la maîtrise de la contamination aéroportée. Technical Report NF S 90-351, Agence Française de NORmalisation, Apr 2013.
- [AKM94] Edward Amoroso, W. E. Kleppinger, and David Majette. An engineering approach to secure system analysis, design, and integration. *AT&T Technical Journal*, 73(5) :40–51, 1994.
- [Alv04] Harald Tveit Alvestrand. RFC 3935 — A Mission Statement for the IETF. *RFC*, 2004.
- [Ama15] Amazon Web Services. *Amazon DynamoDB*. Amazon, Aug. 2015.
<http://aws.amazon.com/fr/dynamodb>.
- [AMR⁺14] Jose Aguilar, Gabriela Marin, L.F. Rincón, G.L. Giraldo, R. Mazo, and C. Salinesi. An ontological rule-based approach for analyzing dead and false optional features in feature models. *Electronic Notes in Theoretical Computer Science*, 302 :111 – 132, 2014.
- [Apa15a] Apache Software Foundation. *Apache Camel*, Jul 2015.
<http://camel.apache.org/index.html>.
- [Apa15b] Apache Software Foundation. *Apache Cassandra*, Aug. 2015.
<http://cassandra.apache.org>.
- [Apa15c] Apache Software Foundation. *Apache CouchDB*, Aug. 2015.
<http://couchdb.apache.org>.
- [Apa15d] Apache Software Foundation. *Apache Felix*, Jul 2015.
<http://felix.apache.org>.
- [Apa15e] Apache Software Foundation. *Apache HBase*, Aug. 2015.
<http://hbase.apache.org>.
- [Apa15f] Apache Software Foundation. *Apache Lucene*, Aug. 2015.
<https://lucene.apache.org>.

- [Apa15g] Apache Software Foundation. *Apache Solr*, Aug. 2015.
<http://lucene.apache.org/solr/>.
- [Ass14] Assurance Maladie. Améliorer la qualité du système de santé et maîtriser les dépenses : propositions de l'Assurance maladie pour 2014. Rapport au ministre chargé de la sécurité sociale et au Parlement sur l'évolution des charges et des produits de l'Assurance maladie au titre de 2014 (loi du 13 août 2004), Sécurité Sociale, 2014.
- [Ass15] Association TEKH!N. *Le travail en milieu confiné*, Jun 2015.
<http://www.tekhin.fr/fr/travail-confine>.
- [Aut15a] AutoDesk. *AutoCAD*, Aug 2015.
<http://www.autodesk.com/products/autocad>.
- [Aut15b] AutoDesk. *AutoCAD MEP*, Aug 2015.
<http://www.autodesk.com/products/autocad-mep>.
- [Aut15c] AutoDesk. *Revit*, Aug 2015.
<http://www.autodesk.fr/products/revit-family>.
- [Aut15d] AutoDesk. *Revit MEP*, Aug 2015.
<http://www.autodesk.com/store/revit-mep/benefits>.
- [AW05] Aybüke Aurum and Claes Wohlin. *Engineering and Managing Software Requirements*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [BABD09] Travis D Breaux, Annie I Antón, Kent Boucher, and Merlin Dorfman. It compliance : Aligning legal and product requirements. *IT professional*, 11(5) :54–58, 2009.
- [BAD08] Travis D. Breaux, Annie I. Antón, and Jon Doyle. Semantic parameterization : A process for modeling domain descriptions. *ACM Trans. Softw. Eng. Methodol.*, 18(2) :5 :1–5 :27, November 2008.
- [BBF06] Nelly Bencomo, Gordon S. Blair, and Robert B. France. Summary of the workshop models@run.time at models 2006. In *Models in Software Engineering, Workshops and Symposia at MoDELS 2006, Genoa, Italy, October 1-6, 2006, Reports and Revised Selected Papers*, pages 227–231, 2006.
- [BBF09] Gordon S. Blair, Nelly Bencomo, and Robert B. France. Models@run.time. *IEEE Computer*, 42(10) :22–27, 2009.
- [BBHL95] Barry Boehm, Prasanta Bose, Ellis Horowitz, and Ming June Lee. Software requirements negotiation and renegotiation aids : A theory-w based spiral approach. In *Software Engineering, 1995. ICSE 1995. 17th International Conference on*, pages 243–243. IEEE, 1995.
- [BCS⁺12] E. Braun, N. Cartwright, A. Shamsaei, S. A. Behnam, G. Richards, G. Mussbacher, M. Alhaj, and R. Tawhid. Drafting and modeling of regulations : Is it being done backwards ? In *Requirements Engineering and Law (RELaw), 2012 Fifth International Workshop on*, pages 1–6, Sept 2012.

- [BDC15] Stéphane Billaud, Nicolas Daclin, and Vincent Chapurlat. *Interoperability as a Key Concept for the Control and Evolution of the System of Systems (SoS)*, pages 53–63. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [BFCA14] Nelly Bencomo, Robert B. France, Betty H.C. Cheng, and Uwe Aßmann, editors. *Models@run.time – Foundations, Applications, and Roadmaps*, volume 8378 of *Programming and Software Engineering*. Springer International Publishing, 1re edition, 2014.
- [BIM16a] BIM CONFERENCE 2016. *BIM and LEAN construction — 4th conference*, May 2016.
<http://bimconference.me>.
- [BIM16b] BIM WORLD. *BIM WORLD 2017*, May 2016.
<http://www.bim-w.com>.
- [BJ80] Janis A Bubenko Jr. Information modeling in the context of system development. In *IFIP Congress*, volume 1980, pages 395–411, 1980.
- [BJG09] Jørgen Bang-Jensen and Gregory Z Gutin. *Digraphs : theory, algorithms and applications*. Springer Monographs in Mathematics. Springer-Verlag London, second edition, 2009.
- [BJV04] Jean Bézivin, F. Jouault, and P. Valduriez. On the need for Megamodels. In *Conference on Object-Oriented Programming Systems, Languages, and Applications*, 2004.
- [BL91] Valdis Berzins and Lucia Luqi. *Software Engineering with Abstractions*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1991.
- [BMP⁺14] P. Biron, M. H. Metzger, C. Pezet, C. Sebban, E. Barthuet, and T. Durand. An information retrieval system for computerized patient records in the context of a daily hospital practice : the example of the léon bérard cancer center (france). *Applied Clinical Informatics*, 5(1) :191–205, 2014.
- [BMPC98] R. W. Butler, S. P. Miller, J. N. Potts, and V. A. Carreno. A formal methods approach to the analysis of mode confusion. In *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, volume 1, pages C41/1, C41/8 vol.1–, Oct 1998.
- [Boe81] Barry W. Boehm. *Software Engineering Economics*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1981.
- [Bra97] Scott Bradner. RFC 2119 — Key words for use in RFCs to Indicate Requirement Levels. 1997.
- [BRJ96] Grady Booch, James Rumbaugh, and Ivar Jacobson. The unified modeling language. *Unix Review*, 14(13) :5, 1996.
- [Bro09] Paul Browne. *JBoss Drools business rules*. Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, 2009.

- [BT76] Thomas E Bell and T A Thayer. Software requirements : Are they really a problem ? *Proceedings of the 2nd international conference on Software engineering*, pages 62–68, 1976.
- [BVA06] Travis D. Breaux, Matthew W. Vail, and Annie I. Anton. Towards regulatory compliance : Extracting rights and obligations to align requirements with regulations. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE '06*, pages 46–55, Washington, DC, USA, 2006. IEEE Computer Society.
- [CB74] Donald D. Chamberlin and Raymond F. Boyce. Sequel : A structured english query language. In *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control*, pages 249–264. ACM, 1974.
- [CCL15] CCLIN et Arlin du Sud-Est. *Définition des Infections Nosocomiales*. Coordination de la lutte contre les infections nosocomiales & associées aux soins, Jul 2015.
http://cclin-sudest.chu-lyon.fr/CCLIN/definition_in.html.
- [CCP10] Benoît Combemale, Xavier Crégut, and Marc Pantel. Transformation de modèles — principes, standards et exemples. Support de cours, Laboratoire CNRS IRISA, Université Rennes 1, 2010.
- [CD15] Vincent Chapurlat and Nicolas Daclin. *System of Systems Architecting : A Behavioural and Properties Based Approach for SoS « -ilities » Modelling and Analysis*, pages 591–603. Springer International Publishing, Cham, 2015.
- [CDB15] Vincent Chapurlat, Nicolas Daclin, and Stéphane Billaud. *An Interface Pattern Model for Supporting Design of Natively Interoperable Systems*, pages 171–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [CDI15] CDISC. *CDISC : Standards & Implementations*, Jul 2015.
<http://www.cdisc.org/standards-and-implementations>.
- [Cen02] Centre de Coordination de la Lutte contre les Infections Nosocomiales du Sud-Ouest. Recommandations pour la manipulation des médicaments cytotoxiques dans les établissements de santé. Recommandations, C.CLIN Sud-Ouest, Bordeaux, Jul 2002.
- [Che76] Peter Pin-shan Chen. The entity-relationship model : Toward a unified view of data. *ACM Transactions on Database Systems*, 1 :9–36, 1976.
- [CL72] CL Chang and RCT Lee. Problem-solving methods in artificial intelligence (nils j. nilsson), 1972.
- [Cla95] Tom Clancy. The standish group report. *Chaos Report*, 20 :2008, 1995.
- [Cod70] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6) :377–387, 1970.
- [Com07] Comité Technique des Infections Nosocomiales et des Infections Liées aux Soins. Définition des infections associées aux soins.

- Rapport, Ministère de la santé, de la jeunesse et des sports, May 2007.
- [Com08] Benoît Combemale. *Approche de métamodélisation pour la simulation et la vérification de modèle — Application à l'ingénierie des procédés*. PhD thesis, Institut National Polytechnique de Toulouse, 2008.
- [CW14] E. Cambria and B. White. Jumping NLP Curves : A Review of Natural Language Processing Research. *IEEE Computational Intelligence Magazine*, 9(2) :48–57, May 2014.
- [DAC99] Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *Proceedings of the 21st International Conference on Software Engineering, ICSE '99*, pages 411–420, New York, NY, USA, 1999. ACM.
- [Dat71] Data Base Task Group (DBTG). CODASYL database model. Report, ACM, Apr 1971.
- [Dea15] Simon Deakin. Juridical ontology : The evolution of legal form. *Historical Social Research*, 40(1), 2015.
- [DeM79] Tom DeMarco. *Structured Analysis and System Specification*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1979.
- [DFF⁺10] Zoé Drey, Cyril Faucher, Franck Fleurey, Vincent Mahé, and Didier Vojtisek. Kermeta language. Reference manual, Triskell, Novembre 2010.
- [DFVL91] Anne Dardenne, Stephen Fickas, and Axel Van Lamsweerde. Goal-directed concept acquisition in requirements elicitation. In *Proceedings of the 6th international workshop on Software specification and design*, pages 14–21. IEEE Computer Society Press, 1991.
- [Dir01] Direction de l'Hospitalisation et de l'Organisation des Soins. Bonnes Pratiques de Pharmacie Hospitalière. Technical report, Ministère de l'Emploi et de la Solidarité – Ministère délégué à la santé, Jun 2001.
- [Div15] DiverSE — Diversity-Centric Software Engineering. *Kermeta*, Aug 2015.
<http://diverse.irisa.fr/software/>.
- [dlVPW13] Jose Luis de la Vara and Rajwinder Kaur Panesar-Walawege. *SafetyMet : A Metamodel for Safety Standards*, pages 69–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [dlVRA⁺16] Jose Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Kaur Panesar-Walawege, Ángel López, Idoya del Río, and Tim Kelly. Model-based specification of safety compliance needs for critical systems : A holistic generic metamodel. *Information & Software Technology*, 72 :16–30, 2016.
- [dRAdOCTQ07] Isa Mara da Rosa Alves, Rove Luiza de Oliveira Chishman, Paulo Miguel Torres, and Duarte Quaresma. The construction

- of a juridical ontology. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL '07*, pages 93–94, New York, NY, USA, 2007. ACM.
- [DS06] O. Djebbi and C. Salinesi. Criteria for comparing requirements variability modeling notations for product lines. In *Fourth International Workshop on Comparative Evaluation in Requirements Engineering (CERE'06 - RE'06 Workshop)*, pages 20–35, Sept 2006.
- [Dum14] Alexandre Dumont. *Classification des systèmes de stockage NoSQL*. Sogilis, Nov 2014.
<http://blog.sogilis.com/post/102957493551/classification-des-systèmes-de-stockage-nosql>.
- [DvL96] Robert Darimont and Axel van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. *SIGSOFT Softw. Eng. Notes*, 21(6) :179–190, October 1996.
- [DVL93] Anne Dardenne, Axel Van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1) :3–50, 1993.
- [Eas94] Steve Easterbrook. *Requirements Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1994.
- [Ecl16a] Eclipse Modeling Framework (EMF). *EMF Core (ECORE)*, Juin 2016.
<http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html>.
- [Ecl16b] Eclipse Modeling Project. *Eclipse Modeling Framework (EMF)*, Juin 2016.
<http://www.eclipse.org/modeling/emf>.
- [Ela15] Elastic. *ElasticSearch*, Aug. 2015.
<https://www.elastic.co>.
- [Eur07] European Parliament and Council. Directive 2007/47/EC of the european parliament and of the council of 5 september 2007 amending council directive 90/385/EEC on the approximation of the laws of the member states relating to active implantable medical devices, council directive 93/42/EEC concerning medical devices and directive 98/8/EC concerning the placing of biocidal products on the market. Directive 2007/47/CE, European Union, Sep 2007.
- [fCRO02] Office for Civil Rights (OCR). *Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy Rule*. U.S. Department of Health and Human Services (HHS), Aout 2002.
- [fCRO03] Office for Civil Rights (OCR). *Health Insurance Portability and Accountability Act of 1996 (HIPAA) Security Rule*. U.S. Department of Health and Human Services (HHS), Fev 2003.
- [fCRO09] Office for Civil Rights (OCR). *Health Information Technology for Economic and Clinical Health (HITECH) Act*. U.S. Department of Health and Human Services (HHS), Fev 2009.

- [fCRO13] Office for Civil Rights (OCR). *Modifications to the HIPAA Privacy, Security, Enforcement, and Breach Notification Rules under the Health Information Technology for Economic and Clinical Health Act and the Genetic Information Nondiscrimination Act ; Other Modifications to the HIPAA Rules*. U.S. Department of Health and Human Services (HHS), Jan 2013.
- [Fea87] Martin S. Feather. Language support for the specification and development of composite systems. *ACM Trans. Program. Lang. Syst.*, 9(2) :198–234, March 1987.
- [Fig10] Michaël Figuière. *NoSQL Europe : Tour d’horizon des bases de données NoSQL*. Xebia France, Apr 2010.
<http://blog.xebia.fr/2010/04/21>.
- [FMD15] FMDS industrie. *CEI 61508 & CEI 61511 — Systèmes et fonctions instrumentés de sécurité*, Aug 2015.
<http://www.suretedefonctionnement.fr/p/cei-61508.html>.
- [Fou13] François Fouquet. *Model@Runtime for continuous development of heterogeneous distributed adaptive systems*. Theses, Université Rennes 1, March 2013.
- [FP87] Anthony Finkelstein and Colin Potts. Building formal specifications using structured common sense. 1987.
- [Fra03] République Française. Loi n° 2003-699 du 30 juillet 2003 relative à la prévention des risques technologiques et naturels et à la réparation des dommages. Loi DEVX0200176L, Jul 2003.
- [Fra06] République Française. Code des marchés publics. Code, 2006.
- [Fra10] République Française. Loi n° 85-704 du 12 juillet 1985 relative à la maîtrise d’ouvrage publique et à ses rapports avec la maîtrise d’oeuvre privée. Loi, Dec 2010.
- [GB12] D. G. Gordon and T. D. Breaux. Reconciling multi-jurisdictional legal requirements : A case study in requirements water marking. In *2012 20th IEEE International Requirements Engineering Conference (RE)*, pages 91–100, Sept 2012.
- [GB13] David G Gordon and Travis D Breaux. A cross-domain empirical study and legal evaluation of the requirements water marking method. *Requirements Engineering*, 18(2) :147–173, 2013.
- [Gel14] Anthony Gelibert. Vers une aide pour la qualification énergétique des bâtiments intelligents. In *Journées Nationales des Communications dans les Transports Terrestres*, 2014.
- [GF95] Orlena Gotel and Anthony Finkelstein. Contribution structures [requirements artifacts]. In *Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on*, pages 100–107. IEEE, 1995.
- [GJGC12] Anthony Gelibert, Sébastien Jean, and Denis Genon-Catalot. Plateforme Ouverte de Supervision et de Traçabilité pour les

- Environnements Confinés. In *Actes de la conférence MajecSTIC 2012*, 2012.
- [GMB82] Sol J. Greenspan, John Mylopoulos, and Alex Borgida. Capturing more world knowledge in the requirements specification. In *Proceedings of the 6th International Conference on Software Engineering, ICSE '82*, pages 225–234, Los Alamitos, CA, USA, 1982. IEEE Computer Society Press.
- [GMUW09] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *DATABASE SYSTEMS : The Complete Book*. Pearson Education Hall, Upper Saddle River, NJ 07458, second edition, 2009.
- [Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199–220, June 1993.
- [Gru15] Sumit Gupta. *Neo4j Essentials — Leverage the power of Neo4j to design, implement, and deliver top-notch projects*. Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, essentials edition, Feb 2015.
- [Gup15] Sumit Gupta. *Building Web Applications with Python and Neo4j — Develop exciting real-world Python-based web applications with Neo4j using frameworks such as Flask, Py2neo, and Django*. Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, July 2015.
- [Hau14] Haute Autorité de Santé. *Scope Santé*, Nov 2014. <http://www.scopesante.fr/>.
- [HBF12] François Hennecart, Alain Bretto, and Alain Faisant. *Éléments de théorie des graphes*. IRIS. Springer, 2012.
- [HM15] Brad Hardin and Dave McCool. *BIM and Construction Management : Proven Tools, Methods, and Workflows*. Sybex, 2nd edition, Jun 2015.
- [Hoe94] T Hoet. Le concept de l'asepsie progressive et son impact sur le comportement dans le bloc opératoire. *Inter Bloc*, pages 24–27, 1994.
- [Hol16] Dominik Holzer. *The BIM Manager's Handbook : Guidance for Professionals in Architecture, Engineering and Construction*. John Wiley & Sons, Inc., Mar 2016.
- [HTC74] GF Hice, WS Turner, and LF Cashwell. System development methodology. amsterdam and new york, 1974.
- [IA10] Claus Ibsen and Jonathan Anstey. *Camel in action*. Manning Publications Co., 20 Baldwin Road, PO box 761, Shelter Island, NY 11964, 2010.
- [II12] Inspection Générale des Finances (IGF) and Inspection Générale des Affaires Sociales (IGAS). Propositions pour la maîtrise de l'ONDAM. Rapport, Jun 2012.
- [Ins07] Institut de Veille Sanitaire. Prévalence des infections nosocomiales, France, 2006. *Bulletin épidémiologique hebdomadaire*, 51-52 :429, 2007.

- [Ins12] Inspection Générale des Affaires Sociales (IGAS). L'Hôpital. Rapport, La documentation Française, Nov 2012.
- [Int99a] International Organization for Standardization. Cleanrooms and associated controlled environments — Part 1 : Classification of air cleanliness. ISO 14644-1, 1999.
- [Int99b] International Organization for Standardization. Information technology — Database languages – SQL. ISO 9075, Dec 1999.
- [Int03] International Organization for Standardization. Cleanrooms and associated controlled environments – Biocontamination control – Part 1 : General principles and methods. ISO 14698-1, 2003.
- [Int04] International Organization for Standardization. Cleanrooms and associated controlled environments — Part 5 : Operations. ISO 14644-5, 2004.
- [Int05] Health Level 7 International. Health Level 7 (HL7) Standards. Standards, 2005.
- [Int10] Intermittent Preventive Treatment in Infants. WHO Policy recommendation on Intermittent Preventive Treatment during infancy with sulphadoxine-pyrimethamine (SP-IPTi) for *Plasmodium falciparum* malaria control in Africa. Who policy recommendation, World Health Organization, Mar 2010.
- [Int11] International Organization for Standardization. Information technology — Database languages – SQL. ISO 9075, Dec 2011.
- [IR96] Milagros Ibanez and Helmut Rempp. European user survey analysis. *ESPITI Project Report*, 1996.
- [Jac95] Michael Jackson. *Software Requirements & Specifications : A Lexicon of Practice, Principles and Prejudices*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995.
- [JBC98] Matthias Jarke, Tung X. Bui, and M. John Carroll. Scenario management : An interdisciplinary approach. *Requirements Engineering*, 3(3) :155–173, 1998.
- [JBF11] Jean-Marc Jézéquel, Olivier Barais, and Franck Fleurey. Model driven language engineering with kermeta. In *Generative and Transformational Techniques in Software Engineering III*, pages 201–221. Springer, 2011.
- [JBo15] JBoss Community. *Drools - The Business Logic integration Platform*, Jul 2015.
<http://www.jboss.org/drools>.
- [JCB⁺15] Jean-Marc Jézéquel, Benoît Combemale, Olivier Barais, Martin Monperrus, and François Fouquet. Mashup of Meta-Languages and its Implementation in the Kermeta Language Workbench. *Software and Systems Modeling*, 14(2) :905–920, 2015.
- [Jor15] Gregory Jordan. *Practical Neo4j*. Apress, Jan 2015.

- [JS93] Andrew JI Jones and Marek Sergot. On the characterisation of law and computer systems : The normative systems perspective. *Deontic logic in computer science : normative system specification*, pages 275–307, 1993.
- [Kah62] A. B. Kahn. Topological sorting of large networks. *Commun. ACM*, 5(11) :558–562, November 1962.
- [Ken15] Karen Kensek. *Manuel BIM : Théorie et applications*. Blanche BTP. Eyrolles, Aug 2015.
- [Kid13] R. Kiddle. Introduction to Natural Language Processing. Undergraduate study in computing and related programmes, University of London International Programmes, 2013.
- [KMQ93] SJH Kent, TSE Maibaum, and William J Quirk. Formally specifying temporal constraints and error recovery. In *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 208–215. IEEE, 1993.
- [Koy92] Ron Koymans. *Specifying Message Passing and Time-Critical Systems with Temporal Logic*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1992.
- [KZB⁺07] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D. Breaux, Annie I. Antón, James R. Cordy, Luisa Mich, and John Mylopoulos. Extracting rights and obligations from regulations : Toward a tool-supported process. In *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering, ASE '07*, pages 429–432, New York, NY, USA, 2007. ACM.
- [KZB⁺08] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D. Breaux, Annie I. Antón, James R. Cordy, Luisa Mich, and John Mylopoulos. *Automating the Extraction of Rights and Obligations for Regulatory Compliance*, pages 154–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [KZC⁺09] Nadzeya Kiyavitskaya, Nicola Zeni, James R. Cordy, Luisa Mich, and John Mylopoulos. Cerno : Light-weight tool support for semantic annotation of textual documents. *Data & Knowledge Engineering*, 68(12) :1470 – 1492, 2009. Including Special Section : 21st {IEEE} International Symposium on Computer-Based Medical Systems (IEEE {CBMS} 2008) – Seven selected and extended papers on Biomedical Data Mining.
- [Lam00] Axel van Lamsweerde. Formal specification : A roadmap. In *Proceedings of the Conference on The Future of Software Engineering, ICSE '00*, pages 147–159, New York, NY, USA, 2000. ACM.
- [Lau13] Anne Laude. La judiciarisation de la santé. In *Questions de santé publique*, volume 20. Institut de la Recherche en Santé Publique (GIS-IRéSP), 2013.
- [LBG⁺16] F.-L. Li, A. Borgida, G. Guizzardi, J. Horkoff, L. Liu, and J. Mylopoulos. Desiree : a Refinement Calculus for Requirements Problems. *ArXiv e-prints*, May 2016.

- [Les16] Les éditions Tissot. *La fiche de poste*, may 2016.
http://www.editions-tissot.fr/actualite/droit-du-travail-article.aspx?secteur=PME&id_art=5612&titre=La+fiche+de+poste.
- [Lev95] Nancy G Leveson. *Safeware : system safety and computers*. Addison-Wesley Professional, 1995.
- [LL98] A. Van Lamsweerde and E. Letier. Integrating obstacles in goal-driven requirements engineering. In *Software Engineering, 1998. Proceedings of the 1998 International Conference on*, pages 53–62, Apr 1998.
- [LS16] Eric Lebègue and José Antonio Cuba Segura. *Conduire un projet de construction à l'aide du BIM*. Eyrolles, Jan 2016.
- [Lut93] R. R. Lutz. Analyzing software requirements errors in safety-critical, embedded systems. In *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 126–133, Jan 1993.
- [MA09] J. C. Maxwell and A. I. Anton. Developing production rule models to aid in acquiring requirements from legal texts. In *2009 17th IEEE International Requirements Engineering Conference*, pages 101–110, Aug 2009.
- [MAS11] J. C. Maxwell, A. I. Antón, and P. Swire. A legal cross-references taxonomy for identifying conflicting software requirements. In *2011 IEEE 19th International Requirements Engineering Conference*, pages 197–206, Aug 2011.
- [MAS12a] J. C. Maxwell, A. I. Anton, and P. Swire. Managing changing compliance requirements by predicting regulatory evolution. In *2012 20th IEEE International Requirements Engineering Conference (RE)*, pages 101–110, Sept 2012.
- [MAS⁺12b] Jeremy C. Maxwell, Annie I. Antón, Peter Swire, Maria Riaz, and Christopher M. McCraw. A legal cross-references taxonomy for reasoning about compliance requirements. *Requirements Engineering*, 17(2) :99–115, 2012.
- [MBJ⁺09] Brice Morin, Olivier Barais, Jean-Marc Jezequel, Franck Fleurey, and Arnor Solberg. Models@ run.time to support dynamic adaptation. *Computer*, 42(10) :44–51, 2009.
- [MCN92] John Mylopoulos, Lawrence Chung, and Brian Nixon. Representing and using nonfunctional requirements : A process-oriented approach. *Software Engineering, IEEE Transactions on*, 18(6) :483–497, 1992.
- [MDCV15] Sihem Mallek, Nicolas Daclin, Vincent Chapurlat, and Bruno Vallespir. Enabling model checking for collaborative process analysis : from bpmn to ‘network of timed automata’. *Enterprise Information Systems*, 9(3) :279–299, 2015.
- [Mem15] MemCached. *MemCached*, Aug. 2015.
<http://memcached.org>.
- [Mey85] B. Meyer. On formalism in specifications. *IEEE Softw.*, 2(1) :6–26, January 1985.

- [MFJ05] Pierre-Alain Muller, Franck Fleurey, and Jean-Marc Jézéquel. Weaving executability into object-oriented meta-languages. In *Model Driven Engineering Languages and Systems*, pages 264–278. Springer, 2005.
- [Min] Ministère des affaires sociales, de la santé et des droits des femmes. Code de la Santé Publique (CSP). Technical report, Ministère des affaires sociales, de la santé et des droits des femmes.
- [Min08] Ministère de l'économie, de l'industrie et de l'emploi. Décret n° 2008-1401 du 19 décembre 2008 relatif à l'accréditation et à l'évaluation de conformité pris en application de l'article 137 de la loi n° 2008-776 du 4 août 2008 de modernisation de l'économie. Décret ECEI0819386D, dec 2008.
- [Min09] Ministère de la santé, de la jeunesse et des sports. Infections nosocomiales : le dossier. Dossier, Jan 2009.
- [Min12] Ministère du travail, de l'emploi et de la santé. Programme Hôpital numérique — Guide des indicateurs des pré-requis et des domaines prioritaires du socle commun. Guide, Direction Générale de l'Offre de Soins (DGOS), Apr 2012.
- [Min14] Ministère des Affaires sociales, de la Santé et des Droits des femmes. Programme Hôpital numérique — La politique nationale relative aux systèmes d'information hospitaliers 2012 - 2017. Programme, Direction Générale de l'Offre de Soins (DGOS), May 2014.
- [MLHS⁺11] R. Mazo, R. E. Lopez-Herrejon, C. Salinesi, D. Diaz, and A. Egyed. Conformance checking with constraint logic programming : The case of feature models. In *2011 IEEE 35th Annual Computer Software and Applications Conference*, pages 456–465, July 2011.
- [MOA09] A. K. Massey, P. N. Otto, and A. I. Antón. Prioritizing legal requirements. In *Requirements Engineering and Law (RELAW), 2009 Second International Workshop on*, pages 27–32, Sept 2009.
- [MOHA10] Aaron K. Massey, Paul N. Otto, Lauren J. Hayward, and Annie I. Antón. Evaluating existing security and privacy requirements for legal compliance. *Requirements Engineering*, 15(1) :119–137, 2010.
- [Mol05] Eric Molinié. L'hôpital public en france : bilan et perspectives. Etude, Conseil Economique et Social, Jun 2005.
- [Mon15] MongoDB Inc. *MongoDB*, Aug. 2015. <https://www.mongodb.org>.
- [MP92] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.
- [MRT12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

- [MS92] Neil Maiden and Alistair Sutcliffe. Exploiting reusable specifications through analogy. *Commun. ACM*, 35(4) :55–64, April 1992.
- [MSOA11] A. K. Massey, B. Smith, P. N. Otto, and A. I. Antón. Assessing the accuracy of legal implementation readiness decisions. In *2011 IEEE 19th International Requirements Engineering Conference*, pages 207–216, Aug 2011.
- [Mum81] Enid Mumford. Participative systems design : Structure and method. *Systems, Objectives, Solutions*, 1(1) :5–19, 1981.
- [MVG06] Tom Mens and Pieter Van Gorp. A taxonomy of model transformation. *Electron. Notes Theor. Comput. Sci.*, 152 :125–142, March 2006.
- [MvL97] P. Massonet and A. van Lamsweerde. Analogical reuse of requirements frameworks. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 26–37, Jan 1997.
- [Nem15a] Nemetschek. *Allplan Architecture*, Aug 2015. <https://www.allplan.com/fr/software/architecture/allplan-architecture.html>.
- [Nem15b] Nemetschek. *Allplan Engineering*, Aug 2015. <http://www.allplan.com/en/software/engineering/allplan-engineering.html>.
- [Neo15a] Neo4J. *Cypher Query Language : introduction*. Neo4J, Dec. 2015. <http://neo4j.com/docs/stable/cypher-introduction.html/>.
- [Neo15b] Neo4J. *Neo4J*, Aug 2015. <http://neo4j.com>.
- [Nil14] Nils J Nilsson. *Principles of artificial intelligence*. Morgan Kaufmann, 2014.
- [Nix93] B. A. Nixon. Dealing with performance requirements during the development of information systems. In *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 42–49, Jan 1993.
- [NKF94] Bashar Nuseibeh, Jeff Kramer, and Anthony Finkelstein. A framework for expressing the relationships between multiple views in requirements specification. *Software Engineering, IEEE Transactions on*, 20(10) :760–773, 1994.
- [NMS89] Celso Niskier, T. S. E. Maibaum, and Daniel Schwabe. A pluralistic knowledge-based approach to software specification. In *Proceedings of the 2Nd European Software Engineering Conference, ESEC '89*, pages 411–423, London, UK, UK, 1989. Springer-Verlag.
- [Noc15] Nocosium SAS. *Nocosium SAS*, Jul 2015. <http://www.nocosium.fr/>.
- [Obj15a] Object Management Group. Meta Object Facility (MOF) 2.0 Query/View/Transformation (QVT). Specification version 1.2, Object Management Group, fev 2015.

- [Obj15b] Object Management Group. Meta Object Facility (MOF) Core. Specification version 2.5, Object Management Group, juin 2015.
- [Obs15] Observatoire National Interministériel de la Sécurité Routière. La sécurité routière en France — Bilan de l'accidentalité de l'année 2013. Bilan annuel, La documentation Française, 2015.
- [Opp04] Andrew Opper. *Databases demystified*. McGraw-Hill, Inc., 2004.
- [Per82] Chaïm Perelman. Juridical ontology and sources of law. *N. Ky. L. Rev.*, 10 :387, 1982.
- [PHG⁺12] D. Proux, C. Hagège, Q. Gicquel, I. Kergourlay, S. Pereira, G. Rondeau, S. Darmoni, F. Segond, and M.-H. Metzger. Aladin : développement d'un outil sémantique d'analyse des documents textuels médicaux pour la détection d'infections associées aux soins. *{IRBM}*, 33(2) :137 – 142, 2012. Numéro spécial {ANR} {TECSANTechnologie} pour la santé et l'autonomie.
- [PM95] David Lorge Parnas and Jan Madey. Functional documents for computer systems. *Science of Computer Programming*, 25 :41–61, 1995.
- [Pot95] Colin Potts. Using schematic scenarios to understand user needs. In *Proceedings of the 1st Conference on Designing Interactive Systems : Processes, Practices, Methods, & Techniques*, DIS '95, pages 247–256, New York, NY, USA, 1995. ACM.
- [Pro16a] Projet National *MINⁿD*. *EDUBIM 2016*, May 2016. <http://www.minnd.fr/edubim/edubim-2016/>.
- [Pro16b] Projet National *MINⁿD*. *Projet National Modélisation des Informations INteropérables pour les INfrastructures Durables*, May 2016. <http://www.minnd.fr>.
- [PTA94] Colin Potts, Kenji Takahashi, and Annie I. Antón. Inquiry-based requirements analysis. *IEEE Softw.*, 11(2) :21–32, March 1994.
- [PW12] R.K. Panesar-Walawege. *Using Model-Driven Engineering to Support the Certification of Safety-Critical Systems*. PhD thesis, University of Oslo, Jul 2012.
- [PWSB11] R.K. Panesar-Walawege, M. Sabetzadeh, and L. Briand. A Model-Driven Engineering Approach to Support the Verification of Compliance to Safety Standards. In *IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE)*, pages 30–39, 2011.
- [RBP⁺91] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William E. Lorensen, et al. *Object-oriented modeling and design*, volume 199. Prentice-hall Englewood Cliffs, NJ, 1991.
- [Red12] Redis DB. *Youporn.com is now a 100% Redis site*, Feb 2012. <https://groups.google.com/forum/?fromgroups=!topic/redis-db/d4QcWV0p-YM>.
- [Red15] Redis DB. *ReDis*, Aug. 2015. <http://redis.io>.

- [Ret15] RethinkDB. *RethinkDB*, Aug. 2015. <http://rethinkdb.com>.
- [RG92] Kenneth S. Rubin and Adele Goldberg. Object behavior analysis. *Commun. ACM*, 35(9) :48–62, September 1992.
- [RJB04] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
- [RKB13] E Rescorla, S Kent, and R Barnes. RFC 6919 — Further Key Words for Use in RFCs to Indicate Requirement Levels. 2013.
- [RMTV11] M. Raatikainen, T. Männistö, T. Tommila, and J. Valkonen. Challenges of requirements engineering – a case study in nuclear energy domain. In *2011 IEEE 19th International Requirements Engineering Conference*, pages 253–258, Aug 2011.
- [Rob89] W. N. Robinson. Integrating multiple specifications using domain goals. *SIGSOFT Softw. Eng. Notes*, 14(3) :219–226, April 1989.
- [Ros77] Douglas T. Ross. Structured analysis (sa) : A language for communicating ideas. *IEEE Trans. Software Eng.*, 3 :16–34, 1977.
- [Rou08] Frédéric Rouvillois. *La société au risque de la judiciarisation*. Colloques & Débats. LexisNexis, 1re edition, Apr 2008.
- [Rou09] Guillaume Rousset. Judiciarisation et juridicisation de la santé : entre mythe et réalité. In *Carnets de Bord*, volume 16, pages 23–28, 2009.
- [RS77] D.T. Ross and K.E. Schoman. Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, 3(1) :6–15, 1977.
- [RS05] Colette Rolland and Camille Salinesi. *Modeling Goals and Reasoning with Them*, pages 189 – 217. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [RW91] H. B. Reubenstein and R. C. Waters. The requirements apprentice : automated assistance for requirements acquisition. *IEEE Transactions on Software Engineering*, 17(3) :226–240, Mar 1991.
- [RWE15] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. O’Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, second edition, Jun 2015.
- [San13] Nicolas Sannier. *INCREMENT an Hybrid approach for modeling and analyzing regulatory safety requirements in the large*. Theses, Université Rennes 1, December 2013.
- [SB14] Nicolas Sannier and Benoit Baudry. INCREMENT : A mixed MDE-IR approach for regulatory requirements modeling and analysis. In *Requirements Engineering : Foundation for Software Quality - 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings*, pages 135–151, 2014.

- [SBMP08] Dave Steinberg, Frank Budinsky, Ed Merks, and Marcelo Paterostro. *EMF : eclipse modeling framework*. Pearson Education, 2008.
- [SJI⁺12] Alberto Siena, Ivan Jureta, Silvia Ingolfo, Angelo Susi, Anna Perini, and John Mylopoulos. *Capturing Variability of Law with N6mos 2*, pages 383–396. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [SMD⁺12] P. Sawyer, R. Mazo, D. Diaz, C. Salinesi, and D. Hughes. Using constraint programming to manage configurations in self-adaptive systems. *Computer*, 45(10) :56–63, Oct 2012.
- [SR07] Matt Stephens and Doug Rosenberg. *Use Case Driven Object Modeling with UML — Theory and Practice*. Apress, Berkeley, jan 2007.
- [SSK⁺06] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, and M. Wimmer. A survey on aspect-oriented modeling approaches. Technical report, 2006.
- [Sut97] A. Sutcliffe. A technique combination approach to requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 65–74, Jan 1997.
- [TA12] Anne Terraz-Ayela. Utilisation d’un entrep6t de donn6es cliniques pour l’6laboration et l’6valuation d’un syst6me d6cisionnel : application 6 la pr6vention des infections du site op6ratoire. Technical report, INSERM UMR S 872 Equipe 22, 2012.
- [Tea13] CDISC XML Technologies Team. CDISC Operational Data Model (ODM). Standards, CDISC, Dec 2013.
- [The12] The 451 Group. *Updated database landscape graphic*, Nov 2012.
https://blogs.the451group.com/information_management/2012/11/02/updated-database-landscape-graphic.
- [The15a] The Apache DB Project. *Apache Derby*. Apache Software Foundation, Aug 2015.
<https://db.apache.org/derby/>.
- [The15b] The HyperSQL Development Group. *HyperSQL*, Aug 2015.
<http://hsqldb.org>.
- [Tri15] Triskell. *Kermeta 2*, Aug 2015. <http://www.kermeta.org>.
- [Tri16] Triskell. *Liste de fichiers du projet KerMeta2*. INRIA, Juin 2016.
http://gforge.inria.fr/frs/?group_id=4312.
- [URMT11] E. Uusitalo, M. Raatikainen, T. M6nnist6, and T. Tommila. Structured natural language requirements in nuclear energy domain towards improving regulatory guidelines. In *Requirements Engineering and Law (RELAW), 2011 Fourth International Workshop on*, pages 67–73, Aug 2011.

- [U.S97] U.S. Food and Drug Administration. Part 11 Electronic Records; Electronic Signatures. *CFR - Code of Federal Regulations Title 21*, Mar 1997.
- [Val10] Antonio Vallecillo. On the combination of domain specific modeling languages. In *European Conference on Modelling Foundations and Applications*, pages 305–320. Springer, 2010.
- [Val16] Clément Valente. *Définition du BIM*. Méthodes BTP, Apr. 2016.
<https://bimbtp.com/pourquoi-le-bim/definition-du-bim/>.
- [Vas06] Alain Vasselle. Rapport sur la politique de lutte contre les infections nosocomiales. Santé publique, Office Parlementaire d’Evaluation des Politiques de Santé, Jun 2006.
- [vL00] Axel van Lamsweerde. Requirements engineering in the year 00 : a research perspective. In *Proceedings of the 22nd international conference on Software engineering, ICSE ’00*, pages 5–19, New York, NY, USA, 2000. ACM.
- [vL08] Axel van Lamsweerde. Requirements engineering : From craft to discipline. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, SIGSOFT ’08/FSE-16*, pages 238–249, New York, NY, USA, 2008. ACM.
- [VL+09] Axel Van Lamsweerde et al. *Requirements engineering : from system goals to UML models to software specifications*. John Wiley & Sons, Inc., 2009.
- [vLDDD91] Axel van Lamsweerde, A Dardenne, B Delcourt, and F Dubisy. The KAOS Project : Knowledge Acquisition in Automated Specification of Software. In *AAAI Spring Symp. Series, Track : Design of Composite Systems*, pages 59–62. Stanford University, Mar 1991.
- [VLDL98] Axel Van Lamsweerde, Robert Darimont, and Emmanuel Letier. Managing conflicts in goal-driven requirements engineering. *Software Engineering, IEEE Transactions on*, 24(11) :908–926, 1998.
- [vLW98] A. van Lamsweerde and L. Willemet. Inferring declarative requirements specifications from operational scenarios. *IEEE Transactions on Software Engineering*, 24(12) :1089–1114, Dec 1998.
- [VWA+15] Aleksa Vukotic, Nicki Watt, Tareq Abedrabbo, Dominic Fox, and Jonas Partner. *Neo4j in Action*. Manning Publications Company, 20 Baldwin Road, PO box 761, Shelter Island, NY 11964, 2015.
- [Wik15a] Wikipédia. *Building Information Modeling*, Aug 2015.
http://www.wikiwand.com/fr/Building_Information_Modeling.
- [Wik15b] Wikipédia. *DO-178*, Aug 2015.
<http://www.wikiwand.com/fr/DO-178>.

- [Wik15c] Wikipédia. *List of CAx companies*, Aug 2015. http://www.wikiwand.com/en/List_of_CAx_companies.
- [WLK⁺14] Keith F. Woeltje, Michael Y. Lin, Michael Klompas, Marc Oliver Wright, Gianna Zuccotti, and William E. Trick. Data requirements for electronic surveillance of healthcare-associated infections. *Infection Control and Hospital Epidemiology*, 35(9) :1083–1091, 2014.
- [WMJC12] Manuel Wimmer, Salvador Martínez, Frédéric Jouault, and Jordi Cabot. A catalogue of refactorings for model-to-model transformations. *Journal of Object Technology*, 11(2) :2 :1–40, August 2012.
- [WPJH98] K. Weidenhaupt, K. Pohl, M. Jarke, and P. Haumer. Scenarios in system development : current practice. *IEEE Software*, 15(2) :34–45, Mar 1998.
- [WS80] Anthony I Wasserman and Susan K Stinson. A specification method for interactive medical information systems. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 3 :1471–1478, 11 1980.
- [YA10] J. D. Young and A. I. Anton. A method for identifying software requirements based on policy commitments. In *2010 18th IEEE International Requirements Engineering Conference*, pages 47–56, Sept 2010.
- [YGMM11] Eric Yu, Paolo Giorgini, Neil Maiden, and John Mylopoulos. *Social Modeling for Requirements Engineering*. The MIT Press, 2011.
- [YM94] Eric Yu and John Mylopoulos. Understanding « why » in software process modelling, analysis, and design. In *Proceedings of the 16th International Conference on Software Engineering, ICSE '94*, pages 159–168, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
- [Yu97] Eric Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 226–235, Jan 1997.
- [Yue87] Kaizhi Yue. What does it mean to say that a specification is complete? In *Proc. IWSSD-4, Fourth International Workshop on Software Specification and Design*, 1987.
- [Zav97] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29(4) :315–321, December 1997.
- [ZBL07] Gregory Zoughbi, Lionel Briand, and Yvan Labiche. *A UML Profile for Developing Airworthiness-Compliant (RTCA DO-178B), Safety-Critical Software*, pages 574–588. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [ZBL11] Gregory Zoughbi, Lionel Briand, and Yvan Labiche. Modeling safety and airworthiness (rtca do-178b) information : Conceptual model and uml profile. *Softw. Syst. Model.*, 10(3) :337–367, July 2011.

- [ZC05] Didar Zowghi and Chad Coulin. *Requirements Elicitation : A Survey of Techniques, Approaches, and Tools*. Springer Berlin Heidelberg, 2005.
- [Zei06] Kurt Zeilenga. RFC 4510 — Lightweight Directory Access Protocol (LDAP) : Technical specification road map. *RFC*, 2006.
- [ZJ93] Pamela Zave and Michael Jackson. Conjunction as composition. *ACM Trans. Softw. Eng. Methodol.*, 2(4) :379–411, October 1993.

Table des figures

1.1	Illustration de la « normalisation » de notre société par l'AFNOR	8
2.1	Objectif industriel	20
2.2	Objectif industriel « réel »	24
3.1	Positionnement thématique de ces travaux	31
3.2	Analyse « verticale » d'un problème	35
3.3	Réalisation traditionnelle d'un environnement	35
4.1	Rappel de la problématique réelle	55
5.1	Version générique de la méthodologie proposée	64
5.2	Analyse du cadre normatif : principe général	65
5.3	Déroulement de l'analyse du cadre normatif	66
5.4	Aide à la qualification : principe général	68
5.5	Déroulement de l'aide à la qualification	69
5.6	Qualification de l'usage : principe général	72
5.7	Déroulement de la qualification de l'usage	72
5.8	Version détaillée de la méthodologie proposée	75
6.1	Carte euristique pour l'analyse d'une norme	79
6.2	Les métadonnées d'une norme sous forme graphique	81
6.3	Forme « générale » des contraintes pesant sur un environnement normé	86
6.4	Forme « générique » d'un graphe de normes	87
6.5	Illustration de la modélisation réticulaire de normes	89
6.6	Illustration d'une modélisation « filtrée » de ses redondances	90
6.7	Illustration d'une modélisation « inversée »	91
6.8	Exemple : Type d'environnement 1 — graphe normatif	92
6.9	Exemple : Type d'environnement 1 — graphe normatif filtré	92
6.10	Exemple : Type d'environnement 2 — graphe normatif	92
6.11	Exemple : Type d'environnement 2 — graphe normatif filtré	93
6.12	Illustration d'un parcours de graph en largeur	94
6.13	Exemple : Type d'environnement 2 — graphe normatif trié en largeur	95
6.14	Illustration d'un parcours de graphe en profondeur	96
6.15	Exemple : Type d'environnement 2 — graphe normatif trié en profondeur	96

7.1	Structure générale d'un « questionnaire »	101
7.2	Fonctionnement « schématique » de l'outil d'aide à la qualification	105
7.3	Fonctionnement précis et technique de l'outil d'aide à la qualification	107
8.1	Panorama des SGBD	117
8.2	Panorama des SGBD « NoSQL »	119
8.3	CYPHER : exemple de graphe	126
8.4	CYPHER : exemple de requête sur un graphe (1/2)	126
8.5	CYPHER : exemple de requête sur un graphe (2/2)	127
8.6	Graphe de normes : représentation visuelle	134
8.7	Graphe de normes : liste des nœuds utilisés pour l'illustration de la construction d'un graphe	135
8.8	Graphe de normes : construction (étape 1)	136
8.9	Graphe de normes : construction (étape 2)	136
8.10	Graphe de normes : construction (étape 3)	137
8.11	Graphe de normes : construction (étape 4)	138
8.12	Graphe de normes : construction (étape 5)	138
8.13	Graphe de normes : exemple de graphe complet	139
9.1	Architecture à mettre en œuvre pour l'outil d'aide à la qualification	144
9.2	Rappel du fonctionnement attendu de l'outil d'aide à la qualification	145
9.3	MDA : Un processus en Y dirigé par les modèles	146
9.4	Classes de transformation en IDM	147
9.5	Modélisation d'un environnement normé par tissages successifs .	148
9.6	Positionnement de KerMeta	149
9.7	Modèle UML simpliste d'une librairie	152
9.8	Architecture de notre mise en œuvre de l'outil d'aide à la qualifi- cation d'un projet d'environnement normé	154
10.1	Structure de l'environnement normé	159
10.2	Hierarchie normative	161
10.3	Illustration : graphe après le CSP	164
10.4	Illustration : graphe après le Décret n° 2000-1316	164
10.5	Illustration : graphe après les BPPH	165
10.6	Illustration : graphe après les BPP	165
10.7	Illustration : graphe après les recommandations cytotoxiques . .	166
10.8	Illustration : graphe après l'ISO 14644-1	167
10.9	Illustration : graphe après la NF S 90-351	167
10.10	Illustration : graphe complet dans Neo4J	169
10.11	Illustration : carte euristique simplifiée d'une URCC	170
10.12	Illustration : sous-graphe concernant les Pharmacies à Usage Intérieur	172
10.13	Illustration : graphe concernant les URCC	172
10.14	Illustration : nettoyage du graphe	173
10.15	Illustration : inversion du graphe	174
10.16	Illustration : préparation au tri topologique du graphe	174
10.17	Illustration : tri topologique du graphe	175
10.18	Illustration : tri topologique alternatif du graphe	175
10.19	Illustration : architecture informatique employée	176
10.20	Illustration : algorithme de modélisation	179

10.21	Illustration : modèle initial du bâti	180
10.22	Illustration : modèle du bâti enrichi de l'instrumentation	180
10.23	Recommandations Cytotoxiques : descriptif des locaux en fonction du type d'équipement pour la reconstitution des cytotoxiques . . .	182
10.24	NF S 90-351 : points de vigilance	183
10.25	NF S 90-351 : classes de risque en fonction du type d'activité . . .	184
10.26	NF S 90-351 : valeurs guide de performance au repos	185
10.27	NF S 90-351 : cascades de pression	186
10.28	NF S 90-351 : schéma de principe cascade de pressions avec un local à risque recevant une activité à risque	186
11.1	Mise en œuvre « complète » de notre approche	191
11.2	Plateforme de traçabilité : une approche orientée services	194
11.3	Mise en œuvre « simplifiée » de notre approche, sans la phase de qualification d'un projet	197
12.1	Méthodologie proposée	204
13.1	Illustration de l'impossibilité d'appliquer un tri topologique sur un graphe cyclique	208
13.2	Illustration d'un conflit de contraintes normatives	210
A.1	Modélisation réticulaire des normes : représentation UML	225
A.2	Graphe de normes pour l'illustration des différents types de repré- sentation	226
A.3	Graphe de normes : représentation relationnelle (1/2)	227
A.4	Graphe de normes : représentation relationnelle (2/2)	228
A.5	Graphe de normes : dépendances en diamant	230
A.6	Graphe de normes : filtrage des diamants	231
A.7	Graphe de normes : sous-graphe pour illustrer le calcul de chemins	232
B.1	Illustration : métamodèle du bâti (diagramme de classe UML) . . .	240

Table des codes sources

8.1	Requête CYPHER identifiant des cycles de longueur arbitraire dans un graphe quelconque	125
8.2	API pour la représentation d'une norme	128
8.3	API pour la supervision générale d'un serveur NEO4J	129
8.4	API pour les stratégies de parcours du graphe de norme	130
8.5	API pour un scénario de modélisation	131
8.6	Classe abstraite pour un scénario de modélisation	132
8.7	Fichier CSS employé pour notre représentation du graphe de normes	133
9.1	Exemple de l'utilisation des concepts « objet » basiques dans KerMeta	151
9.2	Exemple de vérifications basiques dans KerMeta en utilisant des exceptions	151
9.3	Représentation du modèle UML d'une librairie	152
A.1	Exemple d'inversion de dépendance pour le nœud « Bâtiment » .	231
A.2	Détection d'un chemin direct entre « Instrumentation » et « Bâtiment »	233
A.3	Détection d'un chemin direct entre « Bâtiment » et « Instrumentation »	233
A.4	Détection d'un chemin de longueur 2 entre « Norme Métier 1 » et « Instrumentation »	234
A.5	Détection d'un chemin de longueur 2 entre « Instrumentation » et « Norme Métier 1 »	235
A.6	Détermination de l'ensemble des nœuds présent sur les chemins partant de « Type d'environnement 1 »	236
B.1	Illustration : application d'insertion et d'interrogation du graphe	238
B.2	Illustration : récupération des fichiers KerMeta	239
B.3	Illustration : métamodèle du bâti	241
B.4	Illustration : aspect « instrumentation »	242

Liste des acronymes

- A -

AFNOR	Association Française de Normalisation.
ANRT	Association Nationale de la Recherche et de la Technologie.
AOM	Aspect-Oriented Modeling.
AOP	Aspect-Oriented Programming.
ARS	Agences Régionales de Santé.

- B -

BBC	Bâtiment Basse-Consommation.
BIM	Building Information Model.
BPF	Bonnes Pratiques de Fabrication.
BPP	Bonnes Pratiques de Préparation.
BPPH	Bonnes Pratiques de Pharmacie Hospitalière.

- C -

CAAD	Computer-Aided Architectural Design.
CAD	Computer-Aided Design.
CAO	Conception Assistée par Ordinateur.
CEE	Communauté Economique Européenne.
CIFRE	Conventions Industrielles de Formation par la REcherche.
CIM	Computational Independant Model.
CLIN	Comité de Lutte contre les Infections Nosocomiales.
COFRAC	COmité FRançais d'ACcréditation.
CSP	Code de la Santé Publique.
CTE	Common Table Expressions.
CTIN	Comité Technique national des Infections Nosocomiales.

- D -

DAG	Directed Acyclic Graph.
DAO	Dessin Assisté par Ordinateur.
DBMS	DataBase Management System.
DGOS	Direction Générale de l'Offre de Soins.

DGS	Direction Générale de la Santé.
DH	Direction des Hôpitaux.
DMP	Dossier Médical Personnel.
DSL	Domain-Specific Language.
DSML	Domain-Specific Modeling Language.

- E -

E/E/PE	Electrique/Electronique/Electronique Programmable.
ECA	Event-Condition-Action.
EMF	Eclipse Modeling Framework.
EMP	Eclipse Modeling Project.
EMOF	Essential MOF.
ETP	Equivalent Temps-Plein.

- G -

GORE	Goal-Oriented Requirements Engineering.
-------------	---

- H -

HEPA	High Efficiency Particulate Air.
HHS	Health and Human Services.
HIPAA	Health Insurance Privacy Accessibility Act of 1996.
HTTP	Hypertext Transfer Protocol.

- I -

IAO	Ingénierie Assistée par Ordinateur.
IAS	Infections Associées aux Soins.
IDM	Ingénierie Dirigée par les Modèles.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
IN	Infections Nosocomiales.
InVS	Institut de Veille Sanitaire.
ISO	International Organization for Standardization.

- J -

JVM	Java Virtual Machine.
------------	-----------------------

- K -

KAOS	Knowledge Acquisition in autOdated Specification.
-------------	---

- L -

LCIS Laboratoire de Conception et d'Intégration de Systèmes.

- M -

MDA Model-Driven Architecture.
MDE Model-Driven Engineering.
MIB Modèle d'Information du Bâtiment.
MNB Maquette Numérique du Bâtiment.
MOF MetaObject Facility.

- N -

NFR Non-Functional Requirements.

- O -

OCL Object Constraint Language.
OCP Open/Closed Principle.
OCR Office for Civil Rights.
OMG Object Management Group, Inc.
OMS Organisation Mondiale de la Santé.
OSGi Open Service Gateway initiative.

- P -

PDM Platform Dependant Model.
PIM Platform Independant Model.
POA Programmation Orientée Aspect.
POO Programmation Orientée Objet.
PSM Platform Specific Model.
PUI Pharmacie à Usage Intérieur.

- Q -

QVT Query/View/Transformation.

- R -

RCTE Recursive Common Table Expressions.
RE Requirements Engineering.

- S -

SADT	Structured Analysis and Design Technic.
SEQUEL	Structured English QUery Language.
SGBD	Système de Gestion de Base de Données.
SGBDR	Système de Gestion de Base de Données Relationnelles.
SI	Système d'Information.
SIH	Système d'Information Hospitalier.
SIS	Software-Intensive System.
SNS	Stratégie Nationale de Santé.
SOA	Service Oriented Architecture.
SQL	Structured Query Language.

- T -

TALN	Traitement Automatique du Langage Naturel.
TPE	Très Petite Entreprise.

- U -

UML	Unified Modeling Language.
URCC	Unité de Reconstitution Centralisée des Cytotoxiques.

- X -

XML	eXtensible Markup Language.
------------	-----------------------------

Résumé

Les environnements confinés, tels que les blocs chirurgicaux ou les salles blanches, hébergent des processus complexes auxquels sont associés de nombreux risques. Leur conception, leur réalisation et leur exploitation sont complexes, de par les très nombreuses normes les encadrant.

La qualification de ces « environnements normés », afin d'en garantir la qualité de conception, requiert une expertise fine du métier et souffre du manque d'outil en permettant l'automatisation.

Partant de ce constat, nous proposons une approche unifiée visant à faciliter la qualification des environnements normés. Celle-ci s'appuie sur une représentation du contexte normatif sous la forme d'un graphe unique, ainsi que sur une modélisation de l'environnement et son objet final par étapes successives permettant une vérification incrémentale de même que la production d'informations nécessaires à la traçabilité lors de l'exploitation.

Cette démarche, illustrée au travers du domaine des environnements confinés médicaux, est générique et peut s'appliquer à l'ensemble des environnements normés.

Abstract

Industrial clean rooms or operating rooms are critical places often hosting dangerous or complex processes. Their design, building and use are thus difficult and constrained by a large amount of standards and rules. Qualifying these environments, in order to ensure their quality, consequently requires a high level of expertise and lacks assisting tools.

This leads us to propose a unified approach aiming at easing the qualification process of standardized environments. It relies on a graph-based representation of the set of standards and rules that apply to a specific case, as well as on step-by-step modelling of the whole target environment. The verification process is then eased as it becomes incremental. During each stage, relevant information can also be gathered in order to ensure environment traceability during its use.

This approach, applied to medical environments for validation purposes, remains generic and can be applied to any kind of standardized environment.