

Vers une CAO guidée par la connaissance des fabricants

Mael Hilléreau

▶ **To cite this version:**

Mael Hilléreau. Vers une CAO guidée par la connaissance des fabricants. Autre [cs.OH]. Université de Reims - Champagne Ardenne, 2008. Français. NNT : . tel-00411785

HAL Id: tel-00411785
https://theses.hal.science/tel-00411785

Submitted on 28 Aug 2009

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.

Institut de Formation Technique Supérieur
Charleville-Mézières

École Doctorale Sciences, Technologie, Santé

Vers une CAO guidée par la connaissance des fabricants

THÈSE

présentée et soutenue publiquement le 5 septembre 2008

pour l'obtention du

Doctorat de l'Université de Reims Champagne-Ardenne

(spécialité informatique)

par

Mael HILLÉREAU

Composition du jury

Président :	M^me^ Parisa Ghodous	Professeure à l'université de Lyon I
Rapporteurs :	M. Dominique Deneux	Professeur à l'université de Valenciennes
	M^me^ Parisa Ghodous	
Examinateurs :	M. Yvon Gardan	Professeur à l'université de Reims
	M. Chrisitian Minich	Maître de Conférences à l'université de Metz
	M^me^ Estelle Perrin	Maître de Conférences à l'université de Reims, HDR

Centre de Recherche en Sciences et Technologies de l'Information et de la Communication

Document mis en page avec la classe LaTeX thloria et rédigé avec le logiciel LyX.

Remerciements

Les travaux présentés dans ce manuscrit ont été menés au sein de l'équipe CMCAO (Concepts et Modèles pour la Conception Assistée par Ordinateur) de l'IFTS (Institut de Formation Technique Supérieur) de Chalreville-Mézières, sous la direction du professeur Yvon Gardan, et ont été partiellement financés par le Conseil Général des Ardennes.

Je remercie en premier lieu mon directeur de thèse, M. Yvon Gardan, pour m'avoir fait confiance en m'accueillant au sein de l'équipe CMCAO, et pour m'avoir fait profiter d'un encadrement de qualité et de l'ambiance cordiale qu'il sait faire régner dans son équipe.

Je remercie également de tout cœur Estelle Perrin, M^{e} de Conférences à l'Université de Reims, habilitée à diriger les recherches, qui a co-encadré cette thèse. J'ai apprécié sa grande disponibilité, sa patience, ses conseils avisés et son savoir-être, tout particulièrement utiles dans les moments difficiles.

Je remercie respectueusement M^{me} Parisa Ghodous, professeure au LIRIS (Laboratoire d'InfoRmatique en Images et Systèmes d'information) de l'Université de Lyon I, et M. Dominique Deneux, professeur au LAMIH (Laboratoire d'Automatique, de Mécanique, et d'Informatique industrielles et Humaines) de l'Université de Valenciennes, qui ont accepté d'endosser le rôle de rapporteur. Je les remercie pour le temps qu'ils ont consacré à la lecture de ce manuscrit, et l'attention qu'ils ont portée à mon travail.

Je témoigne également ma reconnaissance à M^{me} Parisa Ghodous, M. Dominique Deneux, M. Christian Minich, M^{e} de conférences à l'Université de Metz, M^{me} Estelle Perrin et M. Yvon Gardan, pour avoir accepté de participer au jury de soutenance, et de juger mon travail.

Je souhaite bien sûr remercier tous les membres de l'équipe CMCAO et des autres équipes que j'ai pu croiser à l'IFTS : Estelle, Christian, Fred H., Fred D., Nicolas, Reim, Laurent, Étienne, Julien, Oana, Erwan, Stéphanie, Erasmus, Mohand, Philippe, Fadila, Slah, Walid, Siham, Benoît. Merci à tous.

Merci à ma famille qui m'a toujours soutenu, en particulier mes parents qui m'ont toujours fait confiance. Merci à وفاء pour son soutien et sa patience, et à Yann pour ses sourires et sa contribution à la réduction de mon temps de sommeil à certains moments clefs !

Enfin, merci à tout le personnel de l'IFTS que j'ai cotoyé dans le cadre des cours ou de l'organisation de la licence professionnelle TIC.

À mes parents. . .

Table des matières

Table des figures **xiii**

Liste des tableaux **xvii**

Liste des algorithmes **xix**

Introduction **1**

Chapitre 1 Conception intégrée de produits **5**

1.1 Introduction 6

1.2 Le processus de conception 7

1.2.1 Du besoin aux fonctions 9

1.2.1.1 Notion de fonction 10

1.2.1.2 Analyse fonctionnelle 11

1.2.1.3 Méthodes d'analyse et de décomposition fonctionnelles 12

1.2.2 Des fonctions à la forme 14

1.2.2.1 Point de vue techno-mécaniste 15

1.2.2.2 Point de vue mathématique 17

1.2.2.3 Point de vue psycho-cognitif 20

1.2.2.4 Point de vue informatique 20

1.2.3 Articulation avec l'analyse d'ingénierie 26

1.2.4 Conclusion 28

1.3 Modélisation d'un produit 29

1.3.1 Modèle géométrique et modèle produit 29

1.3.1.1 Modèle géométrique 29

1.3.1.2 Modèle produit 34

1.3.2 Techniques de modélisation 37

1.3.2.1 Modélisation constructive 37

1.3.2.2 Modélisation modificative . . . 37
1.3.2.3 Modélisation descriptive . . . 38
1.3.3 Conclusion . . . 39
1.4 Vers le modèle global . . . 40
1.4.1 Gestion du cycle de vie d'un produit (PLM) . . . 42
1.4.2 Ingénierie simultanée . . . 44
1.4.3 Travail collaboratif . . . 46
1.4.4 Gestion des connaissances . . . 50
1.4.5 Conclusion . . . 52
1.5 Synthèse . . . 53

Chapitre 2 Conception guidée par la fabricabilité : état de l'art 55
2.1 Introduction . . . 56
2.2 Caractérisation de la fabricabilité . . . 59
2.2.1 Indicateurs de fabricabilité . . . 59
2.2.1.1 Degré de détermination . . . 60
2.2.1.2 Indicateurs technologiques . . . 60
2.2.1.3 Indicateurs économiques . . . 62
2.2.2 Référentiel . . . 63
2.2.2.1 Caractère absolu ou relatif . . . 64
2.2.2.2 Caractère concret ou abstrait . . . 64
2.2.2.3 Autres caractéristiques . . . 65
2.2.3 Règles de fabricabilité . . . 65
2.2.4 Conclusion . . . 67
2.3 Méthodes de prise en compte de la fabricabilité . . . 67
2.3.1 Introduction . . . 67
2.3.2 Aspect décisionnel . . . 69
2.3.3 Politique de calcul . . . 72
2.3.3.1 Type d'approche . . . 72
2.3.3.2 Niveau d'automatisation . . . 72
2.3.4 Démarche de réutilisation des connaissances . . . 73
2.3.4.1 Approches génératives . . . 73
2.3.4.2 Approches par variation . . . 76
2.3.4.3 Autres méthodes . . . 78
2.3.4.4 Bilan . . . 79
2.3.5 Intégration et interaction avec les principes de modélisation . . . 81
2.3.5.1 Utilisation des caractéristiques . . . 81

2.3.5.2 Utilisation de contraintes 84
2.3.5.3 Approches multi-vues 86
2.4 Synthèse 89

Chapitre 3 Un outil de modélisation guidée par le savoir-faire des fabricants 93
3.1 Introduction 95
3.2 Concepts fondateurs 96
3.2.1 Le projet DIJA 96
3.2.1.1 Présentation 96
3.2.1.2 Architecture 99
3.2.1.3 Processus de conception 104
3.2.2 Analyse et positionnement 106
3.3 Proposition pour la représentation des connaissances liées à la fabrication dans DIJA 110
3.3.1 Modélisation des règles de fabricabilité 110
3.3.2 Notion d'état pour la conception 114
3.3.3 Interactivité 116
3.3.4 Synthèse 117
3.4 Gestion de la fabricabilité 118
3.4.1 Processus d'évaluation 119
3.4.2 Prise en compte statique des règles 119
3.4.2.1 Graphe des grandeurs de fabrication 120
3.4.2.2 Graphes des états 122
3.4.2.3 Synthèse 126
3.4.3 Prise en compte dynamique des règles 126
3.4.3.1 Outils de conception DIJA 126
3.4.3.2 Transformations structurelles 127
3.4.3.3 Propagation 134
3.4.3.4 Insertion de règles spécifiques 139
3.4.4 Synthèse 141
3.5 Bilan 141

Chapitre 4 Mise en œuvre informatique 143
4.1 Introduction 144
4.2 Contexte d'implémentation 144
4.2.1 Environnement DIJA 144
4.2.2 Limites de la maquette 145

4.3 Étude de l'implémentation . . . 146
4.3.1 Architecture logicielle . . . 146
4.3.2 Éléments relatifs à la représentation des connaissances . . . 148
4.3.2.1 Modèle de conception . . . 149
4.3.2.2 Expressions . . . 151
4.3.2.3 Graphes . . . 153
4.3.2.4 Enregistrement des règles . . . 156
4.3.3 Répercussion des règles de fabricabilité . . . 157
4.3.4 Mécanismes mis en œuvre pour l'évolution des connaissances . . . 158
4.4 Présentation de la maquette . . . 160
4.4.1 Interface . . . 160
4.4.2 Étude de cas . . . 161
4.4.2.1 Exemple illustratif . . . 162
4.4.2.2 Phase de capitalisation . . . 162
4.4.2.3 Phase de réutilisation . . . 164
4.5 Bilan . . . 169

Conclusion **173**

Annexes **177**

Annexe A Représentation par les frontières et topologie **177**
A.1 Relation d'Euler-Poincaré . . . 178
A.2 Opérateurs d'Euler . . . 178

Annexe B Exemple de conception utilisant l'approche synthétique **181**

Annexe C Algorithmes utilisés pour la propagation **185**
C.1 Propagation . . . 186
C.2 Division . . . 186
C.3 Fusion . . . 194

Annexe D Calcul de distances dans l'espace **201**
D.1 Mesures devant être effectuées . . . 202
D.2 Calculs sous-jacents . . . 203
D.2.1 Distance entre deux points dans l'espace . . . 203
D.2.2 Distance entre deux segments coplanaires . . . 203
D.2.3 Distance entre deux contours polygonaux . . . 204
D.2.3.1 Distance entre deux droites . . . 205

D.2.3.2 Distance entre deux segments non coplanaires 207
D.2.3.3 Distance entre deux contours polygonaux 207
D.2.4 Coûts 208
D.3 Remarques 208

Annexe E Format de représentation XML **211**
E.1 Schémas XML 212
E.1.1 Représentation des GGF 212
E.1.2 Représentation des GE 216
E.1.3 Représentation des expressions 218
E.2 Exemple d'utilisation 220
E.2.1 Exemple de GGF 220
E.2.2 Exemple de GE 222

Sigles et acronymes **225**

Bibliographie **231**

Index des références bibliographiques **255**

Table des figures

1.1 Processus et modèle de conception. 7
1.2 Phases en amont et en aval du processus de conception. 8
1.3 Caractérisation, hiérarchisation et raffinement des fonctions. 11
1.4 Exemple de diagramme pieuvre. 13
1.5 Actigramme de la méthode SADT. 14
1.6 Relations entre fonctions, comportements et structure 16
1.7 Approches systémiques du processus de conception. 18
1.8 Les quatre domaines de la conception axiomatique 19
1.9 Architecture d'un système à base de connaissances. 21
1.10 Cycle du raisonnement par cas 22
1.11 Cycle de conception sous contraintes. 24
1.12 Architecture d'un système multi-agents. 25
1.13 Simulation en fonderie : le logiciel *Experto* 27
1.14 Paradigme général en modélisation géométrique. 30
1.15 Exemple de modèle par les frontières. 32
1.16 Exemple d'arbre de construction. 32
1.17 Courbe et surface à pôles. 33
1.18 Exemples de caractéristiques. 34
1.19 Différence entre problèmes paramétriques et variationnels 38
1.20 Diagramme de Pareto. 41
1.21 Gestion du cycle de vie du produit (PLM). 42
1.22 Principe du gel progressif des activités 45
1.23 L'outil de visualisation et d'annotation *Autovue* 49

1.24 Valeur de l'information 52

2.1 Boucles de re-conception. 57
2.2 Exemples de pièces non fabricables par usinage 62
2.3 Logiciel DFMA®. 68
2.4 Mise au point de fonctions d'estimation 78
2.5 Interprétations multiples en termes de caractéristiques d'usinage 81
2.6 Extraction de caractéristiques. 82
2.7 Reconnaissance incrémentale de caractéristiques de fabrication. 83

3.1 Approche synthétique de la conception. 97
3.2 Opérations de transformation élémentaires de DIJA. 98
3.3 Éléments de dialogue. 99
3.4 Décomposition modulaire de DIJA. 100
3.5 Les différents niveaux d'abstraction de DIJA. 102
3.6 Architecture du module de connaissances. 105
3.7 Système de remplissage en fonderie et *workflow* associé 106
3.8 Approche pour la mise à jour des règles de fabricabilité. 107
3.9 Approche modulaire pour la gestion de la fabricabilité. 109
3.10 Mémorisation des interactions. 117
3.11 Répercussions après modification au niveau dialogue. 118
3.12 Processus d'évaluation de règles instanciées. 120
3.13 Exemples de valeurs de fabrication. 121
3.14 Exemple de GGF (une seule règle est présentée). 122
3.15 Exemples de graphes des états relatifs au GGF de la figure 3.14. 124
3.16 Processus de changement d'état. 125
3.17 Application d'un outil DIJA. 127
3.18 Cas général d'application d'un outil DIJA (la partie grisée E^s n'est pas incluse dans E_f). 128
3.19 Exemple de déformation et GGF associés 129
3.20 Fonction de correspondance f_c (seuls les fibres et un contour sont indiqués). . . . 129
3.21 Exemple transformation de GGF où une fonction de référencement est nécessaire ($D_1^{\min}$ et $D_2^{\min}$ sont les diamètres minimaux). 130

3.22 Ensembles impliqués dans les deux fonctions. 132
3.23 Exemple de fonction de référencement (représentation partielle). 133
3.24 Adaptation lors de la division d'un ED (e_a se divise en e_a et e'_a). Le raisonnement est similaire pour la division d'un ED en plus de deux parties. 136
3.25 L'adaptation consiste en une conservation si les règles sont disjointes. 137
3.26 Différents types de jonctions possibles avant fusion. 138
3.27 Graphe après fusion (adaptation) de e_a et e_b sur la figure 3.26d (en considérant qu'e_a doit être le fruit de la fusion). 138

4.1 Architecture générale de l'application (les flèches correspondent à la notion d'importation). 147
4.2 Modèle de représentation par les frontières (B-Rep étendu) 150
4.3 Diagramme de classes UML représentant les expressions logiques et arithmétiques (les noms de classes sont abrégés pour une question de clarté). 152
4.4 Diagramme de classes UML représentant l'interface `Evaluable`. 153
4.5 Diagramme de classes UML représentant les graphes des grandeurs. 154
4.6 Diagramme de classes UML représentant les graphes des états. 156
4.7 Diagramme d'activité UML décrivant le processus de changement d'état. 158
4.8 Diagramme de classes UML représentant les fonctions f_c et f_r. 159
4.9 Interface graphique de l'application. 161
4.10 Exemple de pièce de révolution. 163
4.11 Exemple de scénario de conception. 165
4.11 Exemple de scénario de conception (suite). 166
4.11 Exemple de scénario de conception (suite). 167
4.11 Exemple de scénario de conception (suite). 168
4.12 Graphes des grandeurs avant et après division de la fibre de révolution (cf. figure 4.11). 170

A.1 Construction d'un tétraèdre à l'aide des opérateurs d'Euler 179

B.1 Conception d'un pichet basée sur la méthode synthétique. 182
B.2 Différentes solutions pour le bec verseur. 184

C.1 Valeur de fabrication déjà divisée. 191

C.2 Grandeur de fabrication déjà divisée. 192
C.3 Règle déjà « divisée » (instance clonée). 193
C.4 VF, GF et règle non encore divisées. 193
C.5 Une fusion peut encore survenir au niveau des VF. 198
C.6 Une fusion peut encore survenir au niveau des GF. 198
C.7 Une fusion peut encore survenir au niveau des instances de la règle. 199
C.8 Dernière fusion possible d'un ensemble homogène. 199

D.1 Différents calculs de distance. 202
D.2 Distance entre un point et un segment dans l'espace. 203
D.3 Distance entre deux segments dans l'espace. 205
D.4 Domaine de validité des paramètres s et t. 207

E.1 Diagramme de paquets UML représentant les dépendances entre les trois fichiers de description *XML Schema*. 213
E.2 Représentation graphique des graphes de l'exemple. 221

Liste des tableaux

1.1 Entités manipulées en modélisation sous-contraintes 40
1.2 Matrice de Robert Johansen 47
1.3 Collaboration et éloignement physique 47

2.1 Connaissances et démarches de réutilisation 80
2.2 Résumé des avantages et inconvénients des démarches de réutilisation. 80

3.1 Types de propagation possibles pour chaque opérateur. 134

A.1 Principaux opérateurs d'Euler. 179

C.1 Variables et procédures définies globalement. 189

Liste des algorithmes

3.1 Traitement d'une instruction i par le superviseur. 103
3.2 Propagation. 140
C.1 Propagation (reformulation). 187
C.2 Division d'un ED. 188
C.3 Division d'une VF. 190
C.4 Division d'une GF. 191
C.5 Division d'une règle. 192
C.6 Marquage de la fusion pour un ensemble d'ED. 195
C.7 Marquage de la fusion pour un ensemble de VF. 196
C.8 Marquage de la fusion pour un ensemble de GF. 197
C.9 Marquage de la fusion pour un ensemble de règles. 197
C.10 Fusion des ensembles marqués. 200
D.1 Prise en compte du cas où x est en dehors de l'espace paramétrique des deux segments pour le calcul de la distance. 208

Introduction

Durant ces dernières décennies, l'essor des réseaux informatiques et notamment de l'Internet a indiscutablement révolutionné bon nombre de métiers, en plus d'en voir créé de nouveaux. Ceci tient au fait que les réseaux ont ouvert la voie à de nouvelles façons d'appréhender le travail, notamment à travers une multiplication et une rapidité accrue des échanges. L'avènement du numérique a peu à peu permis le développement de nouveaux modes d'interaction : multiplication des portails collaboratifs, démocratisation de la visioconférence, déploiement des technologies sans fil... Mais le travail en réseau n'en est qu'à ses balbutiements et d'importantes mutations concernant les méthodes et les outils de travail en groupe sont encore à prévoir.

Dans les faits, les outils de travail collaboratif ne sont encore que peu utilisés dans les grandes entreprises, et pour ainsi dire pas du tout dans la plupart des petites structures. Deux facteurs expliquent cela. Le premier est socio-culturel, et concerne les réticences qu'ont les employés à modifier leurs modes de travail. Le second est technologique et englobe des problèmes liés à la mise en place de ces nouveaux outils, en particulier vis-à-vis de l'interopérabilité.

Les progrès liés aux nouvelles technologies de l'information et de la communication (NTIC) s'accompagnent d'une tendance à passer du mode d'organisation séquentiel à un mode d'organisation parallèle et distribué. Ceci s'explique en premier lieu par la recherche d'une meilleure compétitivité (réduction des délais de mise sur le marché, amélioration de la qualité). Mais c'est la réunion de groupes d'entreprises autour de projets communs et le développement du concept d'entreprise étendue, qui en a été le véritable catalyseur.

Ces évolutions organisationnelles valent pour un large panel de métiers et donc en particulier pour ceux de l'industrie des produits manufacturés à laquelle nous nous intéressons notamment dans ce manuscrit. Si dans ce domaine, les méthodes de travail ont déjà été radicalement transformées par l'expansion et la généralisation de l'informatique dans les années

soixante-dix et quatre-vingts (apparition des outils de CAO [1], de FAO [2] et de simulation), elles vont l'être à nouveau avec la naissance du travail de groupe en réseau. Beaucoup de concepts relevant des aspects distribués et simultanés du développement d'un produit font en effet l'objet de recherches accrues depuis une dizaine d'années. Tenons-en pour témoins les travaux récents gravitant autour du vaste concept qu'est le PLM *(Product Lifecycle Management)* et qui, au-delà d'être un acronyme à la mode, remet en question l'organisation même des entreprises en touchant à l'ensemble de la chaîne numérique. Le PLM consiste en une gestion informatique optimale de toutes les étapes de la vie du produit, depuis l'expression du besoin, jusqu'au retrait du marché. La mise en place du PLM est une tâche complexe qui requiert une interopérabilité entre tous les acteurs d'une entreprise. Les outils informatiques doivent donc être présents et opérationnels à chaque étape, et favoriser les échanges entre acteurs intervenant dans une même phase de développement, mais aussi et surtout entre acteurs de compétences différentes.

Les travaux de thèse présentés dans ce manuscrit portent plus spécifiquement sur la question de l'interopérabilité des connaissances des fabricants avec l'activité de conception du produit, ce qui s'inscrit dans une démarche de conception guidée par la fabricabilité. La conception guidée par la fabricabilité a pour objectifs une amélioration des boucles de reconception et de la qualité du produit par l'échange de connaissances entre concepteurs et fabricants, ces flux d'information pouvant être en partie automatisés. La conception guidée n'est pas un thème nouveau. Mais les approches et outils existants ne sont pas satisfaisants parce qu'ils ne sont pas génériques, et parce qu'ils ne sont pas intégrés aux logiciels de CAO. Dans l'idéal, la CAO devrait en effet contribuer au PLM en permettant d'optimiser le produit, y-compris en regard des autres phases du cycle de vie. Ceci inclut notamment une gestion de la fabricabilité dès les premières phases de la conception. Une adaptation des outils de conception existants est donc requise en vue d'y intégrer la connaissance de fabrication (analyse de fabricabilité), et la notion de collaboration avec les fabricants, tout en garantissant le respect de la confidentialité, élément crucial dans un contexte d'entreprise étendue.

Ces travaux prennent part au projet DIJA de l'équipe CMCAO, qui vise à définir un environnement de CAO utilisable en réseau et orienté métier. Cet environnement est conçu de manière à se décliner en différentes variantes destinées à des domaines d'application et à des métiers distincts. Le public auquel s'adresse DIJA est donc potentiellement novice (en informa-

1. Conception Assistée par Ordinateur.
2. Fabrication Assistée par Ordinateur.

tique, mais bien sûr pas dans le métier concerné). C'est pourquoi nous privilégions une interface homme-machine (IHM) « intuitive » [3], contrairement aux IHM mettant en jeu des contrôles multiples, confus ou insignifiants aux yeux des non-initiés.

Un autre enjeu majeur de ce projet réside dans la prise en compte des aspects fonctionnels de l'activité de conception. Nous estimons en effet que les fonctionnalités d'un produit devraient pouvoir être considérées dans toutes les phases du développement du produit. Idéalement, partant d'une décomposition fonctionnelle, nous devrions être en mesure de raffiner le modèle jusqu'à arriver à la forme, caractérisée par un modèle géométrique. Chaque élément géométrique s'appuie sur une fonction qui justifie sa présence et permet une traçabilité de l'intention de conception. Ces aspects ne sont que trop peu considérés par les logiciels de CAO actuels qui se cantonnent bien souvent à la modélisation géométrique du produit. À ces fins, l'approche de conception utilisée par DIJA, dite synthétique, est descendante, s'opposant ainsi aux approches constructives (p. ex. : utilisation courante des CSG [4], modèles paramétriques...). La mise en place d'un tel projet fait appel à des domaines aussi variés que la modélisation géométrique, les architectures distribuées, les IHM, le travail collaboratif, la conception fonctionnelle ou la gestion des connaissances. C'est un projet fédérateur au sens où les travaux menés par les membres de l'équipe CMCAO sont dédiés à des aspects complémentaires. Les travaux menés au cours de cette thèse concernent des aspects relatifs à l'ingénierie simultanée, à la gestion des connaissances et au travail collaboratif.

Ce manuscrit s'organise comme suit. Le chapitre 1 *(Conception intégrée de produits)* présente l'activité de conception intégrée d'un point de vue général. Dans un premier temps, le processus de conception est défini à travers sa finalité, les différents types d'approches existantes, ainsi que le rapport entretenu avec les phases en amont et en aval de la conception. La façon de considérer le processus de conception influe sur la technique de modélisation utilisée qui est elle même étroitement liée au modèle sous-jacent. Dans un deuxième temps, nous présentons donc les différents modèles existants, puis nous faisons le tour des différentes techniques de modélisation. Nous présentons ensuite les concepts de PLM, de travail collaboratif, et de gestion des connaissances qui sont étroitement liés et qui induisent une évolution du modèle géométrique vers le modèle produit, puis vers le modèle global. Nous concluons ce chapitre en précisant l'objectif de la thèse.

3. Peut être faudrait-il dire « une IHM intuitive pour un public le plus large possible » ; ce qui est intuitif pour un individu n'étant par définition pas nécessairement intuitif pour un autre individu.

4. *Constructive Solid Geometry*.

Le chapitre 2 *(Conception guidée par la fabricabilité : état de l'art)* offre un panorama des recherches existantes en matière de conception guidée par la fabricabilité. Les critères de fabricabilité d'un produit sont présentés de manière globale. Vient ensuite une classification des méthodes qui permettent d'évaluer ces différents critères et des outils déjà développés. Pour conclure ce chapitre, les intérêts et les défauts de ces outils sont mis en exergue ; les principaux inconvénients étant qu'ils ne sont pas suffisamment génériques et pas actifs en permanence au cours de la conception.

Le chapitre 3 *(Un outil de modélisation guidée par le savoir-faire des fabricants)* présente notre proposition de modèle d'intégration du savoir-faire en commençant par poser les choix fondamentaux et par définir le lien avec les autres travaux de l'équipe, autour du projet DIJA. Le principe de fonctionnement est ensuite présenté d'un point de vue général (distinction du modèle de conception du modèle de fabrication), puis les aspects statique (graphes de dépendances et d'états) et dynamique (évolution des règles de fabricabilité) du modèle sont détaillés. L'ensemble de l'approche est récapitulé dans une synthèse.

Le chapitre 4 *(Mise en œuvre informatique)* présente la validation des concepts théoriques. Les choix d'implantation et l'organisation de la maquette (librairies, diagrammes) sont exposés avant de présenter l'application à travers un exemple de scénario collaboratif illustrant les aspects développés dans le chapitre 3. Un bilan conclue ce chapitre.

Le chapitre *Conclusion* clôture ce mémoire en rappelant ce qui a été présenté, l'apport de la proposition ainsi que ses limites. Pour finir, nous y soumettons également des directions de recherche qu'il semble intéressant d'explorer par la suite.

Chapitre I

Conception intégrée de produits

Sommaire

1.1 Introduction 6

1.2 Le processus de conception 7

1.2.1 Du besoin aux fonctions 9

1.2.2 Des fonctions à la forme 14

1.2.3 Articulation avec l'analyse d'ingénierie 26

1.2.4 Conclusion 28

1.3 Modélisation d'un produit 29

1.3.1 Modèle géométrique et modèle produit 29

1.3.2 Techniques de modélisation 37

1.3.3 Conclusion 39

1.4 Vers le modèle global 40

1.4.1 Gestion du cycle de vie d'un produit (PLM) 42

1.4.2 Ingénierie simultanée 44

1.4.3 Travail collaboratif 46

1.4.4 Gestion des connaissances 50

1.4.5 Conclusion 52

1.5 Synthèse 53

1.1 Introduction

La conception de produits manufacturés a bien évolué depuis l'apparition, dans les années soixante-dix, des premiers outils informatiques dédiés. Les premiers logiciels ont permis d'assister le concepteur dans sa tâche qui consistait alors à établir à la main l'ensemble de vues et de coupes bidimensionnelles représentant la géométrie d'un produit et nécessaires à sa fabrication. L'inconvénient majeur de ces outils était que, de même que pour le dessin sur planche, il était tout à fait possible de représenter des figures incohérentes ou ambiguës [5]. Mais de plus en plus d'aspects liés au métier de la conception (opérations géométriques d'assistance au dessin, cotation, tolérancement) ont peu à peu été pris en charge. C'est dans les années quatre-vingt, lorsque l'agencement spatial des entités formant le modèle de la pièce a été considéré à travers l'outil informatique, que la représentation de pièces inconsistantes a directement été rendue impossible (en principe) par le logiciel [6]. Cela a aussi permis l'apport de nouveaux services : perspectives, coupes, calculs de volumes, d'inertie, rendu réaliste, etc. quelle que soit la forme de l'objet [Blain, 1990]. Par la suite, d'autres améliorations comme les approches paramétrique et variationnelle ou l'introduction des caractéristiques ont contribué à améliorer la CAO. L'avènement de ces outils n'est devenu possible que par la réduction du coût de mise en place des stations de travail qui était au départ rédhibitoire pour les petites et moyennes industries.

Parallèlement à ces outils de conception, se sont développés d'autres outils informatiques permettant de mener à bien les autres phases du développement du produit. Dans le domaine de la fabrication, l'utilisation, dès le milieu des années soixante-dix, de machines-outils à commande numérique a donné lieu au développement de logiciels de fabrication assistée par ordinateur (FAO) permettant la mise en place de séquences d'usinage. Les logiciels de calcul et de simulation ont également connu un net essor, plus particulièrement depuis une dizaine d'années.

En outre, tous ces développements permettent aujourd'hui d'envisager le produit à travers sa maquette numérique, continuum reliant les premières phases de la conception à la livraison du produit fini. La maquette numérique tend progressivement à se substituer à la maquette physique (plans, prototypes) dont la réalisation est de plus en plus une étape finale, voire élu-

5. Il s'entend que celles-ci pouvaient résulter d'une exécution respectueuse des règles du dessin.

6. C'est ainsi que les logiciels, qui étaient jusqu'alors des applications de dessin assisté par ordinateur (DAO), ont été rebaptisés logiciels de CAO. Cela coïncide aussi avec une transition des modes de définition en 2D vers des modes de définition en 3D, levant ainsi les ambiguïtés induites par les représentations bidimensionnelles.

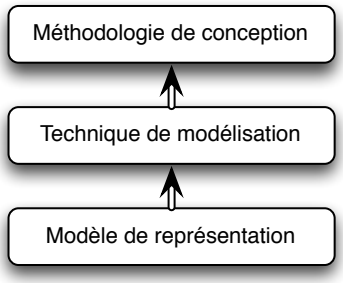


FIGURE 1.1: Processus et modèle de conception.

dée. Les modèles informatiques, au départ cantonnés à la description de la géométrie, évoluent afin d'englober de plus en plus d'informations relatives au produit, dans le but de les rendre centrales et explicites. Mais si nous observons un enrichissement des modèles de représentation, c'est aussi parce que les processus de conception sont devenus plus complexes. Les changements méthodologiques observés influent en effet sur la technique de modélisation, elle-même s'appuyant sur le modèle de représentation utilisé (cf. figure 1.1).

Les travaux présentés dans ce manuscrit concernent en particulier la phase de conception, étape clef du développement d'un produit. Plus précisément, nous souhaitons intervenir dans le processus de conception en vue de le rattacher à des connaissances du domaine de la fabrication. Nous proposons donc d'étudier le processus de conception, notamment à travers la méthodologie de conception, la technique de modélisation et le modèle de représentation (cf. figure 1.1), de façon à poser les bases de notre approche.

Dans ce chapitre, nous présentons les différentes approches existantes permettant d'appréhender l'activité de conception (sect. 1.2), ainsi que la modélisation du produit (sect. 1.3). Nous détaillons ensuite les tendances d'évolution, et les nouveaux besoins qui en découlent (sect. 1.4). Nous concluons par une synthèse faisant ressortir les aspects qui nous intéressent plus particulièrement dans le cadre de notre problématique.

1.2 Le processus de conception

Si pour l'utilisateur un produit est un moyen de satisfaire un besoin [7], pour l'entreprise, un produit est le fruit d'un (ou de plusieurs) processus. D'un point de vue général, nous consi-

7. L'Afnor (Association française de normalisation) définit le besoin comme ceci : « Nécessité ou désir éprouvé par un utilisateur. » [NF X 50-150, 1990]. Il peut être explicite ou implicite, existant ou potentiel [www-AFAV].

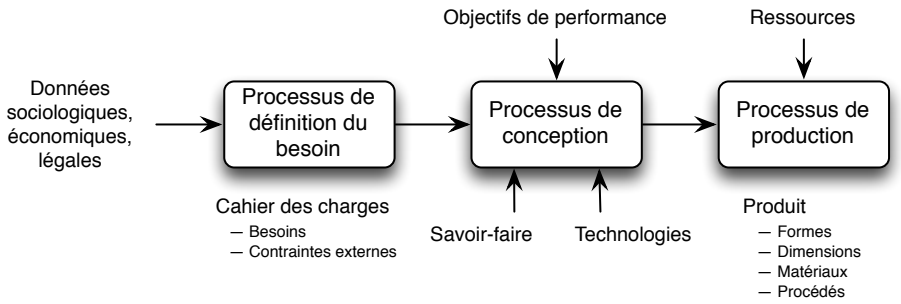


FIGURE 1.2: Phases en amont et en aval du processus de conception.

dérons que le processus de conception a pour but la définition d'un produit fini respectant un ensemble de contraintes [8] décrites dans un cahier des charges (cf. figure 1.2). Ces contraintes initiales sont définies en amont de la conception par une étude de marché (avant-projet). Celle-ci prend notamment en considération des paramètres économiques (conjoncture, concurrence, stratégie), sociologiques (statistiques liées à la demande et au besoin) ou encore législatifs (normes de sécurité, respect de l'environnement). En aval de la conception, le produit doit passer de l'état virtuel (modèle de représentation) à l'état de réalité physique. La solution adoptée est ainsi évaluée en vue de sa fabrication. Ce sont là des critères technologiques et liés au savoir-faire et aux ressources disponibles qui déterminent la capacité du produit à être fabriqué dans des conditions conformes au cahier des charges. Ainsi, différents acteurs influencent directement ou indirectement le processus de conception : les concepteurs ; les décideurs, les financiers et les services de commercialisation (en amont) ; et les fabricants (en aval). Cette influence s'exerce à travers les décisions de conception. La conception est une évolution technique et sociale de l'information, ponctuée par des prises de décisions [Ullman, 2002]. [Girard, 2001] note une distinction entre décisions de conduite et décisions technologiques. Les décisions technologiques correspondent à des choix de solutions techniques et sont donc souvent liées à des connaissances formalisables (p. ex. le choix d'une dimension ou d'un procédé de fabrication). Les décisions de conduite sont quant à elles beaucoup plus subjectives et sont nécessairement prises par l'humain car difficilement formalisables (p. ex. la favorisation de la créativité ou la maîtrise des coûts).

Lors de la phase de conception, il est recherché une solution optimale à un problème spé-

8. Ici, cette notion n'est pas considérée au simple sens qu'une contrainte puisse être soit satisfaite, soit violée : différents niveaux de satisfaction peuvent être envisagés.

cifique. L'objectif est donc non seulement de définir le produit, mais également de trouver le meilleur chemin aboutissant à cette définition [Girard, 2001]. Le caractère optimal de la solution obtenue est principalement évalué suivant trois axes : le coût de fabrication et de développement du produit ; la qualité du produit (son adéquation au besoin) ; et le délai nécessaire à son obtention. Trois types de conception sont habituellement distinguées [Girard, 2001] :

la conception créative qui correspond aux situations où l'on ne connaît pas *a priori* de décomposition du problème. Il faut donc déterminer le plan d'activités, et les connaissances mises en jeu ne sont pas nécessairement connues et maîtrisées ;

la conception innovante qui se base sur un produit existant, dont une décomposition structurelle est donc connue. L'innovation porte sur un ou plusieurs aspect(s) technologique(s) du produit. Elle est en principe planifiée ;

la conception routinière qui correspond aux cas où l'on connaît la stratégie et les connaissances nécessaires. Les décisions prises pour ce type de conception sont essentiellement technologiques.

[Brown, 1998] préfère appréhender ces catégories de façon plus fine, en les situant selon deux axes : l'axe conceptuel-paramétrique qui définit le degré d'abstraction du problème de conception ; et l'axe éprouvé-inconnu qui définit l'expérience que l'on en a. Ceci révèle qu'il n'existe pas de frontières franches entre les trois catégories mentionnées ci-avant. Naturellement, le déroulement de la conception se fait aussi et surtout en fonction de la nature et du domaine du problème. Ainsi, toutes les entreprises ont leurs particularités et donc leurs propres visions du processus de conception. Dans tous les cas, la finalité est de partir du besoin pour en retirer la définition d'artefacts, ce qui nécessite une étape intermédiaire : l'analyse fonctionnelle.

Dans la suite, nous décrivons les approches centrées sur la notion de fonction et de décomposition fonctionnelle (§ 1.2.1). Un panorama des approches permettant de passer des fonctions à la forme du produit est ensuite proposé (§ 1.2.2). Puis, nous abordons l'analyse d'ingénierie, notamment utilisée pour vérifier la validité fonctionnelle d'un produit (§ 1.2.3).

1.2.1 Du besoin aux fonctions

Il est indispensable de définir et d'analyser le besoin de l'utilisateur, initialement enregistré sous forme purement textuelle, afin d'en déduire la forme d'une pièce. Le cahier des charges fonctionnel permet de traduire le besoin sous une forme plus rationnelle, donc moins implicite,

c'est-à-dire en termes de fonctionnalités. Notamment cela permet de s'assurer que le besoin est totalement défini, et d'apporter une référence aussi bien technique que juridique (contrat). Le raffinage des fonctions, qui permet d'entrevoir une ou plusieur(s) solution(s) satisfaisant le besoin, est capital du fait de la non unicité de la décomposition fonctionnelle.

Dans la suite de ce paragraphe, nous explorons la notion de fonction et ses propriétés (§ 1.2.1.1), et l'analyse fonctionnelle (§ 1.2.1.2). Quelques méthodes d'analyse et de décomposition fonctionnelles sont également abordées (§ 1.2.1.3).

1.2.1.1 Notion de fonction

Il est difficile de donner une définition objective de la notion de fonction du fait qu'elle dépend fortement de l'intention du concepteur [Umeda, 1997]. Le dictionnaire de l'Académie française définit une fonction en tant que « rôle caractéristique joué par un élément au sein d'un ensemble » [Druon, 2001]. Dans le domaine de l'industrie, nous pouvons citer l'Afnor qui définit une fonction comme étant « l'action d'un produit ou de l'un de ses constituants exprimée uniquement en terme de finalité » [NF X 50-150, 1990].

Les fonctions peuvent être classifiées suivant leurs propriétés. [Minich et Pallez, 1999] proposent la classification suivante :

fonctions objectives ou subjectives : une fonction est objective si l'on peut juger sans équivoque de sa présence (p. ex. communiquer pour un téléphone). À l'inverse, une fonction subjective est liée à l'interprétation personnelle qu'en fait le concepteur (p. ex. ce qui a trait à l'esthétique). Dans une certaine mesure, il est possible d'« objectiver » les fonctions subjectives ;

fonctions intentionnelles ou déduites : les fonctions intentionnelles dérivent de l'intention du concepteur et donc du besoin exprimé dans le cahier des charges (p. ex. brûler du carburant pour un véhicule). Les fonctions déduites sont des fonctions collatérales des concepts utilisés pour répondre aux fonctions intentionnelles (p. ex. le bruit ou la pollution induits par la carburation) ;

fonctions bénéfiques ou néfastes : les fonctions requises dans le cahier des charges sont bénéfiques au produit. Les fonctions déduites peuvent quant à elles être soit bénéfiques, soit néfastes, d'où l'importance de la décomposition fonctionnelle.

Dans le paragraphe suivant, nous décrivons l'analyse fonctionnelle, qui permet de déterminer,

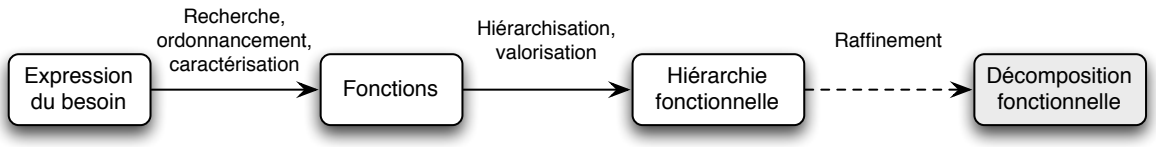


FIGURE 1.3: Caractérisation, hiérarchisation et raffinement des fonctions.

de comprendre et d'étudier les fonctions que doit remplir un produit.

1.2.1.2 Analyse fonctionnelle

Selon l'Afnor, « L'analyse fonctionnelle consiste à identifier les fonctions, à les valider au moyen d'une rationalité explicite et à les caractériser » [NF X 50-100, 1996], ou encore, « l'analyse fonctionnelle est une démarche qui consiste à rechercher, ordonner, caractériser, hiérarchiser et/ou valoriser les fonctions » [NF X 50-150, 1990]. Elle donne donc lieu à une caractérisation et à une hiérarchisation des fonctions (cf. figure 1.3). La recherche des fonctions se base sur le besoin initial, exprimé dans le cahier des charges. Plusieurs approches, souvent complémentaires, sont disponibles. Nous pouvons citer la recherche intuitive (plus connue sous le nom de *brainstorming*), l'inventaire systématique du milieu, l'étude du comportement de l'utilisateur, l'analyse des séquences (méthode SAFE [9]), ou la recherche des insatisfactions des clients. L'ordonnancement des fonctions aide à les caractériser. Pour caractériser les fonctions, le concepteur doit partir de l'expression du besoin pour aboutir à une définition fonctionnelle du produit, le besoin étant traditionnellement exprimé en langage naturel. Il s'agit d'énoncer des critères d'appréciation, des niveaux d'appréciation et des niveaux de flexibilité (marges de manœuvre) pour chacune des fonctions. La hiérarchisation des fonctions consiste à leur donner des niveaux d'importance les unes par rapport aux autres en vue de les valoriser – la hiérarchisation est qualitative et la valorisation quantitative. L'analyse fonctionnelle permet de rendre explicite et de préciser le besoin afin de rendre possible l'étude de différentes solutions, ce qui s'effectue à travers le raffinement des fonctions. Un vaste panel de méthodes existe pour analyser et décomposer les fonctions. [Minich et Pallez, 1999] expliquent qu'une méthode se définit par la donnée d'un formalisme et d'une démarche. Le formalisme permet la représentation de l'objet de la réflexion tandis que la démarche décrit la manière de la faire évoluer. Les fonctions seront donc représentées grâce à un formalisme spécifique, souvent lié à la méthode employée

9. *Sequential Analysis of Functional Elements*.

pour les analyser et les décomposer. Quelques méthodes d'analyse et de décomposition fonctionnelle sont présentées ci-après.

1.2.1.3 Méthodes d'analyse et de décomposition fonctionnelles

Beaucoup de méthodes de représentation et de raffinement des fonctions existent. Nous en présentons ici brièvement quelques-unes.

L'analyse de la valeur (AV) est une méthode américaine développée en 1945 par Lawrence D. Miles, alors responsable des approvisionnements de la *General Electric Company*. L'AV est une méthode d'aide à la conception de produits qui est à l'origine de l'analyse fonctionnelle, cette dernière ayant par la suite acquis une certaine autonomie. L'AV a une triple dimension : fonctionnelle, économique et pluridisciplinaire. Elle peut être présentée comme étant la définition d'une interface entre les deux faces d'un produit : sa face fonctions-valeurs (cachée) et sa face artefacts-coûts (visible) [Perrin, 2005]. Elle détermine les fonctions du produit grâce à une caractérisation des liens existant entre celui-ci et son environnement. Sont distinguées :

- les fonctions de *service*. Une fonction de service est une « action attendue d'un produit (ou réalisée par lui) pour répondre à un élément du besoin d'un utilisateur donné » [NF X 50-150, 1990] – elle peut être d'usage ou d'estime ;
- les fonctions *contraintes*. Une fonction contrainte « vient de l'environnement, de la technologie, du marché, de la situation et des choix de l'entreprise » [NF X 50-150, 1990] ;
- et les fonctions *techniques* qui sont des « actions internes au produit (entre ses constituants), choisies par le concepteur-réalisateur, dans le cadre d'une solution, pour assurer des fonctions de service » [NF X 50-150, 1990].

Le célèbre diagramme pieuvre permet de représenter le produit, son environnement et ses fonctions de service et contraintes (cf. figure 1.4).

La conception à coût objectif (CCO) [10] ou *Design to Cost* [NF X 50-156, 2003] est un complément de l'analyse de la valeur permettant de concevoir un produit en fonction d'une cible de clients potentiels. Ce concept a par exemple été introduit chez Renault à partir de 1992 pour la conception de la Twingo. Des indicateurs assurent le suivi de la progression vers l'objectif, ce qui permet d'orienter l'arbitrage et la conduite du projet [www-AFAV].

La matrice QFD [11], aussi appelée « maison de la qualité » est un complément de l'analyse

10. On parle plus généralement de conception à objectif désigné (COD).
11. *Quality Function Deployment*.

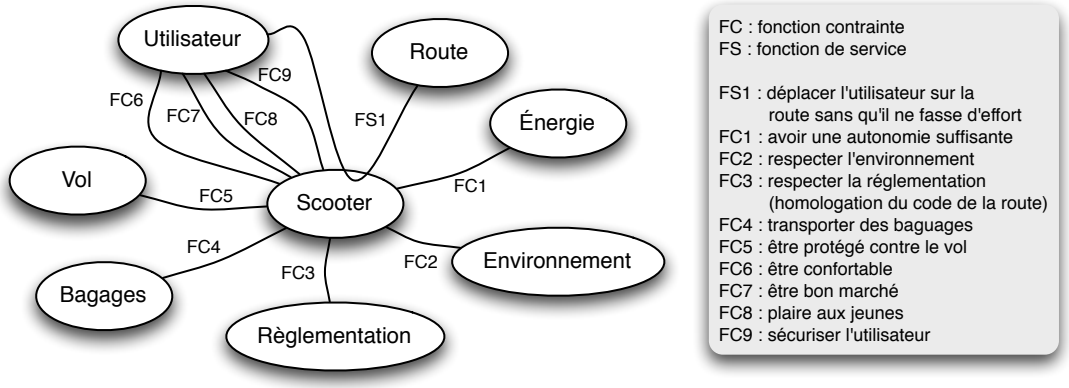


FIGURE 1.4: Exemple de diagramme pieuvre.

de la valeur permettant la prise en compte de l'ensemble des besoins critiques des clients. Le formalisme de représentation le plus usité se présente sous forme d'un dessin en forme de maison (d'où le nom). L'entrée de la maison représente le « quoi » (le besoin) et l'intérieur de la maison le « comment » (les fonctions du produit à réaliser). Le garage représente les produits concurrents, ce qui permet d'y comparer le produit. Le toit représente les dépendances entre les fonctions, et les fondations représentent le « combien », c'est-à-dire l'estimation des fonctions. Il est possible de chaîner plusieurs maisons en liant le « comment » d'une maison au « quoi » d'une autre maison, ce qui permet une certaine décomposition.

La méthode SADT [12], développée par [Ross, 1977], est une méthode d'analyse et de décomposition fonctionnelle systémique. Un modèle SADT se compose d'un ensemble de diagrammes, dont le principal type est l'actigramme, complété par des descriptions textuelles. Les actigrammes décomposent le système en modules. Chaque module est spécifié par des données en entrée, des données en sortie, une fonction de transformation associée, et des données de contrôle (voir figure 1.5). De plus, les modules peuvent être décomposés en sous-modules, permettant le raffinement progressif des fonctions. Des compléments de SADT ont vu le jour, comme par exemple SART *(Structured Analysis for Real Time)* qui apporte un support pour l'aspect temporel.

La méthode FAST [13], dont [Bytheway, 1965] est à l'origine, dérive des travaux de Miles sur l'analyse de la valeur. L'objectif de Bytheway était de stimuler la créativité en posant des ques-

12. *Structured Analysis and Design Technique*.
13. *Functional Analysis System Technique*.

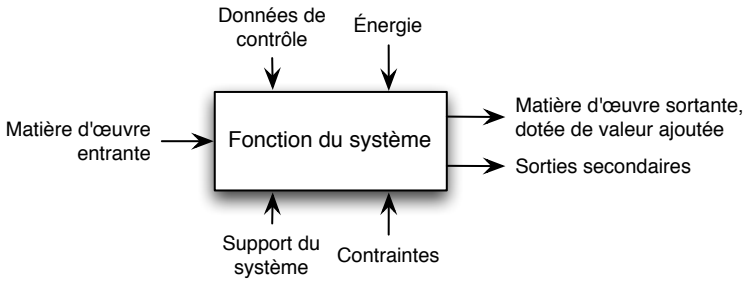


FIGURE 1.5: Actigramme de la méthode SADT.

tions provoquant une remise en cause des choix de conception fonctionnelle. Le modèle FAST permet de décomposer les fonctions selon deux axes : l'axe « pourquoi-comment » qui indique la façon dont une fonction est réalisée et inversement, et l'axe « quand » qui spécifie la décomposition ainsi que les différentes solutions alternatives. Ces dépendances sont représentées sur un graphe de type « et-ou » dont les nœuds représentent les fonctions, et dont les arcs permettent de définir la décomposition en sous-fonctions et les différentes décompositions alternatives pour une même fonction.

À présent que nous avons présenté la notion de fonction, et l'analyse et la décomposition fonctionnelles qui permettent d'aboutir aux fonctions à partir du besoin, nous parcourons dans le paragraphe suivant les approches focalisées sur le passage fonctions-forme.

1.2.2 Des fonctions à la forme

Un certain nombre de recherches, menées depuis plus de trente années, visent à définir des théories, des modèles et des méthodes pour les processus de conception. Les différents points de vue adoptés ont mené à des approches très variées dont il est impossible de faire un inventaire complet. Nous nous contenterons de parcourir quelques-uns des principaux courants, en adoptant successivement des points de vue techno-mécanistes (§ 1.2.2.1), mathématiques (§ 1.2.2.2), cognitivistes (§ 1.2.2.3), et informatiques (§ 1.2.2.4).

Le processus de conception peut être appréhendé à travers des approches descriptives (pour expliquer ce qu'est la conception), prescriptives (pour montrer comment mieux concevoir) ou encore prédictives (pour prévoir les propriétés des objets conçus) [Chen et Doumeingts, 2000]. En outre, certains travaux cherchent à modéliser le processus de conception, d'autres ont pour but de l'automatiser partiellement, et d'autres encore

tendent à l'améliorer par une démarche devant être adoptée par le concepteur. Naturellement, les points de vue décrits ci-après ne sont pas exclusifs et des approches mixtes existent.

1.2.2.1 Point de vue techno-mécaniste

Les approches systémiques considèrent le processus de conception comme une séquence d'opérations permettant d'aboutir à une définition du produit.

Pahl et Beitz proposent à partir des années quatre-vingt une vision systématique du processus de conception [Pahl et Beitz, 1999]. Ce dernier est décomposé en tâches et en sous-tâches. Les quatre phases principales sont la planification et la clarification du besoin, la conception conceptuelle, la conception préliminaire et la conception détaillée. À la fin de chaque étape, une évaluation de la solution en regard des critères définis lors des phases précédentes peut donner lieu à des rebouclages. Un grand nombre d'approches systématiques découlent de cette vision séquentielle du processus de conception, communément reconnue comme référence.

Notamment, [Ben Amara et coll., 1996] proposent avec la *conception par fonctionnalités*, un découpage en trois étapes cycliques :

- la spécification fonctionnelle qui suit les mêmes principes que l'analyse de la valeur ;
- la conception préliminaire qui consiste en une énumération, une classification et une validation de solutions sur le plan de la cinématique ;
- enfin, la conception détaillée qui consiste à habiller la chaîne cinématique obtenue à l'étape précédente.

Dans ce même esprit, le modèle SATT [14] [Clément et coll., 1996] utilise la notion de flux entre le produit et l'environnement afin d'en déduire automatiquement des informations topologiques. Un flux ne traversant que deux surfaces fonctionnelles, le raffinement des flux permet de déduire de nouvelles surfaces associées technologiquement et topologiquement aux précédentes, ce qui mène à une description topologique du produit à travers un graphe de surfaces fonctionnelles typées (linéaire, cylindrique, etc.). Le concepteur peut ensuite définir les surfaces manquantes et ainsi raccorder les SATT.

Le modèle FBS [15] [Umeda et coll., 1990], et la méthode associée FEP [16] [Shimomura et coll., 1996], visent à définir un cadre pour la conception à travers un triple point de vue : les *fonctions* du produit représentent ce qu'il doit faire (le problème) ;

14. Surfaces Associées Technologiquement et Topologiquement.
15. *Function, Behaviour, Structure.*
16. *Functional Evolution Process*.

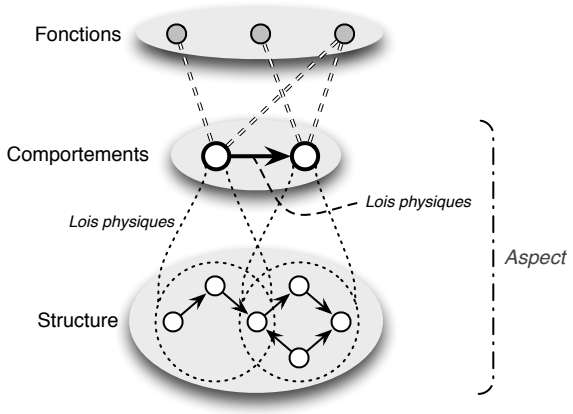


FIGURE 1.6: Relations entre fonctions, comportements et structure [Takeda et coll., 1997].

ses *comportements* sont les façons dont il réalise les fonctions (la solution) ; et sa *structure* est sa représentation interne (les entités, leurs attributs et leurs relations). La figure 1.6 présente les liens existant entre ces différentes notions. Selon les auteurs, les fonctions sont liées à la perception de l'objet qu'ont les concepteurs, alors que le comportement peut être déduit de la structure et de l'environnement. À travers la structure et les comportements, regroupés par aspects, les auteurs tendent ainsi à réaliser une liaison entre fonctions et forme. Mais ils se limitent néanmoins au passage des fonctions à la structure, qui n'est d'ailleurs pas automatique. [Ranta et coll., 1996] proposent un passage semi-automatisé entre fonctions et forme avec un couplage entre le modeleur FBS SYSFUND [17] et le modeleur produit MCOES [18]. La structure fonctionnelle est créée avec le modeleur FBS puis est confiée au modeleur produit qui la transforme en une famille d'assemblages. Le concepteur a ainsi la faculté d'identifier les fonctions associées aux entités directement depuis le modeleur.

D'autres découpages du processus de conception, proches de celui proposé par Pahl et Beitz, existent [Pugh, 1991 ; Ullman, 2002 ; Ulrich et Eppinger, 2003]. Les cinq phases suivantes, tirées de [Gardan, 2003], sont assez représentatives :

1. L'*analyse* et la *conception fonctionnelles* permettent d'identifier et de structurer les fonctions.
2. La *conception conceptuelle* consiste à faire émerger des concepts (structures physiques) qui répondent aux fonctions établies lors de la première étape.
3. La *conception intégrée* consiste à associer chaque entité de la structure à la ou les fonction(s)

17. *SYStematization tool of FUNctional knowledge for Design*.
18. *Manufacturing Cell Operator's Expert System*.

correspondante(s) dans la hiérarchie fonctionnelle. Ceci permet de vérifier qu'en principe, les fonctions sont bien assurées par les concepts.

4. La *conception détaillée* a pour objectif de raffiner la description des concepts jusqu'à arriver à une définition suffisamment précise pour envisager la fabrication.
5. L'*analyse d'ingénierie* permet de vérifier que la solution développée remplit bien les fonctions qui lui ont été attribuées. Des rebouclages plus ou moins importants sont possibles.

Dès lors, deux types d'approches sont possibles [Gardan, 2003] :

les approches descendantes où l'on cherche à modéliser le processus de conception de façon hiérarchique pour aboutir à la forme par raffinements successifs (cf. figure 1.7a) ;

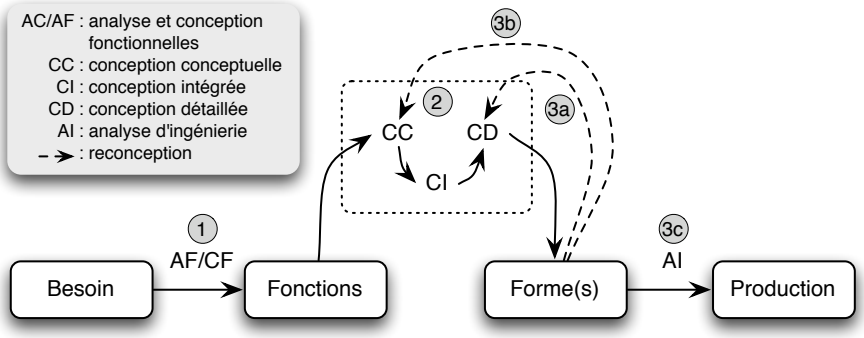
les approches ascendantes où c'est le concepteur qui génère directement les concepts jusqu'à satisfaction des fonctions (cf. figure 1.7b). Le lien avec les fonctions est rendu possible par l'ajout d'un niveau intermédiaire entre fonctions et formes (p. ex. des caractéristiques). On ne cherche donc pas à systématiser l'élaboration des formes à partir des fonctions.

Dans ces approches, la conception est considérée comme un processus cyclique. L'approche co-évolutive de la conception [Lonchampt, 2004] est centrée sur cet aspect. Elle est fondée sur l'idée qu'il est nécessaire de reformuler le besoin pour parvenir à une solution (évolution conjointe du besoin et de la solution). L'auteur définit ainsi un cycle élémentaire de conception en quatre phases : conjecture ; définition d'une solution ; évaluation ; et reformulation du besoin. Le cycle s'arrête lorsque l'évaluation de la solution est satisfaisante.

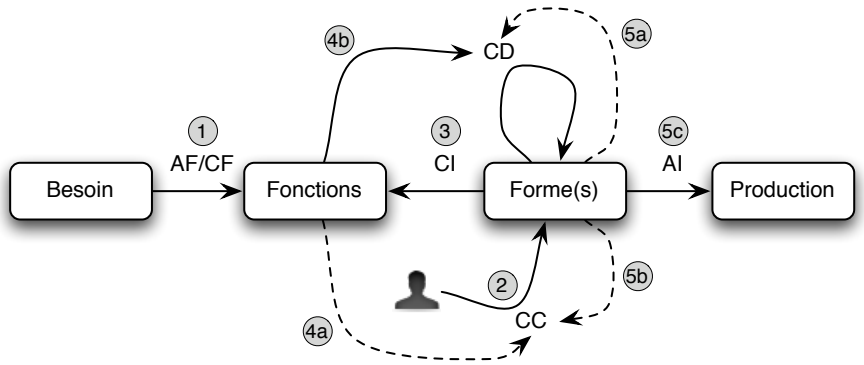
1.2.2.2 Point de vue mathématique

L'élaboration d'une théorie de la conception est une préoccupation récurrente depuis le début des années soixante-dix. Nous en présentons quelques-unes dans ce paragraphe.

[Yoshikawa, 1981] a introduit la première théorie générale en conception, nommée GDT *(General Design Theory)* . Elle se base sur la théorie des ensembles et la topologie pour modéliser les connaissances et le processus de conception. Cette théorie, fondée sur des principes mathématiques, est en pratique limitée dans le sens où elle considère un processus de conception et des connaissances idéaux, ce qui la rend difficilement applicable à des cas de conception réels [Reich, 1995]. La théorie mathématique de la conception [Braha et Maimon, 1998] est, d'un point de vue conceptuel, similaire à la GDT, mais le processus de conception réel est défini en se basant sur la théorie des automates finis.



(a) Approche descendante



(b) Approche ascendante

FIGURE 1.7: Approches systémiques du processus de conception.

La théorie universelle de la conception ou UDT *(Universal Design Theory)* [Lossack et Grabowski, 2000] définit le processus de conception comme un ensemble fini de niveaux d'abstraction (spécification, conception fonctionnelle, couche physique et réalisation) et d'étapes à effectuer. La connaissance et les solutions sont modélisés à travers les concepts d'objet, de processus et de solution qui sont décomposables et réutilisables.

L'approche axiomatique de la conception [Suh, 1990] se fonde sur l'hypothèse qu'il existe des principes généraux (les axiomes [19]) pouvant s'appliquer à la conception de produits. L'espace de conception est partagé en quatre domaines, dont les trois derniers sont hiérarchisés (cf. figure 1.8) : besoins du client, besoins fonctionnels, paramètres de conception et paramètres de fabrication. Les deux axiomes sont l'axiome d'*indépendance*, qui tend à maintenir l'indépendance des besoins fonctionnels, et l'axiome d'*information* qui vise à minimiser la redondance fonctionnelle. Ils sont utilisés pour connaître les solutions qui sont acceptables parmi celles

19. « Prémisse considérée comme évidente et reçue pour vraie sans démonstration. » [Druon, 2001].

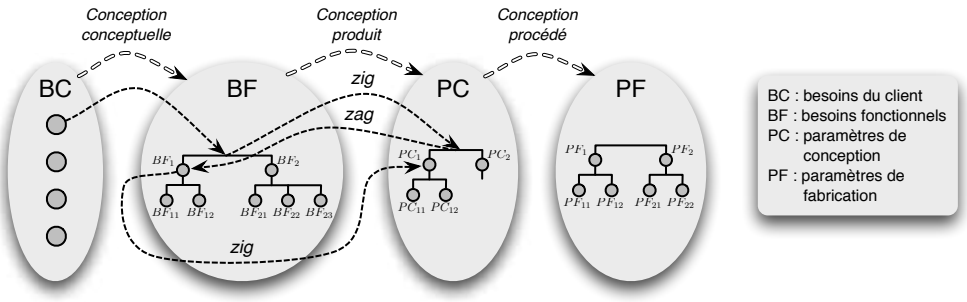


FIGURE 1.8: Les quatre domaines de la conception axiomatique [Cochran et Reynal, 1996].

imaginées par le concepteur. La conception est basée sur un modèle de raisonnement en zig-zags, alternant phases de synthèse (les « zigs ») et d'analyse (les « zags »), tout en progressant dans les hiérarchies. Les besoins fonctionnels et les paramètres de conception étant représentés sous forme de vecteurs, la correspondance entre ces domaines s'effectue via la matrice de conception : $\{BF\} = [MC]\{PC\}$. Chaque élément de la matrice $MC_{ij} = \frac{\partial BF_i}{\partial PC_j}$ est une constante correspondant à la contribution du paramètre de conception PC_j au besoin fonctionnel BF_i. Certaines propriétés de MC correspondent à des types de conception particuliers [20]. Afin de respecter l'axiome d'indépendance, MC doit ainsi être diagonale ou triangulaire. Pour respecter l'axiome d'information, la somme d'information $I = \sum_{i=1}^{n} \log\left(\frac{1}{p_i}\right)$ contenue dans la conception doit être minimisée, p_i étant la probabilité que le BF_i soit satisfait (p. ex. les tolérances géométriques sont des facteurs de satisfaction). Cette approche ne dit pas comment effectuer les décompositions, mais il est possible de l'utiliser en vue de comparer différentes alternatives de conception [Lee et coll., 2004]. En outre, l'axiome d'indépendance s'oppose à l'intégration des fonctions, ce qui n'est pas nécessairement souhaitable dans tous les cas. Il existe des variantes utilisant des axiomes différents [www-Axiomatic].

Enfin, la théorie de la résolution de problème inventive, mieux connue sous son acronyme russe TRIZ [Altshuller et Shapiro, 1956], tend à stimuler la conception créative et est à ce titre prescriptive. Selon TRIZ, des lois décrivent l'évolution des systèmes techniques, et permettent, par la considération d'un produit idéal et par l'étude des obstacles à la créativité, de placer le concepteur dans une démarche innovante. Un algorithme en neuf étapes, nommé ARIZ [21] [Altshuller, 1992], utilise les principes de TRIZ pour guider la conception.

20. Notamment, si MC est pleine, la conception est dite couplée, si elle est triangulaire, la conception est dite découplée, et si elle est diagonale, la conception est dite non couplée.

21. ARIZ est aussi un acronyme russe, signifiant « algorithme de résolution de problème inventive ».

1.2.2.3 Point de vue psycho-cognitif

Les approches psycho-cognitives étudient la façon dont pensent les concepteurs afin d'identifier leur activité mentale. L'objectif est de mieux comprendre la façon dont un concepteur conçoit, ce qui correspond à une démarche descriptive. Ces approches se fondent principalement sur l'observation directe de l'activité du concepteur. À titre d'exemple, [Yoshikawa, 1989] découpe le processus mental d'un concepteur en cinq phases qui forment un cycle : la prise de conscience du problème, la suggestion, le développement, l'évaluation et la conclusion. Nous ne développons pas d'avantage ce type d'approche du fait qu'elles sont purement descriptives et ne présentent pas un intérêt immédiat dans ce manuscrit. La psychologie cognitive a néanmoins des intérêts divers en conception [22], le premier étant qu'elle constitue une source d'inspiration importante pour la définition d'autres approches.

1.2.2.4 Point de vue informatique

Les approches informatiques de la conception relèvent de l'intelligence artificielle (IA). Elles tentent donc d'imiter le raisonnement et la façon de se représenter la connaissance qu'a l'être humain. Dans ces approches, les concepts de connaissance et de raisonnement tiennent une place centrale [Haton et Haton, 2000 ; Gero et Maher, 1990]. Certains auteurs combinent plusieurs approches d'IA [23] afin de diversifier connaissances et raisonnement (p. ex. [Zakarian et Kaiser, 1999]). Dans la suite, nous abordons les systèmes à base de connaissances, le raisonnement par cas, la conception sous contraintes, les systèmes multi-agents et les systèmes d'apprentissage.

Systèmes à base de connaissances (SBC) [24]. Les applications des SBC concernent tous les domaines d'expertise et sont principalement destinées à répondre à des problèmes de diagnostic, de classification et de décision. Une caractéristique des SBC est la séparation des connaissances et du mécanisme de raisonnement (cf. figure 1.9). Deux formes de connaissance cohabitent :

22. Notamment l'étude des rapports que l'usager entretient avec le produit (p. ex. pour définir le besoin), ou dans le domaine des IHM, afin de guider la définition des interfaces d'utilisation des outils de conception.

23. D'autres auteurs lient des approches d'IA avec d'autres paradigmes, comme p. ex. [Boyle et coll., 2004] qui utilisent les principes de la conception axiomatique (cf. § 1.2.2.2) dans une approche de conception à base de cas (cf. page 21) pour des mécanismes de bridage. L'intérêt de cette approche réside dans l'indexation des cas qui prend en compte la décomposition fonctionnelle plutôt que l'expression des besoins.

24. Le terme de système expert tend à s'effacer devant celui de système à base de connaissances. Ceci tient au fait que la problématique originelle, centrée sur un expert humain, a fait place, avec la diversification des modes de représentation et d'acquisition des connaissances, à une problématique centrée sur la connaissance.

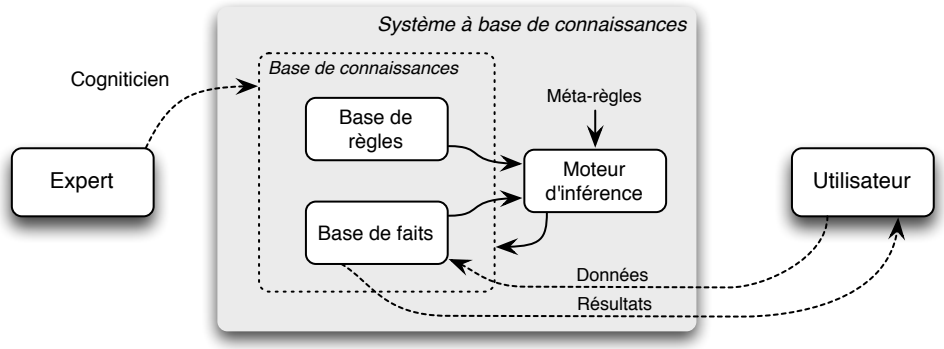


FIGURE 1.9: Architecture d'un système à base de connaissances.

les règles, et les faits. Selon un cycle, le moteur d'inférence peut raisonner par des déductions, inductions ou abductions (des inférences) conduisant à de nouvelles connaissances. Des méta-règles définissent la conduite du raisonnement (résolution de conflits, contrôle). De nombreux SBC sont utilisés dans l'industrie manufacturière, ainsi que dans d'autres domaines [25]. À titre d'exemple, dans le champ de la CAO, nous pouvons citer des applications concernant la conception de dispositifs de contrôle [Lin et Shen, 1995] ou de produits en matière plastique [Borg et MacCallum, 1995], l'optimisation de produit [Chau et Albermani, 2003] ou la conception d'assemblages [Myung et Han, 2001]. L'avantage avec un SBC est qu'il est en principe possible de faire évoluer la connaissance sans avoir à agir sur le mécanisme de raisonnement. Mais en réalité, il est souvent nécessaire de réfléchir à la structuration des connaissances et de définir des stratégies d'utilisation des règles. En outre, les SBC semblent montrer leurs limites dès qu'ils doivent gérer de gros volumes d'information (construction, maintenance) [Haton et Haton, 2000], ce qui est particulièrement vrai en conception où l'on manipule des informations spatiales. Ce manque est en partie lié aux limites de la logique (des propositions ou des prédicats) sur laquelle ils sont fondés.

Raisonnement par cas. Les approches à base de cas ou CBR [26] dérivent des travaux de [Schank et Abelson, 1977] sur la notion de script en tant que représentation des connaissances. Par opposition au courant de pensée cognitiviste, ces auteurs font l'hypothèse que notre connaissance générale des situations auxquelles nous sommes confrontés est enregistrée sous forme de scripts (p. ex. aller au restaurant ou consulter un médecin). Cependant,

25. Un panorama des applications des systèmes experts est disponible dans [Liao, 2005].
26. *Case-Based Reasoning*.

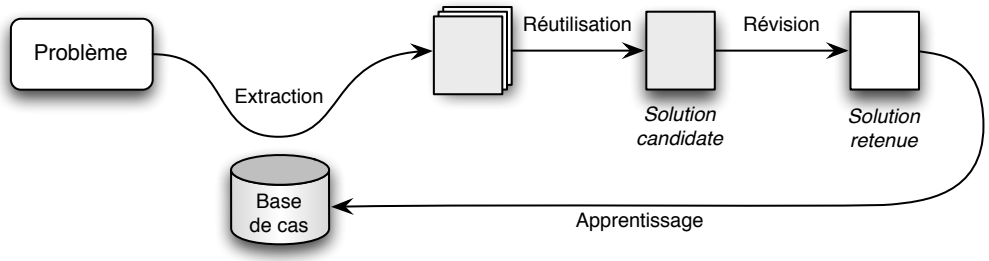


FIGURE 1.10: Cycle du raisonnement par cas (inspiré de [Aamodt et Plaza, 1994]).

l'expérience a montré qu'ils ne constituaient pas une théorie complète de représentation de la mémoire, du fait que certaines personnes confondaient souvent des événements ayant des scripts similaires (p. ex. : visite chez le dentiste ou chez le médecin). Néanmoins, le CBR permet de combler certains manques des SBC en représentant la connaissance en extension plutôt qu'en intention [Váncza, 1999]. En ingénierie, le raisonnement par cas préconise la réutilisation de solutions de conception déjà éprouvées. De nombreuses mises en œuvre du CBR ont vu le jour, notamment à partir du début des années quatre-vingt. Par exemple, le système de [Qin et Regli, 2000] utilise une approche par cas pour aider à la conception de roulements ; [Faltings et Sun, 1996] donnent un exemple d'utilisation du CBR en conception innovante (par réinterprétation des cas). Les approches de CBR suivent un cycle (cf. figure 1.10), qui nécessite très souvent l'intervention humaine, en particulier lors des phases de réutilisation et de révision. Ce cycle nécessite de pouvoir décrire les cas (problème, solution et résultat), les rechercher (indexation, comparaison, évaluation) et les adapter (transformation, dérivation). Le raisonnement par cas n'est possible que pour des conceptions appartenant à une même classe de problèmes. L'avantage est que la résolution est assimilée à de l'apprentissage, ce qui va vers une certaine évolutivité. Mais le succès dépend directement du volume de la base de cas. En outre, ce type d'approche limite la créativité dans la mesure où les nouvelles solutions se basent sur les anciennes.

Conception sous (ou avec) contraintes. Cette approche consiste à considérer la conception comme un problème de satisfaction de contraintes, ou CSP[27]. Dans cette approche, initialement proposée par [Simon, 1969], la conception est vue comme un cycle d'élaboration et de satisfaction de contraintes (cf. figure 1.11). Le problème étant au départ sous-contraint, l'ob-

27. *Constraint Satisfaction Problem*.

jectif du concepteur est de le transformer en un problème bien-contraint par ajout ou retrait de contraintes. La conception sous contraintes n'est donc pas un problème d'optimisation du fait qu'il est recherché une solution à un CSP, sans qu'elle ne soit nécessairement optimale [28]. Le problème de satisfaction d'un CSP étant NP-dur, des optimisations sont utilisées (filtrage, élagage ou décomposition). L'application des CSP à la CAO [29] se retrouve également à travers la modélisation descriptive : modélisation paramétrique ou variationnelle (cf. § 1.3.2.3) ; ou recherche de solutions à un problème de modélisation déclarative [30] [Le Roux et coll., 2001]. La modélisation déclarative vise à caractériser des solutions à partir de propriétés imprécises, principalement morphologiques, données par le concepteur [Lucas et coll., 1990]. Le système propose à l'utilisateur de faire son choix dans l'espace des solutions. La forme finale est obtenue de façon incrémentale via des cycles de description, génération et proposition. Ce type d'approche a connu de nombreuses applications, mais reste dédié à des domaines très particuliers [31], et le lien entre les propriétés et les fonctions n'est pas nécessairement trivial. Notons également que des méthodes de résolution des CSP (notamment numériques) peuvent aussi être utilisées en vue de résoudre des problèmes d'ingénierie mécanique [Chung et coll., 2000], ou de prendre en compte des contraintes géométriques liées à la fabrication du produit (cf. § 2.3.5.2). Néanmoins, afin d'être viables, ces approches doivent permettre d'exprimer le besoin en termes de contraintes, ce qui n'est à notre sens possible que pour des problèmes particuliers, ou à partir d'un certain stade de raffinement des fonctions. Comme le note [Váncza, 1999], les CSP ne sont pas une solution d'automatisation totale de la conception ; ils sont plutôt employés dans le but de maintenir la consistance de solutions partielles, de réduire les domaines des variables ou de vérifier la complétude de solutions définitives.

Systèmes multi-agents (SMA). Le concept d'agent, dont le développement est relativement récent, dérive des idées de Minsky dont l'hypothèse est que l'esprit est une société d'agents dont la collaboration fait émerger notre comportement, sans pour autant nécessiter de mécanisme de contrôle centralisé [Minsky, 1988]. Le principe est ici de distribuer les compé-

28. C'est là une définition s'opposant à celle de [Girard, 2001] (cf. page 8), l'optimalité ne pouvant être considérée comme une contrainte au sens des CSP.

29. Une étude complète de l'application des CSP à la CAO est disponible dans [Kwaiter et Gaildrat, 1998].

30. Les CSP ne sont pas une solution unique pour la modélisation déclarative (cf. approches évolutionnaires, page 25).

31. Par exemple, la génération de polyèdres [Martin et Martin, 1988], la génération automatique de courbes à partir de spécifications floues [Rossignol, 2001], la génération de modèles architecturaux à partir de propriétés d'ensoleillement [Hégron, 2002] ou l'agencement spatial d'éléments architecturaux [Medjdoub et Yannou, 2000].

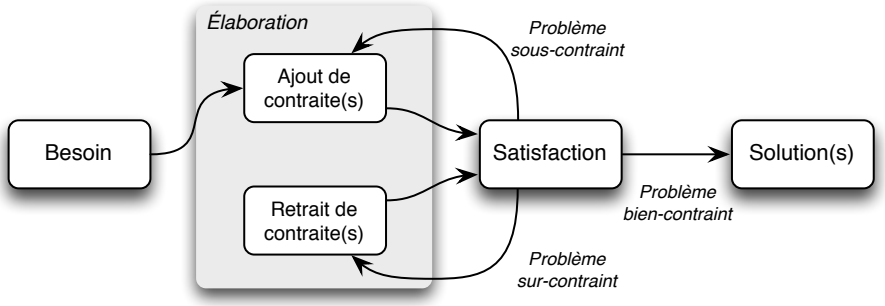
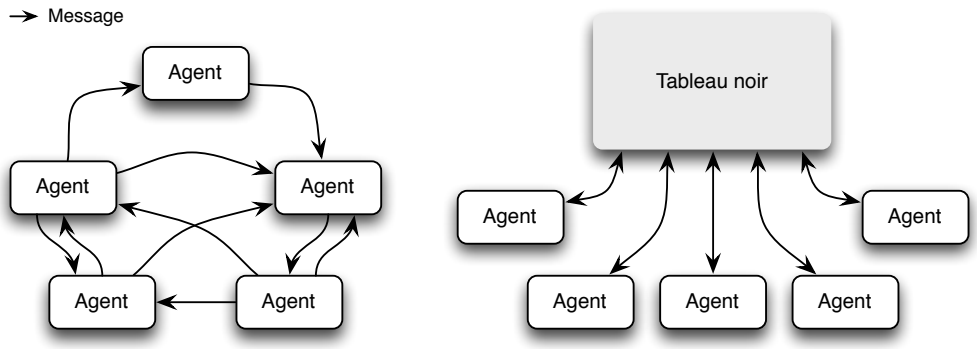


FIGURE 1.11: Cycle de conception sous contraintes.

tences de résolution du problème à plusieurs entités (humaines ou artificielles) compétentes sur un domaine particulier, ce qui tend vers une certaine « socialisation » des théories de l'IA [Drogoul, 2005]. Les agents disposent de connaissances relatives au monde extérieur et aux autres agents. Deux types d'agents sont distingués : les agents dits « cognitifs », dotés d'une représentation symbolique du monde ; et les agents dits « réactifs », qui sont influencés par des *stimuli*. Les agents peuvent interagir par envoi de messages (cf. figure 1.12a), ou via un tableau noir (cf. figure 1.12b). Les applications des SMA concernent principalement la simulation, la résolution de problèmes, et le domaine du génie logiciel [Haton et Haton, 2000]. Dans le domaine de la conception, ils peuvent donc être utilisés :

- en vue de simuler l'activité de conception. Par exemple, [Shakeri et Brown, 2004] utilisent un SMA pour « simuler » le processus de conception en vue de l'améliorer ;
- au même titre que les SBC, en tant que systèmes de résolution de problèmes de conception, afin de délimiter l'espace des solutions. Par exemple, [Grecu et Brown, 1996] ont développé un système de conception paramétrique de ressorts de compression hélicoïdaux. Les agents sont des SiFA *(Single Function Agents)* qui tendent à favoriser une fonction particulière du produit. D'autres applications portent par exemple sur la planification du processus de conception [Madhusudan, 2005], ou sur la notion de « raisonnement réfléchi » qui, pour chaque agent, consiste en une prise en compte du point de vue qu'ont les autres agents sur le problème [Brazier et coll., 2001] ;
- afin de fournir un support pour l'interopérabilité dans un contexte collaboratif [McAlinden et coll., 1998 ; Brown, 1998]. Ils servent alors d'enveloppes pour les systèmes existants.



(a) Envoi de messages (b) Tableau noir

FIGURE 1.12: Architecture d'un système multi-agents.

Un avantage des SMA est qu'il existe des solutions de calcul parallèle ou distribué particulièrement bien adaptées à leur mise en œuvre. L'autre avantage est qu'intrinsèquement, ils proposent une structure qui correspond bien aux problèmes multi-tâches, dont la conception fait partie. Nous développerons les aspects distribué et parallèle de la conception dans le § 1.4.3.

Apprentissage automatique. Dans ce type d'approche, le système a la charge de trouver la solution au problème, notamment à l'aide de la théorie de l'évolution, des algorithmes génétiques ou des réseaux neuronaux artificiels. Par exemple, [Rosenman, 1997] utilise une approche évolutionnaire en vue de générer des plans bidimensionnels représentant l'agencement des pièces d'un logement. [Jakiela et coll., 2000] utilisent une approche similaire pour générer une forme optimisée structurellement en regard de contraintes mécaniques. [Liu et coll., 2002] utilisent quant à eux un algorithme génétique combiné avec des agents pour générer différents *designs* pour un boîtier de téléphone portable. La construction de l'objet s'effectue en partant d'une forme de base puis en opérant par croisements ou combinaisons, le paramétrage consistant simplement à borner l'espace des solutions. Les approches d'apprentissage par renforcement consistent à apprendre un comportement à partir d'observations. Par exemple, [Zakarian et Kaiser, 1999] combinent un système expert avec des réseaux neuronaux pour assister le concepteur dans le paramétrage de moteurs électriques. L'atout de ces méthodes est leur possibilité d'être innovantes. Leur principal défaut est qu'afin d'être innovantes, leur espace de solutions doit être suffisamment vaste, ce qui pose un problème de performance : parcourir cet espace est particulièrement difficile lorsqu'il s'agit de formes géomé-

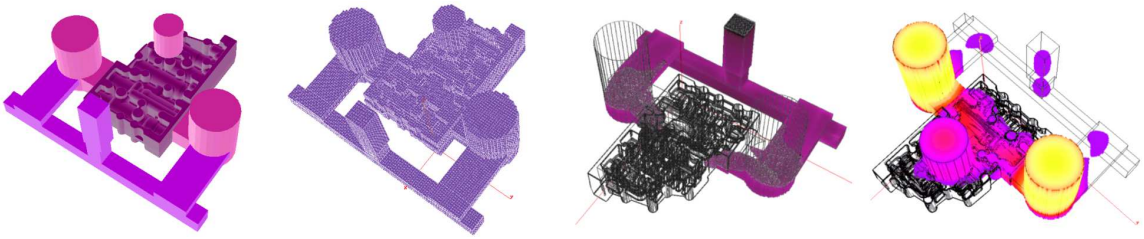
triques [Gardan, 2003].

À présent que nous avons parcouru les approches permettant de passer des fonctions à la forme, nous souhaitons aborder dans le paragraphe suivant, l'analyse d'ingénierie. Cette activité entretient un rapport particulier avec la conception du produit, en ce sens qu'elle permet la validation de ses fonctions.

1.2.3 Articulation avec l'analyse d'ingénierie

L'analyse d'ingénierie consiste à vérifier qu'un produit répond bien aux fonctions qui lui sont associées. Historiquement, cette activité s'est effectuée à travers la réalisation de prototypes ou de maquettes physiques. Elle est d'une importance capitale car elle consiste à vérifier que le comportement réel d'un produit est conforme à son comportement attendu, ce qui est rarement directement prévisible par le concepteur. Le comportement d'un produit peut en effet être complexe (résistance mécanique, cinématique, comportement dynamique du matériau, résistance thermique, résistance à l'usure…) car de très nombreux paramètres sont susceptibles de l'influencer (propriétés physiques du matériau et de l'environnement). Le champ d'étude du comportement physique d'un produit est donc très vaste. Une grande variété de modèles mathématiques entrant dans différentes catégories (mécanique des structures et des fluides, thermique, électromagnétique, thermodynamique, rhéologie…) ont été développés pour répondre à ces problématiques. Les méthodes de calcul sont également diverses : résistance des matériaux, méthode des éléments finis… Ces travaux théoriques couplés au développement de logiciels de simulation (cf. figure 1.13) et de moyens matériels de calcul (notamment distribué) contribuent à la disparition progressive de la maquette physique du produit. Par analogie avec le modèle du produit qui est aujourd'hui numérique, le prototypage, au départ physique, tend en effet à devenir virtuel, cette optimisation ayant déjà été amorcée par l'apparition du prototypage rapide. La qualité des produits en est aussi améliorée car de plus en plus d'aspects peuvent être simulés (p. ex. l'ergonomie), et simuler plusieurs fois est beaucoup moins coûteux en temps et en argent que de réaliser plusieurs tests réels.

L'expérience a montré que la simulation numérique avait un impact certain en termes de coût et de délais. Par exemple, l'utilisation exclusive de la simulation dans le projet X90 du groupe Renault, débuté en 2000, a permis la sortie en 2004 de la Dacia Logan, une voiture à 5 euros le kilo [Dacia, 2004]. Depuis une quinzaine d'années, la simulation se diffuse ainsi massivement dans l'industrie, ce qui la place aujourd'hui dans une position de prédominance.



(a) Modèle surfacique (b) Modèle volumique (c) Remplissage du moule (d) Solidification

FIGURE 1.13: Simulation de la réalisation d'une pièce en fonderie : le logiciel *Experto* du CRIF-WTCM [www-CRIF-WTCM].

[Pourroy, 1999] distingue deux usages différents de la simulation numérique :

le calcul de validation qui intervient à un stade avancé du développement du projet, afin de valider l'ensemble des choix de conception relativement aux spécifications fonctionnelles. On s'intéresse là au caractère absolu du résultat du calcul ;

l'aide au choix. Les concepteurs sont fréquemment confrontés à différentes alternatives de conception ou différents procédés de fabrication. L'analyse d'ingénierie a de ce point de vue un rôle d'aide au choix. C'est cette fois un résultat relatif qui est déterminant (comparaison des alternatives). Dans ce type d'utilisation, la précision du calcul importe donc beaucoup moins, et des modèles simplifiés peuvent alors être utilisés. Sur le plan pratique, le couplage entre logiciels de simulation et de CAO, proposé par certains éditeurs logiciels, est donc pertinent pour ce type d'utilisation.

Le calcul de validation vient donc remplacer la réalisation d'une maquette physique, tandis que l'aide au choix permet au concepteur de se baser sur des éléments plus tangibles. Il est à noter que la simulation numérique est également utilisée pour mettre en place le procédé de fabrication (p. ex. : détermination du retrait en fonderie, du lopin en forge…). Nous reviendrons sur ce point dans le chapitre 2.

Malgré les nombreux progrès effectués dans les méthodes numériques de simulation, il existe encore des cas où le passage par un prototype physique est inévitable. En effet,

- les modèles de calcul ou les lois de comportement sont parfois limités face à la complexité de la situation. C'est notamment le cas lorsque des phénomènes de natures différentes (p. ex. thermo-mécaniques) interviennent en même temps ;
- certains modèles théoriques sont trop peu fiables, du fait que des paramètres du phéno-

mène sont inconnus. Par exemple les phénomènes mécaniques de fatigue [32] et de fluage [33] sont encore mal compris, et les modèles existants sont imprécis, ce qui oblige les concepteurs à prévoir des facteurs de sécurité.

D'autres difficultés, technologiques, sont liées à l'utilisation des méthodes de simulation. Elles nécessitent souvent l'utilisation de maillages volumiques [34], alors que les modèles polyédriques utilisés en CAO sont généralement basés sur des surfaces. Il est donc indispensable d'effectuer un remaillage du modèle, afin qu'il soit exploitable pour la simulation. Cette opération, appelée « mise en données » [35], n'est pas simple car le maillage, pour donner de bons résultats, doit posséder certaines caractéristiques qui dépendent du cas traité [36]. Par exemple, lorsque deux pièces d'un mécanisme sont en contact, leurs maillages doivent être congruents [Gardan, 2005].

1.2.4 Conclusion

Dans cette section, nous avons étudié le processus de conception à travers les différentes activités qu'il engendre dans différents domaines. Nous avons effectué un panorama des moyens qui permettent de passer du besoin à la forme finale du produit. Ainsi, différentes approches du processus de conception, impulsées par une vision technologique ou théorique, de nature descriptive, prescriptive ou productive ont été présentées. Il n'existe pas à l'heure actuelle de méthode universelle permettant un passage automatisé entre besoin et fonctions, pas plus qu'entre fonctions et forme. En premier lieu, ceci est dû au fait que la compréhension du besoin par la machine est pour l'instant une question hors de portée du monde informatique [Pallez, 2000]. Une autre raison réside dans le constat qu'il semble extrêmement difficile de rationaliser l'ensemble des compétences nécessaires à la réalisation d'un produit. Par conséquent, bien souvent, les approches présentées ne constituent au mieux qu'un support sur lequel les concepteurs peuvent s'appuyer pour mener leur activité.

Après avoir passé en revue les approches du processus de conception, nous souhaitons présenter dans la section 1.3, les différents modèles de représentation informatique du produit, ainsi que les techniques qui permettent de les manipuler.

32. Répétition cyclique et prolongée d'une sollicitation mécanique.
33. Application prolongée d'une sollicitation mécanique constante.
34. Les maillages surfaciques, destinés à simplifier les calculs, ont tendance à être de moins en moins utilisés.
35. En réalité, la mise en données comprend aussi une étape de spécification des conditions aux limites (p. ex. propriétés des matériaux, contraintes mécaniques appliquées…).
36. Un panorama des problèmes relatifs à l'intégration de la simulation et de la CAO est disponible dans [Adams, 2006].

1.3 Modélisation d'un produit

La modélisation est l'activité d'élaboration d'un modèle informatique représentant le produit. L'approche de conception, décrite à la section 1.2, est centrée sur le problème de conception. Mais le modèle est une représentation de l'objet de la conception, et la technique de modélisation définit des outils permettant de manipuler cette représentation. La modélisation est donc davantage centrée sur la représentation du problème que sur sa résolution. Ces deux aspects (conception et modélisation) sont en principe indépendants, même si l'un est le support de l'autre.

Ainsi, dans cette section, nous présentons le modèle de représentation (§ 1.3.1), et la technique de modélisation (§ 1.3.2) qui donne les moyens de façonner le modèle.

1.3.1 Modèle géométrique et modèle produit

Le concept de modèle est très vaste, dépassant l'unique cadre de la conception. Il est utilisé dans de nombreuses sciences, qu'elles soient théoriques (mathématiques, physique théorique, logique), humaines (sciences économiques, politiques) ou pratiques (sciences de l'ingénieur) [Nouvel, 2002]. Dans le cadre de la CAO, un modèle est une représentation informatique du produit, utilisée primairement pour sa spécification spatiale ou comportementale, et pouvant en principe être exploité (au moins) aussi bien que le serait une instance réelle du produit [Gardan, 1991], par exemple pour la simulation ou la définition d'un processus de fabrication. [Bouleau, 2002] définit un « modèle en modélisation » en tant que « [...] l'élaboration par un ingénieur, un expert ou un bureau d'étude, d'une représentation destinée à éclairer et à préparer la décision d'un acteur spécifié ou de plusieurs acteurs en situation de dialogue. ». Ainsi, fondamentalement, un modèle est donc un moyen plus qu'une fin.

Historiquement, différents types de modèles ont successivement été utilisés en CAO. Nous décrivons ici brièvement les différents modèles géométriques ayant d'abord été proposés, puis la notion de modèle produit, plus récente, orientée vers une logique de métier, et donc vecteur potentiel d'une sémantique plus riche (p. ex. liée au mode de fabrication du produit).

1.3.1.1 Modèle géométrique

Un modèle *géométrique* est une représentation numérique et partielle d'un objet réel, « [...] ne prenant pas en compte ses aspects physiques, sauf sa forme » [Gardan, 1991]. Très

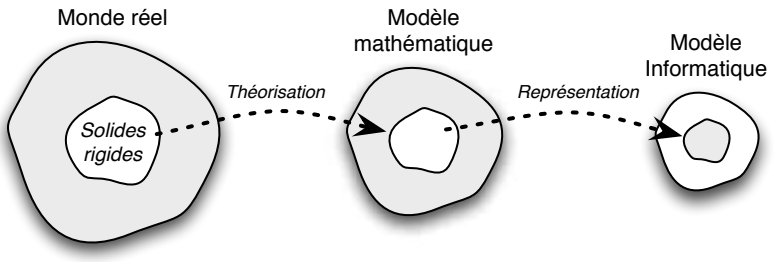


FIGURE 1.14: Paradigme général en modélisation géométrique.

tôt, les modèles géométriques ont été classés suivant qu'ils étaient bidimensionnels ou tridimensionnels, par analogie avec les branches de la géométrie plane et de la géométrie dans l'espace en mathématiques. Nous ne nous intéresserons pas ici aux modèles bidimensionnels, dont l'utilisation exclusive est aujourd'hui devenue obsolète [37]. Trois principales familles de modèles tridimensionnels peuvent être distinguées [Gardan, 2002] :

- les modèles « fil de fer » qui ne considèrent un objet qu'à travers les sommets et les arêtes qui le forment. Ils induisent des ambiguïtés [38], et ne sont plus utilisés ;
- les modèles surfaciques qui ne représentent l'objet qu'à travers une ou plusieurs surfaces le délimitant ;
- et les modèles de solides qui connaissent l'objet à travers la matière qui le compose (surface et intérieur sont donc connus).

En CAO, nous utilisons principalement des modèles de solides, bien qu'historiquement, les modèles surfaciques aient aussi été utilisés de façon autonome. Ces derniers sont aujourd'hui plutôt utilisés en combinaison avec les modèles de solides, ou intégrés en tant que sur-couche, pour modéliser les surfaces gauches. En CAO, ces solides sont implicitement non déformables, contrairement aux modèles dédiés à la simulation. Afin d'être viable, un modèle de solide doit être valide (correspondre à un objet réel), non ambigu, et permettre toute mesure pouvant être effectuée sur l'objet réel qu'il représente [Shapiro, 2001]. D'autres propriétés sont notables : finitude, homogénéité, partition de l'espace en deux classes d'équivalence.

Les modèles géométriques s'appuient sur un paradigme général consistant à partir d'un modèle mathématique représentant une partie de la réalité, pour en tirer une représentation

37. Ils se sont effacés dès lors que les modèles tridimensionnels ont eu acquis une certaine maturité, et qu'ils ont ainsi pu être réutilisés ailleurs qu'en conception, notamment pour la FAO ou la simulation. Ils restent encore aujourd'hui utilisés, mais uniquement dans le but de construire des objets tridimensionnels [Gardan, 2002].

38. Notamment visuelles, mais aussi lors des calculs (p. ex. celui du volume).

informatique (cf. figure 1.14). Les modèles informatiques qui ne sont pas valides au sens décrit ci-dessus, rendent possible la représentation d'éléments qui ne sont pas des images du modèle mathématique. À l'opposé, la puissance d'un modèle est sa capacité à représenter un grand nombre d'objets. Comme le montre [Gardan, 1991], le passage d'un modèle mathématique à une représentation numérique n'est pas simple. Les modèles numériques, en tant que représentations discrètes d'un espace continu, posent en effet des problèmes de précision liés aux erreurs d'approximation, en particulier lors des opérations géométriques ou des conversions. Ainsi, tout modèle n'est qu'une approximation. Deux catégories de modèles de représentation existent [Shapiro, 2001] : les modèles *explicites* qui conservent le résultat des opérations effectuées par l'utilisateur ; et les modèles *implicites* qui conservent la manière dont le résultat a pu être obtenu. Différentes solutions de représentation informatique, entrant dans l'une ou l'autre de ces deux catégories, ont été proposées : modèles par les frontières, modèles à historique de construction, modèles mathématiques, modèles paramétriques…

Modèles par les frontières. Les modèles par les frontières ou B-Rep [39], connaissent l'objet par l'intermédiaire de sa « peau » (sa limite). En réalité, la peau de l'objet est approximée, le plus souvent à l'aide de facettes planes, ce qui permet une représentation réaliste des polyèdres, et approchée pour les autres solides. Ce modèle enregistre des informations géométriques qui permettent le positionnement des objets, et des informations topologiques [40] qui relient les informations géométriques entre elles, afin de compléter la définition de la forme. Elles sont généralement stockées sous forme d'un graphe de type face, arrête, sommet (FAS). Ce principe est illustré sur la figure 1.15. La règle de Möbius [41] permet pour chaque face, de savoir de quel côté se trouve la matière. Cet artifice a ainsi permis aux B-Rep d'entrer dans la catégorie des modèles de solides. Les B-Rep jouent un rôle central dans les systèmes de CAO car ils jouissent d'avantages non négligeables, notamment en ce qui concerne la visualisation en temps réel. Leur inconvénient majeur provient des difficultés de modification qui les caractérise. Certaines techniques de modélisation (notamment modificatrices, cf. § 1.3.2) apportent néanmoins une réponse satisfaisante à ce problème.

39. *Boundary Representation*.
40. Voir à ce sujet l'annexe A qui traite de la topologie des modèles de représentation par les frontières.
41. La règle de Möbius stipule que chaque arête d'un solide non eulérien appartient à deux faces et que l'orientation de chaque arête s'inverse pour ses deux faces.

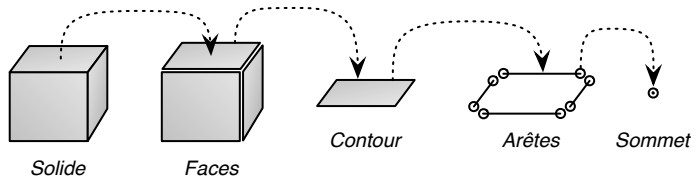


FIGURE 1.15: Exemple de modèle par les frontières.

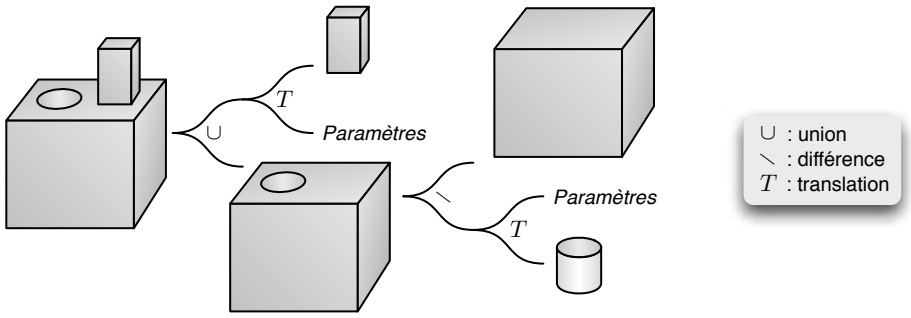


FIGURE 1.16: Exemple d'arbre de construction.

Modèles à historique de construction. Les modèles à historique de construction ou CSG [42], représentent l'objet à travers l'ensemble des opérations qui l'ont généré (cf. figure 1.16). Il s'agit souvent d'un arbre binaire dont les nœuds sont les opérations en question (opérations booléennes, transformations affines), et les feuilles des primitives géométriques paramétrables, généralement définies sous forme implicite. La modification est, avec ce type de modèle, facilitée par le fait qu'il suffit de réévaluer l'arbre de construction pour prendre en compte les changements opérés. De plus, le concepteur peut à tout moment revenir facilement sur ses choix, sans conséquences sur la qualité du modèle. En revanche, l'évaluation de ce type de modèle est lourde en temps de calcul, ce qui conduit souvent à maintenir en parallèle un modèle à facettes planes pour le dialogue et l'affichage [Gardan, 2002]. À l'instar des modèles B-Rep, les modèles CSG ne vérifient pas la propriété d'unicité (plusieurs modèles peuvent représenter le même objet).

Modèles mathématiques. Il s'agit essentiellement de modèles de courbes et de surfaces. Ces modèles découlent des travaux de Steven A. Coons, Pierre Bézier et Paul de Casteljau sur les

42. *Constructive Solid Geometry*.

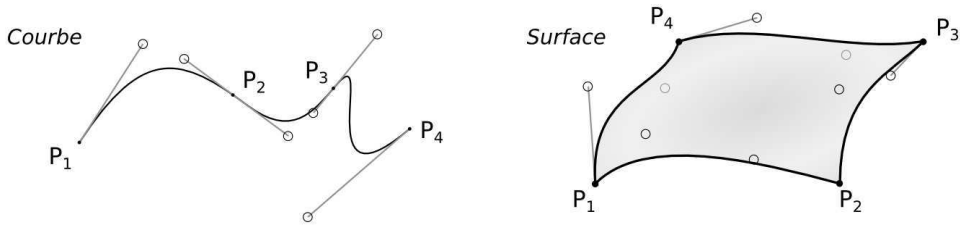


FIGURE 1.17: Courbe et surface à pôles.

courbes et surfaces à pôles [43] (cf. figure 1.17), dont les débuts remontent aux années cinquante. La quasi-totalité des modeleurs actuels utilisent encore ce type de modèle s'appuyant sur des polynômes paramétriques pour représenter les surfaces gauches, utiles dans de nombreux domaines d'application (automobile, aéronautique...). Les B-spline [44], puis les NURBS [45], plus récentes, ont permis de pallier certains manques [46] des courbes et surfaces de Bézier. Ces modèles sont aujourd'hui utilisés en combinaison avec d'autres modèles, de type B-Rep ou CSG, ce qui ne se fait pas sans poser quelques problèmes (calcul d'intersection de surfaces, approximations ou incohérences lors des opérations booléennes). Cela étant, l'approximation d'une courbe ou d'une surface paramétrique par des facettes planes est très naturelle (polygonalisation basée sur une variation discrète des paramètres).

Modèles paramétriques. Ces modèles permettent la création d'objets par un langage et des algorithmes paramétrés. Ce type de modèle se prête bien à la conception de familles de pièces connues, dont la géométrie est parfaitement maîtrisée ; en général des pièces normalisées (roulements, vis, engrenages...) [Gardan, 1991]. Il apparaît que par leur côté procédural, les modèles paramétriques rejoignent les modèles à historique de construction. La mise en œuvre du paramétrage peut aussi être effectuée par le dialogue qui est dual du langage [Gardan, 2002]. Ce point de vue a permis d'entrevoir une paramétrisation interactive des formes. Deux possibilités sont offertes [Gardan, 1991] : le paramétrage par structuration des informations, et la géométrie variationnelle (cf. § 1.3.2).

Autres modèles. Il existe d'autres types de modèles, moins utilisés en CAO :

43. Le terme point de contrôle est aussi utilisé.
44. *Basis spline*.
45. *Non-Uniform Rational B-splines*.
46. Notamment la nécessité d'avoir un contrôle local pour les B-spline, ou l'impossibilité de représenter les quadriques pour les NURBS.

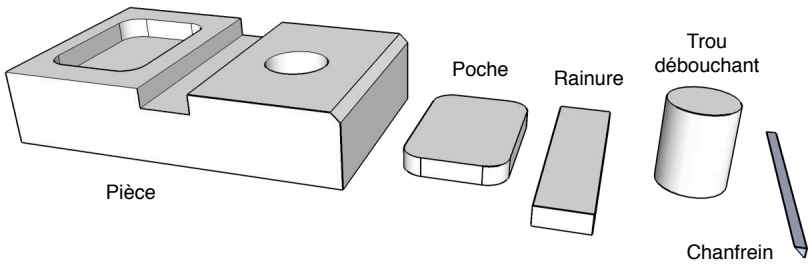


FIGURE 1.18: Exemples de caractéristiques.

– l'extrusion généralisée définit un objet comme étant l'ensemble des points intérieurs à un contour qui suit une trajectoire, éventuellement en se déformant. Les solides de révolution ou les formes prismatiques sont de bons candidats à ce type de représentation ;
– les modèles spatiaux divisent l'espace en portions élémentaires (des voxels [47]) pour lesquelles le système sait s'il y a présence de matière ou non. L'utilisation d'arbres octaux permet de minimiser la quantité d'informations à conserver. En ingénierie mécanique, sauf exception, ces modèles sont utilisés pour le calcul.

Dans le paragraphe suivant, nous présentons l'évolution du modèle géométrique vers le modèle produit, ce qui correspond à la mise en relation d'entités géométriques et d'informations sémantiques liées aux fonctions du produit.

1.3.1.2 Modèle produit

Le modèle géométrique a longtemps été le thème central des recherches et développements effectués pour les applications de CAO. Comme le note [Gardan, 1991], l'idée de ne pas limiter les modèles à la géométrie date pourtant de la fin des années soixante-dix. En effet, dans un contexte industriel, de nombreuses informations gravitent autour du modèle géométrique, en particulier des informations fonctionnelles ou liées à la production. Il paraît donc naturel de considérer qu'un modèle ne doive pas se limiter au pur aspect géométrique.

Dans l'idéal, un modèle devrait être capable de prendre en compte des informations très hétérogènes : géométriques, technologiques, liées à la précision, aux matériaux, aux assemblages, aux moyens de fabrication ou encore administratives [Gardan, 2003]. Un modèle prenant en compte ce type d'aspect est qualifié de *modèle produit* car il est tourné vers le produit dans sa globalité. C'est l'orientation prise par la plupart des systèmes de CAO à partir de la fin des

47. Contraction de *volumetric pixel* (pixel étant la contraction de *picture element*).

années quatre-vingt. Peu à peu, les modeleurs ont ainsi intégré des informations ayant une connotation métier [48]. Cette transition s'est effectuée à travers le paramétrage, et le développement des caractéristiques (*features*), concept introduit par [Grayer, 1976] dans le but de générer automatiquement des gammes d'usinage. Il a ensuite été développé intensivement et intégré à la CAO au cours des années quatre-vingt, notamment au travers des travaux de [Shah, 1991]. Une caractéristique est une abstraction morphologique ayant une sémantique propre, pouvant être manipulée et réutilisée en tant que telle, notamment grâce à des paramètres. Quelques exemples sont présentés sur la figure 1.18. Les caractéristiques permettent de relier les informations fonctionnelles et morphologiques [Feng et coll., 1996]. Elles sont classifiées et hiérarchisées sous forme de taxinomies. Ainsi, elles peuvent comporter des aspects génériques (communs à plusieurs métiers) ou spécifiques. Il est donc possible de traiter différemment un concept suivant le métier auquel il se rapporte. Par exemple, une fourche [49] en menuiserie sera considérée comme une rainure en mécanique. À l'inverse, il est aussi possible d'interpréter différemment des concepts portant le même nom. Par exemple, le terme « cannelure » n'a pas la même signification en architecture [50], en mécanique [51] ou dans l'industrie papetière [52]. Les caractéristiques sont souvent catégorisées suivant leur champ d'application :

- les caractéristiques de *forme* décrivent des formes d'un point de vue géométrique ;
- les caractéristiques de *fabrication* représentent des paramètres liés à un procédé ;
- les caractéristiques de *précision* indiquent des tolérances [53] ;
- les caractéristiques d'*assemblage* décrivent des liaisons entre pièces ;
- les caractéristiques *fonctionnelles* relient des informations fonctionnelles à la géométrie ;
- les caractéristiques de *matériau* décrivent la matière dont une pièce est constituée ;
- les caractéristiques de *calcul* identifient des éléments pertinents pour le calcul [54].

Généralement, les caractéristiques sont spécifiées en s'appuyant sur le modèle géométrique. Par exemple, un trou borgne ou une poche sont portés par une surface. La définition de cer-

48. Par métier, nous entendons l'exercice d'un savoir-faire dans un domaine d'ingénierie.
49. Élément utilisé en combinaison avec un tenon pour encastrer deux pièces perpendiculairement.
50. Sillon creusé sur le fût d'une colonne ou la face d'un pilastre [Druon, 2001]. Il s'agit donc d'un élément de décoration.
51. Rainure pratiquée longitudinalement dans une pièce mécanique cylindrique. Chaque pièce cannelée comporte un ensemble de cannelures réparties uniformément autour de son axe [www-GDT]. La cannelure permet donc d'accoupler deux pièces mécaniques.
52. Concernant le carton ondulé. La cannelure assure une liaison rigide des semelles. Elle joue le rôle d'entretoise, maintient un écartement constant [www-GDT]. La cannelure permet donc de rigidifier le carton.
53. Variations géométriques admissibles sur la pièce, une fois qu'elle aura été réalisée.
54. Par exemple, des zones potentielles de concentration de contraintes.

taines caractéristiques doit s'appuyer sur d'autres caractéristiques : par exemple, le trou de la figure 1.18 aurait pu être pratiqué sur le fond de la poche. En outre, la question du maintien de la cohérence est fondamentale. [Maculet et Daniel, 2003] considèrent que le modèle produit est constitué d'une juxtaposition de modèles spécifiques correspondant aux différents points de vue qu'ont les acteurs sur le produit. Les caractéristiques ont ainsi un rôle double :

- permettre aux acteurs de s'exprimer dans un langage métier, sémantiquement plus riche que celui de la géométrie ;
- supporter l'échange, la communication et la cohérence entre deux types de métiers. Le plus souvent, il s'agit d'une part des métiers de la conception qui utilisent des caractéristiques fonctionnelles, de forme, de précision ou d'assemblage, et d'autre part des métiers de la fabrication qui utilisent des caractéristiques de fabrication ou de matériau (p. ex. le CAPP [55]). Le concept de caractéristique peut ainsi jouer un rôle dans le cadre de la prise en compte de critères de fabrication au moment de la conception (cf. chapitre 2).

Pour finir, nous notons que ce concept est généralement exploité selon deux approches complémentaires [Han, 1996] :

1. L'extraction de caractéristiques qui consiste à rechercher des configurations particulières dans le modèle géométrique (de façon automatique ou interactive) afin de construire un modèle à base de caractéristiques dont on peut ensuite tirer parti ;

2. La conception par caractéristiques qui s'assimile à de la réutilisation d'entités technologiques par le logiciel de CAO.

Globalement, les modeleurs actuels gèrent le modèle produit de façon plus ou moins satisfaisante. La notion de caractéristique de forme y est habituellement bien intégrée (souvent à travers un arbre de construction), et certains logiciels permettent même la définition de caractéristiques de forme personnalisées. Cependant, les aspects supportés se limitent d'ordinaire aux caractéristiques de forme, d'assemblage ou de précision, et la cohérence entre les métiers est quasiment inexistante (il faut souvent recourir à d'autres outils).

Dans ce paragraphe, nous avons passé en revue des représentations du produit sémantiquement plus ou moins riches. Le paragraphe suivant présente les techniques de modélisation qui permettent de manipuler ces modèles.

55. *Computer-Aided Process Planning*.

1.3.2 Techniques de modélisation

Une technique de modélisation définit les moyens dont dispose le système pour répercuter les actions du concepteur sur le modèle. Les techniques de modélisation existantes peuvent être classifiées selon trois grandes catégories [Danesi, 2002] : les techniques constructives, modificatives et descriptives. À l'instar des modèles de représentation, l'utilisation qui en est faite est en réalité souvent hybride. Dans les paragraphes suivants, nous tentons néanmoins de les dissocier à des fins de clarté.

1.3.2.1 Modélisation constructive

Les techniques constructives sont ascendantes : le concepteur ne partant de rien, il doit utiliser des outils de génération (formes primitives, caractéristiques) et de combinaison (opérations booléennes, extrusions…) pour construire la pièce. À chaque étape, il peut ajouter de la matière (protusions) ou en retirer (dépressions). Les protusions et dépressions sont habituellement créées par l'application d'opérations booléennes d'union et de différence. Les CSG sont ainsi une solution de représentation privilégiée pour la modélisation constructive. La majorité des modeleurs de CAO actuels utilisent ce type de technique mais les opérations booléennes n'y sont plus utilisées directement, mais plutôt par l'intermédiaire de caractéristiques de forme. Les modeleurs paramétriques et variationnels (cf. plus bas) y trouvent des applications intéressantes, notamment à travers l'utilisation conjointe de *sketchers* [56] et d'opérateurs d'extrusion. L'inconvénient majeur des techniques constructives est qu'elles requièrent une forte expertise logicielle de la part du concepteur. Il y a deux raisons à cela :

- les outils de construction sont souvent très nombreux et varient d'un logiciel à l'autre ;
- la nature de l'approche constructive oblige le concepteur à se représenter mentalement l'ensemble des opérations géométriques nécessaires à l'obtention de la pièce. Son mode de réflexion s'en trouve donc perturbé.

1.3.2.2 Modélisation modificative

La métaphore de ce type d'approche est celle d'un sculpteur disposant d'un bloc de matière et d'outils pour la travailler. Le principe est donc de partir d'une pièce correspondant très grossièrement à l'objet à réaliser, puis de la raffiner par déformations consécutives. Celles-ci

56. Modeleurs d'esquisse.

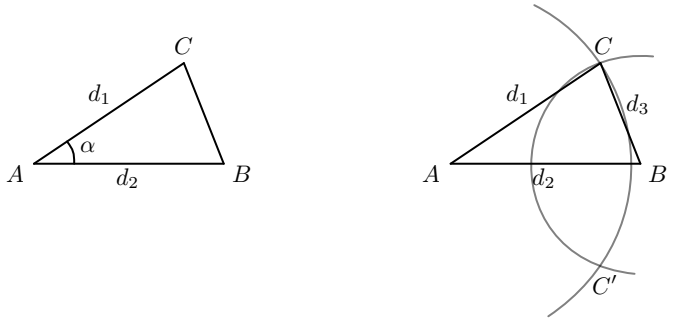


FIGURE 1.19: Différence entre problèmes paramétriques et variationnels (tiré de [Maculet et Daniel, 2003]).

peuvent être locales ou globales et s'appliquent aux surfaces de l'objet considéré. Du fait de l'aspect très interactif de ce type d'approche, les IHM y jouent un rôle important. Deux grandes familles de déformations sont disponibles :

- les déformations normalisées qui s'appliquent au modèle implicite de la pièce. L'exemple typique est celui de la manipulation de points de contrôle pour paramétrer des courbes et surfaces gauches. L'inconvénient de cette façon d'opérer est qu'elle est peu intuitive, nécessitant une connaissance (au moins partielle) du modèle mathématique sous-jacent ;
- les déformations non normalisées qui s'appliquent au modèle explicite de la pièce, en général un B-Rep. Ce type d'approche a le mérite de s'ouvrir à des méthodes d'interaction plus intuitives.

Bien que la modélisation modificative soit traditionnellement plutôt utilisée en synthèse d'images ou en animation, il paraît prometteur de l'appliquer à la CAO [Danesi, 2002]. En effet, il semble très intéressant de faire coïncider le raffinement de la forme avec celui du problème de conception. Le projet DIJA [Danesi et coll., 2002 ; Denis et coll., 2004] développé par l'équipe CMCAO a pour vocation d'explorer cette piste à travers le développement d'un système de CAO distribué, simultané et interactif. Ce système est centré sur une approche dite « synthétique » de la modélisation. Nous reviendrons plus en détail sur les différents aspects du système DIJA au chapitre 3.

1.3.2.3 Modélisation descriptive

La modélisation descriptive ne repose pas, dans son principe, sur une séquence déterminée d'opérations. Le concepteur décrit son but de manière implicite, et le système met en œuvre

des moyens permettant d'en déduire une représentation explicite. Les approches de modélisation paramétrique et variationnelle peuvent être considérées comme des applications des CSP (cf. § 1.2.2.4) à la modélisation géométrique descriptive [57]. Les contraintes correspondent alors à des dimensions et à des relations entre entités géométriques. Un problème est dit paramétrique s'il peut être décomposé en un ensemble de sous-problèmes ne définissant qu'une seule variable par rapport aux autres [Hoffmann et Joan-Arinyo, 2002], ce qui induit une résolution séquentielle. Ce type d'approche est très utilisé en CAO, notamment dans tous les modeleurs à historique de construction [Lesage, 2002]. À l'inverse, un problème est dit variationnel s'il nécessite que plusieurs variables soient placées simultanément en relation les unes par rapport aux autres [Hoffmann et Joan-Arinyo, 2002]. La figure 1.19 illustre la différence entre les deux approches. Le triangle de gauche peut être construit par une approche paramétrique, ce qui n'est pas le cas de celui de droite, car les distances d_1 et d_3 doivent être considérées simultanément. Les problèmes variationnels étant plus difficiles que les problèmes paramétriques, en modélisation géométrique, ils sont surtout exploités dans le plan [58], notamment à travers les *sketchers* qui apportent une aide interactive en « devinant » l'intention du concepteur lors de la construction de profils ou de contours plans. Ces deux techniques sont bien adaptées à la prise en compte de la cotation. L'approche de modélisation déclarative (cf. § 1.2.2.4) pourrait être qualifiée de descriptive. Elle n'est pourtant pas, à notre sens, une technique de modélisation car elle manipule des concepts et des propriétés abstraits, et apporte directement une solution au problème de conception, le modèle sous-jacent n'étant pas fixé. La table 1.1 présente les entités manipulées ainsi que le niveau d'abstraction dans chacune des modélisations que nous venons d'aborder. Certaines approches, également descriptives, permettent, à partir de croquis effectués par l'utilisateur, de déduire un modèle tridimensionnel [Danesi, 2002]. Elles peuvent jouer un rôle intéressant en conception, en particulier du point de vue du *design*.

1.3.3 Conclusion

Dans cette section, nous avons présenté les différents aspects relevant de la modélisation, principalement géométrique, d'un produit. Nous avons aussi montré l'apport du concept générique de caractéristique, permettant la définition et la réutilisation d'unités ayant une di-

57. Bien qu'il existe d'autres méthodes de résolution, différentes de celles des CSP [Maculet et Daniel, 2003 ; Lesage, 2002].

58. Bien qu'il existe des travaux concernant les approches variationnelles dans l'espace [Chung et coll., 2000 ; Lesage, 2002].

59. *Geometrical Constraint Satisfaction Problem*.

TABLE 1.1: Entités manipulées en modélisation sous-contraintes (d'après [Maculet et Daniel, 2003]).

	Type de modélisation		Niveau d'abstraction	Entités manipulées
CSP	Paramétrique		0	variables mathématiques
GCSP[59]		Variationnelle	1	éléments géométriques
			2	caractéristiques de forme
	Déclarative		3	concepts, propriétés

mension métier. Cependant, nous avons noté que le lien entre caractéristiques de forme et de fabrication était souvent difficile à établir, ce problème étant le reflet de différences essentielles existant entre les activités de conception et de fabrication.

Comme évoqué au début de ce chapitre, un enrichissement des méthodes, des modèles et des techniques de modélisation est nécessaire. Nous tenterons d'en expliquer les tenants et les aboutissants dans la section 1.4.

1.4 Vers le modèle global

Depuis une quinzaine d'années déjà, de plus en plus d'entreprises tendent à passer du mode d'organisation traditionnel (séquentiel, systémique, hiérarchisé) à un mode d'organisation parallèle, simultané et distribué. Il y a plusieurs raisons à cela. La recherche d'une meilleure compétitivité en est la raison primaire, mais c'est la réunion de groupes d'entreprises autour de projets communs et le développement du concept d'entreprise étendue qui en a été le véritable catalyseur.

Ainsi, les industriels recherchent des solutions de plus en plus performantes, ce qui passe par la réduction des coûts, l'amélioration de la qualité, et la réduction des délais de mise sur le marché. Ils sont dès lors résolus à détecter les problèmes et à faire intervenir les corrections au plus tôt dans le cycle de développement du produit. Les premières étapes de la conception sont en effet les plus déterminantes pour le coût du produit, sa longévité et son adaptation aux exigences fonctionnelles [Ullman, 2002]. La figure 1.20 illustre le fait que les $4/5$ du coût global d'un produit sont engagés lors des phases de développement, alors même que seulement $1/5$ de ce coût est effectivement dépensé. Sans doute pourrions-nous faire un constat similaire concernant les délais de mise sur le marché. Ainsi, plus la définition du produit se précise, plus les

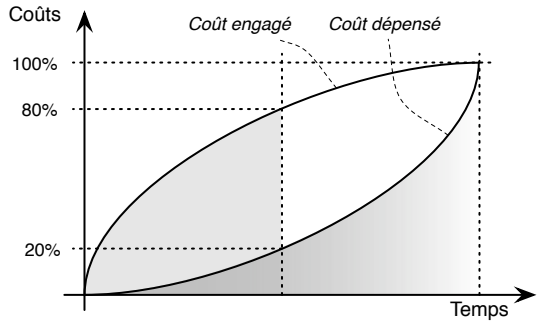


FIGURE 1.20: Diagramme de Pareto.

retours en arrière sont difficiles et donc coûteux. En outre, les phases en aval de la conception imposent de plus en plus de contraintes, ce qui vient renforcer cette tendance. Prenons-en pour témoin les questions environnementales sur lesquelles la législation se montre de plus en plus présente [Barnabé et coll., 2003].

Par ailleurs, la complexification et la diversification des produits ainsi que le contexte de compétitivité ont mené les industriels à se spécialiser davantage, ce qui a conduit à de nouvelles organisations où l'entreprise, seule, n'est plus l'unique acteur du développement d'un produit et de sa réussite. Ces organisations ne sont plus centrées sur l'entreprise, ses compétences et sa propre vision du produit, mais sur le projet, unité fédératrice à laquelle contribuent activement de nombreux intervenants (donneurs d'ordre, sous-traitants, fournisseurs…). Ainsi, l'entreprise ne doit plus seulement être considérée en tant que telle, mais également à travers les échanges et collaborations qu'elle entretient avec ses partenaires – c'est ainsi que l'on parle d'*entreprise étendue*.

Toutes ces mutations organisationnelles induisent des changements dans les méthodes de travail, qui se traduisent par un besoin d'évolution des outils informatiques, et en particulier des outils de CAO. Pour que l'adaptation soit réussie, tous les nouveaux facteurs doivent être intégrés au modèle produit. C'est en ce sens que l'on se dirige peu à peu vers la définition et l'utilisation d'un modèle désormais qualifié de *global*.

Dans la suite de cette section, nous décrivons les évolutions envisagées d'un point de vue général à travers la présentation du concept de PLM *(Product Lifecycle Management)* (§ 1.4.1), puis nous focalisons sur trois aspects fondamentaux en ce qui concerne la conception intégrée de produits : l'ingénierie simultanée (§ 1.4.2), le travail collaboratif (§ 1.4.3) et la gestion des

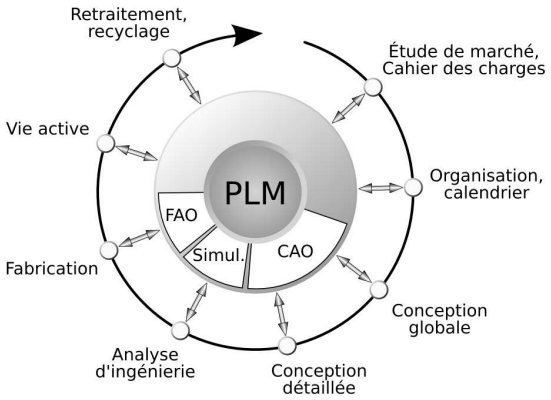


FIGURE 1.21: Gestion du cycle de vie du produit (PLM).

connaissances (§ 1.4.4).

1.4.1 Gestion du cycle de vie d'un produit (PLM)

Un certain nombre de méthodes permettant d'agir pour améliorer le trio « coûts, qualité, délais » ont progressivement vu le jour [Decreuse et Feschotte, 1998]. Par exemple, la réduction des coûts peut être facilitée par l'étalonnage concurrentiel *(benchmarking)*, consistant à positionner le produit par rapport à la concurrence. La qualité peut être améliorée par l'utilisation de méthodes telles le TQM [60] (cercles de qualité pluridisciplinaires), le QFD [61] ou la mise en conformité aux standards de l'ISO [62] (certifications de la série ISO 9000). Les délais peuvent être réduits par le « *just in time* » (réduction des stocks), le DFMA [63] (réduction de la complexité des produits en regard de critères de fabrication) ou l'ingénierie simultanée.

À la base, la plupart de ces méthodes sont plutôt informelles, ce qui rend leur application tributaire d'une forte sensibilisation des acteurs des différents métiers, condition traditionnellement difficile à établir, et encore plus dans un contexte distribué. Les outils informatiques ont donc clairement un rôle important à jouer de ce point de vue.

C'est dans ce cadre général qu'émerge le concept de PLM qui consiste en une gestion informatique optimale de toutes les étapes de la vie d'un produit, depuis l'expression du besoin jusqu'au retrait du marché (voir figure 1.21). Ce concept est très vaste car il se rapporte en principe à tous les secteurs gravitant autour du produit. Sont donc concernés les décideurs, les

60. *Total Quality Management*.
61. *Quality Function Deployment*.
62. *International Organization for Standardization*.
63. *Design for Manufacture and Assembly*.

gestionnaires, les concepteurs, les fabricants, mais aussi les fournisseurs, les sous-traitants, les clients, les vendeurs, les techniciens de maintenance... Selon [CIMdata, Inc., 2002], les concepts essentiels caractérisant le PLM sont :

- une utilisation universelle et un accès sécurisé et authentifié aux informations définissant le produit ;
- un maintien de l'intégrité de cette définition et des informations connexes, tout au long des cycles de vie du produit et de la chaîne de fabrication ;
- et une gestion et une maintenance des processus métier utilisés pour créer, gérer, diffuser, partager et exploiter l'information.

Par conséquent, le PLM n'est pas une technologie particulière mais plutôt une approche globale [CIMdata, Inc., 2002]. Sa mise en place est une tâche complexe qui requiert une interopérabilité entre tous les acteurs intervenant dans la vie du produit. Les outils informatiques doivent donc être présents et opérationnels à chaque étape, et favoriser les échanges entre acteurs intervenant dans une même phase de vie, mais aussi et surtout entre acteurs de compétences différentes. Cela implique d'intégrer les nouvelles contraintes organisationnelles des entreprises (distribution géographique, distribution des compétences, confidentialité, etc.).

Ainsi, le PLM sert d'assise à différentes approches d'optimisation du cycle de vie, dont font partie l'ingénierie simultanée, le travail collaboratif et la gestion des connaissances (qui sont présentés ci-après). Nous considérons que les trois points suivants sont fondamentaux pour la mise en place d'une démarche de PLM :

1. Faire en sorte que les systèmes informatiques offrent une interopérabilité à la fois syntaxique et sémantique (standards, ontologies...) ;
2. Rendre possible la communication entre les différents acteurs d'un projet, même s'ils sont distants physiquement et même s'ils ne partagent pas le même langage métier ;
3. Permettre de capitaliser et de réutiliser la connaissance, ce qui va de pair avec la traçabilité des décisions de conception et de leurs motivations.

Un certain nombre de travaux tentent de répondre aux défis posés dans ces domaines [Subrahmanian et coll., 2005 ; Kim et coll., 2006 ; Kesavadas et coll., 2005 ; Rachuri et coll., 2005 ; Mervyn et coll., 2004 ; Xue et Yang, 2004 ; Gomes et coll., 2005]. Mais la notion de confidentialité, la différence du langage et des préoccupations des différents acteurs métier, la séparation géographique et l'éclatement organisationnel sont autant de contraintes

qui rendent encore plus complexe une problématique qui l'est déjà par essence.

Sur la figure 1.21, la flèche circulaire représente un processus de conception tel qu'il serait idéalement, semblable à une stricte succession de tâches. Or dans la réalité, comme nous l'avons noté à plusieurs reprises dans la section 1.2, le développement d'un produit nécessite de nombreux rebouclages. Par exemple, si, lors de la phase d'analyse d'ingénierie, il est déterminé qu'une pièce ne satisfait pas une contrainte fonctionnelle, la spécification du produit devra être révisée afin de prendre en compte ce fait. Le PLM est sensé fournir un support optimal pour ces boucles de réitération et de reformulation du fait qu'il permet aux intervenants de concilier leurs points de vue sur le produit en partageant leurs compétences respectives. Un grand nombre d'outils informatiques devraient être en mesure de contribuer au PLM, tels les outils de travail collaboratif, les SGDT [64], la CAO, les logiciels de simulation, de FAO, de GPAO [65], mais aussi les PGI [66], le SCM [67], la GRC [68]... Cependant, bien souvent, les formats de représentation usités sont disparates, ce qui pose des problèmes d'interopérabilité syntaxique ; [Decreuse et Feschotte, 1998] n'hésitent pas à parler de « tour de Babel des outils logiciels » pour pointer ces manques en terme d'utilisation de standards. Mais afin que la communication puisse exister, l'interopérabilité doit de surcroît être sémantique, ce qui, comme nous le verrons dans le chapitre 2, pose d'autres problèmes, plus profonds.

Dans le paragraphe suivant, nous insistons sur l'aspect pluridisciplinaire de la conception à travers le concept d'ingénierie simultanée.

1.4.2 Ingénierie simultanée

L'ingénierie simultanée est à opposer à l'organisation séquentielle de la conception héritée du taylorisme où le développement du produit est vu comme une succession d'étapes indépendantes. Les approches séquentielles se caractérisent par un double cloisonnement. Le premier, horizontal, provient du découpage par activités ou compétences : chaque service fonctionne de manière autonome, en faisant abstraction des contraintes des autres services. Le second, vertical, oblige les informations à remonter dans la hiérarchie afin de pouvoir être répercutées. Cet isolement entre les métiers donne lieu à une détection tardive des conflits (ralentissement des

64. Système de Gestion de Données Techniques.
65. Gestion de Production Assistée par Ordinateur.
66. Progiciel de Gestion Intégré.
67. *Supply Chain Management*.
68. Gestion de la Relation avec la Clientèle.

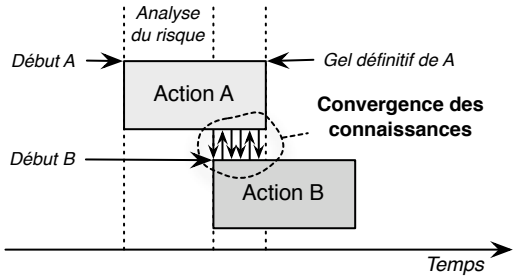


FIGURE 1.22: Principe du gel progressif des activités (tiré de [Levan, 2004]).

rebouclages), qui est elle-même une source d'incohérences [69] [Decreuse et Feschotte, 1998].

À l'inverse, l'ingénierie simultanée consiste à faire intervenir les différents corps de métier dès le début du développement, ce qui se caractérise par un chevauchement des phases de vie. Elle est née dans les années quatre-vingt aux États-Unis mais s'est inspirée des modes de travail de l'industrie automobile japonaise (notamment Toyota et Honda), qui faisait intervenir des comités multi-fonctionnels et un coordinateur de projet lors du développement de nouveaux véhicules. Elle est d'ailleurs souvent associée à la notion de projet, même si celle-ci est en réalité plus large et plus ancienne (elle remonte aux années cinquante) [Poveda, 2001]. Un projet peut en effet être mené dans une optique exclusive de coordination, et donc sans pour autant impliquer une coopération directe, aspect fondamental en ingénierie simultanée.

Le traitement transversal des informations permet de considérer un maximum de contraintes au plus tôt, en passant outre les méandres de la hiérarchie. De plus, l'approche simultanée favorise l'innovation en synthétisant les compétences et les idées d'acteurs de sensibilités différentes, et valorise ainsi le travail des services en aval de la conception. L'ingénierie simultanée peut être qualifiée de parallèle (chevauchement temporel), concourante (objectif commun), pluridisciplinaire (partage des compétences) et multi-entreprises (entreprise étendue). Elle entretient une relation forte avec la notion de projet, vouée à catalyser la collaboration. Des revues de projet périodiques permettent de faire le point et de confronter les idées des différents spécialistes. Cela fait généralement apparaître des antagonismes dûs aux divergences des contraintes propres à chaque domaine. La conciliation et le compromis sont donc nécessaires pour aboutir à un choix qui soit globalement optimal. La figure 1.22 illustre ce principe de « gel progressif » des activités, nécessitant une certaine convergence des connaissances.

69. Les spécifications du produit utilisées en production sont ainsi souvent en retard par rapport à celles définies en conception.

[Poveda, 2001] indique quatre archétypes d'organisation de projet :

- la coordination hiérarchique rend le projet très fortement dépendant de la volonté des différents chefs de service. Cette organisation est la moins favorable au projet ;
- l'existence d'un coordinateur permet une meilleure prise en compte des préoccupations relatives au projet ;
- la nomination d'un directeur de projet, donne au projet plus d'importance encore en le plaçant au même niveau hiérarchique que les autres services ;
- enfin, l'organisation en « plateau projet » isole définitivement le projet des autres organes de l'entreprise. Les acteurs sont alors uniquement dévolus au projet.

Le plateau projet, forme la plus aboutie, a été adopté par certains grands groupes dans les années quatre-vingt dix. Ce principe a permis d'ancrer une plus forte dynamique de collaboration dans les pratiques des industriels occidentaux, culturellement moins enclins à la coopération que leurs homologues asiatiques [Levan, 2004]. Par exemple, chez Renault, les activités de recherche et de développement ont été délocalisées sur un site unique, le *Technocentre*, situé en région parisienne et rassemblant environ 11 000 personnes, dont 3 000 extérieures à Renault.

Si le plateau projet a d'abord été physique, l'apparition des outils de travail distant, permettant une économie notable sur les frais de déplacement, a fait émerger un nouveau concept qui est aujourd'hui prépondérant : le plateau projet virtuel. Celui-ci est voué à permettre aux différents acteurs de collaborer en se passant de la proximité géographique. Ainsi, les outils de travail collaboratif ont pu contribuer à une évolution des pratiques d'ingénierie simultanée.

1.4.3 Travail collaboratif

Les technologies de l'information et de la communication (TIC) sont utilisées depuis déjà longtemps en tant que support au travail. Mais l'avènement des réseaux a entrouvert de nouvelles possibilités en la matière, en particulier à travers la multiplication des outils de travail collaboratif (TC). Les préceptes du travail collaboratif ont été introduits par [Engelbart, 1962], conceptualisés par [Johnson-Lenz et Johnson-Lenz, 1980] qui ont inventé le terme de *groupware*[70] (plus tard traduit par « collecticiel »), puis approfondis par la communauté du CSCW[71] à partir du milieu des années quatre-vingt. Le TC utilise des ressources

70. Ils définissent le *groupware* comme l'ensemble des processus et procédures d'un groupe de travail devant atteindre un objectif particulier, plus les logiciels conçus pour faciliter ce travail de groupe [Johnson-Lenz et Johnson-Lenz, 1989].
71. *Computer-Supported Cooperative Work*.

TABLE 1.2: Matrice de Robert Johansen [Johansen et coll., 1988].

Espaces / Temps	Identiques (lieu unique)	Différents (distribué)
Identiques (communication synchrone)	Interactions en face à face : écrans partagés ; tableaux électroniques ; support à la décision en groupe…	Interactions à distance : systèmes de bureau partagé ; édition collaborative ; visioconférence ; partage d'applications…
Différents (communication asynchrone)	Espaces et tâches communs : salles d'équipe ; affichages de groupe ; collecticiels de travail différé ; gestion de projet…	Communication et coordination : messagerie électronique ; conférences asynchrones ; *workflows* ; contrôle de version ; planificateurs de réunions ; hypertextes collaboratifs…

TABLE 1.3: Collaboration et éloignement physique (tiré de [Levan, 2004]).

Localisation des bureaux	Probabilité de collaboration
Dans le même couloir	10.3 %
Sur le même étage	1.9 %
Dans des étages différents	0.3 %
Dans des immeubles différents	0.4 %

informatiques dans le contexte d'un projet réalisé par les membres d'un groupe de travail reliés en réseau [www-GDT]. Il se caractérise par une action selon trois axes essentiels : la communication, la coopération et la coordination, *alias* « les trois C ».

Même si les outils informatiques de travail collaboratif se doivent d'apporter un soutien au travail du groupe, ils ne peuvent en aucun cas exclure la nécessité pour ses membres d'avoir et de cultiver des compétences collaboratives. La notion de TC a ainsi une dimension humaine et organisationnelle, et une dimension technique. Les approches de travail en groupe sont souvent classifiées par l'intermédiaire de la matrice de Johansen (cf. table 1.2) [Johansen et coll., 1988]. Celle-ci les situe selon deux axes : l'axe synchrone-asynchrone qui détermine la répartition de la collaboration dans le temps ; et l'axe distribué-localisé qui définit sa répartition dans l'espace.

Le TC permet en principe de briser les contraintes temporelles et spatiales, ce qui se traduit par des gains significatifs en termes de temps et d'argent. Les outils de TC permettent aussi de favoriser la coopération. En effet, avec les modes de communication directe, plus l'éloignement géographique entre deux acteurs est grand, moins les interactions collaboratives surviennent. Une étude menée par [Allen, 1984] a montré qu'au delà de trente mètres, le taux d'interaction

collaborative est le même qu'à mille cinq cents mètres (cf. table 1.3). Ce problème peut au moins partiellement être compensé par l'utilisation des réseaux. Afin de transposer certains aspects du monde du travail dans les espaces numériques, les collecticiels doivent mettre en place des métaphores évoluées dont voici quelques exemples [Levan, 2004] :

- la salle virtuelle (non limitée dans l'espace) ;
- la réunion virtuelle qui s'affranchit des contraintes de distance (et éventuellement de temps en mode asynchrone) ;
- l'armoire virtuelle qui peut contenir un nombre quelconque de « tiroirs » (ce qui correspond à la notion de répertoire partagé ou de dossier virtuel) ;
- le calendrier numérique qui permet de gérer un grand nombre de ressources à la fois et qui est utilisé afin de coordonner le groupe et ses accès à des ressources partagées ;
- le bloc-notes numérique (liste de tâches, croquis...).

Dans le cadre du développement de produits, toutes ces métaphores pourraient être considérées comme participant à celle, plus générale, de plateau projet virtuel (cf. § 1.4.2).

Le degré d'interaction collaborative supporté varie en fonction des outils. Ainsi, les outils de TC peuvent être distingués suivant le degré d'assistance qu'ils apportent du point de vue de la coopération [Gardan, 2005] :

les outils de communication sont basés sur les notions d'échange et de langage (écrit, oral), et sont aujourd'hui maîtrisés : téléconférence, clavardage[72], messagerie électronique... Ils peuvent être utilisés dans des contextes très variés. De par la souplesse de leur mise en œuvre, ils sont maintenant bien implantés dans les pratiques industrielles ;

les outils de coordination permettent une gestion des activités du groupe et un partage structuré des données. Dans cette catégorie, nous trouvons donc les PGI, les SGDT (partage de documents techniques, contrôle de version, gestion d'assemblages en CAO) ou les portails collaboratifs offrant par exemple des possibilités de synchronisation de tâches (routage, chaînes de validation, procédures qualité) ou de notification automatique, mis en place grâce aux *workflows*[73]. Les *workflows* sont aussi utilisés dans le cadre de la gestion documentaire, de la GRC (gestion commerciale, *marketing*) et de la CRM[74] [Pontacq, 2000]. Les outils de coordination permettent une meilleure maîtrise des

72. Terme d'origine québécoise donné en équivalent au terme anglais *chat*.
73. Un *workflow* est un enchaînement ordonné de tâches et de procédures, selon des règles prédéfinies, pour la réalisation d'un produit ou d'un service [Pontacq, 2000].
74. *Customer Relationship Management*.

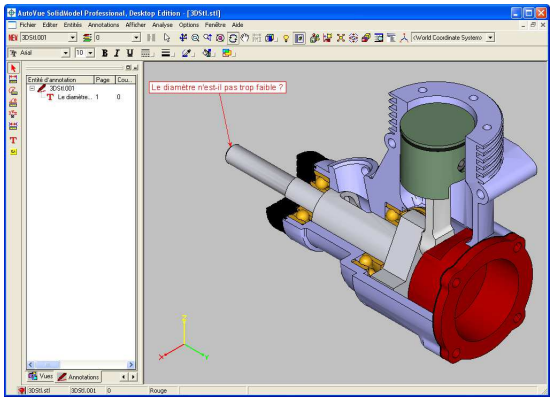


FIGURE 1.23: L'outil de visualisation et d'annotation *Autovue* [www-Cimmetry].

processus métier ;

les outils de collaboration offrent une plus grande intégration avec les concepts métier. Ici, les interactions ne se limitent plus à la gestion des données et au routage, mais consistent à partager des outils métier dans une optique de travail commun. Entrent donc dans cette catégorie les outils de partage d'application, et plus spécifiquement dans les domaines de la productique, les outils de visualisation et d'annotation de modèles géométriques [75] (cf. figure 1.23), ainsi que les outils de co-conception. L'utilisation de ces outils, en particulier de co-conception, se fait encore assez rare dans l'industrie.

Les problématiques posées par la conception collaborative peuvent être entrevues au travers de trois questions simples :

Qui ? La collaboration est possible entre différents concepteurs ou bien entre concepteurs et fabricants, apportant dans le second cas un support pour l'ingénierie simultanée ;

Quand ? Les modes de modélisation synchrone et asynchrone sont possibles. La co-conception est en général synchrone lorsqu'elle concerne un composant unique, et asynchrone lorsqu'elle se rapporte à un assemblage ;

Où ? Aujourd'hui, la conception est souvent répartie sur plusieurs sites. Cela pose des problèmes de confidentialité, d'intégrité, de vitesse de transfert et de gestion des conflits [Esposito, 2002].

75. Par exemple, le logiciel *Autovue* [www-Cimmetry] permet de partager la visualisation d'un modèle géométrique, avec la possibilité de l'annoter en temps-réel (voir figure 1.23). Ce logiciel supporte un grand nombre de formats natifs, ce qui l'affranchit des problèmes d'interopérabilité.

Bien sûr, tous ces aspects sont complémentaires et interreliés. Du fait de la bande passante limitée de l'Internet, les outils de visualisation nécessitent l'utilisation d'approches de transfert progressif des données *(3D streaming)* initialement utilisées pour les flux audio et vidéo. Deux principes opposés et complémentaires sont utilisés pour obtenir un transfert rapide des données géométriques [Li et coll., 2005] : la simplification qui consiste à réduire la complexité du modèle ; et le raffinement qui consiste à augmenter progressivement son niveau de détail. Dans les approches de co-conception, les protagonistes se voient allouer un accès partiel au modèle, avec un découpage temporel ou géométrique. Aussi, le modèle géométrique peut être centralisé sur un serveur, répliqué ou semi-répliqué. Trois architectures de réseau typiques permettent la prise en charge des approches de co-conception [Fuh et Li, 2005] :

- les architectures client-serveur, avec deux politiques de gestion envisageables :
  - l'utilisation de clients lourds coordonnés par un serveur. Il s'agit de systèmes maître-esclave où chaque client détient le contrôle d'un sous-ensemble, et a la possibilité de visualiser ceux des autres intervenants [Tay et Roy, 2003],
  - l'utilisation de clients légers. Le modèle est ici centralisé et les clients y accèdent à tour de rôle, par exemple par un mécanisme de jeton [Li et coll., 2004] ;
- les architectures d'égal à égal. Dans ce type d'application, tous les clients ont les mêmes possibilités et offrent des services aux autres clients. Un exemple typique est l'utilisation d'agents pour gérer la négociation des choix de conception [Mori et Cutkosky, 1998].

Afin d'être efficace, un système de conception collaborative doit faire intervenir des moyens correspondant à des niveaux d'abstraction différents. Notamment en milieu industriel, nous pourrons rencontrer : des outils de visualisation et d'annotation associés à des outils de visioconférence pour mener les revues de projet ; des outils de co-conception pour l'intégration des assemblages et la conception des parties critiques ; et des SGDT afin de maintenir la consistance globale du produit et de garantir une certaine traçabilité.

Le paragraphe suivant traite de la gestion des connaissances, un concept sur lequel les outils interactifs de travail collaboratif et d'ingénierie simultanée reposent nécessairement.

1.4.4 Gestion des connaissances

Depuis le début des années quatre-vingt dix, la gestion des connaissances prend une dimension de plus en plus importante dans les entreprises [Serenko et Bontis, 2004]. Nous pouvons expliquer cette marque d'intérêt accrue par la complexification des produits, la diversification

des savoir-faire[76], et l'évolution des connaissances de manière générale, mais aussi par le fait que la gestion des connaissances est devenue un avantage compétitif et stratégique déterminant [Economist Intelligence Unit, 2005]. La gestion des connaissances trouve son origine dans le constat que la connaissance (technologies, savoir, savoir-faire[77], brevets...), qui est la principale richesse d'une entreprise, est principalement détenue par les employés, et que les départs à la retraite, les mutations, les démissions, etc. sont autant de facteurs susceptibles de remettre en question sa pérennité. La notion de mémoire d'entreprise est néanmoins représentée depuis longtemps à travers l'ensemble des documents produits par l'entreprise. Ces documents sont aujourd'hui très variés dans leurs formes (documents électroniques, rapports imprimés, données audio et vidéo, courrier électronique, plaquettes...), leurs contenus (technique, administratif, commercial...) et le public auquel ils s'adressent dans l'organisation ou à l'extérieur de celle-ci. Néanmoins, représenter l'information ou la connaissance n'est bénéfique que si elle est représentée sous une forme compréhensible et exploitable [Bellinger, 2004]. Ainsi, il existe deux objectifs fondamentaux :

Capitaliser la connaissance, ce qui nécessite d'être en mesure de créer, d'extraire, d'analyser, de structurer et de formaliser la connaissance, puis de pouvoir l'actualiser en temps opportun. Le formalisme choisi conditionne fortement l'utilisation qui pourra être faite de la connaissance par la suite ;

Réutiliser la connaissance, ce qui implique d'avoir des moyens de diffusion, de protection, de recherche et de mise en œuvre de la connaissance.

Un certain nombre de difficultés freinent une gestion des connaissances efficace. Notamment, les connaissances ne sont pas toujours conscientes ; [Roucoules, 1999] indique que la part consciente du savoir peut être assimilée au sommet d'un iceberg. En outre, le savoir-faire est généralement tacite et parfois difficile à formaliser, sans compter la réticence que peuvent avoir les employés à le partager. Ainsi, la mise en place d'une politique de gestion des connaissances est coûteuse en temps et en argent.

Dans le domaine du développement de produits manufacturés, l'utilisation des bibliothèques de composants industriels, de caractéristiques, et aujourd'hui des informations rela-

76. Ce qui est dû à l'apparition de nouvelles technologies (p. ex. l'informatique embarquée), de nouveaux matériaux (p. ex. dans le domaine des matériaux composites) et de nouveaux procédés de fabrication (p. ex. les procédés de fabrication par ajout de matière).

77. Une nuance existe entre savoir et savoir-faire : le *savoir* regroupe un ensemble de connaissances explicites acquises par l'étude ou l'expérience, et le plus souvent théoriques ; le *savoir-faire* désigne quant à lui des connaissances directement applicables à des tâches spécifiques.

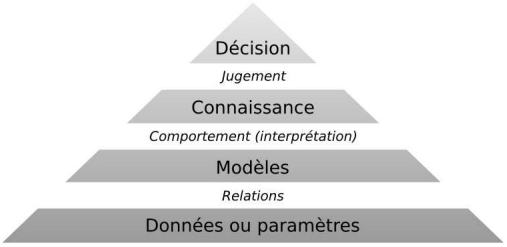


FIGURE 1.24: Valeur de l'information selon [Ullman, 2002] (croissante de bas en haut).

tives aux processus de conception et de fabrication, sont des démarches qui entrent dans le champ de la gestion des connaissances. [Roucoules, 1999] catégorise les formalismes de représentation des connaissances en deux grandes familles :

- les formalismes représentant les liaisons de cause à effet qui correspondent aux règles de production et aux différentes formes de raisonnement de la logique ;
- les formalismes décrivant les faits du monde réel qui correspondent aux représentations d'entités (réseaux sémantiques, objets, représentations des logiques non-classiques, et caractéristiques dans le domaine de la CAO).

En CAO, la gestion des connaissances devrait pouvoir se rapporter à l'ensemble des informations relatives au produit. Cela constitue l'un des buts fondamentaux du PLM. Comme le montre Ullman, selon le degré d'abstraction, l'information utile à la conception d'un produit n'a pas la même « valeur » (cf. figure 1.24) [Ullman, 2002]. L'objectif essentiel de la gestion des connaissances est ici de permettre une réutilisation de la rationalité de l'acte de conception, ce qui, si l'on veut être complet, passe par la prise en compte des informations provenant des quatre niveaux d'abstraction représentés sur la figure 1.24. La représentation des connaissances liées à la fabrication sera abordée dans le chapitre 2.

1.4.5 Conclusion

Dans cette section, nous avons abordé les préoccupations relatives à la prise en compte des évolutions organisationnelles des entreprises, et dues au durcissement du contexte économique de celles-ci. Ces thèmes poussent la CAO vers une meilleure prise en compte des différents aspects du cycle de vie, dans le but de l'optimiser. Nous avons vu qu'une intégration de l'écrasement de la structure hiérarchique, de la pluridisciplinarité, des concepts relatifs au travail de groupe en réseau, et du savoir-faire était vraisemblablement l'élément déterminant dans le succès de la prochaine génération de systèmes de CAO.

1.5 Synthèse

En plus de trente ans, un certain nombre d'avancées ont conduit les premiers outils de CAO, à ceux que nous connaissons aujourd'hui. Le concept de caractéristique permet de prendre en compte un certain nombre de notions fonctionnelles, souvent reliées à la forme (caractéristiques de forme, de fabrication, de précision, d'assemblage…). Même si ces évolutions ont apporté des gains remarquables, il n'en reste pas moins que les défis posés par le contexte économique actuel d'une part, et l'avènement des réseaux d'autre part, nous montrent que beaucoup de chemin reste encore à parcourir avant d'aboutir à une gestion intégrée de type PLM.

Tout d'abord, le processus de conception lui-même est amené à évoluer afin d'intégrer les contraintes fortes de l'ingénierie simultanée et de la gestion des connaissances. Les théories de la conception semblent à l'heure actuelle encore trop éloignées de la réalité industrielle pour prendre en compte ce type de paramètre. Comme le note [Váncza, 1999], notre compréhension du mécanisme de synthèse étant encore insuffisante, la conception collaborative et l'ingénierie concourante sont considérées comme le meilleur cadre d'utilisation de cette faible connaissance. Une plus grande souplesse est nécessaire. Aujourd'hui, la plupart des entreprises utilisent une combinaison des modèles systémique (besoin de planification) et de gestion économique et temporelle de projet (nécessité de collaborer pour atteindre l'objectif) [Eynard, 2005]. De plus en plus souvent, les phases jalonnant le processus de conception doivent ainsi être reconsidérées au cas par cas pour chaque projet et même évoluer au cours d'un projet afin d'adapter et d'optimiser la coordination. Les outils de travail collaboratif peuvent jouer un rôle de support à cette souplesse.

En second lieu, nous avons noté au § 1.3.1.2 que la notion d'intégrité entre caractéristiques de conception et de fabrication était un élément clef, non encore géré par les systèmes de CAO. Cette notion d'intégrité technologique est une répercussion directe de la logique d'ingénierie simultanée ; elle consiste à faire converger les points de vue et les connaissances correspondant aux différents métiers, en faisant intervenir la conciliation lorsque cela est nécessaire. Sous cet angle, la conception assistée par ordinateur pourrait évoluer vers une conception assistée par les connaissances.

Dans le cadre de cette thèse, nous souhaitons proposer une approche apportant une réponse au problème d'intégrité entre connaissances de conception (méthodologie, caractéristiques de forme) et de fabrication (savoir-faire lié aux procédés). Plus précisément, l'approche proposée

vise à intégrer au logiciel de CAO des critères liés au savoir-faire des fabricants. Ces critères sont utilisés pour guider et assister le concepteur, afin que des règles dites de « fabricabilité » soient respectées dès les premières phases de la conception. Malgré le caractère automatique de l'approche, l'intervention directe du fabricant n'est pas exclue (compromis ou recherche d'une solution à un problème spécifique). Ainsi, ce travail s'inscrit dans une logique de conception guidée par la fabricabilité et constitue, dans une certaine mesure, un support pour l'ingénierie simultanée. Afin de mieux cerner cette problématique, nous effectuons dans le chapitre 2, un état de l'art des différents travaux de recherche dans le domaine.

Chapitre II

Conception guidée par la fabricabilité : état de l'art

Sommaire

2.1 Introduction 56

2.2 Caractérisation de la fabricabilité 59

2.2.1 Indicateurs de fabricabilité 59

2.2.2 Référentiel 63

2.2.3 Règles de fabricabilité 65

2.2.4 Conclusion 67

2.3 Méthodes de prise en compte de la fabricabilité 67

2.3.1 Introduction 67

2.3.2 Aspect décisionnel 69

2.3.3 Politique de calcul 72

2.3.4 Démarche de réutilisation des connaissances 73

2.3.5 Intégration et interaction avec les principes de modélisation 81

2.4 Synthèse 89

2.1 Introduction

Dans le chapitre 1, nous avons passé en revue un certain nombre d'aspects relevant de la conception de produits manufacturés, en particulier vis-à-vis des apports présents et possibles de l'informatique dans ce domaine. Nous avons défini l'activité de conception comme la synthèse d'un artefact ou d'un assemblage d'artefacts répondant à un cahier des charges, avec pour finalité la réalisation d'un produit. Dans sa tâche, le concepteur doit ainsi prendre en considération des contraintes issues de deux domaines :

- les contraintes des activités en amont de la conception, que l'on qualifie de fonctionnelles ;
- et les contraintes des activités en aval de la conception, qui proviennent notamment des procédés de fabrication utilisés.

Pour être concis, le rôle du concepteur consiste à rendre explicites les interdépendances existant entre les contraintes de ces deux domaines. Par exemple, le cahier des charges fixe habituellement une fourchette de prix pour le produit, en fonction du public auquel il s'adresse. Il est évident que le procédé de fabrication utilisé pour fabriquer le produit aura un impact sur son prix de vente. Afin d'obtenir un produit viable, le concepteur doit alors s'assurer que le produit puisse être fabriqué pour un coût compatible avec le cahier des charges, et faire en sorte de limiter autant que possible les coûts de fabrication. Il doit donc mettre en corrélation le coût évalué et le coût visé.

Or, si le concepteur n'est pas en mesure d'effectuer cette vérification, ce qui est souvent le cas du fait de sa connaissance partielle des procédés, ou plus simplement parce que la production est sous-traitée, c'est au moment de la fabrication que les incohérences apparaissent (cf. figure 2.1). Les conséquences se caractérisent alors par une augmentation des coûts de développement du produit : il faut revenir en arrière pour faire la modification, ce qui a, la plupart du temps, un impact sur les choix de conception effectués entre-temps. À l'inverse, une considération directe de la fabrication au moment de la conception permet d'éviter ce type de rebouclages. Mais cela implique que le concepteur soit effectivement conscient des conséquences de ses choix en termes de fabricabilité.

Cette idée de tenir compte dès la conception des préoccupations non directement liées à la conception se retrouve dans un certain nombre de recherches menées depuis plus de vingt années sous le thème de la conception orientée ou guidée [Kuo et coll., 2001]. Des travaux ont

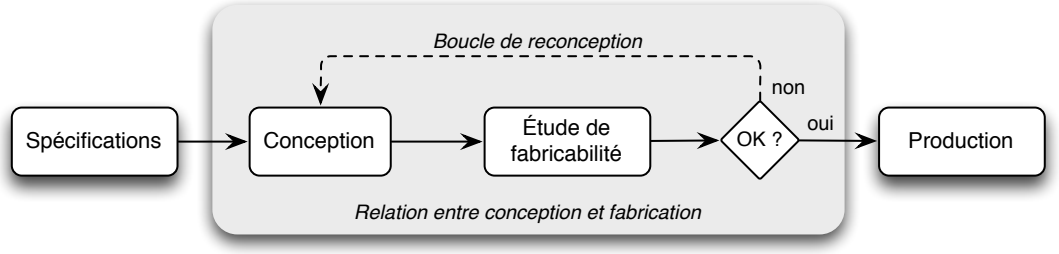


FIGURE 2.1: Boucles de re-conception.

été réalisés dans ce sens, relativement à divers domaines du cycle de vie [78] :

Conception du produit. Un certain nombre de préoccupations liées à la phase de développement du produit peuvent être considérées lors de la conception. Par exemple, la conception orientée vers la fiabilité consiste à concevoir un produit en considérant les différents facteurs susceptibles d'améliorer sa fiabilité, c'est-à-dire sa capacité à remplir ses fonctions durant une durée donnée et dans des conditions déterminées (sanité des mécanismes, résistance à l'usure...) [Kuo et coll., 2001]. La conception pour la diversité permet de prendre en compte les coûts liés à la personnalisation de masse [Martin et Ishii, 1997]. La conception pour la simulation [79] consiste quant à elle à concevoir en respectant les besoins de la simulation, notamment en termes de propriétés du modèle de représentation [Craveur et Marceau, 2001 ; Adams, 2006] ;

Production. La conception en vue de la fabrication et de l'assemblage (DFMA [80]) consiste à s'assurer que la production est possible et à minimiser les coûts associés [Boothroyd et coll., 2001 ; Edwards, 2002]. Le DFM [81] tend à réduire les coûts de fabrication de chaque élément indépendamment des autres, par exemple en évitant les formes trop complexes ou trop difficiles à obtenir et en choisissant des procédés stables et peu coûteux [O'Driscoll, 2001]. Le DFA [82] permet de réduire les coûts de montage, principalement en réduisant le nombre de composants, en étudiant la fa-

78. Notons que certaines de ces approches s'appliquent à plusieurs domaines à la fois. Par exemple, la conception orientée vers la fiabilité (voir dans la suite) fait intervenir des notions relatives à la conception ainsi qu'à la fabrication. Cette classification doit donc être considérée comme non unique.

79. Celle-ci s'inscrit dans le cadre plus général du *simulation-based design* qui consiste à utiliser la simulation comme outil d'aide aux choix de conception.

80. *Design for Manufacture and Assembly*.

81. *Design for Manufacture*.

82. *Design for Assembly*.

çon dont ils s'agencent, et en facilitant la préhension dans le cas d'un assemblage manuel [Huang et Mak, 1999 ; Beiter et coll., 2000]. Plus globalement le DFC [83] et le DFQ [84] permettent de réduire les coûts de production et d'améliorer la qualité des produits. Enfin, la conception guidée par le contrôle vise à faciliter les tests de qualité, p. ex. en électronique afin d'améliorer la facilité avec laquelle un circuit intégré peut être testé [85] [Russell et Sayers, 1986] ;

Vie du produit. Lors de l'utilisation du produit, il faut pouvoir en effectuer la maintenance. La conception orientée vers la maintenance a pour objectif de faciliter ces interventions en améliorant l'identification et le remplacement des pièces défectueuses, permettant des interventions de maintenance rapides [Slavila et coll., 2005 ; Desai et Mital, 2006 ; Kuo et coll., 2001] ;

Disparition du produit. L'importance grandissante des questions environnementales pousse de plus en plus les industriels à considérer au plus tôt le retraitement des produits et leur impact sur l'environnement (recyclage) [Kuo et coll., 2001]. L'écoconception et la conception pour le cycle de vie permettent ainsi de prendre en compte les coûts directs et indirects du produit qui sont liés au principe pollueur-payeur [86].

Il est important de noter que ces principes de conception entretiennent des rapports de dépendance (p. ex. entre écoconception et DFC [Chen, 2004]), et que certains s'entrecoupent dans leurs finalités (p. ex. DFC et DFM). Toutes ces approches s'insèrent dans la logique de l'ingénierie simultanée. Elles mettent en relief des critères qui devraient d'emblée être considérés par les concepteurs, mais qui, en pratique, du fait des contraintes temporelles fortes des projets, préfèrent souvent se concentrer sur le respect des fonctions primaires du produit à concevoir, et « oublient » les aspects qui ne sont pas immédiatement mesurables.

Dans ce manuscrit, nous nous focalisons sur la conception guidée par la fabricabilité. La fabricabilité est l'aptitude d'un produit à être réalisé facilement, de façon constante et avec une excellente qualité [www-GDT]. Beaucoup d'approches et d'outils de conception guidée par la fabricabilité existent à l'heure actuelle, mais ils souffrent d'un certain nombre d'inconvénients que nous nous proposons de souligner dans ce chapitre. Dans la suite, nous présentons

83. *Design for cost*.
84. *Design for Quality*.
85. Ce concept est également présent en informatique (génie logiciel) [Pettichord, 2002].
86. ce principe consiste à imputer aux entreprises les coûts liés à la prévention et à la lutte contre la pollution qui résulte de leurs activités. Les informations officielles à ce sujet sont disponibles sur le site [www-Ecologie].

le concept de fabricabilité et différentes manières de l'appréhender (§ 2.2). Puis, nous faisons un tour d'horizon des différentes méthodes de prise en compte de la fabricabilité (§ 2.3). Enfin, nous clôturons le chapitre par une synthèse.

2.2 Caractérisation de la fabricabilité

La notion de fabricabilité englobe une prise en compte de l'ensemble des contraintes liées à la production d'un produit, incluant donc normalement les processus de fabrication et d'assemblage. Ainsi, la conception guidée par la fabricabilité contribue globalement à créer des produits qui donnent de bons résultats en fabrication.

Elle consiste à :

- intervenir au plus tôt dans le cycle de développement et utiliser des procédés rapides à mettre en place afin d'améliorer les délais de mise sur le marché ;
- mieux maîtriser les coûts émanant des procédés de fabrication (utilisation de procédés peu coûteux, ayant un bon rapport qualité/coût) ;
- mieux maîtriser la qualité (adéquation entre la qualité requise par les fonctionnalités et celle obtenue par les procédés de fabrication).

[Gupta et coll., 1995] définissent la conception guidée par la fabricabilité comme devant considérer simultanément les objectifs de conception et les contraintes de fabrication afin d'identifier les problèmes liés à la fabrication et de faciliter leur résolution lors de la conception. Guider le concepteur nécessite de pouvoir évaluer la fabricabilité de la pièce au cours de la conception. Évaluer cette fabricabilité revient à quantifier des indicateurs dits de fabricabilité, permettant par exemple de comparer différentes solutions.

Dans la suite de cette section, nous évoquons différents indicateurs relatifs à la fabricabilité (§ 2.2.1) et différentes façons de les mesurer (§ 2.2.2). Nous y présentons également la notion de règle de fabricabilité (§ 2.2.3) qui, comme nous le verrons, paraît incontournable dans une approche de conception guidée. Des exemples de mise en œuvre de ces différents indicateurs existent dans la littérature référencée dans la section 2.3.

2.2.1 Indicateurs de fabricabilité

Nous définissons un indicateur de fabricabilité comme une grandeur qui, une fois évaluée, contribue à caractériser le fait qu'une pièce est facilement ou difficilement fabricable. Les adjec-

tifs « facilement » et « difficilement » peuvent ici être interprétés comme « simple ou complexe techniquement », comme « rapide ou long à mettre en place », ou bien comme « peu ou très onéreux ». Nous décrivons successivement ces différents points de vue dans ce paragraphe.

Il faut avoir à l'esprit que les différents indicateurs mentionnés dans la suite s'influencent les uns les autres. Notamment, un procédé de fabrication complexe techniquement sera très souvent plus coûteux économiquement qu'un procédé plus simple à mettre en œuvre (mais pas nécessairement, ceci étant lié au cas traité). Quoi qu'il en soit, la fabricabilité d'un produit est toujours déterminée par la mise en corrélation des connaissances relatives aux ressources de fabrication, avec les spécifications (ou les modèles) du produit dont on dispose [Zhao et Shah, 2002].

2.2.1.1 Degré de détermination

L'objectif de l'analyse de fabricabilité est d'aider les acteurs du projet à faire des choix. Nombre d'auteurs ayant travaillé sur le DFM s'accordent à dire que la fabricabilité doit comporter différents degrés de détermination ou différentes phases d'analyse [Boothroyd et coll., 2001 ; Gupta et coll., 1997 ; van Vliet et coll., 1999 ; Zhao et Shah, 2002]. Ainsi, globalement, nous distinguons deux objectifs fondamentaux en analyse de fabricabilité :

- l'analyse de faisabilité qui est purement qualitative ;
- et l'analyse comparative qui est quant à elle quantitative et fait donc intervenir des évaluations plus précises.

L'analyse de faisabilité doit permettre aux acteurs de savoir si la fabrication du produit est envisageable à la fois techniquement et économiquement, alors que l'analyse comparative permet de situer le produit, tantôt par rapport à d'autres alternatives de conception, tantôt par rapport aux spécifications du cahier des charges, et là encore, en tenant compte des points de vue technique et économique. Naturellement, pour une conception donnée, l'analyse quantitative n'est nécessaire que si l'analyse qualitative a été menée avec succès [Zhao et Shah, 2002].

2.2.1.2 Indicateurs technologiques

Il existe un grand nombre d'indicateurs propres à la technologie utilisée, ayant un impact sur la fabricabilité d'un produit. Ces facteurs peuvent être classés dans trois catégories : les facteurs liés aux ressources de fabrication disponibles ; les facteurs liés aux caractéristiques du produit que l'on cherche à évaluer ; et enfin, les facteurs liés aux exigences du procédé. Nous les détaillons ci-après.

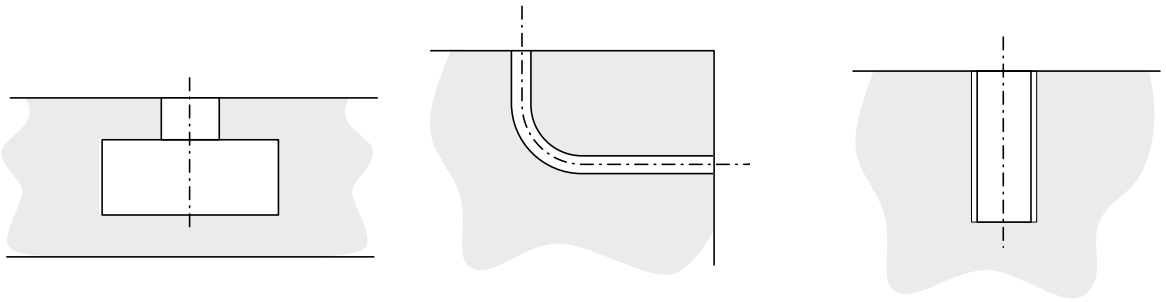
Facteurs liés aux ressources de fabrication. Il est nécessaire de prendre en compte les capacités de production pour évaluer la fabricabilité d'une pièce [Fortin et Huet, 2007 ; Sly, 2004]. Les capacités de production consistent en un ensemble de ressources de production : les machines (machines-outils à commande numérique, robots d'assemblage, fours, presses à injecter…) ; les outils (fraises, forets, plaquettes en carbure de tungstène, moules, empreintes…) ; les techniciens ; les opérateurs ; etc. Chacune de ces ressources possède des capacités techniques qui définissent le type des pièces qui peuvent être fabriquées, et a, dans ce cadre, un impact sur la fabricabilité du produit. Par exemple, dans le cas de l'usinage d'une pièce, une caractéristique de forme donnée peut nécessiter un outillage spécifique non directement disponible, ce qui engendre une phase de mise en place plus complexe : achat ou fabrication d'un outillage ; sous-traitement de la fabrication de la pièce ; ou abandon de la pièce ou du mode de fabrication envisagé. Les ressources de production auront le plus souvent un impact sur la faisabilité.

Facteurs liés aux caractéristiques du produit. Dès lors que les ressources de fabrication sont disponibles, c'est la forme du produit qui va avoir un impact direct sur ses possibilités de fabrication. Nous répartissons les caractéristiques technologiques du produit ayant un impact sur sa fabricabilité en trois catégories :

la forme de la pièce est l'un des facteurs les plus déterminants. De plus, son impact est différent selon le procédé de fabrication ou le matériau utilisé : une forme difficile à obtenir avec un procédé de fabrication A, peut être beaucoup plus simple à obtenir en utilisant un procédé B. Par exemple, la réalisation d'une pièce ayant des parois fines est souvent bien plus simple par pliage ou par coulée sous pression que par usinage [Boothroyd et coll., 2001]. Par ailleurs, certaines formes peuvent être impossibles à obtenir avec un procédé de fabrication donné : p. ex., en usinage, il est impossible d'effectuer un filetage borgne à fond plat, ou un trou ayant un axe courbé [Boothroyd et coll., 2001]. La figure 2.2 présente quelques exemples de pièces non fabricables par usinage ;

la matière choisie pour le produit peut avoir un impact sur sa fabricabilité. Par exemple, en usinage, certains aciers sont plus difficiles à usiner que d'autres [Boothroyd et coll., 2001]. En outre, tous les procédés ne permettent pas de travailler tous les matériaux ;

la qualité requise pour la pièce est aussi un élément déterminant [Boothroyd et coll., 2001]. La plupart du temps, la mise en place d'un procédé sera plus difficile et son coût plus



(a) Rainure intérieure (b) Trou à axe courbé (c) Filetage borgne à fond plat

FIGURE 2.2: Exemples de pièces non fabricables par usinage (ici en vue de coupe ; tirés de [Boothroyd et coll., 2001]).

important si la pièce doit avoir une qualité élevée. Requérir des tolérances trop serrées engendre donc des surcoûts inutiles. À l'inverse, un tolérancement trop lâche risque de faire perdre au produit des propriétés fonctionnelles. De plus, les machines sont limitées quant à leurs possibilités d'obtenir une qualité donnée.

Facteurs liés aux exigences du procédé. Beaucoup de procédés nécessitent de la pièce qu'elle ait certaines propriétés de forme. Par exemple, en forgeage, il est nécessaire de prévoir des dépouilles afin de faciliter l'extraction des pièces. Ces propriétés géométriques peuvent engendrer des étapes supplémentaires dans le processus de fabrication, ou dans certains cas, aller à l'encontre de certaines contraintes fonctionnelles de la pièce.

Dans le paragraphe suivant, nous abordons d'autres indicateurs, basés sur des critères économiques.

2.2.1.3 Indicateurs économiques

Nous distinguons deux types d'indicateurs économiques :

les coûts de fonctionnement d'un procédé de fabrication, de la matière première utilisée, des outillages, sont autant d'indicateurs de la fabricabilité d'une pièce. Une classification des différents coûts de fabrication est disponible dans [Chan et Salustri, 2003]. L'emploi de tel ou de tel autre procédé se fait en adéquation avec le coût visé ;

les délais de préparation nécessaires à certains procédés de fabrication sont importants. La

pièce doit être adaptée au procédé afin de pouvoir être fabriquée (p. ex. par la création de dépouilles, ou la détermination du brut en usinage ou du lopin en forge), ou à l'inverse, le procédé doit être adapté à la pièce (p. ex. par la conception de moules, de mécanismes de bridage...). En outre, les temps de fabrication et d'assemblage de la pièce peuvent être plus ou moins longs, ce qui a aussi un impact sur son coût.

Les moyens qu'il est possible d'utiliser pour déterminer ces indicateurs sont bien évidemment liés à la technologie utilisée.

2.2.2 Référentiel

Au début du § 2.2.1, nous avons défini un indicateur de fabricabilité comme étant une grandeur [87]. Or, par définition, selon le [Bureau International des Poids et Mesures, 2006], nous avons :

$$Grandeur = Nombre \times Unité$$

où l'unité n'est qu'un exemple particulier de la grandeur concernée, utilisé comme référence. Le nombre est le rapport entre la valeur de la grandeur en question et l'unité. Le référentiel relatif à une grandeur consiste ainsi en :

- une valeur de référence qui constitue le point de repère à partir duquel la mesure est effectuée. Par exemple, en thermométrie, il est possible d'utiliser 0 °C ou bien 0 °F comme valeur de référence (ici, $0\,°\mathrm{C} = 32\,°\mathrm{F}$ et $0\,°\mathrm{F} = -17,\bar{7}\,°\mathrm{C}$) ;
- et une unité de mesure qui permet de quantifier physiquement la grandeur. L'équivalence entre les degrés Celcius et Fahrenheit de l'exemple précédent est donnée par la formule $°\mathrm{F} = °\mathrm{C} \times \frac{9}{5} + 32$, ce qui veut dire que l'écart entre deux températures n'est pas le même suivant que l'on utilise l'une ou l'autre des échelles. L'unité de mesure joue un rôle déterminant quant à la lisibilité des résultats.

En fonction du point de vue, plusieurs types de référentiels sont donc possibles pour spécifier une même grandeur. Il est possible de les classifier selon plusieurs critères que nous indiquons brièvement dans les paragraphes suivants : caractère absolu ou relatif, abstrait ou concret, incertain, lié à un optimum, etc.

87. Dans la littérature étudiée, les auteurs choisissent d'emblée un référentiel en fonction du cadre particulier de l'étude, mais il semble que la question du référentiel soit considérée comme annexe. Il est vrai que les conversions sont triviales. Néanmoins, il nous semble important de ne pas omettre cet aspect.

2.2.2.1 Caractère absolu ou relatif

Pour toute mesure, deux types de référentiels sont envisageables : les référentiels absolus, où chaque élément est quantifié indépendamment des autres, et les référentiels relatifs, où les différents éléments sont évalués les uns par rapport aux autres. Ces deux types de référentiels ont des avantages et inconvénients, que nous exposons brièvement ci-après.

Mesures absolues. L'échelle est ici indépendante des instances que l'on mesure. Elle est donc plus objective qu'une échelle relative dans la mesure où elle fixée à l'avance, mais elle rend la comparaison entre plusieurs alternatives moins immédiate du fait qu'elle nécessite une connaissance des évaluations habituellement constatées : p. ex., situer $P_a = 12$ par rapport à $P_b = 8$ est difficile si l'on ne sait pas que la valeur moyenne de P est 9. Du point de vue de la fabricabilité, les causes des difficultés sont alors plus difficiles à identifier, et l'on a pas nécessairement conscience des évolutions (amélioration ou détérioration de la fabricabilité).

Mesures relatives. L'échelle est cette fois dépendante d'une ou de plusieurs valeurs antérieures, ce qui veut dire qu'elle évolue dans le temps. L'avantage des mesures relatives est qu'elles permettent à une personne non spécialiste de situer la grandeur et d'identifier assez facilement les situations anormales. En revanche, lorsqu'il s'agit de mettre en parallèle l'objectif visé et la situation courante, ce type d'approche est moins adéquat car il est alors nécessaire de situer l'objectif sur l'échelle relative, ce qui est peu naturel.

2.2.2.2 Caractère concret ou abstrait

Quel que soit le type de système de référence choisi (absolu ou relatif), il est possible d'utiliser des unités de mesure concrètes ou abstraites. L'utilisation d'une unité de mesure concrète permet de représenter des valeurs précises et rend la grandeur « palpable » pour un expert. Le système international d'unités du [Bureau International des Poids et Mesures, 2006] comprenant par exemple le degré Celcius, le mètre ou le kilogramme, constitue un jeu d'unités de mesure concrètes. Mais certains auteurs préfèrent utiliser des mesures dites abstraites qui sont souvent plus faciles à interpréter pour une personne non experte. Opter pour une échelle abstraite revient à utiliser une grandeur de dimension un [88], ce qui peut par exemple être utile pour regrouper plusieurs indicateurs de fabricabilité sous la forme d'une mesure

88. On utilise alors comme unité l'unité SI « un » [Bureau International des Poids et Mesures, 2006].

unique. Ainsi, une échelle abstraite pourra par exemple être constituée de $\mathbb{N}^{+\star}$ (échelle abstraite quantitative), ou d'un ensemble du type $\{1, 2, 3, 4, 5\}$, ce qui s'apparente cette fois plus à une caractérisation qualitative utilisée par certains auteurs [Gupta et coll., 1995] : par exemple {médiocre,moyen, satifaisant, excellent} ou plus simplement {non fabricable,fabricable}.

2.2.2.3 Autres caractéristiques

Il est possible de trouver d'autres éléments permettant de discriminer les référentiels. Par exemple, le fait que la valeur de référence soit liée ou non à un optimum, ou le fait d'utiliser une échelle linéaire ou non-linéaire [89]. Un autre aspect intéressant est celui de la précision ou de la certitude qu'il est possible d'accorder à une mesure. Mais ces points vont au delà du champ de cette étude et nous ne décrirons pas plus en détail le vaste sujet qu'est celui de la métrologie dans ce manuscrit.

2.2.3 Règles de fabricabilité

Les connaissances relatives à la fabricabilité des pièces sont la plupart du temps enregistrées sous forme de règles de production [90] [Zhao et Shah, 2004 ; Boothroyd et coll., 2001]. Une raison à cela est que les règles de production sont suffisamment souples pour permettre de représenter à la fois des connaissances déclaratives et procédurales. Elles sont beaucoup utilisées en intelligence artificielle, notamment dans les systèmes experts, et sont particulièrement bien adaptées à la représentation des règles de fabricabilité. Elles se présentent sous la forme suivante :

si *conditions* **alors**
  conclusions
fin si

La partie *conditions* correspond à des valeurs de paramètres (dimensions, matériau, procédé, etc.) qui déterminent la fabricabilité de la pièce. La partie *conclusions* permet de prendre en compte l'appréciation de la fabricabilité.

89. Nous pensons aux échelles logarithmiques comme par exemple celle des Décibels ($dB = 10 \log_{10} \left(\frac{P_1}{P_0}\right)$ où P_0 et P_1 sont des puissances).

90. Une règle de production (ou règle de type « *if-then* ») est un quantum de connaissance déclaratif et autonome [Haton et Haton, 2000] comportant deux parties : une condition et une conclusion. La véracité de la condition dicte la survenance de la conclusion. Un coefficient de vraisemblance peut être adjoint à la règle afin de signifier qu'elle est entachée d'incertitude.

[Zhao et Shah, 2004] distinguent deux types de règles de fabricabilité : les règles simples qui sont directement associées aux caractéristiques de forme ; et les règles complexes qui nécessitent un traitement particulier. Un exemple de règle simple pourrait être : pour la tôle métallique, le diamètre d'un trou (débouchant) ne doit pas être inférieur à l'épaisseur de la tôle. Un exemple de règle complexe pourrait être : pour la tôle métallique, la distance entre deux trous doit être supérieure à deux ou trois fois l'épaisseur de la tôle. Dans ce second exemple, le traitement nécessaire est le calcul de la distance entre les deux trous, élément qui n'existe pas par avance dans le modèle numérique de la pièce [91].

Voici quelques exemples supplémentaires de règles de fabricabilité [92] tirés de [Zhao et Shah, 2004] :

- en moulage par injection :
  - l'épaisseur des parois formant la pièce ne doit pas dépasser 3,5 mm si le matériau utilisé est de l'ABS [93],
  - pour les trous borgnes, le rapport de la hauteur au diamètre doit être inférieur à deux si le diamètre du trou est inférieur à 3,5 mm, ou à trois sinon ;
- en fraisage :
  - les surfaces fraisées doivent être telles que le rapport de la hauteur au diamètre de la fraise n'excède pas trois,
  - pour les coins intérieurs ayant trois arêtes, au moins une arête doit porter un congé dont le rayon soit au moins égal à celui de la fraise utilisée.

À travers ces exemples, nous constatons que seuls, les indicateurs de fabricabilité (ici l'épaisseur, le rapport hauteur/diamètre, et le rayon du congé) ne suffiraient pas à déterminer la fabricabilité d'une pièce : leur cadre d'utilisation joue également un rôle fondamental. De par leur structure, les règles de fabricabilité sont à même de remplir cette fonction de contextualisation.

Il est à noter que ces règles peuvent être difficiles à évaluer de par le fait qu'elles font intervenir des informations diverses, pas toujours connues lors de la conception (notamment relatives au procédé).

91. Du moins, pas dans le cas général.
92. Pour des procédés différents il existe aussi des règles comparables. Par exemple, les procédés de repoussage au tour (*spinning*) et de profilage (*rollforming*) font l'objet de règles de fabricabilité [Ramana et coll., 2006].
93. Acrylonitrile-Butadiène-Styrène (matériau thermoplastique très résistant).

2.2.4 Conclusion

Dans cette section, nous avons défini ce qu'est la fabricabilité et identifié différents types d'indicateurs permettant de l'évaluer. Nous avons observé que différentes façons de mesurer la fabricabilité étaient possibles, en utilisant des échelles absolues ou relatives, abstraites ou concrètes, et basées sur des critères économiques ou technologiques. Nous avons également identifié deux exploitations différentes de l'analyse de fabricabilité : l'étude de faisabilité qui nécessite une évaluation qualitative, et l'étude comparative qui nécessite quant à elle une évaluation quantitative. Enfin, nous avons mis en évidence le fait que le contexte d'utilisation des indicateurs de fabricabilité jouait un rôle déterminant, et qu'afin d'en tenir compte, il était nécessaire de représenter les connaissances liées à la fabricabilité sous forme de règles. Dans la suite, nous parcourons différentes méthodes de prise en compte des règles de fabricabilité.

2.3 Méthodes de prise en compte de la fabricabilité

Dans cette section, nous faisons un tour d'horizon des méthodes existantes en matière de gestion de la fabricabilité durant la phase de conception.

2.3.1 Introduction

Prendre en compte des règles de fabricabilité au moment de la conception est une question à laquelle trois réponses possibles ont successivement été proposées [Zhao et Shah, 2002] :

1. L'utilisation de manuels de DFM et de DFA par le concepteur. Un certain nombre d'ouvrages de référence sont disponibles, dont deux des principaux sont [Boothroyd et coll., 2001] et [Bralla, 1999]. Dans ce cas, l'analyse de fabricabilité est effectuée directement par le concepteur. La plupart du temps, cela revient à respecter des recommandations (p. ex. minimiser le nombre de composants) et à valider des listes de critères (p. ex. la conformité de l'épaisseur des parois d'une pièce moulée). Des problèmes majeurs existent avec ce type d'approche : d'une part, toutes les vérifications (et les calculs qui s'y rapportent) doivent être faits à la main ; d'autre part, le concepteur ne peut plus se concentrer sur son activité de conception et doit de surcroît engranger une foultitude de connaissances qui ne font pas partie de son domaine. En outre, les connaissances de fabrication étant très vastes, un seul homme ne peut à lui seul en maîtriser

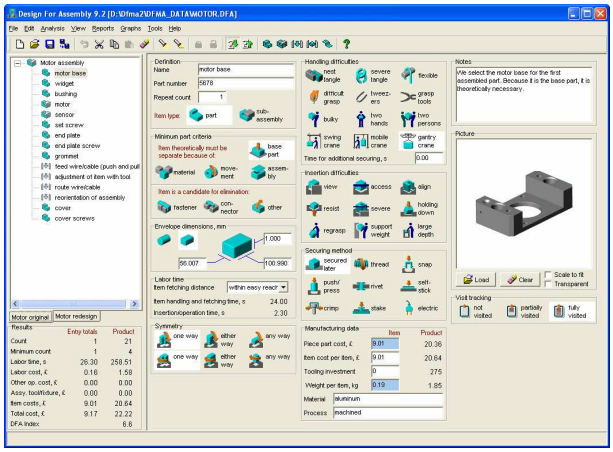


FIGURE 2.3: Logiciel DFMA®.

l'exploitation [Gupta et coll., 1995]. Bien que faciles d'accès, les manuels présentent donc des inconvénients incontestables ;

2. L'utilisation de logiciels de DFM et de DFA. Le développement des techniques de modélisation des solides et des capacités de calcul géométrique des ordinateurs a abouti sur des outils logiciels permettant de pallier partiellement les inconvénients relatifs à l'utilisation de manuels. Les calculs sont alors effectués par l'ordinateur qui, d'une manière ou d'une autre, porte à la connaissance du concepteur les critères de fabricabilité. Des outils commerciaux sont actuellement disponibles : p. ex., la suite logicielle commerciale DFMA® (cf. figure 2.3) proposée par Boothroyd et Dewhurst [www-DFMA] permet d'estimer la fabricabilité de pièces réalisées par divers procédés de fabrication (usinage, moulage par injection…). Mais ces outils sont souvent dédiés à des domaines spécifiques et ils ne sont pas toujours faciles à adapter aux spécificités de chaque entreprise. En outre, ils requièrent de l'utilisateur d'interpréter et de stipuler de lui-même certains indicateurs de fabricabilité ;

3. L'adoption d'organisations de type ingénierie simultanée (voir le § 1.4.2 pour une description). Cependant, les outils logiciels prenant en compte ce type d'organisation sont encore trop peu développés pour offrir un véritable support à l'analyse de fabricabilité. A fortiori, ils ne permettent pas une automatisation, même partielle, dans ce domaine. Les outils informatiques dédiés à l'ingénierie simultanée sont en règle générale des outils de travail collaboratif.

Dans la suite de cette section, nous apportons quelques précisions sur l'aspect décisionnel qui a son importance relativement à l'analyse de fabricabilité (§ 2.3.2). Ensuite, nous décrivons l'analyse de fabricabilité selon différents aspects : les différentes stratégies de calcul utilisées pour évaluer la fabricabilité (§ 2.3.3) ; la démarche de réutilisation du savoir-faire (§ 2.3.4) ; et enfin, l'intégration et les interactions des différentes approches avec les techniques de modélisation (§ 2.3.5).

2.3.2 Aspect décisionnel

Nous avons vu au chapitre 1 que le processus mental de conception d'un produit s'effectue selon un cycle comprenant différentes phases (voir notamment le § 1.2.2.3). D'un point de vue général, nous pouvons considérer trois phases principales dans le cycle de réflexion du concepteur :

1. Génération d'une ou de plusieurs solution(s) ;
2. Évaluation de chaque solution ;
3. Décision : rejet, adoption de la solution si elle est unique, ou bien choix de l'une des alternatives dans le cas contraire.

Ce cycle peut intervenir à différents niveaux d'abstraction et des rebouclages sont naturellement possibles à la fin de chaque cycle. À un niveau d'abstraction faible (caractéristiques de forme, composants élémentaires), il est fréquent que plusieurs solutions soient envisagées, au moins dans les premières phases de la conception (lorsque les définitions du produit ne sont pas encore gelées). En revanche, au niveau d'abstraction le plus élevé (celui du produit dans son ensemble), la plupart du temps, une seule solution est développée à la fois. Seules certaines firmes comme Toyota font exception à la règle et vont presque jusqu'à finaliser plusieurs solutions de manière concurrente pour n'en choisir qu'une à la fin[94]. C'est ainsi que [Zhao et Shah, 2002] opposent la conception « ponctuelle » *(point-based design)* pour laquelle une solution unique est développée de manière itérative, à la conception par ensembles *(set-based design)* [Sobek et Ward, 1996] pour laquelle plusieurs solutions sont développées en parallèle.

Dans le cas général, la première phase du cycle (génération de solutions) s'effectue par le concepteur lui même (sauf dans les systèmes de modélisation déclarative ; cf. § 1.2.2.4). La

94. [Sobek et Ward, 1996] proposent de composer la solution finale à partir de briques élémentaires choisies parmi les différentes alternatives développées, de façon à l'optimiser.

deuxième phase permet d'évaluer le produit selon des critères fonctionnels et comprend donc la conception fonctionnelle mais aussi l'analyse de fabricabilité ; la simulation peut être considérée comme un outil contribuant à la réalisation de cette phase. La troisième phase (décision) est traditionnellement à la charge du concepteur. Cependant, un certain nombre de travaux, principalement issus des domaines de la gestion et de l'économie, ont été effectués dans le but de modéliser la prise de décision en conception de produits [Zhao et Shah, 2002]. Mais ce processus est complexe pour plusieurs raisons :

- les critères de décision sont multiples ;
- ils sont interdépendants ;
- certains sont antagonistes et nécessitent donc d'établir des compromis.

C'est d'ailleurs pour cela que le processus de conception est majoritairement considéré comme itératif (et unanimement comme cyclique). Les approches d'aide à la décision en conception doivent donc prendre en compte le fait que plusieurs critères interdépendants interviennent au moment où un choix doit être effectué.

Les principales approches existantes sont : la théorie de l'utilité multi-attributs [Keeney et Raiffa, 1976] – appliquée à la conception par [Thurston, 1991] (pour les problèmes de jugement liés aux critères subjectifs, notamment) – qui dérive de la théorie de l'utilité de [Morgenstern et von Neumann, 1947] (notion d'utilité espérée) ; la méthode AHP [95] (et plus globalement ANP [96]), élaborée par [Saaty, 1980], qui se focalise sur la notion d'objectif ; l'approche BCA [97] dont l'origine remonte au milieu du XIX$^{\text{ième}}$ siècle, et qui s'appuie sur la comparaison du bénéfice économique au coût économique (notion d'utilité marginale) ; et l'approche DBD *(Decision-Based Design)* [Hazelrigg, 1998] et les méthodes qui en découlent (voir p. ex. [Wassenaar et Chen, 2001]), qui considèrent l'activité de conception à travers des options (ensembles de paramètres), des espérances et des valeurs.

La plupart de ces approches se basent sur une comparaison entre plusieurs alternatives et s'appliquent donc plus naturellement dans le cadre du *set-based design*. Elles sont particulièrement intéressantes lorsque les critères sont subjectifs ou incertains, et apportent alors une aide réelle pour mesurer le risque lié à une décision ou pour synthétiser la valeur d'une alternative en se basant sur une évaluation selon plusieurs critères. Mais elles ne sont exploitables que lorsque les critères pertinents sont clairement identifiés et positionnés les uns par rapport aux

95. *Analytic Hierarchy Process*.
96. *Analytic Network Process*.
97. *Benefit-Cost Analysis*.

autres (notion de priorité). Elles ne permettent pas une automatisation de la phase de décision dans le cadre de la conception, car les critères sont souvent trop nombreux et ils sont amenés à changer d'une conception à l'autre. Surmonter ces difficultés nécessite donc dans tous les cas l'intervention de l'humain [Hazelrigg, 1998]. L'étude des approches de modélisation de la décision sortant du cadre de cette thèse (qui se focalise sur les problèmes d'interopérabilité entre deux domaines), nous ne les détaillons pas davantage. Nous nous cantonnons ainsi à l'évaluation de la fabricabilité, qui reste nécessaire à toute décision, et considérons que le concepteur prend de bonnes décisions dès lors qu'il dispose des bonnes informations, ce qui n'interdit pas, par ailleurs, l'utilisation de méthodes du type de celles que nous venons d'évoquer. Selon [Gupta et coll., 1997], les trois objectifs fondamentaux de l'analyse de fabricabilité sont :

1. Déterminer si le produit est réalisable ;
2. Si tel est le cas, évaluer quantitativement la facilité avec laquelle la pièce pourra être fabriquée ;
3. Sinon, identifier les attributs de conception qui posent des problèmes de fabricabilité.

Ces trois aspects fondamentaux sont ainsi indispensables à la prise de décision. Nous pourrions ajouter une quatrième phase consistant à proposer des modifications visant à améliorer la fabricabilité.

[Duverlie et coll., 1999] indiquent qu'une méthode de prédétermination (p. ex. une estimation de coût) peut être qualifiée, similairement à un appareil de mesure, selon trois critères :

fidélité : capacité à donner toujours le même résultat pour le même produit ;

sensibilité : capacité à tenir compte des variations de données ;

justesse : capacité à fournir des résultats exacts.

Il est difficile d'attendre d'une analyse de fabricabilité un résultat juste car la fabricabilité est une notion qui fait intervenir un grand nombre de facteurs qui ne sont pas tous maîtrisés. C'est pourquoi nous parlons d'évaluation plutôt que de calcul précis. Toutefois, certains facteurs n'influent que très peu sur les critères de choix. L'exactitude ne présente donc pas nécessairement un intérêt du point de vue de la prise de décision : les notions de fidélité et de sensibilité sont en revanche toujours importantes. Par ailleurs, logiquement, l'acuité d'une prédétermination ne pourra être élevée que si les informations disponibles sont suffisamment précises.

Dans la suite de cette section nous explorons différents aspects relatifs à l'analyse de fabricabilité : la politique de calcul, la démarche de réutilisation, puis l'intégration avec les principes

de modélisation. Ces axes permettent de discriminer les recherches ayant trait à l'analyse de fabricabilité.

2.3.3 Politique de calcul

2.3.3.1 Type d'approche

Il est possible de distinguer deux familles de politiques de calcul selon que celles-ci sont directes ou indirectes [Gupta et coll., 1997] :

Les méthodes à base de règles brutes consistent à identifier des attributs de conception pertinents du point de vue de la fabricabilité et ce, par une inspection directe du modèle de conception. Comme nous l'avons vu, il existe en effet des règles établies qui permettent d'évaluer la fabricabilité d'une pièce en se basant sur des variables de conception. Ce type d'approche est néanmoins réservé à certains aspects correspondant à des procédés ou des règles spécifiques, et ne permet pas de prendre en compte tous les aspects de la fabricabilité notamment dans le cadre des procédés d'usinage.

Les méthodes guidées par la planification font intervenir des traitements intermédiaires. Elles génèrent un ou plusieurs plans de fabrication (séquences d'usinage et/ou de montage), puis les optimisent, et enfin évaluent les plus prometteurs. Elles mettent donc en action des traitements de type CAPP [Cutkosky et Tenenbaum, 1990]. En somme, ces méthodes consistent à prédire ce que seront les choix de fabrication afin de s'en « inspirer » pour évaluer la fabricabilité. Elles sont nettement moins légères que les premières en termes de calcul, mais permettent, lorsqu'elles sont applicables, d'obtenir des résultats plus précis, notamment vis-à-vis des procédés d'assemblage ou d'usinage. Cependant, le séquençage n'est jamais unique : pour une même pièce, plusieurs plans de fabrication sont toujours possibles. L'évaluation de fabricabilité n'est donc valable que sous réserve que le plan choisi soit réellement adopté, ce qui n'est pas garanti pour différentes raisons (modification de la pièce, du matériau, fluctuations de production, indisponibilité des ressources…).

2.3.3.2 Niveau d'automatisation

Le niveau d'automatisation est un élément qui varie suivant les approches [Gupta et coll., 1997]. Les interactions avec l'utilisateur peuvent être plus ou moins

importantes : certains systèmes proposent par exemple une extraction automatique des caractéristiques de fabrication, alors que d'autres requièrent une intervention humaine. Un autre aspect intéressant est la façon dont la fabricabilité est présentée à l'utilisateur : il est possible de n'afficher qu'un indicateur global, ou bien des indicateurs relatifs à chaque caractéristique de fabrication. De même, les suggestions de modifications peuvent concerner des attributs de conception spécifiques ou bien l'ensemble de la pièce.

2.3.4 Démarche de réutilisation des connaissances

Les données nécessaires à une évaluation de fabricabilité proviennent de deux sources : le modèle du produit en l'état (caractéristiques principalement fonctionnelles et morpho-dimensionnelles) et les connaissances de fabrication. Ces dernières existent sous différentes formes, ce qui conduit à différentes démarches de réutilisation. Nous distinguons deux types de connaissances liées à la fabrication :

- les connaissances qui concernent les procédés ;
- et l'expérience (les données constituant l'historique de l'entreprise).

Deux approches distinctes sont donc possibles pour réutiliser le savoir-faire de fabrication : utiliser la connaissance que l'on a des procédés afin d'évaluer la fabricabilité (déduction de l'information par génération ou calcul) ; ou utiliser l'expérience passée pour estimer la fabricabilité (déduction de l'information par variation). Nous décrivons ces deux approches dans la suite. Néanmoins, nous remarquons qu'elles ne sont pas exclusives et que dans certains cas, il peut être judicieux de les combiner (nous pensons à des cas où le procédé ou le produit ne sont que partiellement connus). Cette combinaison peut par exemple s'effectuer via un SBC.

2.3.4.1 Approches génératives

Les approches génératives permettent d'évaluer la fabricabilité en se basant d'une part, sur ce qui est connu du produit (à travers ses différents modèles), et d'autre part, sur ce qui est connu du procédé de fabrication (fonctionnement, coûts, configuration, paramétrage, qualité à attendre…). Il existe deux familles de méthodes génératives : les méthodes analytiques et les méthodes heuristiques. Nous les décrivons rapidement ci-dessous.

Méthodes analytiques. Les méthodes analytiques considèrent la rationalité du processus de fabrication pour déterminer ce qu'il est possible de fabriquer, et les coûts qui en découlent.

Afin d'évaluer la fabricabilité d'une pièce de manière analytique, il faut donc suffisamment connaître la nature du procédé de fabrication ainsi que la définition de la pièce. Les méthodes analytiques sont ainsi principalement utilisées afin d'obtenir des évaluations précises, une fois que la définition de la pièce est suffisamment mature.

Ces méthodes utilisent un processus de calcul déterminé qui prend en compte divers aspects afin d'obtenir un résultat le plus proche possible de la réalité. Il s'agit de méthodes de « simulation », qui peuvent faire appel à des techniques plus ou moins poussées :

- calcul de trajectoire d'une machine-outil (systèmes de CAPP et apparentés) [Chen et coll., 2002] ;
- étude analytique des coûts basée sur des propriétés technologiques des pièces [Dissinger et Magrab, 1996 ; Sánchez et coll., 1997] ;
- méthodes de simulation numérique. Celles-ci permettent en général d'obtenir des paramètres nécessaires à une estimation. Par exemple, le volume de matière pour une pièce moulée est nécessaire à une estimation de son coût.

Nous pourrions englober les méthodes analytiques dans ce que l'on appelle l'usine numérique [98]. En général, elles consistent à décomposer le procédé de fabrication et/ou la définition de la pièce en opérations ou en entités élémentaires, auxquelles une évaluation peut être plus facilement associée (grâce à l'isolement des phénomènes). Cette évaluation peut concerner des aspects divers de la fabricabilité (faisabilité, coût, qualité, délais…).

Par exemple, le système de [Dissinger et Magrab, 1996] décompose une pièce devant être fabriquée par frittage de poudre, selon des critères géométriques (décomposition par couches, étude des angles, des épaisseurs…) afin d'évaluer la fabricabilité de la pièce, et, en se basant sur cette décomposition, propose des modifications visant à l'améliorer (ajout de chanfreins, de dépouilles…). Ce dernier aspect est très intéressant car il permet au concepteur de comprendre les raisons d'une mauvaise fabricabilité. Dans un autre domaine, celui de l'usinage, le système proposé par [Sánchez et coll., 1997] prend en entrée un ensemble de spécifications (géométrie, tolérances, états de surface, dimensions du brut…) pour y appliquer un algorithme de reconnaissance de formes visant à définir des vecteurs d'approche pour outils coupants et un ensemble d'éléments de volumes d'enlèvement de matière (MRSEV [99]). Ce processus de reconnaissance peut être à la fois automatique et interactif. Tous les MRSEV sont ensuite exploités

98. Maquette numérique d'une usine exploitée pour détecter les problèmes liés au fonctionnement de l'usine.
99. *Material Removal Shape Element Volume*.

pour déterminer la fabricabilité de la pièce (coûts et disponibilité des ressources matérielles) en envisageant différents outils pour usiner une forme, et différentes séquences d'usinage pour les ébauches et la finition. Il y a certaines limites à ce type d'approche :

- les formes doivent toutes être connues à l'avance ;
- la reconnaissance de formes n'est pas complète et impose certaines limites (p. ex., dans le cas de [Sánchez et coll., 1997], des caractéristiques de forme parallèles aux axes) ;
- certaines informations relatives à la fabrication doivent être entrées par le concepteur (p. ex., dans le cas de [Sánchez et coll., 1997], le brut ou certaines caractéristiques de fabrication non reconnues).

Méthodes heuristiques. Contrairement aux méthodes analytiques, les méthodes heuristiques se basent sur des règles qui dérivent de l'expérience directe du fabricant et qui sont souvent définies de manière plus ou moins formelle. Elles sont donc souvent propres à chaque entreprise. Les méthodes heuristiques utilisent des règles qu'il est bon de respecter afin d'augmenter la fabricabilité des pièces. Ces règles ne se basent pas sur une analyse complète mais représentent un savoir-faire acquis au fil du temps, très souvent issu de l'observation. Elles sont donc valides au sens de l'expérience dans un contexte donné, mais non appuyées par une loi exhaustive. Les méthodes heuristiques consistent à détecter dans le modèle de la pièce des configurations particulières (des caractéristiques de fabrication) auxquelles il est possible d'associer une fabricabilité. Les règles de production sont très souvent utilisées avec ces méthodes.

Les exemples de règles heuristiques sont nombreux dans la littérature (les livres tels que [Boothroyd et coll., 2001] et [Bralla, 1999] en contiennent à profusion). Nous ne nous attarderons pas sur ce point (voir les quelques exemples de la section 2.2.3 pour une illustration). En ce qui concerne les méthodes de prise en compte des règles heuristiques, nous pouvons distinguer deux approches :

les approches qualitatives indiquent une à une les causes d'une mauvaise (ou d'une non-) fabricabilité. Par exemple, le système défini par [Ramana et coll., 2006] prend en entrée la définition d'une pièce, l'analyse, puis, en résultat, donne au concepteur des indications telles que : « la pièce a un angle vif entre les caractéristiques 1 et 2. Un congé ou rayon de courbure devrait être appliqué » ;

les approches quantitatives évaluent la fabricabilité d'un point de vue global par combinaison d'évaluations quantitatives locales (elles-mêmes évaluées de façon heuristique). Par

exemple, [Ong et coll., 2003] utilisent des indices de fabricabilité relatifs compris entre 0 et 1 pour déterminer la fabricabilité de pièces de révolution réalisées par tournage [100]. Des facteurs de coût, prédéterminés pour un ensemble de caractéristiques de forme, de tolérances, et de mécanismes de bridage et de support, sont transformés en indices de fabricabilité via des fonctions floues. Ces indices sont ensuite combinés par la méthode AHP de [Saaty, 1980] afin d'obtenir une évaluation globale de la fabricabilité tenant compte de l'importance relative des différentes caractéristiques (notamment à travers leurs fonctions), permettant ainsi la comparaison de solutions alternatives. Les auteurs considèrent trois types de coûts : les coûts de serrage [101], les coûts de support [102] et les coûts des caractéristiques de forme. L'indice de fabricabilité de la pièce est déterminé comme suit :

$$MI = \sum_{i=1}^{n} (W_i \times MI_{F_i}) + MI_{Support} + MI_{Serrage}$$

où les MI sont les indices de fabricabilité, les F_i les caractéristiques de forme, et les W_i les poids attribués par la méthode AHP. Pour chaque caractéristique F_i, l'indice de fabricabilité est déterminé en associant une opération d'usinage, et un type d'opération (élément déterminant la qualité).

2.3.4.2 Approches par variation

Les approches par variation sont quant à elles exclusivement basées sur des données héritées de conceptions d'anciennes pièces. Elles consistent donc à trouver des configurations au sein de la nouvelle pièce, déjà rencontrées pour d'autres pièces, sans qu'il n'y ait pour autant de règle ou de loi explicitement connue. Les approches par variation sont ainsi plutôt destinées à la conception routinière.

Méthodes analogiques. La technologie de groupe ou le raisonnement par cas (cf. § 1.2.2.4) sont des cadres privilégiés pour l'utilisation de telles approches, car si une base de cas existe déjà pour l'activité de conception, il est assez naturel d'y adjoindre des informations relatives à la fabricabilité des pièces. Notons que les paramètres pertinents pour l'évaluation de la fabricabilité (ceux qui forment l'index de recherche) ne sont cependant pas toujours les mêmes que

100. Ce travail se base sur une étude menée en atelier de fabrication.
101. Il s'agit des coûts provenant du mandrin.
102. Il s'agit des coûts provenant des mécanismes de type « contre-poupée » utilisés pour les pièces qui ne peuvent supporter la flèche induite par l'outil coupant (pièces pas suffisamment rigides).

pour la conception.

Dans ce type d'approche, la fabricabilité d'une pièce est déterminée en se basant sur les difficultés (ou facilités selon le point de vue) qui ont été rencontrées lors de la fabrication d'anciennes pièces de la même famille (cas de la technologie de groupe) ou de familles éventuellement différentes (cas du CBR) [Shah et Wright, 2000]. Les pièces doivent alors être décomposées en éléments pertinents du point de vue de l'objectif visé. Par exemple, le système de [Nedeß et Jacob, 1997] permet de concevoir un moule de presse à injecter en tenant compte des problèmes de fabricabilité déjà rencontrés, notamment du point de vue de la qualité. Le système proposé par [Chou, 2004] est aussi de type CBR. Les cas sont indexés sous forme de *clusters* fonction/activité auxquels sont associés des coûts (via des tables FACT [103]). [Ficko et coll., 2005] proposent un outil pour la prédiction du coût de l'outillage permettant l'estampage de feuilles métalliques. Le système sélectionne un ensemble de cas similaires afin d'améliorer l'acuité de la prédiction qui ici, est réalisée par un module basé sur un algorithme génétique. [Lee et Kazuhiro, 2002] proposent d'exploiter la technologie de groupe afin d'optimiser l'utilisation des ressources de l'atelier en regroupant des familles de pièces dans des cellules d'usinage différentes. Leur système propose aux concepteurs des suggestions de modification en fonction de critères qui améliorent la fabricabilité à travers une diminution des coûts liés aux changements de réglages des machines d'usinage.

Méthodes paramétriques. Les méthodes paramétriques utilisent des fonctions mathématiques (linéaires, exponentielles, logarithmiques…), appelées fonctions d'estimation, pour exprimer la fabricabilité relativement à un ou plusieurs paramètres indépendants liés à la pièce et/ou au procédé (p. ex. la masse ou des paramètres provenant des caractéristiques de fabrication). Ces méthodes se basent sur l'analyse de données héritées pour trouver une corrélation entre un ou plusieurs paramètres et les coûts constatés. La formulation de ces fonctions statistiques est donc l'étape centrale dans ces approches, leur exploitation restant simple. [Farineau et coll., 2001] proposent une méthode paramétrique appliquée à l'estimation de coûts d'usinage. Le principe est présenté sur la figure 2.4 : un ou des calculs de régression (p. ex. « moindres carrés ») sont appliqués aux données initiales avant que les fonctions d'estimation alors obtenues soient évaluées, affinées, puis validées par un expert. Ainsi, le processus de définition des formules d'estimation est incrémental et nécessite l'intervention d'experts en

103. *Function Activity Cost Tables*.

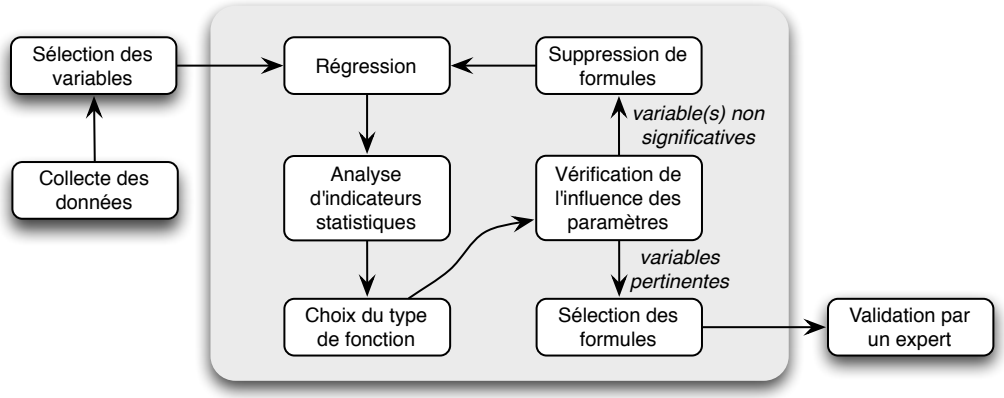


FIGURE 2.4: Mise au point de fonctions d'estimation (tiré de [Farineau et coll., 2001]).

fabrication et en méthodes de calcul de régression mathématique.

Dans ce type d'approche, les connaissances ne sont pas représentées sous forme de cas, mais synthétisées au moment de l'établissement des fonctions d'estimation. L'information recherchée est donc obtenue « à l'aveugle » par extrapolation plutôt que par dérivation de cas explicitement connus. Les paramètres les plus influents sont en revanche connus, ce qui permet d'indiquer au concepteur les causes d'une mauvaise fabricabilité. Selon [Farineau et coll., 2001], les paramètres utilisés peuvent être classifiés dans trois catégories : les entités de fabrication (p. ex. les types d'opérations d'usinage), les fonctions techniques, ou un paramètre de complexité (masse, cadence, nombre de directions d'usinage, etc.). Dans cette étude, les auteurs indiquent que les résultats obtenus sont satisfaisants et obtiennent des écarts normalisés variant entre 6.3 % et 10.9 % suivant le type de paramètre utilisé.

2.3.4.3 Autres méthodes

Il est possible de rapprocher des systèmes paramétriques les outils basés sur des réseaux de neurones artificiels, dont le fonctionnement est inspiré de celui des neurones du vivant. Partant de variables technologiques en entrée, ils utilisent des méthodes d'apprentissage de type statistique pour déduire une information, cette dernière pouvant par exemple être un coût. Il s'agit finalement, encore davantage que pour les systèmes paramétriques, de s'appuyer sur la notion de perception.

Notons, pour finir, qu'il existe des systèmes experts orientés vers le calcul de fabricabilité. Étant donné que les règles d'inférence utilisées sont censées pouvoir reproduire l'expertise

d'un spécialiste, nous pouvons considérer que ces systèmes experts utilisent l'une ou l'autre des approches génératives, suivant le type des connaissances sur lesquelles le raisonnement est effectué. Aussi, comme nous l'avons indiqué plus haut, rien n'interdit d'utiliser des approches mixtes, et c'est par exemple l'objet des travaux de [Rush et Roy, 2001] qui analysent les mécanismes d'inférence de spécialistes en estimation de coût dans le but de les intégrer à un système expert prenant en compte l'expérience passée.

2.3.4.4 Bilan

Dans cette section, nous avons décrit différents principes de réutilisation des connaissances exploitables dans le cadre d'une évaluation de fabricabilité. Nous avons distingué les approches génératives (analytiques et heuristiques) et les approches par variation (analogiques, paramétriques et neuronales). Nous avons également noté que ces différentes approches pouvaient être combinées. Les connaissances exploitables peuvent se rapporter au produit, à travers ses définitions fonctionnelles et morpho-dimensionnelles, au(x)s procédé(s) de fabrication utilisé(s) et à leur agencement dans le temps et dans l'espace (cas des cellules d'usinage), ou bien à l'expérience passée de l'entreprise.

Les méthodes analytiques se basent sur des lois physiques. Elles fournissent donc des résultats précis. Mais en contrepartie, elles requièrent des informations qui ne sont connues que tard dans le cycle de vie. En outre, les méthodes de calcul utilisées nécessitent des ressources importantes (p. ex. pour générer des séquences d'usinage), rendant leur exploitation en temps réel trop difficile. Or, l'un des objectifs fondamentaux de l'ingénierie simultanée est d'aboutir à un système de CAO qui intègre les informations relatives à la fabrication. La capacité de ce dernier à interagir directement avec l'utilisateur sur des questions de fabricabilité est donc primordiale. Les méthodes paramétriques et neuronales sont quant à elles plus intéressantes de ce point de vue. Cependant, elles nécessitent une phase de mise en place (ou d'apprentissage) qui n'est pas simple à mener, et qui est déterminante pour le succès de l'outil. Dans l'absolu, les systèmes experts permettent de combiner les trois types de connaissances évoqués plus haut, mais là encore, la phase de capitalisation des connaissances (en particulier les mécanismes d'inférence sous-jacents) requiert un investissement non négligeable. Les ressources de calcul nécessaires sont en outre assez importantes. Les méthodes analogiques répertorient l'historique de l'entreprise et fournissent des mécanismes de recherche afin de réutiliser les cas rencontrés par le passé. Cela implique d'une part que la quantité d'informations à manipuler

TABLE 2.1: Connaissances et démarches de réutilisation

	Analytiques	Heuristiques	Paramétriques, neuronales	Analogiques	Expertise
Lois physiques	♡				♡
Règles métier		♡			♡
Lois statistiques			♡		♡
Historique				♡	♡

TABLE 2.2: Résumé des avantages et inconvénients des démarches de réutilisation.

Méthodes	Avantages	Inconvénients
Analytiques	Précision.	Nécessite des informations détaillées. Calculs lourds (nécessite des mécanismes asynchrones).
Heuristiques	Rapidité de calcul ; déjà présent « sur le terrain ».	Pas prouvé rationnellement.
Analogiques	Réutilisation de l'historique, assez précis.	Nécessite expertise (phase d'adaptation). Base de données lourde. Dédié à la conception routinière.
Paramétriques, neuronales	Rapidité de calcul.	Difficile à mettre en place (phase d'apprentissage).

est importante, et d'autre part qu'il faut adapter chaque expérience au cas présent, ce qui nécessite une expertise qui n'est pas à la portée d'un système informatique. Les règles heuristiques se basent sur des connaissances métier. Même si elles sont a priori moins précises, elles ont l'avantage d'être simples à mettre en œuvre et de se baser sur certains paramètres disponibles dès la conception, ce qui autorise à envisager un traitement en temps réel. L'avantage principal des règles heuristiques est que dans la pratique, elles sont le moyen le plus utilisé pour représenter les connaissances relatives aux métiers de la fabrication (voir les exemples indiqués dans la section 2.2.3).

La table 2.1 résume les différents types de connaissances employés pour chaque démarche de réutilisation. La table 2.2 résume les principaux avantages et inconvénients des différentes approches évoquées. Dans la suite nous passons en revue différents principes de modélisation afin de mieux cerner leurs interactions possibles avec l'analyse de fabricabilité ; cette dernière s'appuie en effet sur des informations présentes dans le modèle.

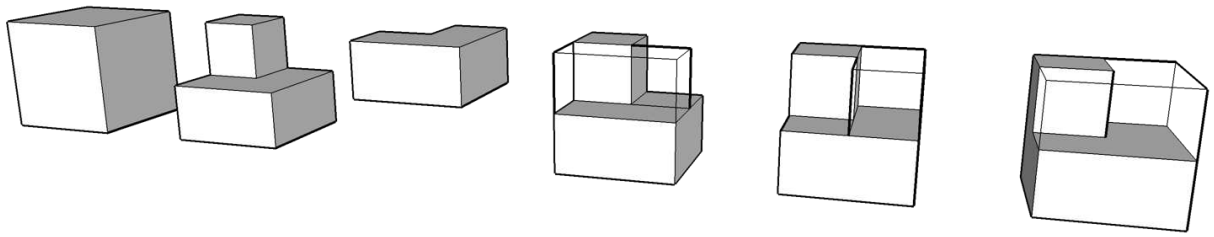


FIGURE 2.5: Interprétations multiples en termes de caractéristiques d'usinage ; tiré de [Han et coll., 2000].

2.3.5 Intégration et interaction avec les principes de modélisation

Dans toutes les approches que nous avons parcourues, les critères de fabricabilité sont systématiquement considérés à travers des attributs de la pièce extraits du modèle produit. Il existe donc un rapport important entre l'aptitude à prendre en compte ces critères, et la technique de modélisation. Dans la suite de ce paragraphe, nous considérons la fabricabilité en regard de différents principes de modélisation : les caractéristiques, les contraintes, et les modèles multi-vues.

2.3.5.1 Utilisation des caractéristiques

Du fait que les concepteurs utilisent des modèles différents de ceux des fabricants, l'accès aux attributs nécessaires à une évaluation de fabricabilité n'est pas immédiat. Notamment, les caractéristiques d'usinage sont souvent très différentes des caractéristiques de forme utilisées par les concepteurs. Une solution simple à ce problème a d'abord consisté à fournir au concepteur des outils de modélisation calqués sur les outils de fabrication (notamment avec le *design by machining features* [Cutkosky et Tenenbaum, 1990]). Il s'agissait ainsi de concevoir à l'aide de caractéristiques de fabrication, mais comme nous l'avons évoqué auparavant à propos des manuels de DFM (§ 2.3.1), ce n'est pas réaliste : ce qui est important pour un fabricant ne l'est pas nécessairement pour un concepteur, et vice-versa [Gardan, 2003]. En particulier, le concepteur ne saura pas choisir la meilleure alternative parmi des interprétations multiples, par exemple dans le cas de caractéristiques d'usinage (voir figure 2.5).

La conception par caractéristiques ne peut donc se faire qu'à partir de caractéristiques de conception (de forme, fonctionnelles…). Or, à l'égard de la fabricabilité, les entités importantes sont nécessairement les caractéristiques de fabrication. Se pose ainsi un problème de correspon-

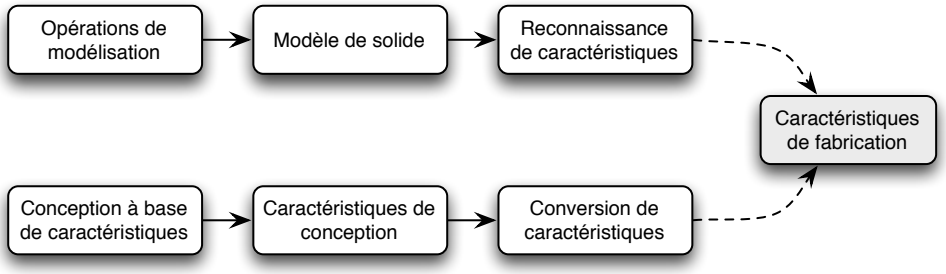


FIGURE 2.6: Extraction de caractéristiques.

dance auquel les approches d'extraction de caractéristiques tentent d'apporter une réponse.

Essentiellement, l'extraction de caractéristiques joue un rôle d'interface entre CAO et FAO. [Han et coll., 2000] distinguent la reconnaissance de caractéristiques qui se base sur un modèle de solide (p. ex. un B-Rep), et la conversion de caractéristiques qui se base sur un modèle à base de caractéristiques de conception, généralement de type CSG enrichi (voir figure 2.6). La conversion de caractéristiques exploite la non-unicité des représentations de type CSG. Elle consiste à traduire des entités de conception en entités de fabrication (souvent de type DSG [104]), ce qui parfois se fait facilement (p. ex. dans le cas d'un trou), et dans d'autres cas, nécessite de passer par une phase de raisonnement géométrique pour identifier les caractéristiques [Li et coll., 2002]. L'extraction de caractéristiques peut être appréhendée par des techniques de manipulation de graphes, de décomposition volumique, ou par des méthodes de raisonnement géométrique basées sur des indices (raisonnement sur les traces « laissées » par les caractéristiques de fabrication) [Han et coll., 2000]. Certains algorithmes nécessitent une intervention de l'utilisateur pour aider à identifier certaines caractéristiques ou pour choisir parmi plusieurs alternatives.

En réalité, les deux approches présentées sur la figure 2.6 ne sont pas exclusives et l'obtention d'un modèle à base de caractéristiques de fabrication se fait plus efficacement par leur utilisation conjointe. Par exemple, [Han et Requicha, 1997] proposent le système IF^2 *(Integrated Incremental Feature Finder)* qui permet de tirer parti d'un modèle B-Rep ainsi que d'un modèle à base de caractéristiques de formes, pour en extraire des caractéristiques de fabrication. Le système emploie une architecture de type tableau noir en combinaison avec des facteurs d'incertitude qui sont voués à favoriser les interprétations semblant les plus pertinentes (utilisation

104. *Destructive Solid Geometry*.

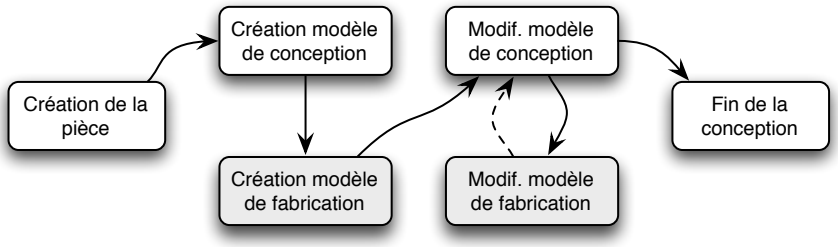


FIGURE 2.7: Reconnaissance incrémentale de caractéristiques de fabrication.

d'heuristiques).

Les algorithmes de reconnaissance de caractéristiques de fabrication sont par essence lourds et figent une vision de la façon dont la pièce sera fabriquée, même si certains systèmes proposent plusieurs interprétations à l'utilisateur. Ils s'apparentent donc aux approches guidées par la planification (cf. § 2.3.3.1) qui font appel à une certaine expertise et à des informations qui ne sont souvent disponibles qu'à un stade avancé du cycle de vie.

Néanmoins, afin de découper le calcul, des approches effectuant l'extraction des caractéristiques de manière incrémentale ont été développées à partir du milieu des années quatre vingt dix[105]. Ces approches consistent à construire le modèle d'usinage au fur et à mesure que le modèle de conception évolue (voir figure 2.7). Elles maintiennent généralement à jour une structure proche de la géométrie reliant caractéristiques de forme et d'usinage ; il s'agit p. ex. d'un graphe retraçant l'évolution des faces complété par des informations d'adjacence pour [Lee et Kim, 1998], et d'une liste de liens dynamiques pour [Chen et coll., 2005]. L'intérêt majeur avec ce type d'algorithme, est que la quantité de calcul devant être effectuée en une fois est grandement réduite. En effet, après chaque opération de modélisation, la zone à analyser étant moindre, les algorithmes d'extraction sont bien plus efficaces.

Mais la reconnaissance de caractéristiques rencontre d'autres problèmes. Lorsque des interférences apparaissent entre caractéristiques, les algorithmes doivent réinterpréter le modèle. Cette réinterprétation est problématique dans la mesure où dans certains cas, une nouvelle caractéristique de fabrication peut apparaître, ce qui est parfois difficile à détecter et nécessite souvent l'utilisation d'heuristiques.

Notons enfin que les caractéristiques extraites, même si elles sont disponibles, ne suffisent pas en tant que telles à définir la fabricabilité dans son ensemble. Quelquefois, en fonction des

105. Les premières recherches dans ce sens semblent être celles de [Laakko et Mäntylä, 1993].

heuristiques utilisées, l'extraction peut donc ne pas être adéquate (des caractéristiques de fabrication sont mal interprétées). Une exploitation de la seule géométrie ne permet pas d'obtenir toutes les informations utiles : notamment, les fonctions et les tolérances sont des éléments déterminants [Gardan, 2003]. Par exemple, le fait que deux trous soient trop proches l'un de l'autre n'est pas une information directement présente dans un modèle à caractéristiques d'usinage (cf. règle complexe, § 2.2.3). Ainsi, le fait de connaître la géométrie des trous contribuera par exemple à l'attribution d'un coût à l'opération de perçage, mais ne permettra pas de savoir que la paroi les séparant risque de se disloquer lors du second perçage. Enfin, dans le cas de l'usinage, un algorithme d'extraction ne définit que partiellement les caractéristiques de fabrication, à savoir principalement leur forme, là où un véritable système CAPP ira jusqu'à déterminer les volumes balayés, l'ordre dans lequel les opérations seront exécutées, et le fait que l'outil est adéquat et disponible en atelier. Mais la fabricabilité des caractéristiques est une question à part entière [Han et coll., 2000]. Une phase d'analyse après extraction est donc toujours indispensable.

2.3.5.2 Utilisation de contraintes

L'utilisation de contraintes est fréquente dans les modeleurs commerciaux [www-CATIA ; www-NX ; www-Pro/E], spécialement à travers la géométrie paramétrique (cf. § 1.3.2.3). Les contraintes y sont essentiellement utilisées pour décrire des relations géométriques (positionnement, dimensionnement) entre éléments géométriques et caractéristiques de forme. La modélisation sous-contraintes passe alors par trois phases [Anderl et Mendgen, 1996] : le croquis ; la détection (définition) de contraintes ; et la résolution. Mais les contraintes peuvent être appliquées à d'autres fins : modélisation par variante (modèles paramétrés), contraintes d'ingénierie (p. ex. dimensionnement de roulements), respect de normes (vis, écrous…), simulations cinématiques, dynamiques ou élastiques. Ces applications peuvent ainsi mener à l'adjonction de variables non géométriques (p. ex. : vitesse de rotation, masse…). Les contraintes d'ingénierie sont représentées par des fonctions (équations mathématiques) ou des formalismes logiques tels les règles de production [Anderl et Mendgen, 1996].

Le système proposé par [Chung et coll., 2000] est un modeleur variationnel tridimensionnel permettant d'effectuer des simulations cinématiques et dynamiques, et d'analyser des chaînes de tolérancement, en particulier suite à une modification (calcul de dérivées partielles). Afin de permettre des interactions en temps réel (principe *drag and drop*), le solveur utilise une méthode

de résolution mixte [106] (décomposition algébrique couplée à une résolution numérique).

Les équations étant éventuellement liées à des paramètres d'ingénierie, elles peuvent par conséquent être exploitées dans l'optique de la fabricabilité. [Anderl et Mendgen, 1996] indiquent en effet également une utilisation possible de la conception sous contraintes dans le but de prendre en compte des critères liés aux métiers post-conception. Par exemple, [Feng et Kusiak, 1995] proposent de représenter des contraintes de fabrication dans le cadre d'un système de conception par caractéristiques de fabrication. Sept classes différentes de caractéristiques de fabrication ayant des topologies différentes sont considérées. Les contraintes de fabrication sont modélisées par des règles de production qui sont évaluées après chaque ajout de caractéristique par le concepteur. Un aspect intéressant de cette étude est que les règles tiennent compte du contexte d'utilisation de chaque entité – notamment la classe de l'entité parente – et adaptent l'analyse en conséquence. Ceci permet entre autres de stipuler le fait qu'un trou ne doive pas être tangent à l'entité qui le porte. Cette idée, très intéressante dans le principe, gagnerait à être plus générique, les seules relations considérées ici étant les relations de type « père-fils ».

La conception sous contraintes est donc intéressante en regard de la fabricabilité, en premier lieu parce qu'elle permet d'enregistrer au sein du modèle produit des relations implicites entre entités géométriques. Ceci apporte un plus par rapport au simple fait d'attribuer des dimensions à des paramètres particuliers car la pièce peut alors être modifiée sans que le concepteur n'ait à répercuter lui-même les changements sur les autres entités (principe de la propagation de contraintes). Les relations alors formées ne sont pas figées. Dans le dernier exemple du paragraphe précédent, nous évoquions la nécessité d'analyser le positionnement relatif de deux trous juxtaposés. Ce type de vérification aurait ainsi facilement pu être effectué par l'adjonction de contraintes au modèle géométrique du produit.

Cependant, la réutilisation des contraintes est parfois problématique : lorsqu'un concepteur reprend le modèle créé par un autre concepteur, les contraintes existantes ne sont pas nécessairement parlantes pour celui-ci, notamment parce que plusieurs jeux de contraintes différents peuvent avoir des effets identiques sur une même pièce (chaque concepteur ayant sa propre façon de contraindre la géométrie). Néanmoins, un certain nombre de contraintes peuvent être

106. Les méthodes de résolution peuvent être symboliques (constructions à la règle et au compas, bases de Gröbner) ou numériques (Newton/Raphson), et éventuellement basées sur une décomposition [Maculet et Daniel, 2003]. Les méthodes de résolution numériques ont l'avantage d'être plus rapides, mais posent des problèmes de convergence et ne sont pas exactes. Les combiner avec une méthode de décomposition algébrique permet d'amoindrir ces difficultés.

intégrées aux librairies de caractéristiques de formes, de façon à pouvoir être adaptées de façon adéquate après instanciation. La réutilisation est alors en partie facilitée dans le sens où chaque caractéristique de forme devient autonome. Mais il reste indispensable de définir des contraintes inter-caractéristiques…

Parmi les modeleurs du marché, certains permettent de stipuler des règles directement dans le modèle produit. Par exemple, le logiciel CATIA (dans sa version 5) dispose de deux modules, nommés *Knowledge Expert* et *Knoledge Advisor*, autorisant l'introduction de règles [www-CATIA]. À titre d'exemple, ces modules permettent de prendre en compte le fait qu'en forge, l'utilisation d'un marteau-pilon ou d'une presse donne lieu à des attributs différents pour les dépouilles [107]. L'utilisation de ces modules permet, lorsque plusieurs concepteurs interviennent successivement sur la même pièce, de capturer la logique de conception afin de mettre à jour la géométrie de façon consistante. La réutilisation de cette connaissance pour de nouvelles pièces est néanmoins peu naturelle : les règles se basent sur les noms des entités formant les pièces qui, dans le cas général, varient d'une pièce à l'autre (en genre et en nombre).

Concernant les contraintes de fabricabilité, d'autres complications sont à prévoir. En effet, les contraintes de fabricabilité peuvent aller à l'encontre des autres contraintes (géométriques et d'ingénierie). Des mécanismes supplémentaires sont donc nécessaires pour gérer ce type de conflit. En outre, jusqu'à présent, c'est le concepteur qui est chargé d'injecter les contraintes dans le modèle, donc en particulier celles liées à la fabricabilité. Or, comme nous l'avons noté, celui-ci n'est pas compétent pour cette activité. Des moyens seraient ainsi bienvenus afin de permettre au fabricant d'introduire les contraintes de fabricabilité dans le modèle produit au fur et à mesure de son élaboration. Cette possibilité est envisagée dans le paragraphe suivant.

2.3.5.3 Approches multi-vues

La conception multi-vues consiste à maintenir cohérentes différentes vues du produit. À l'instar des systèmes multi-agents, la modélisation multi-vues permet ainsi une prise en compte du caractère distribué de l'activité de conception [108]. Une vue correspond à la façon qu'a un spécialiste d'un domaine particulier de voir le produit [de Kraker et coll., 1997 ; Rosenman et Gero, 1996]. La plupart des systèmes commerciaux permettent de créer plusieurs vues d'un même produit, mais pas d'en maintenir la cohérence. Ceci est problématique car,

107. L'extraction de la pièce étant différente du fait des frappes multiples dans le cas du marteau-pilon.
108. Les différentes vues métier peuvent en effet être assimilées à des agents communicants (voir p. ex. [Medani et Ratchev, 2006]).

après modification d'une vue, il faut répercuter à la main les modifications dans les autres vues, ce qui est une source d'incohérences.

Différentes vues font ainsi référence à des caractéristiques différentes (de forme, de fabrication, de simulation...). Le concept de modélisation multi-vues est donc clairement dédié à l'ingénierie simultanée, et en particulier à la conception guidée. Selon [de Kraker et coll., 1997], deux questions centrales concernant les approches multi-vues sont :

Comment ouvrir une nouvelle vue ? Ceci revient à réinterpréter les caractéristiques avec un point de vue différent ;

Comment propager les changements effectués sur une vue ? Ceci nécessite de maintenir la consistance des contraintes définies dans chaque vue.

Les répercussions peuvent s'effectuer à sens unique *(one-way)* ou de manière multidirectionnelle *(multi-way)* [de Kraker, 1998 ; Hoffmann et Joan-Arinyo, 1998]. Dans ce dernier cas, cela permet par exemple à l'ingénieur de fabrication de modifier le modèle défini par le concepteur. La gestion des différentes vues est généralement réalisée à travers un modèle central, utilisé pour répercuter les changements effectués dans une vue sur les autres vues. Ce modèle est générique (représentation canonique) de sorte que chaque vue est bâtie en se référant aux constituants du modèle central.

[Lee, 2005] applique par exemple ce principe afin de maintenir la consistance entre la vue conception et la vue simulation. Plusieurs niveaux d'abstraction et de résolution peuvent ainsi être utilisés en simulation [109] et évoluent sans que le modèle de conception n'ait à être reconsidéré directement par l'ingénieur en simulation.

Le modèle maître proposé par [Hoffmann et Joan-Arinyo, 1998] est fondé sur une structure de type B-Rep. Les « éléments de forme » constituant ce B-Rep sont caractérisés par un triplet (i, t, p), où i est un identifiant unique, t le type de l'élément (sommet, arrête ou face), et p un point de l'espace intérieur de l'élément permettant de l'identifier sans ambiguïté dans chaque vue (p. ex. par *picking* [110]). Le modèle maître permet ainsi une associativité avec les modèles spécifiques des différentes vues qui, du fait de sa généricité, peuvent être des systèmes quelconques (notamment des systèmes hérités) qui auront été adaptés afin de communiquer avec celui-ci. Lorsqu'une modification est effectuée dans l'une des vues, elle est dans un premier temps ré-

109. Cet élément est pertinent en simulation dans le cadre de l'utilisation d'hypothèses simplificatrices visant à réduire les calculs.
110. Technique consistant à extraire un élément du modèle par restriction de la zone de recherche (pointage).

percutée sur le modèle de la vue CAO, puis, en cas de succès, est appliquée au modèle maître qui se charge de la diffuser à l'ensemble des autres vues. Un graphe acyclique répertorie alors les différentes opérations [111] appliquées aux éléments de forme affectés afin que, dans chaque vue, l'associativité avec le modèle maître puisse être révisée. Des contraintes peuvent également être appliquées aux vues. Lorsqu'un conflit survient à ce niveau, un mécanisme de conciliation de contraintes est invoqué afin de tenter de maintenir la cohérence globale de l'ensemble des modèles privés. La façon dont chaque vue est mise à jour après modification du modèle maître est fortement dépendante du domaine d'application, et doit donc être définie au cas par cas. Les auteurs proposent quelques exemples d'application, mais indiquent qu'automatiser ces mises à jour n'est pas possible dans tous les cas, et que cela pose de nombreux problèmes, en particulier vis-à-vis de la conciliation des contraintes [Hoffmann et Joan-Arinyo, 2000].

Le modèle de [de Kraker, 1998] est quant à lui basé sur un ensemble de cellules (volumes quasi-disjoints) correspondant à toutes les intersections possibles des caractéristiques instanciées dans les vues. Dans chaque vue, une caractéristique est donc formée à partir d'un ensemble de cellules pleines ou vides, et des contraintes peuvent être définies en se basant sur ces mêmes cellules. Ainsi, lorsqu'une cellule est ajoutée, supprimée ou modifiée par une vue, les autres vues sont ajustées en conséquence [112]. Il peut apparaître deux types de conflits : les conflits de contraintes (p. ex. la modification d'une dimension conduisant à une violation de contrainte dans une autre vue) ; et les conflits topologiques (p. ex. l'ajout ou la suppression explicite d'une caractéristique). Lors d'un conflit topologique, des vues peuvent avoir à être réinterprétées, ou dans certains cas supprimées et remplacées par des vues plus adéquates [113]. Les interprétations correspondant aux différentes vues sont générées automatiquement par des algorithmes de raisonnement géométrique. Une politique de prescription de classes de caractéristiques à rechercher est donc adoptée dans chaque vue. Les instances de caractéristiques sont extraites par comparaison de la configuration des faces de cellules candidates avec celle des faces des classes prescrites. La combinatoire de ce type de traitement est lourde, même si l'arbre de recherche est élagué par des heuristiques prédéfinies pour chaque classe de caractéristique. Cette approche a été étendue afin de prendre en compte les particularités des assemblages [Bronsvoort et Noort, 2004]. Deux problèmes majeurs se posent : tout d'abord, pour des

111. Un élément de forme peut être soumis à l'une des six opérations suivantes : déplacement, effacement, fusion, division, changement de taille, création.
112. Un système de priorité permet néanmoins d'interdire certaines modifications.
113. Ceci qui peut par exemple correspondre à un changement de procédé de fabrication.

pièces complexes, le nombre de cellules explose, ce qui alourdit sévèrement les temps de réinterprétation ; en second lieu, les algorithmes de recherche de caractéristiques peuvent, à notre sens, être difficiles à spécifier par les protagonistes.

Les approches présentées dans ce paragraphe offrent un cadre pertinent pour diffuser des changements opérés dans une vue, et pour détecter les conflits survenant entre les acteurs métier pendant qu'ils travaillent sur la même pièce. Elles fournissent de fait une interopérabilité syntaxique. Cependant, des difficultés incontestables surviennent lorsqu'il faut réinterpréter les changements de manière automatique – il s'agit là du problème d'interopérabilité sémantique évoqué au § 1.4.1. La nécessité de l'intervention d'un utilisateur ne peut donc être définitivement écartée. L'utilisation d'un modèle canonique apporte néanmoins un atout indéniable : le respect de la confidentialité. En effet, le modèle central étant constitué d'entités élémentaires, toutes les données détenues par une vue n'ont pas besoin d'êtres dévoilées aux autres vues pour que la consistance globale puisse être garantie. En somme, le dialogue s'effectue avec des entités de granularité si fine que le savoir-faire est protégé.

2.4 Synthèse

Dans ce chapitre, nous avons défini le concept de fabricabilité à travers différents indicateurs et référentiels qui peuvent être employés. Deux facettes de ce concept doivent être considérées dans le cycle de vie du produit : l'étude de faisabilité et l'analyse quantitative. Nous avons également noté l'importance du concept de règle de fabricabilité, notamment du fait de son omniprésence dans les ouvrages de référence (savoir-faire commun) ainsi que dans l'industrie (savoir-faire privé). Au cours de notre investigation, est apparue la nécessité absolue de l'utiliser afin de prendre en compte le contexte d'utilisation des indicateurs de fabricabilité.

Nous constatons que l'ingénierie simultanée et la conception guidée agissent sur deux registres : le temps de développement, par le chevauchement des phases du cycle de vie ; et la qualité de ce développement, par l'échange des compétences et la concertation. Même si certains vecteurs de l'ingénierie simultanée sont automatisables, l'idée d'une automatisation complète des flux nous paraît irréaliste, et la part décisionnelle doit être soumise à l'utilisateur. Certains aspects peuvent néanmoins être pris en charge par le système de CAO, de manière plus ou moins complète selon la disponibilité et la fiabilité des informations requises pour évaluer les indicateurs.

Nous avons parcouru un certain nombre de méthodes d'évaluation de la fabricabilité, et avons distingué plusieurs catégories de méthodes (analytiques, analogiques, heuristiques, paramétriques et neuronales), tout en remarquant qu'elles ne sont pas exclusives. Après ce panorama, il ressort qu'elles peuvent être situées selon trois axes, ceux-ci étant somme toute assez corrélés :

1. Leur caractère synchrone ou asynchrone. Les méthodes guidées par la planification, hormis peut-être les méthodes incrémentales, ne peuvent à l'évidence pas être mises en œuvre de façon synchrone ;
2. L'état d'avancement nécessaire à l'obtention des informations utiles. Certaines caractéristiques, ne sont en effet totalement connues qu'à un stade avancé du cycle de vie (tolérancement, disponibilité des ressources en atelier...), alors que d'autres le sont plus tôt (topologie, nombre de composants et d'entités, procédé, matériau, masse de la pièce...) ;
3. La précision (justesse, fidélité, sensibilité). Naturellement, plus il y a de paramètres disponibles, plus les estimations ont des chances de se rapprocher de la réalité. Dans le cas général, il est par exemple très difficile de prédire avec acuité les coûts en début de conception. Mais la précision dépend aussi de la finesse de la méthode d'évaluation. Par exemple, les méthodes guidées par la planification permettront d'obtenir plus d'informations.

Pour les méthodes asynchrones, des informations précises sont en général néanmoins requises. Si cet élément est restrictif ou contraignant, nous pouvons prévoir à l'avance la phase du processus de conception à partir de laquelle l'évaluation pourra être effectuée. Ce type d'évaluation s'insérera donc dans le processus de conception à un point prédéterminé. Les méthodes synchrones constituent un complément intéressant du fait qu'elles permettent de traiter les problèmes détectables au plus tôt de la conception. Cette double approche semble être la plus apte à offrir un support optimal de la fabricabilité, évitant de trop longs rebouclages, et validant la conception de manière plus précise à certains points clefs du processus de conception.

En dernier lieu, les différentes méthodes d'évaluation de la fabricabilité nécessitent des informations contenues dans le modèle produit, et il n'est donc pas possible d'envisager l'analyse de fabricabilité sans considérer la technique d'élaboration du modèle produit. Nous avons noté que les techniques d'extraction de caractéristiques rencontrent des difficultés, en particulier vis-à-vis des interprétations multiples, qu'il n'était pas aisé de les mettre en œuvre en

temps réel, et qu'elles ne suffisaient pas pour aboutir à une prise en compte de tous les aspects relatifs à la fabricabilité. Enfin, une tendance générale à l'intégration de la modélisation et de la conversion des caractéristiques a été constatée dans le cadre des systèmes multi-vues. Mais nous avons remarqué que des problèmes subsistaient concernant le caractère automatique des réinterprétations et de la conciliation des contraintes.

Le chapitre 1 a mis en évidence deux points qui nous semblent essentiels :

1. L'importance d'une considération des critères de fabricabilité au plus tôt dans le cycle de vie du produit ;
2. La nécessité de gérer le processus de conception de manière adaptée, en premier lieu dans le contexte de la conception en plateau projet virtuel. Ceci implique une liaison forte entre le produit (fonctions, géométrie…), le processus (besoin de planification) et les procédés (prise en compte des paramètres métier).

Concernant le second point, les travaux de thèse de [Gardan, 2005] ont permis la définition d'une méthodologie de conception collaborative au sein de laquelle les interactions synchrones et asynchrones entre concepteurs et fabricants sont d'ores et déjà prises en compte. Le système DIJA est ainsi en mesure de guider la conception via des interactions collaboratives qui s'inscrivent dans le processus. Ces interactions passent par des paramètres dits grapho-numériques et des outils de communication qui mettent en correspondance les caractéristiques métier émanant des différents protagonistes du projet. Le fait que le fabricant modifie une surface fonctionnelle peut ainsi déclencher un scénario collaboratif.

Dans le cadre de cette thèse nous nous focalisons par conséquent sur une prise en compte de la fabricabilité ne nécessitant pas d'intervention directe des fabricants. Nous souhaitons ainsi offrir au fabricant la possibilité de stipuler des règles de fabricabilité a priori applicables à tout type de procédé, en nous appuyant sur les informations dès qu'elles sont disponibles, et sans que l'on ne sache à l'avance à quel moment elle le seront. Ces informations peuvent donc être de nature fonctionnelle, géométrique, ou liées au procédé. Enfin, nous nous interdisons de dévoiler directement les règles métier au concepteur pour des questions de confidentialité. Ainsi, dans le chapitre 3, nous présentons une approche de modélisation guidée par la fabricabilité s'intégrant au projet DIJA, et rendant interopérables les outils de CAO et des connaissances de fabrication.

Chapitre III

Un outil de modélisation guidée par le savoir-faire des fabricants

Sommaire

3.1 Introduction 95

3.2 Concepts fondateurs 96

3.2.1 Le projet DIJA 96

3.2.2 Analyse et positionnement 106

3.3 Proposition pour la représentation des connaissances liées à la fabrication dans DIJA 110

3.3.1 Modélisation des règles de fabricabilité 110

3.3.2 Notion d'état pour la conception 114

3.3.3 Interactivité 116

3.3.4 Synthèse 117

3.4 Gestion de la fabricabilité 118

3.4.1 Processus d'évaluation 119

3.4.2 Prise en compte statique des règles 119

3.4.3 Prise en compte dynamique des règles 126

3.4.4 Synthèse 141

3.5 Bilan 141

3.1 Introduction

Le chapitre 2 a mis en évidence la nécessité de développer des outils génériques permettant la prise en compte de critères de fabricabilité au plus tôt dans le cycle de vie du produit. Nous y avons caractérisé le concept de fabricabilité et présenté différentes méthodes d'évaluation. Ces dernières ont été situées selon trois axes : le caractère synchrone ou asynchrone ; la maturité des informations requises ; et la précision de l'évaluation.

Dans ce chapitre, nous présentons une approche de prise en compte de règles de fabricabilité basée sur le projet fédérateur DIJA développé au sein de l'équipe CMCAO (projet déjà évoqué en introduction de ce manuscrit). Cette approche est générique au sens où rien n'est présupposé quant aux métiers de fabrication concernés par les règles. De même, tous les types de facteurs de fabricabilité (triptyque coûts, qualité, délais) sont considérés sous une optique commune. En outre, nous proposons un système fonctionnant en continu lors de la conception. De ce fait, le spectre des règles considérées se limite à celles qui ne nécessitent pas de planification : les approches planifiées sous-tendent des traitements qui, pour l'heure, nous paraissent incompatibles avec un service en continu, en particulier à cause de l'existence d'interprétations multiples. L'approche utilisée ici n'interdit cependant pas la prise en compte de telles règles, celles-ci ne donnant alors lieu qu'à des évaluations ponctuelles.

Notre proposition consiste en une modélisation des connaissances de fabrication par l'intermédiaire de règles de production, et en l'utilisation (et la réutilisation) de ce savoir-faire au travers de l'approche de modélisation synthétique du projet DIJA. Le principe adopté consiste à faire en sorte que le système de CAO assiste, guide et sensibilise le concepteur au cours de la conception, en observant l'influence de ses actes sur les possibilités et conditions de fabrication de la pièce. Ainsi, dès qu'un critère de fabricabilité est modifié, le concepteur s'en voit informé par le système qui lui propose alors différents choix, ceux-ci consistant le plus souvent à accepter des conditions pour poursuivre la conception (p. ex. des surcoûts de production ou des délais accrus), ou bien à annuler l'action initiée. Ce processus est mené tout en garantissant le respect de la confidentialité du fait de l'utilisation d'un modèle canonique qui, comme nous l'avons vu au chapitre 2, garantit une séparation nette entre savoir-faire de conception et de fabrication.

Dans la suite, nous présentons les principes fondamentaux du système DIJA sur lequel se base notre approche, et la façon dont nous abordons la gestion du savoir-faire de fabrication à

travers celui-ci (sect. 3.2). La section 3.3 présente la façon dont nous proposons de représenter la connaissance, tandis que la section 3.4 présente le modèle proposé pour prendre en compte les règles de fabricabilité, et maintenir leur cohérence tout au long de la conception. La section 3.5 est un bilan du travail présenté dans ce chapitre.

3.2 Concepts fondateurs

Dans cette section, nous présentons tout d'abord les concepts fondamentaux du projet DIJA sur lequel ce travail de recherche se base (§ 3.2.1), puis nous l'analysons afin de positionner notre approche (§ 3.2.2). Cette présentation nous semble nécessaire avant d'introduire la structure utilisée pour représenter les règles de fabricabilité, en l'occurrence, des règles de production auxquelles nous adjoignons une notion d'état.

3.2.1 Le projet DIJA

Dans cette section, nous présentons l'approche DIJA dans sa globalité avant d'expliciter le modèle d'architecture et les mécanismes internes employés pour mettre en œuvre un processus de conception.

3.2.1.1 Présentation

Les travaux présentés dans ce mémoire s'inscrivent dans le cadre de DIJA. Il s'agit d'un projet fédérateur visant à définir un environnement de CAO distribué accessible par l'Internet. Un des enjeux principaux de ce projet est l'exploitation des NTIC (Nouvelles Technologies de l'Information et de la Communication) dans le cadre de la modélisation fonctionnelle et de l'entreprise étendue (cf. sect. 1.4). Il fait appel à des domaines aussi variés que la modélisation géométrique, les architectures distribuées, les IHM, le travail collaboratif ou la gestion des connaissances, et se construit ainsi autour de différents travaux de thèse [Danesi, 2002 ; Denis, 2003 ; Gardan, 2005 ; Heschung, 2005].

Afin d'obtenir un système le plus intuitif possible, la méthode de conception utilisée, dite synthétique, s'oppose aux approches constructives qui sont ascendantes [114]. Ainsi, la modélisation d'un objet est abordée selon le principe suivant :

114. La méthodologie de conception employée dans le cadre de DIJA est ainsi descendante (cf. § 1.2.2.1).

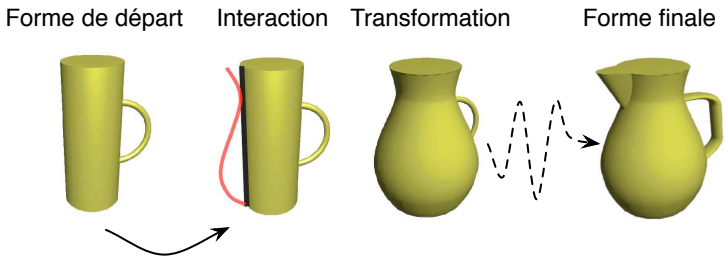


FIGURE 3.1: Approche synthétique de la conception.

– une forme approchée de l'objet est dans un premier temps obtenue soit par esquisse, soit par sélection dans une base de données. La recherche peut alors être effectuée à l'aide de critères liés à la forme de l'objet ou bien à sa fonction ;
– l'objet est ensuite transformé, à l'aide d'outils respectant certaines contraintes fonctionnelles, de façon incrémentale jusqu'à obtention du résultat escompté. Ces opérations conservent la topologie du solide.

La figure 3.1 illustre ce principe à travers la conception d'un pichet. Une forme de départ est choisie dans une librairie métier. Cette forme est ensuite transformée par application successive d'outils dédiés à ce type de pièce, jusqu'à obtention de la forme finale. La phase d'interaction correspond à un paramétrage et peut donc faire appel à d'autres solutions que l'esquisse (p. ex. la saisie de paramètres) ; comme nous le verrons plus loin, ce paramétrage est contrôlé afin de respecter les critères d'un métier de conception donné.

Les opérations élémentaires permettant de transformer la pièce sont au nombre de trois :

division : cette opération permet de séparer l'objet en plusieurs parties ;

déformation : cette opération permet de déformer un objet ou une partie d'un objet, soit en esquissant sa nouvelle forme, soit en la choisissant dans une base de données métier. Dans ce dernier cas, il est souvent nécessaire de paramétrer la forme en question ;

transmutation : cette opération permet de changer la nature d'un objet ou de l'une de ses parties (p. ex. : transformation d'un contour polyédrique en contour circulaire).

La figure 3.2 illustre ces trois types de transformation avec la conception d'un pilastre. La forme de départ est ici un simple cylindre dont la fonction est de supporter une charge. Celui-ci est d'abord divisé en trois parties correspondant à différentes sous-fonctions : le chapiteau (partie supérieure), le fût (partie centrale) et la base (partie inférieure). Le chapiteau et la base

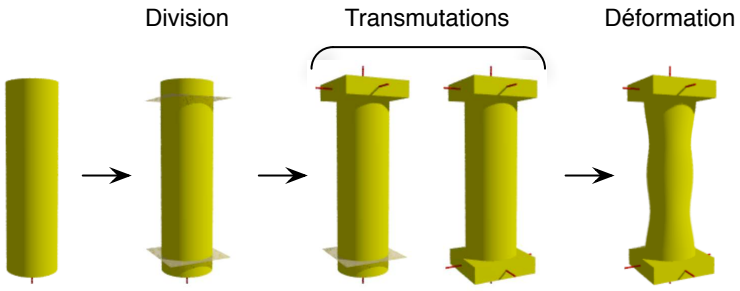


FIGURE 3.2: Opérations de transformation élémentaires de DIJA.

sont destinés à supporter la jonction avec le bâti et la rampe ; les cylindres correspondants sont ainsi transmutés en parallélépipèdes. Pour finir, le fût est déformé par esquisse afin d'obtenir une allure plus élégante.

Bien qu'assez simples, ces exemples illustrent bien la démarche synthétique adoptée (cf. annexe B pour un exemple de modélisation complet). Les fonctions et formes qui constituent l'objet sont raffinées par le concepteur qui les fait évoluer conjointement. Cette façon d'opérer s'adapte bien à la façon dont un concepteur raisonne lorsqu'il conçoit : il part d'une idée globale, assez floue, pour aboutir par affinements successifs au résultat recherché.

Afin de conserver une associativité entre la forme et les fonctions, l'ensemble des actions menées par l'utilisateur sont enregistrées dans un historique. De cette manière, chaque action est révocable et une trace de l'intention du concepteur est conservée.

Le modèle employé dans cette approche est un B-Rep (polygonal ou non), enrichi par des *éléments de dialogue*. Les éléments de dialogue (ED) définissent la géométrie de l'objet, et permettent à l'utilisateur d'interagir avec celui-ci de manière simple et intuitive. Ils représentent ses caractéristiques visibles et sont de trois types (cf. figure 3.3) :

faces : il s'agit de surfaces homogènes de l'objet (donc non nécessairement planes) ;

lignes et contours caractéristiques : il s'agit de lignes visibles à la surface de l'objet ;

fibres : il s'agit de lignes décrivant la silhouette de l'objet. Pour les formes de révolution, elles peuvent être axiales ou de révolution (pour un exemple, voir l'élément à droite sur la figure 3.3).

Les trois transformations de base indiquées précédemment peuvent naturellement être combinées afin de donner naissance à des outils plus complexes, qui auront une sémantique plus élevée, et différente selon le métier concerné. Ces transformations sont paramétrées par des

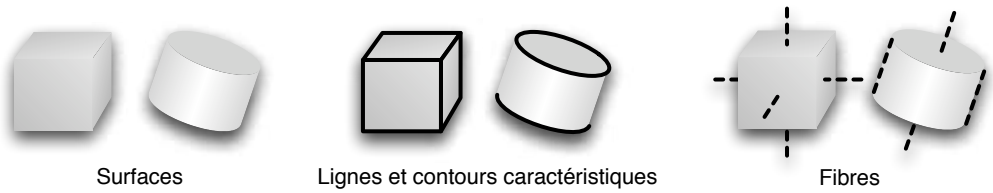


FIGURE 3.3: Éléments de dialogue.

éléments de dialogue, et agissent sur des éléments de dialogue. Ainsi, selon l'outil et le (ou les) élément(s) de dialogue impliqués, il est possible d'agir sur la forme globale de l'objet, ou bien d'opérer des transformations locales. Ceci garantit une représentativité d'un large panel de domaines d'application.

Les outils disponibles et l'ordre dans lequel ceux-ci sont appliqués dépendent du métier mis en œuvre. L'environnement DIJA étant distribué et multi-acteurs, il est prévu la possibilité de définir des processus métier sous forme de *workflows*. Les tâches sont alors décrites à différents niveaux d'abstraction et constituent des scénarii de conception qui peuvent être collaboratifs. Dans le paragraphe suivant, nous décrivons l'architecture qui supporte cette structuration de la conception.

3.2.1.2 Architecture

En tant que système de modélisation fonctionnelle, DIJA doit être en mesure de gérer différents besoins liés à la décomposition fonctionnelle (conception du produit). Notamment il doit gérer les règles de décomposition et de structuration du raisonnement, les aspects liés à l'interaction avec l'utilisateur, ou encore la visualisation du modèle. C'est ainsi que l'environnement de CAO est agencé en cinq modules [Danesi et coll., 2003] ; chaque module a un rôle spécifique et peut communiquer avec un ou plusieurs autres modules (cf. figure 3.4) :

le module de connaissances (MDC) contient les connaissances nécessaires au fonctionnement du logiciel. Le terme connaissances est ici à prendre au sens large : il s'agit de scénarii de collaboration, de règles de conception, de définitions géométriques, de contraintes topologiques… Le MDC définit les conditions et les termes dans lesquels la conception peut s'effectuer ;

le module d'interaction homme-machine (MIHM) permet la communication avec l'utilisa-

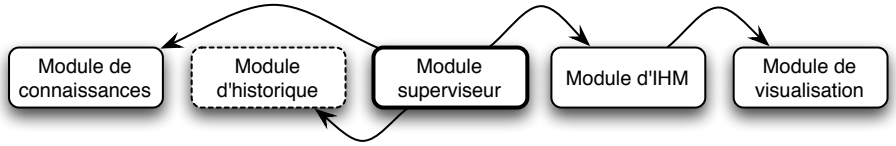


FIGURE 3.4: Décomposition modulaire de DIJA.

teur. Il a un rôle différent suivant le contexte d'utilisation : saisie d'un paramètre, pointage d'une entité visuelle... Ce module constitue le lien entre les connaissances mises en œuvre dans le MDC et l'utilisateur ;

le module de visualisation (MV) permet l'affichage. Là encore, le résultat peut être différent selon le contexte : visualisation plus ou moins détaillée, masquage de certains paramètres...

le module d'historique (MH) enregistre l'ensemble des actions entreprises au niveau de la base de connaissances, afin de pouvoir revenir en arrière, et de garder une trace de l'intention du concepteur. En parallèle de l'historique, est construit un graphe qui permet de maintenir sa consistance (notamment de le réévaluer en cas de modification d'éléments créés antérieurement) [Denis, 2003] ;

le module superviseur (MS) est le centre nerveux de l'architecture. Il assure la communication et la coordination entre les autres modules.

Afin de découper le raisonnement, le système DIJA adopte une telle architecture sur cinq niveaux distincts (cf. figure 3.5) [Danesi et coll., 2003]. Il s'agit de couches correspondant à différents degrés d'abstraction dans le processus de conception. La conception peut ainsi être définie comme une succession d'étapes globales qui sont ensuite décomposées dans les niveaux inférieurs en actions plus spécifiques, ayant un impact sur des fragments du modèle de plus en plus fins. Les niveaux d'abstraction définis sont les suivants (du plus abstrait au plus concret) :

le niveau application décrit les possibilités de l'application. Il permet de décrire des actions telles que sélectionner le métier souhaité, adopter un certain scénario de conception ou charger des règles métier, et d'adapter la constitution de l'interface du logiciel en conséquence ;

le niveau métier permet de définir le vocabulaire, les outils et les règles propres à chaque métier. Ainsi, une opération pourra être décomposée de manière différente suivant le métier

sélectionné au niveau application. Par exemple, opérer un trou dans une pièce donnera lieu à des traitements différents selon que le domaine sélectionné est l'usinage ou la plasturgie ;

le niveau commun comprend des éléments communs à plusieurs métiers. À ce stade de représentation, il est par exemple possible de trouver dans le module de connaissances les caractéristiques de matériaux, ou des scénarii permettant de créer des formes communes à plusieurs métiers (concepts génériques). Ceci évite les définitions redondantes ;

le niveau dialogue est celui des éléments de dialogue. Il permet une interaction intuitive avec l'utilisateur : ce dernier peut manipuler une géométrie complexe à l'aide d'un nombre restreint d'éléments de dialogue. À ce niveau, les entités de haut niveau sémantique sont définies en termes d'éléments de dialogue. Par exemple, créer un trou est décomposé en une suite d'opérations permettant de créer les faces, les coutours, les fibres, etc. qui constituent « l'armature » du trou ;

le niveau géométrique correspond aux entités géométriques de plus bas niveau (facettes, arêtes, sommets…) et peut varier selon le contexte matériel ou le rôle de l'utilisateur du logiciel. Par exemple, les entités géométriques pourront être celles d'un B-Rep sur un poste de travail standard, ou des éléments finis sur un serveur de calcul.

À chaque niveau d'abstraction, l'ensemble des possibilités offertes est assimilé à un jeu d'instructions. Pendant l'exécution, si une instruction n'est pas directement compréhensible par le module de connaissances du niveau concerné, alors le module superviseur la délègue au niveau inférieur, qui l'exécute s'il est en mesure de le faire, ou la délègue à nouveau sinon. Il se charge, le cas échéant, de retransmettre au(x) module(s) supérieur(s) un message de fin d'exécution éventuellement porteur d'un résultat. Suivant le type d'instruction à interpréter, le module superviseur aiguille le traitement vers un module adapté (voir l'algorithme 3.1).

Les différents utilisateurs du système travaillant en réseau, leurs postes sont amenés à communiquer au cours de la conception. Le protocole de communication utilisé est calqué sur cette décomposition fonctionnelle de l'activité de conception [Denis, 2003]. Ainsi, en plus d'être reliés entre eux, les modules superviseurs sont connectés au réseau. Un modèle maître de la pièce est enregistré sur un serveur qui se charge de communiquer aux clients les instructions leur permettant de le reconstruire localement (le plus souvent partiellement). À l'inverse, lorsqu'un client modifie la pièce, il communique les instructions correspondantes au serveur qui

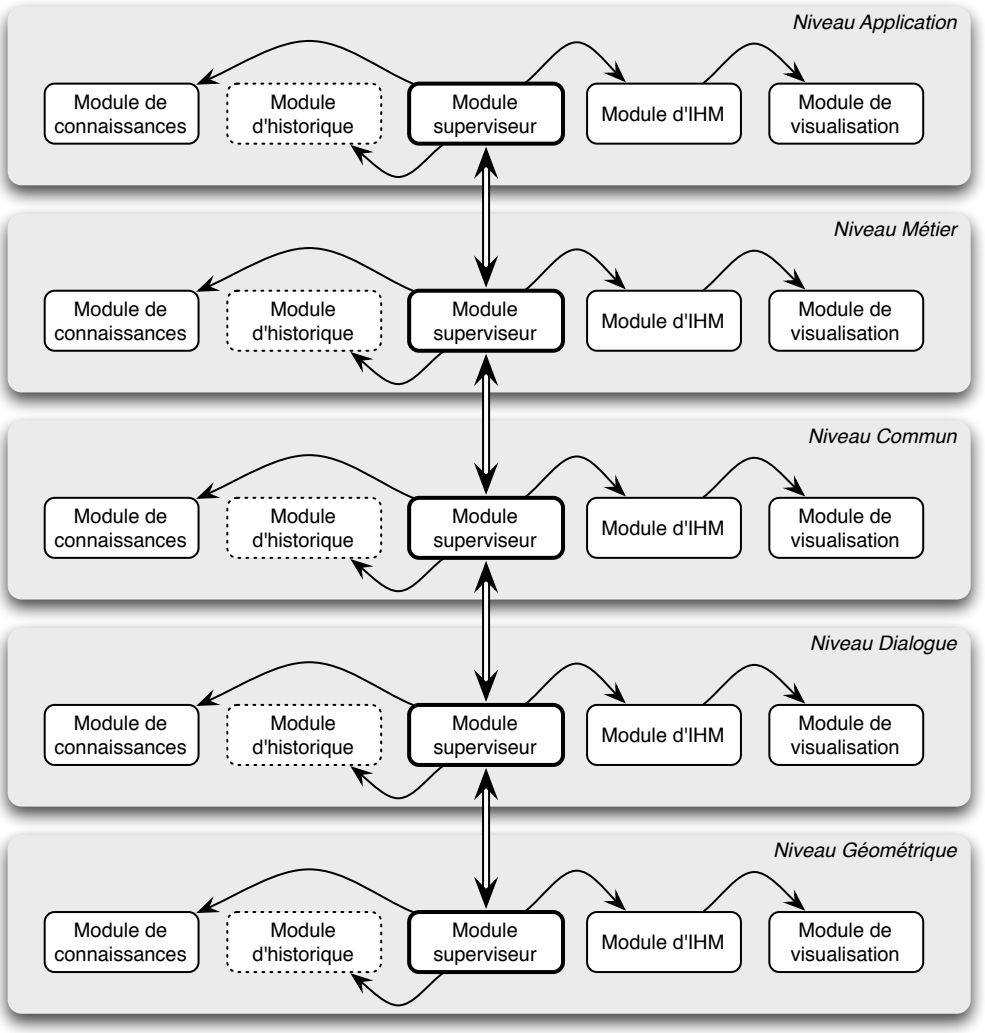


FIGURE 3.5: Les différents niveaux d'abstraction de DIJA.

ALGORITHME 3.1: Traitement d'une instruction i par le superviseur.

procédure traitement(i : instruction)

si niveau(i) = `niveauCourant` **alors**
  envoyer i au MH
fin si
si interactif(i) **alors**
  envoyer i au MIHM
  attendre le résultat r
  traitement(r)
sinon, si executable(i) **alors**
  exécuter i
sinon
  demander au MDC de décomposer i en un scénario s
  pour tout $i' \in s$ **faire**
    traitement(i')
  fin pour
fin si

se charge ensuite de les répercuter sur les autres clients si nécessaire. Les conflits éventuels peuvent être évités par un mécanisme de blocage partiel [Denis, 2003]. De tels conflits peuvent survenir si un élément est modifié localement alors même qu'un traitement distant initié préalablement (p. ex. un calcul) est voué à produire des effets sur ce même élément. Le blocage consiste alors à geler les éléments sensibles tout en laissant la possibilité de modifier le reste de la pièce. À la fin du traitement, le déblocage donne lieu à une re-synchronisation des identifiants des instructions opérées entre-temps. Ce mécanisme de blocage peut par exemple être utilisé dans le cadre d'une évaluation de fabricabilité asynchrone.

Outre le travail en parallèle, le protocole de communication permet de répartir la charge du réseau et la charge de calcul de manière optimale[115]. Nous avons donc un modèle semi-répliqué, avec un degré de réplication qui varie selon le besoin et selon le contexte. Par exemple, un modèle peut être moins détaillé sur un poste a destiné à la visualisation, que sur un poste b destiné à la simulation. De même, selon le métier, des modules différents peuvent être utilisés (p. ex. les termes métiers ou l'IHM peuvent varier), ceux-ci étant chargés à la demande des postes clients depuis le serveur [Denis, 2003]. Le chargement des modules à la volée permet

115. Si les instructions échangées sur le réseau ont un niveau d'abstraction élevé, elles seront moins volumineuses, mais nécessiteront des calculs plus conséquents. À l'inverse, des instructions de bas niveau demanderont moins de calcul mais plus de bande passante.

également des mises à jour plus souples. En somme, l'environnement DIJA peut se décliner en un grand nombre de variantes correspondant à différents savoir-faire et différents points de vue.

Dans le paragraphe suivant, nous décrivons la manière dont l'architecture, que nous venons de présenter, est utilisée pour offrir une gestion collaborative du processus de conception. Cette dernière est définie à travers l'approche baptisée Pro^3.

3.2.1.3 Processus de conception

L'approche Pro^3 (Produit, Procédé, Processus) [Gardan, 2005] est une méthodologie de conception qui définit un mode d'exploitation du MDC visant à mener un processus de conception collaborative. Le MDC (cf. § 3.2.1.2) s'appuie sur des scénarii collaboratifs pour définir le déroulement du *Processus*. Ainsi, l'ensemble des tâches que les acteurs doivent effectuer sont coordonnées par un méta-scénario, qui se matérialise par un *workflow* enregistré au niveau application. C'est l'exécution de ce *workflow* qui aboutit à la construction du modèle du *Produit*, par raffinements successifs dans les niveaux inférieurs. Chaque étape du *workflow* représente un *Procédé* qui peut donc être décomposé à son tour en un scénario. Pour tout scénario, le paramétrage du modèle se fait par l'intermédiaire d'expressions et de paramètres dits grapho-numériques [116]. Ces derniers permettent de définir le lien entre les fonctions, la connaissance et la géométrie de la pièce [Danesi et coll., 2007]. Par exemple, l'expression grapho-numérique (EGN) suivante fait appel à un scénario qui permet de créer un trou à partir de son diamètre et de sa face d'appui :

```
créer|trou(de#diamètre/)(de#face#d#appui/)
```

Dans cet exemple, les paramètres grapho-numériques (PGN) correspondent au diamètre et à la face d'appui. La détermination de ces paramètres peut nécessiter une interaction avec l'utilisateur ou une résolution. Ainsi, les EGN sont des instructions telles qu'évoquées plus haut (exécutables, interactives ou décomposables) et peuvent donc à leur tour servir de paramètres dans d'autres EGN.

Ces expressions sont associées à des contraintes et des règles métier qui, pour chaque niveau d'abstraction, permettent une gestion cohérente de la co-conception (cf. figure 3.6). Celles-

116. L'origine de ce terme tient au caractère tantôt numérique (si la valeur est connue) et tantôt alphanumérique ou géométrique des paramètres manipulés par l'utilisateur d'un système de CAO. Ce terme, initialement proposé par [Gardan, 1982], a été remodelé pour répondre aux besoins spécifiques du projet DIJA [Gardan, 2005].

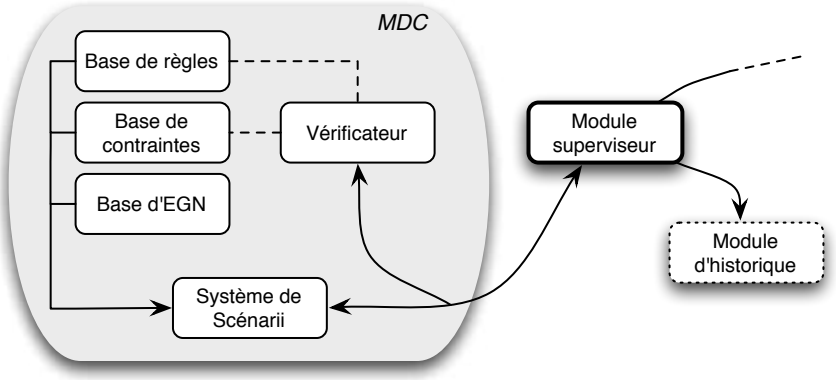


FIGURE 3.6: Architecture du module de connaissances.

ci garantissent le respect de critères liés au métier choisi : elles permettent, via un vérificateur, de s'assurer que les règles et contraintes ne sont pas invalidées par une action entreprise par le superviseur. Cette approche a par exemple permis de définir un logiciel de conception de systèmes de remplissage en fonderie [Gardan et Gardan, 2003]. Le processus de conception a été découpé en scénarii, et les règles de dimensionnement des éléments constitutifs du système de remplissage (canaux, descentes, godets, masselottes...) ont été injectées dans le MDC, de sorte que toute modélisation effectuée via le logiciel les respecte nécessairement (cf. figure 3.7).

L'approche Pro^3 se base sur l'hypothèse que l'on connaît suffisamment le processus de conception pour le découper en sous-tâches jusqu'au niveau géométrique. Ainsi découpée, la conception est relativement déterministe, même si elle est paramétrée par les choix de l'utilisateur et qu'il est possible de laisser le concepteur libre de choisir les outils qu'il applique durant certaines phases. La connaissance enregistrée dans le MDC n'en est pas moins statique, au sens où l'ajout d'EGN, de règles et de contraintes se fait préalablement à la conception. Par conséquent, s'il y a interaction ou interférence entre des critères de conception [117], cela doit être prévu à l'avance et traité à un stade déterminé. La structuration des connaissances doit ainsi être remaniée pour chaque nouveau processus, ce qui, bien sûr, n'interdit pas la réutilisation de scénarii prédéfinis.

La hiérarchie fonctionnelle, telle qu'elle est définie, permet de gérer la connaissance liée à une conception orientée métier. Malgré l'emploi du terme « métier », il faut reconnaître que jusqu'alors, un seul métier est considéré à la fois à un instant t. Le but que nous poursuivons

117. Et à plus forte raison entre des critères de conception et de fabrication...

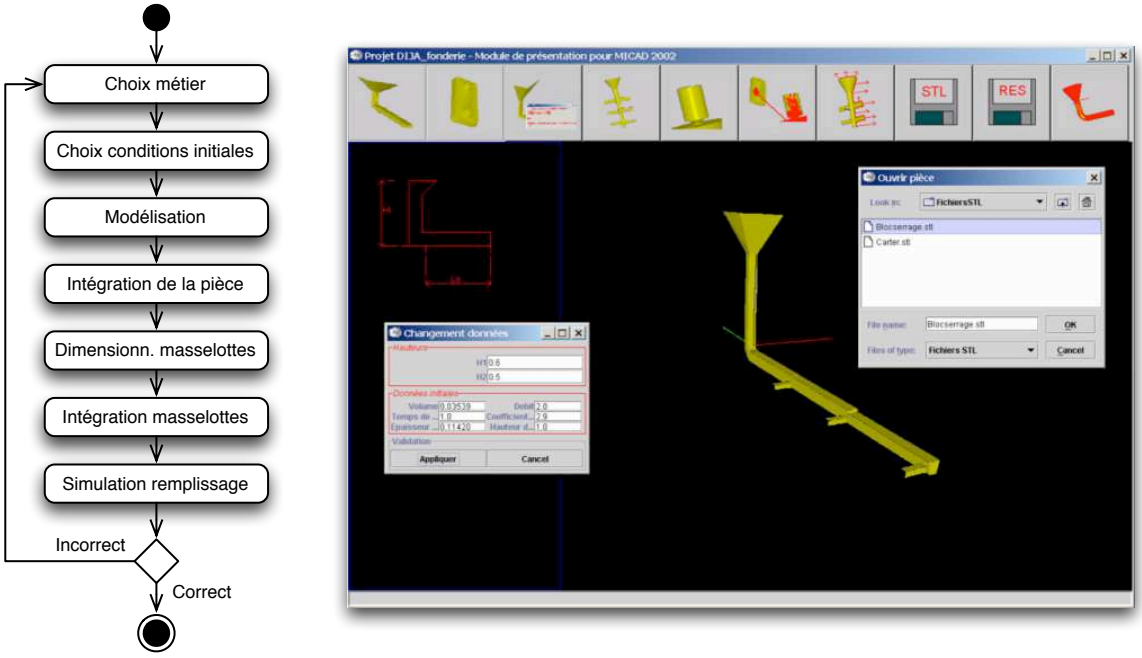


FIGURE 3.7: Système de remplissage en fonderie et *workflow* associé (tiré de [Gardan, 2005]).

est justement une prise en compte de critères émanant de plusieurs métiers au même moment (issus des domaines de la conception et de la fabrication dans notre cas), et dont les objectifs peuvent ainsi être antagonistes. Dans le prochain paragraphe, nous envisageons les possibilités d'y parvenir en nous basant sur l'approche synthétique de DIJA.

3.2.2 Analyse et positionnement

Vis-à-vis de notre problématique, DIJA présente un certain nombre d'atouts que nous nous proposons de souligner dans ce paragraphe.

En premier lieu, le caractère incrémental de l'approche synthétique offre la possibilité de prendre en compte des règles de fabricabilité de manière progressive. Ceci est intéressant dans l'optique d'une analyse de fabricabilité en temps réel : à chaque opération de transformation, un ensemble restreint de règles est à considérer. La figure 3.8 illustre ce principe. Nous considérons que les règles de fabricabilité r_1 sont connues pour l'objet de base o_1. Les outils de transformation de DIJA définissant comment le système calcule la géométrie g_2 à partir de la géométrie g_1, nous souhaiterions définir comment le système peut obtenir les règles r_2 en connaissant les règles r_1 et les géométries g_1 et g_2 de l'objet. Deux approches sont possibles :

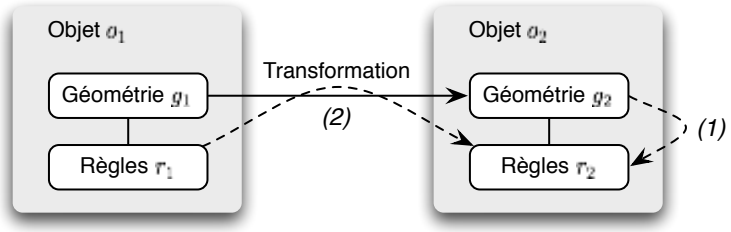


FIGURE 3.8: Approche pour la mise à jour des règles de fabricabilité.

1. Déduire les règles de fabrication r_2 à partir de la géométrie g_2, ce qui revient à déduire directement un ensemble de règles à partir de toute géométrie valide du point de vue du système. Cette approche consisterait à effectuer une extraction de caractéristiques de fabrication en partant du modèle produit ;
2. Recalculer, mettre à jour et maintenir la cohérence des règles r_1 en fonction de l'outil de transformation appliqué par le concepteur.

Nous optons pour la seconde approche pour les raisons suivantes :

- l'approche 1 est complexe à formaliser du fait que l'on ne suppose presque rien de la géométrie g_2 – définir une telle approche revient à faire ingérer au système l'ensemble des facultés d'analyse géométrique des fabricants. Comme nous l'avons vu au chapitre 2, les approches d'extraction pures sont lourdes et posent de sérieux problèmes d'interprétation, ce qui les rend incompatibles avec un traitement en temps réel ;
- l'approche 2 implique que le système connaît l'intention du concepteur (utilisation d'outils orientés métier) et semble donc plus abordable. On ne raisonne plus seulement sur la géométrie pure, mais aussi sur les particularités sémantiques des outils permettant sa transformation, en particulier les caractéristiques des outils influant sur le procédé de fabrication ;
- l'approche 2 offre un environnement plus spécifique à un métier (ensemble d'outils et de formes de base) et peut donc être développée de façon modulaire pour chaque métier, ce qui la rend plus flexible ;
- le fait d'utiliser l'approche 2 n'interdit pas d'utiliser des phases de raisonnement géométrique si cela est nécessaire de façon ponctuelle.

En regard, les approches ascendantes n'offrent pas autant d'avantages :

- les transformations opérées peuvent modifier la topologie du solide, ce qui pousse à re-

considérer complètement la géométrie des parties affectées ;

– les fonctions n'étant pas définies de manière hiérarchique, de nouvelles fonctions peuvent être ajoutées en cours de conception sans qu'elles soient le fruit d'un raffinement, ce qui rend les choix liés aux métiers plus facilement révocables.

En deuxième lieu, les éléments de dialogue utilisés pour représenter et manipuler la géométrie constituent un modèle géométrique canonique : n'importe quel solide peut être totalement représenté par des éléments de dialogue. Par conséquent, les éléments de dialogue offrent un pouvoir d'expression géométrique suffisant pour exprimer toute règle de fabricabilité. Nous pouvons donc les employer en tant qu'éléments de base pour stipuler les règles. Comme nous l'avons souligné dans le chapitre 2, l'adoption d'un modèle canonique permet de respecter la confidentialité, aspect primordial dans le cadre de schémas d'organisation de type entreprise étendue.

En dernier lieu, l'utilisation d'une architecture modulaire permet une séparation des différents métiers de fabrication. La mise à jour des règles de fabricabilité peut donc se faire de manière plus souple. En outre, l'architecture multi-niveaux de DIJA correspond à une approche métier : chaque outil contribue à un vocabulaire métier (une ontologie) et un niveau d'abstraction donné, et peut être décomposé en un ensemble d'actions de niveau inférieur. Ceci est transposé aux règles de fabricabilité, afin d'utiliser des termes propres aux métiers de la fabrication et de factoriser les opérations communes à des règles issues de plusieurs métiers.

L'approche que nous proposons consiste à introduire dans l'architecture DIJA un ensemble de modules de fabricabilité qui agissent en complément des modules de conception existants. Ainsi, il devient possible d'associer aux outils métier s'appliquant (indirectement) aux éléments de dialogue, des actions permettant de vérifier et de maintenir la cohérence des règles de fabricabilité relatives à la pièce en cours de conception. À tout instant, seules les règles pertinentes sont activées (instanciées), ce qui permet de cibler l'analyse. La figure 3.9 illustre ce principe. Le module de fabricabilité du niveau *application* permet d'associer des règles aux différents types de pièces et d'outils proposés dans les modules de conception. Les deux modules de fabricabilité des niveaux *métier* et *commun* permettent de valider la pièce en regard des règles activées.

Parallèlement aux modules de conception qui déterminent le modèle produit, nous introduisons ainsi des modules qui déterminent le modèle de fabricabilité de la pièce. Après chaque opération portant sur un ou plusieurs élément(s) de dialogue, ce modèle évolue de manière à refléter en permanence les règles qui doivent s'appliquer. Comme présenté sur la figure 3.9, les

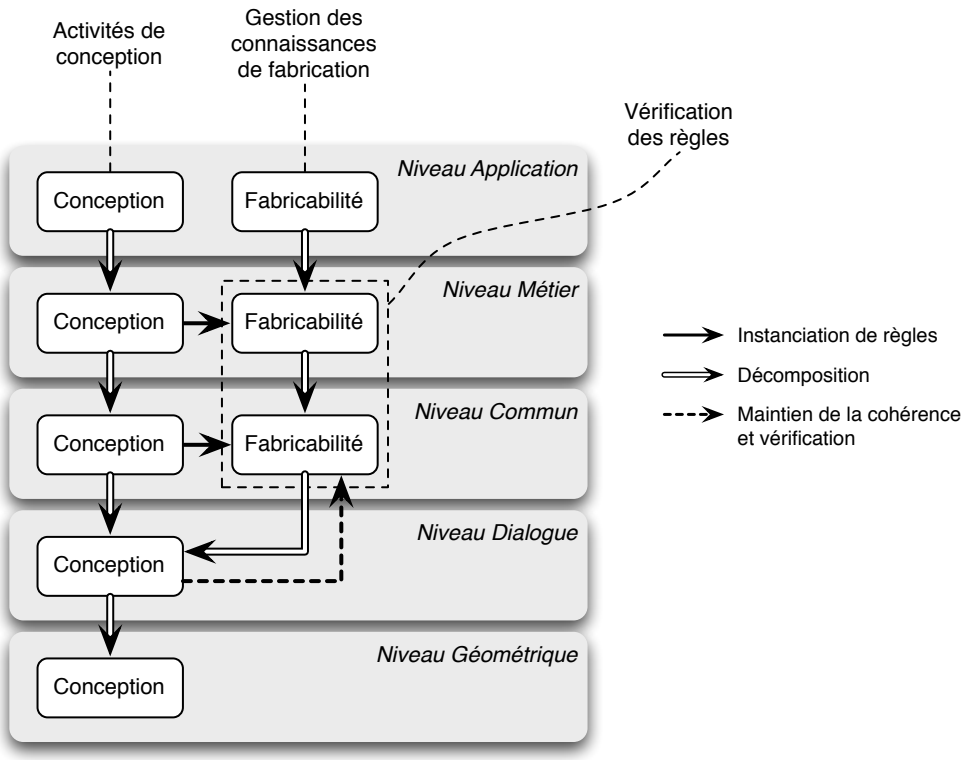


FIGURE 3.9: Approche modulaire pour la gestion de la fabricabilité.

deux modèles se rejoignent au niveau dialogue, de sorte que ce sont les éléments de dialogue qui constituent la jonction entre les métiers de conception et de fabrication. Nous adoptons ici l'idée selon laquelle la place centrale occupée par le modèle géométrique dans les systèmes de CAO permet d'assurer la cohérence du modèle produit [Maculet et Daniel, 2003]. Le maintien de la cohérence étant à la charge du système, le processus de conception peut être structuré sans qu'il faille se soucier des interférences avec la fabricabilité, y compris durant les phases de conception libre où l'ordre d'application des outils n'est pas déterminé.

Dans la section qui suit, le mode de représentation des règles de fabricabilité est introduit. Les conditions d'activation et le maintien de la cohérence des règles seront abordés dans la section 3.4.

3.3 Proposition pour la représentation des connaissances liées à la fabrication dans DIJA

Dans cette section, nous introduisons le mode de représentation proposé pour les règles de fabricabilité. Nous présentons dans un premier temps la représentation interne des règles (§ 3.3.1 et § 3.3.2), puis la façon dont ces règles peuvent être répercutées par des interactions avec l'utilisateur (§ 3.3.3), avant de conclure par une synthèse (§ 3.3.4).

3.3.1 Modélisation des règles de fabricabilité

Nous choisissons de modéliser la fabricabilité par l'intermédiaire de règles de production. Comme nous l'avons vu au chapitre précédent (§ 2.2.3, page 65), ce type de règles, constituées de conditions et de conclusions, sont particulièrement bien adaptées à la représentation du savoir-faire de fabrication :

si *conditions* **alors**
  conclusions
fin si

La partie *conditions* détermine un contexte de déclenchement, et la partie *conclusions* définit l'appréciation de la fabricabilité associée à ce contexte. Implicitement, lorsque la partie *conditions* n'est pas (ou plus) vérifiée, la conclusion ne peut ainsi être tirée, ce que nous pouvons assimiler à une fabricabilité normale (situation où aucun problème particulier n'est décelé). À

titre d'exemple, nous reprenons la règle du § 2.2.3 qui définit les conditions de fabricabilité des trous borgnes en moulage de matière plastique :

> *Moulage par injection : le rapport de la hauteur au diamètre doit être inférieur à deux si le diamètre du trou est inférieur à 3,5 mm, ou à trois sinon.*

Cette règle, exprimée ci-dessus en langage naturel, stipule que pour être fabricable, un trou borgne doit respecter des conditions dépendant de sa hauteur et de son diamètre. Nous pourrions retourner la règle comme ceci :

> *Moulage par injection : le rapport de la hauteur au diamètre ne doit pas être supérieur ou égal à deux si le diamètre du trou est inférieur à 3,5 mm, ou à trois sinon.*

Il est clair que la partie *conditions* s'assimile à une formule logique, celle-ci pouvant être écrite de différentes façons. Le mode de représentation logique correspond au principe de formalisation de la connaissance, qui consiste à partager le réel en un ensemble de cas facilement identifiables, sans ambiguïté. Nous pouvons donc formaliser la partie *conditions* par une formule logique en forme normale disjonctive [118] :

$$conditions \Leftrightarrow \left(e_1^1 \wedge e_2^1 \wedge \ldots \wedge e_a^1\right) \vee \left(e_1^2 \wedge e_2^2 \wedge \ldots \wedge e_b^2\right) \vee \ldots \vee \left(e_1^n \wedge e_2^n \wedge \ldots \wedge e_k^n\right)$$

où les e_j^i sont des expressions mathématiques (celles-ci sont décrites plus loin). Ce formalisme a l'avantage de permettre d'associer facilement une partie *conclusions* à plusieurs contextes différents. Cette écriture se ramène à un ensemble de clauses de Horn strictes [119] :

---

118. Notons que dans le cas où l'on voudrait spécifier les conditions des règles en forme normale conjonctive, il serait toujours possible de les transformer en forme normale disjonctive (loi de dualité de De Morgan) :

$$\begin{aligned} conditions &\Leftrightarrow \left(e_1^1 \vee e_2^1 \vee \ldots \vee e_a^1\right) \wedge \left(e_1^2 \vee e_2^2 \vee \ldots \vee e_b^2\right) \wedge \ldots \wedge \left(e_1^n \vee e_2^n \vee \ldots \vee e_k^n\right) \\ &\Leftrightarrow \neg\left(\left(\neg e_1^1 \wedge \neg e_2^1 \wedge \ldots \wedge \neg e_a^1\right) \vee \left(\neg e_1^2 \wedge \neg e_2^2 \wedge \ldots \wedge \neg e_b^2\right) \vee \ldots \vee \left(\neg e_1^n \wedge \neg e_2^n \wedge \ldots \wedge \neg e_k^n\right)\right) \end{aligned}$$

Ceci revient à renverser la formule, de même que cela a été fait en langage naturel précédemment.

119. Les clauses de Horn (du nom du logicien Alfred Horn) sont un sous-ensemble des formules en forme normale disjonctive : celles qui n'ont qu'un seul terme positif.

$$\begin{array}{rcl} \textit{conclusions} & \vee & \neg e_1^1 \vee \neg e_2^1 \vee \ldots \vee \neg e_a^1 \\ \textit{conclusions} & \vee & \neg e_1^2 \vee \neg e_2^2 \vee \ldots \vee \neg e_b^2 \\ & \vdots & \\ \textit{conclusions} & \vee & \neg e_1^n \vee \neg e_2^n \vee \ldots \vee \neg e_k^n \end{array}$$

Cela revient bien à écrire que les termes formant les conditions modélisent la partie *conclusions* :

$$\begin{array}{rcl} \left\{e_1^1, e_2^1, \ldots, e_a^1\right\} & \models & \textit{conclusions} \\ \left\{e_1^2, e_2^2, \ldots, e_b^2\right\} & \models & \textit{conclusions} \\ & \vdots & \\ \left\{e_1^n, e_2^n, \ldots, e_k^n\right\} & \models & \textit{conclusions} \end{array}$$

Ceci est la formalisation logique habituelle associée aux règles de production [Alliot et Schiex, 2002]. Un avantage résultant de l'utilisation de clauses de Horn est que la résolution est efficace : la satisfiabilité d'un ensemble de clauses de Horn (ou encore HORN-SAT) est de classe $\mathcal{P}$. Notamment, cette propriété pourrait être utilisée dans le but de garantir que les règles introduites sont déclenchables, et donc non inutiles [120].

Ainsi, la règle donnée en exemple précédemment peut être traduite en deux règles de production formelles :

si `diamètre` $< 3,5\,\text{mm} \wedge$ `hauteur/diamètre` > 2 **alors**
  `pièce_non_fabricable`
fin si
si `diamètre` $\geq 3,5\,\text{mm} \wedge$ `hauteur/diamètre` > 3 **alors**
  `pièce_non_fabricable`
fin si

Ou encore, en utilisant une écriture condensée :

120. En supposant que les calculs afférants aux expressions conservent une complexité de classe $\mathcal{P}$.

si $(\texttt{diamètre} < 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 2)$
$\vee\,(\texttt{diamètre} \geq 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 3)$ **alors**
  `pièce_non_fabricable`
fin si

Comme nous l'avons expliqué au § 3.2.2, nous pouvons nous baser sur les éléments de dialogue de DIJA pour stipuler les règles de fabricabilité. Nous constatons que les éléments de dialogue permettront effectivement de calculer les variables relatives à la hauteur et au diamètre du trou dans l'exemple ci-dessus. Nous distinguons ces calculs, de nature géométrique, des calculs relatifs aux expressions, comme par exemple « `hauteur/diamètre` $>$ 2 », qui font quant à elles partie intégrante du véritable savoir-faire de fabrication. L'approche de gestion des calculs s'appliquant aux éléments de dialogue est décrite plus loin (§ 3.4.2.1). Mais nous pouvons d'ores et déjà noter que pour former les expressions e^i_j, il est nécessaire d'utiliser des grandeurs (`hauteur/diamètre` et `diamètre` dans l'exemple) assorties d'opérateurs ($<$ et $\geq$ dans l'exemple) et de mesures (3, 5 mm, 2 et 3 dans l'exemple). Plus précisément, il est possible de définir la syntaxe de ces expressions en se conformant à la grammaire suivante (notation BNF [121]) :

$$
\begin{aligned}
&e^i_j ::= \langle \textit{expr} \rangle\ \langle \textit{op-rel} \rangle\ \langle \textit{expr} \rangle \mid\ \neg\ (\ \langle \textit{expr} \rangle\ \langle \textit{op-rel} \rangle\ \langle \textit{expr} \rangle\ )\\
&\langle \textit{op-rel} \rangle ::= < \mid\ > \mid\ = \mid\ \leq \mid\ \geq \mid\ \neq\\
&\langle \textit{expr} \rangle ::= (\ \langle \textit{expr} \rangle\ \langle \textit{op-b} \rangle\ \langle \textit{expr} \rangle\ ) \mid\ \langle \textit{op-u} \rangle\ (\ \langle \textit{expr} \rangle\ ) \mid\ \text{grandeur}\\
&\langle \textit{op-b} \rangle ::= + \mid\ - \mid\ \div \mid\ \times \mid\ \ldots\\
&\langle \textit{op-u} \rangle ::= \ln \mid\ \sin \mid\ \sqrt{} \mid\ \ldots
\end{aligned}
$$

La forme niée (symbole $\neg$) est permise par simple commodité. La notion de grandeur est à prendre au sens tel que défini au § 2.2.2 : il s'agit d'un produit *Nombre* $\times$ *Unité*, l'unité pouvant être l'unité « un » [Bureau International des Poids et Mesures, 2006], par exemple dans le cas d'un rapport (fraction de grandeurs d'unités identiques). Néanmoins, il peut non seulement s'agir de grandeurs géométriques (correspondant plus exactement au niveau *dialogue* de DIJA), mais également de paramètres du niveau *commun*, comme par exemple les propriétés d'un matériau [122]. Il faut en effet pouvoir utiliser ce type de paramètre pour spécifier le contexte des règles. C'est ainsi que nous préférons adopter le terme de « grandeur de fabrication » qui

121. *Backus-Naur Form*.
122. Il s'agit alors de paramètres grapho-numériques (PGN).

relate bien l'aspect métier. Ce concept, bien que fondé, entre autres, sur la géométrie, est sémantiquement plus élevé que le niveau dialogue et permet de factoriser les calculs géométriques et de s'en abstraire lors de la spécification des règles (du moins à un certain stade). En effet, nous remarquons par exemple que, dans la règle précédente, la grandeur `diamètre` est utilisée dans les deux expressions formant chaque condition. Enfin, mélanger les formules de calcul géométrique, qui ont une écriture souvent lourde, à celles relevant de l'analyse de fabricabilité alourdirait inutilement ces dernières. La grandeur de fabrication est donc un concept du niveau *métier,* qui est relatif au module de fabricabilité (cf. figure 3.9).

Concernant la partie *conclusions* des règles, dans l'exemple présenté, il est indiqué que la pièce n'est pas fabricable. Dans le cas général, la conclusion d'une règle pourra être de nature variée : surcoût de production, délai de préparation ou de fabrication accru, rebus important, qualité moindre, etc. Il est même possible d'envisager qu'une règle de fabricabilité entraîne une conclusion moins immédiate, par exemple en y associant un scénario collaboratif qui fera directement intervenir le fabricant afin que ce dernier et le concepteur puissent trouver un compromis satisfaisant du point de vue de leurs domaines respectifs. C'est à travers ce type d'interaction que l'on constate que les règles de production peuvent revêtir un caractère procédural. Nous proposons d'englober toutes ces possibilités sous le concept générique d'état décrit dans le paragraphe suivant.

3.3.2 Notion d'état pour la conception

Nous introduisons la notion d'état, à l'égard de la pièce en cours de conception, afin de prendre en compte tous les types de critères qui permettent d'apprécier la fabricabilité. Un état correspond à une situation dans laquelle la pièce se trouve, relativement à une règle donnée. Cette situation peut être caractérisée par des indicateurs technologiques ou économiques (coûts, qualités et délais ; se référer au § 2.2.1 pour plus de détails).

Ainsi, l'évaluation d'une condition peut, selon la valeur booléenne obtenue, mener ou ne pas mener à une transition vers un nouvel état pour la pièce. Il est de ce fait naturel de se donner la liberté d'associer à l'arrivée dans un état donné, le déclenchement d'un scénario collaboratif.

Comme nous l'avons déjà indiqué dans le paragraphe précédent, seules les règles pertinentes vis-à-vis de la conception en cours sont activées. Il faut cependant préciser qu'une règle pouvant s'appliquer plusieurs fois sur des parties analogues d'une même pièce, il est nécessaire de l'instancier. Dans le système, une règle doit donc exister à deux niveaux : le niveau

abstrait et le niveau instancié. Au niveau abstrait, la règle est définie en se basant sur des grandeurs de fabrication paramétrées, dont les paramètres doivent être assignés ou valués lors de l'instanciation. Dès lors qu'une règle est instanciée, elle est distinguable des autres instances de la même règle abstraite.

Pour résumer, une règle instanciée se définit grâce à un identifiant, des grandeurs de fabrication instanciées (paramètres connus) et un état propre. Bien évidemment, plusieurs règles instanciées peuvent se retrouver dans le même état. Nous retrouvons donc ce double niveau de représentation pour les états. Pour un état abstrait, nous souhaitons disposer des informations suivantes :

État ::= ⟨*nom*⟩ , ⟨*type*⟩ , ⟨*description*⟩ [, ⟨*action*⟩]
⟨*nom*⟩ ::= ⟨*chaîne-unique*⟩
⟨*type*⟩ ::= fabricable | non-fabricable | surcoût | délai-accru | perte-qualité | ...
⟨*description*⟩ ::= ⟨*chaîne-quelconque*⟩
⟨*action*⟩ ::= ⟨*EGN*⟩ | ⟨*appel-routine*⟩

Le nom d'un état permet de l'identifier sans ambiguïté. Le regroupement des états selon un type permet, par exemple, d'adapter la présentation pour l'utilisateur suivant la nature de l'information. La description de l'état permet de l'assortir de consignes expliquant les raisons de sa présence, ses conséquences ou la façon d'y remédier. Dans ce but, la description peut éventuellement faire référence aux grandeurs de fabrication impliquées. Enfin, le champ *action* correspond aux interactions et aux scénarii collaboratifs. Dans le cas d'une expression graphonumérique (EGN), il est bien sûr nécessaire d'être en mesure de paramétrer l'action en fonction du contexte. Cela s'effectue en référençant les grandeurs de fabrication ou les éléments de dialogue relatifs à la règle concernée. Quant aux routines, elles sont des instructions particulières qui peuvent être utilisées sans passer par la définition d'un scénario (p. ex. des instructions liées à l'IHM : affichage d'un message, etc.).

Dans ce cadre, l'état de la pièce est considéré comme étant l'agrégation des états propres de chacune des règles instanciées :

$$\text{état}(\texttt{pièce}, t) = \bigcup_{i=1}^{k} \{\text{identifiant}(\texttt{règle}_i), \text{nom}(\text{état}(\texttt{règle}_i, t))\}$$

où k est le nombre de règles instanciées à l'instant t. Ainsi, il suffit qu'une seule règle mène à un état de non-fabricabilité pour que la pièce dans son ensemble, soit considérée comme

non-fabricable.

Dans la suite, nous présentons les interactions qui sont à l'origine des évolutions de l'état de la pièce.

3.3.3 Interactivité

Afin que le concepteur en ait connaissance, l'état de la pièce est une information qui doit se retrouver dans l'interface du logiciel. Se pose ainsi la question du mode de présentation de cette information, mais aussi des interactions qui en gouvernent l'évolution.

Lors de la conception, le système observe les actes du concepteur, en déduit l'état de la pièce au sens indiqué ci-avant, et assiste l'utilisateur en regard des critères de fabricabilité définis par les fabricants (règles abstraites). Un changement d'état pourra donc mener à des interactions via le champ *action* associé au nouvel état. Dans le cadre de la fabricabilité, trois types d'interactions sont envisageables :

1. Interdire la conception des pièces non fabricables. Ceci est par exemple possible à travers un scénario qui avertit l'utilisateur, et annule l'acte ayant initié la transition vers le nouvel état (qui est dans ce cas de type « non-fabricable ») ;
2. Informer le concepteur lorsqu'une modification rend plus difficile la fabrication de la pièce. Ceci peut aussi passer par un scénario demandant confirmation à l'utilisateur. L'acte de conception à l'origine de l'avertissement peut alors être confirmé ou annulé. Dans le cas d'une confirmation, il est bon que la nouvelle appréciation de fabricabilité demeure visible (ou du moins accessible, cf. plus bas) et révocable (le fabricant peut alors en être informé) ;
3. Aider le concepteur à rendre la pièce fabricable. Ceci peut passer par des consignes enregistrées dans la description du nouvel état, ou par un scénario collaboratif. Dans le second cas, de nombreuses possibilités sont envisageables : déclenchement d'une phase de négociation avec le fabricant, application d'une procédure spécifique (ajout de surépaisseurs d'usinage, de raidisseurs…), délégation d'une tâche à un autre collaborateur, etc.

Les interactions et la présentation des appréciations de fabricabilité provenant des règles instanciées dont l'état n'est pas « fabricable » posent deux problèmes :

1. La quantité d'informations peut rapidement devenir importante ;

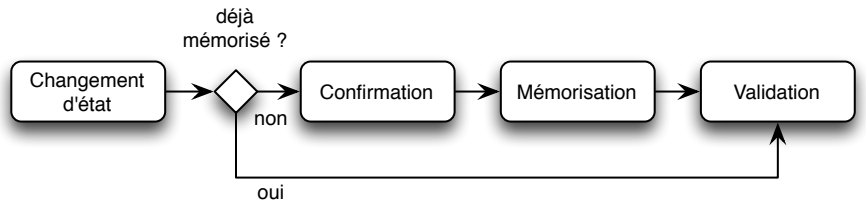


FIGURE 3.10: Mémorisation des interactions.

2. Les interactions peuvent se répéter très fréquemment, ce qui constitue alors un frein au travail du concepteur. Par exemple, si le concepteur valide une condition de fabricabilité puis qu'il revient en arrière, le système pourra être amené à lui demander encore une fois de valider cette même condition (si les critères sont de nouveau remplis).

Le problème de la quantité d'informations peut être atténué en utilisant des solutions de présentation évoluées. Il est possible d'utiliser une présentation condensée, par exemple en regroupant les instances des mêmes règles, ou ayant un même type d'état, tout en laissant la possibilité d'accéder à une description complète. Il est aussi possible de trier les informations en ne présentant en premier que celles qui sont jugées plus importantes (plus récentes, plus coûteuses, etc.).

Afin de solutionner le problème de la répétition, nous proposons de mémoriser les choix de l'utilisateur. Si un choix est fait une première fois, il est mémorisé et réutilisé si nécessaire (voir figure 3.10). Ceci implique naturellement d'offrir l'opportunité de faire « oublier » au système un choix précédemment mémorisé dans le cas où le concepteur aurait changé d'avis.

Le paragraphe suivant est une synthèse du mode de représentation des connaissances de fabricabilité proposé dans cette section.

3.3.4 Synthèse

Dans cette section, nous avons défini un mode de représentation pour les règles de fabricabilité. La description sous forme de règles de production est adoptée, avec, pour la partie *conditions*, une formule logique sous forme de clause de Horn, et pour la partie *conclusions*, une appréciation qui peut s'accompagner d'une interaction. Nous avons pour cela introduit la notion de grandeur de fabrication qui définit tout élément du contexte de conception contribuant à caractériser la fabricabilité, ainsi que la notion d'état qui représente une situation particulière

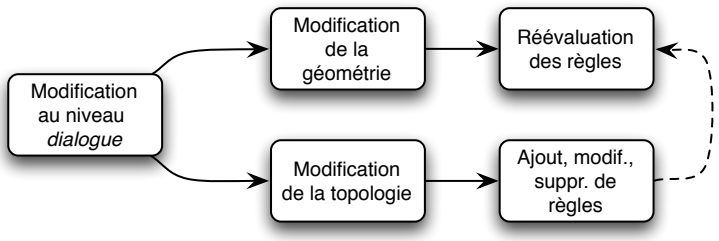


FIGURE 3.11: Répercussions après modification au niveau dialogue.

de la pièce vis-à-vis d'une règle de fabricabilité.

Un atout du concept générique d'état, est qu'il autorise différents types de référentiels pour la fabricabilité. Cette dernière peut ainsi être signalée en employant un système de référence absolu ou relatif, et avec des unités de mesure concrètes ou abstraites (ces notions ont été décrites au § 2.2.2). Le système de référence se traduira dans la description des états, par un paramétrage absolu ou relatif à l'aide de grandeurs de fabrication, tandis que les unités de mesure seront soit concrètes, soit abstraites.

Dans la section suivante, les processus qui permettent d'appliquer, d'instancier et de maintenir la cohérence des règles de fabricabilité au cours de la conception sont présentés.

3.4 Gestion de la fabricabilité

Dans cette section, nous présentons l'approche de gestion des règles de fabricabilité qui, en regard de la section précédente, correspond à l'instanciation et à la manipulation de grandeurs de fabrication et d'états. Nous commençons par remarquer qu'après application d'un outil de conception DIJA, des éléments de dialogue sont toujours modifiés. L'emploi d'un outil peut alors affecter :

- la seule géométrie ;
- ou bien la géométrie, ainsi que la topologie du modèle [123] au niveau *dialogue* (ajout ou suppression d'éléments de dialogue).

Suivant le cas, les répercussions ne sont pas les mêmes du point de vue de la fabricabilité. Le jeu de règles instanciées n'a effectivement pas à être révisé lorsque seule la géométrie est

123. Nous faisons ici référence à la topologie du graphe d'historique associé aux éléments de dialogue, mais pas à la topologie du solide, caractéristique qui reste inchangée de par l'adoption de l'approche synthétique DIJA. Le graphe d'historique est présenté dans [Denis, 2003].

impliquée. En revanche, les règles doivent être réévaluées après chaque modification affectant le niveau *dialogue*. Ceci est illustré sur la figure 3.11.

Ainsi, dans ce qui suit, le processus d'évaluation des règles est tout d'abord défini d'un point de vue général (§ 3.4.1). La prise en compte statique des règles (évolution de la géométrie, mais pas de la topologie) est ensuite détaillée à travers la description de deux types de graphes qui permettent d'optimiser la phase d'évaluation des règles (§ 3.4.2). Puis, l'évolution du jeu de règles instanciées suite aux changements de topologie est présentée (§ 3.4.3). Cette section se conclue par une synthèse (§ 3.4.4) qui rappelle les points essentiels de l'approche développée.

3.4.1 Processus d'évaluation

La figure 3.12 schématise les différentes étapes du processus adopté pour la prise en compte de règles que l'on suppose déjà instanciées. Dans cette section, cet aspect est désigné comme étant « l'évaluation » des règles.

Suite à l'application d'un outil, la partie *conditions* de l'ensemble des règles concernées est dans un premier temps évaluée (ou réévaluée si elle l'a déjà été). Si aucun changement n'est constaté (cas *(a)* de la figure 3.12), l'opération initiée par le concepteur est acceptée, car aucun changement d'état n'interviendra. Dans le cas contraire, l'évaluation de la conclusion de la ou des règle(s) donne lieu à un changement d'état. Si ce changement d'état a déjà été opéré antérieurement, et donc mémorisé (cf. § 3.3.3), l'opération est directement acceptée ou rejetée (cas *(b)* et *(c)* de la figure 3.12). Si au contraire, le changement d'état n'a pas été mémorisé (ou s'il a été réinitialisé), les interactions correspondantes sont opérées, ce qui mène à accepter ou rejeter l'opération selon le déroulement du scénario (cas *(d)* et *(e)* de la figure 3.12).

La plus grande part des calculs survient lors de l'évaluation des règles (éléments grisés sur la figure 3.12). Les conditions des règles sont calculées à partir des grandeurs de fabrication, elles-mêmes obtenues par des calculs géométriques (du moins pour la plupart), en se basant sur les éléments de dialogue. Afin d'optimiser ces enchaînements de calculs (nous souhaitons un service en continu), nous proposons d'utiliser les deux types de graphes présentés dans le paragraphe suivant.

3.4.2 Prise en compte statique des règles

Le processus d'évaluation peut être décomposé en deux étapes :

– l'évaluation des grandeurs de fabrication (GF) ;

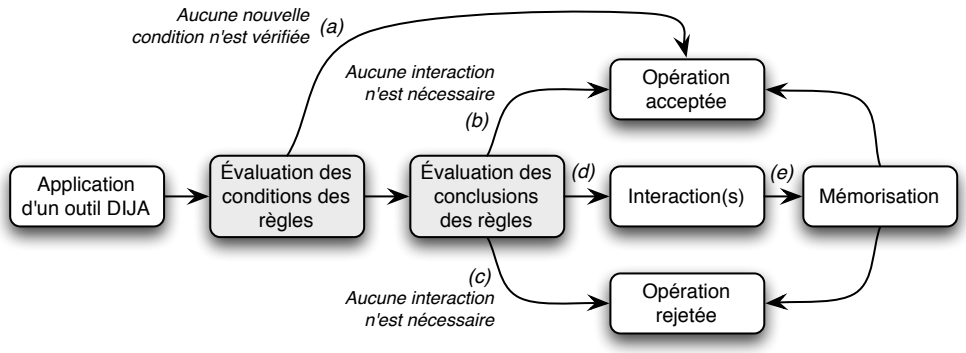


FIGURE 3.12: Processus d'évaluation de règles instanciées.

– l'évaluation des conditions rattachées aux règles qui se basent sur ces GF.

Cette décomposition est intéressante du fait qu'une même GF peut être impliquée dans plusieurs règles. Il est en effet inutile de recalculer plusieurs fois une telle grandeur. L'évaluation des GF se fait donc tout naturellement avant celle des conditions qui s'y rattachent. Nous formalisons ces deux phases de calcul en nous basant sur deux types de graphes :

– un graphe des grandeurs de fabrication (GGF) qui permet de minimiser les calculs lors de l'évaluation des GF ;
– des graphes des états (GE) ; un pour chaque aspect de la fabricabilité ; qui permettent quant à eux de minimiser les traitements liés aux changements d'état.

Ces deux graphes sont présentés en détail dans les paragraphes 3.4.2.1 et 3.4.2.2.

3.4.2.1 Graphe des grandeurs de fabrication

Un graphe des grandeurs de fabrication (GGF) est défini pour chaque conception. Il permet de structurer les calculs afférant aux grandeurs de fabrication afin de minimiser ces derniers. Ainsi, le GGF indique quels éléments de dialogue sont utilisés pour calculer chaque grandeur. De manière à factoriser certains calculs communs à plusieurs GF, la notion de valeur de fabrication est introduite. Une valeur de fabrication (VF) est assimilable à une mesure géométrique élémentaire, que l'on pourrait être à même d'effectuer physiquement sur la pièce si cette dernière existait réellement sous forme d'éléments de dialogue. Par exemple, si nous voulons obtenir le rapport de la hauteur au diamètre d'un trou, nous devons « mesurer » la hauteur et le diamètre de ce trou. La hauteur du trou peut par exemple être obtenue en « mesurant » la distance entre les contours caractéristiques inférieur et supérieur du trou, tandis que le dia-

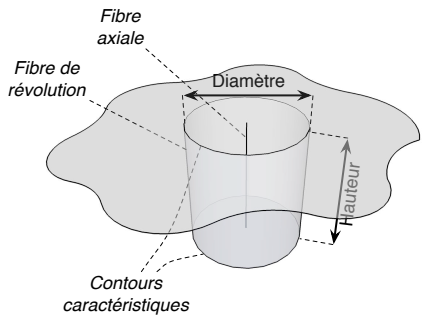


FIGURE 3.13: Exemples de valeurs de fabrication.

mètre peut par exemple être obtenu à partir du rayon, en « mesurant » la distance entre la fibre axiale et l'une des fibres de révolution du trou (cf. figure 3.13). Ainsi, en reprenant l'exemple de règle du § 3.3.1, la VF `diamètre` peut être utilisée pour obtenir les GF `hauteur/diamètre` et `diamètre`. Nous ne nous attarderons pas davantage sur l'aspect géométrique du calcul des VF, qui fait appel à des méthodes de calcul géométrique (pour la plupart connues), mais sortant du cadre de cette thèse : distances entre éléments de dialogue (points, segments, surfaces…) dans l'espace.

Ainsi, le graphe des grandeurs indique les liens de dépendance existant entre quatre ensembles :

- l'ensemble des éléments de dialogue et PGN, noté X^{ED} ;
- l'ensemble des valeurs de fabrication, noté X^{VF} ;
- l'ensemble des grandeurs de fabrication, noté X^{GF} ;
- et l'ensemble des règles de fabricabilité, noté X^{R}.

À chaque fois qu'un sous-ensemble x de X^{ED} est modifié, seul les sous-ensembles de X^{VF}, puis de X^{GF} et de X^{R} qui en dépendent sont recalculés. Un exemple de graphe des grandeurs correspondant à la règle du § 3.3.1 est présenté sur la figure 3.14. Le diamètre du trou est obtenu en calculant la distance entre les fibres f_1^a et f_1^r, tandis que sa hauteur est obtenue en calculant la distance entre les contours caractéristiques c_1 et c_2. Les deux GF sont alors calculées en se basant sur ces deux VF. Ainsi, après modification d'un élément de dialogue (p. ex. f^r), seules les VF et les GF dépendantes sont réévaluées (le diamètre et le rapport hauteur/diamètre dans le cas de f^r). Par conséquent, les règles devant être réexaminées (car susceptibles de changer d'état) sont alors connues après une quantité de calculs minimale.

Le fait de séparer ainsi la géométrie et les règles de fabricabilité autorise en outre à mas-

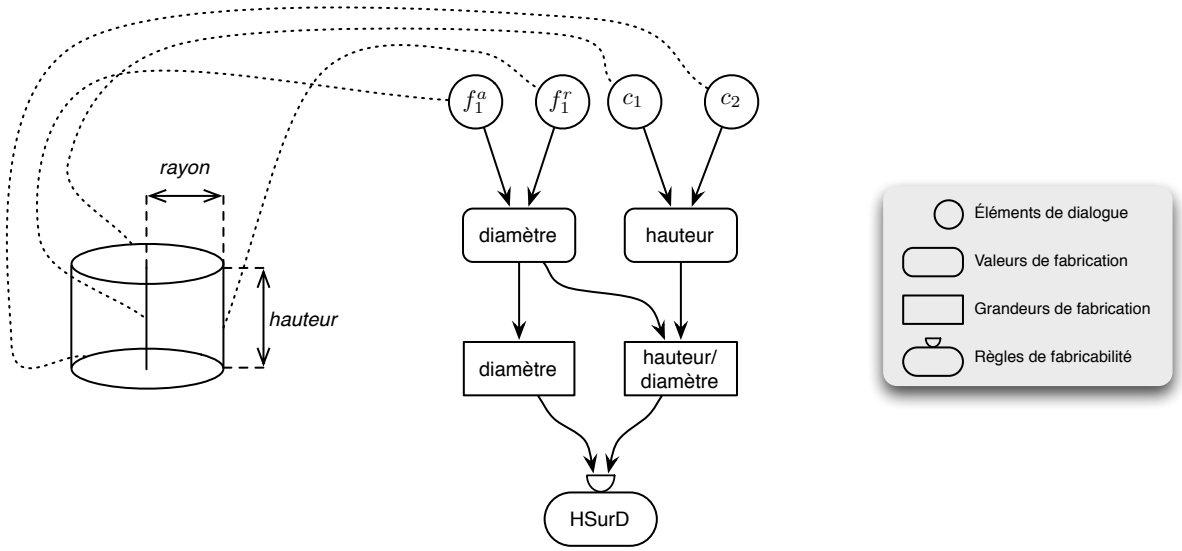


FIGURE 3.14: Exemple de GGF (une seule règle est présentée).

quer le savoir-faire de fabrication. Du point de vue du concepteur, celui-ci ne transparaît alors qu'à travers les interactions qui se déclenchent au moment où un problème de fabricabilité survient. La confidentialité des règles est ainsi assurée et les informations présentées à l'utilisateur peuvent être sélectionnées à la guise du fabricant.

Dans le paragraphe suivant, nous présentons le second graphe, qui permet de représenter les états caractérisant la fabricabilité, et de gérer les changements d'états.

3.4.2.2 Graphes des états

Les parties *conditions* des règles de fabricabilité se basent sur les GF calculées comme indiqué dans le paragraphe précédent. Après toute modification d'élément de dialogue (ED), nous connaissons donc les règles de fabricabilité dont les parties *conditions* doivent être réévaluées. Ces règles sont représentées sous forme de règles de production. Considérons la situation suivante. Soit g une GF. Une règle de fabrication indique que si g a une valeur inférieure à un seuil s, alors la pièce devient non fabricable :

si $g < s$ **alors**
  `pièce_invalide`
fin si

Comme indiqué au § 3.3.1, si la partie *conditions* n'est pas vérifiée, l'état de la pièce relativement à cette règle de fabricabilité est implicitement fabricable. Nous avons donc implicitement :

si $\neg\,(g < s)$ **alors**
  `pièce_valide`
fin si

Il est alors clair que les états `pièce_valide` et `pièce_invalide` s'excluent mutuellement. Cependant, si nous ajoutons un état intermédiaire nommé `surcoût`, correspondant par exemple à un surcoût de fabrication, et survenant lorsque $s \leq g < S$, la situation est moins simple. En effet, il faut alors prendre en compte le fait que les trois états relevant de cet « aspect [124] » doivent être exclusifs, ce qui nous conduit à la situation des règles 3.1a.

RÈGLES 3.1: Règles permettant de représenter des états mutuellement exclusifs.

(a) Cas particulier

si $\neg\,(g < S)$ **alors**
  `pièce_valide`
fin si
si $g < S \wedge \neg(g < s)$ **alors**
  `surcoût`
fin si
si $g < s$ **alors**
  `pièce_invalide`
fin si

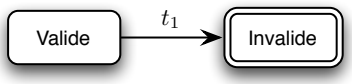
(b) Cas général

si $C_1 \wedge \neg C_2 \wedge \neg C_3 \wedge \ldots \wedge \neg C_n$ **alors**
  `état_1`
fin si
si $\neg C_1 \wedge C_2 \wedge \neg C_3 \wedge \ldots \wedge \neg C_n$ **alors**
  `état_2`
fin si
...
si $\neg C_1 \wedge \neg C_2 \wedge \neg C_3 \wedge \ldots \wedge C_n$ **alors**
  `pièce_invalide`
fin si

Cette situation, relativement fréquente (un état passe rarement du tout au tout), pose deux problèmes. Le premier, purement formel, réside dans le fait qu'il n'est pas naturel de spécifier les règles ainsi. Cependant, cette écriture étant systématique, il serait possible de la faire effectuer automatiquement ; dans le cas général, l'utilisateur ne spécifierait alors que les conditions $C_1 \ldots C_n$, comme illustré dans les règles 3.1b (la condition C_n correspond à une pièce invalide). Il n'en reste pas moins que les calculs qui en découlent sont inutilement alourdis. C'est là le second problème.

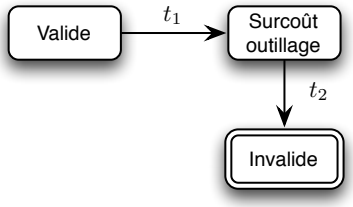
Nous proposons donc de simplifier l'évaluation par l'utilisation d'un graphe des états. Ce graphe simplifie la spécification et permet de n'évaluer l'ensemble des conditions que si cela est nécessaire, contrairement aux règles stipulées comme ci-dessus, où les parties *conditions* sont

124. Dans cet exemple, par « aspect », nous entendons le contexte de conception défini par les valeurs possibles de g.



$$\begin{aligned} C_1 &= (\texttt{diamètre} < 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 2) \\ &\vee (\texttt{diamètre} \geq 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 3) \end{aligned}$$

(a) Aspect regroupant deux états



$$\begin{aligned} C_1 &= (\texttt{diamètre} < 3\,\text{mm} \wedge \texttt{hauteur/diamètre} > 3) \\ &\vee (\texttt{diamètre} \geq 3\,\text{mm} \wedge \texttt{hauteur/diamètre} > 5) \\ C_2 &= (\texttt{diamètre} < 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 2) \\ &\vee (\texttt{diamètre} \geq 3,5\,\text{mm} \wedge \texttt{hauteur/diamètre} > 3) \end{aligned}$$

(b) Aspect regroupant trois états

FIGURE 3.15: Exemples de graphes des états relatifs au GGF de la figure 3.14.

redondantes et doivent être systématiquement considérées une à une.

Afin de limiter les calculs, nous proposons ainsi de regrouper les tests de changement d'état par aspects. Un aspect rassemble plusieurs règles de fabricabilité abstraites dont les conclusions mènent à des états exclusifs. Dans ce cadre, chaque aspect est caractérisé par un graphe des états (GE). Ce dernier détermine quel état peut être atteint à partir de l'état courant (état propre), et quelles conditions déterminent les changements d'état. Ceci traduit l'idée que les changements d'état relatifs aux instances d'une règle abstraite donnée s'effectuent en respectant les mêmes principes, et que les conditions devant être évaluées varient selon l'état courant.

Un GE, est un graphe orienté acyclique constitué de nœuds représentant un ensemble d'états exclusifs (les parties *conclusions* des règles relevant du même aspect), et d'arcs orientés représentant les transitions autorisées entre ces états (les parties *conditions* des règles relevant du même aspect). Les nœuds qui correspondent aux situations où la pièce n'est pas fabricable sont schématisés par un double cerclage. Un exemple de GE correspondant à la règle intitulée `HSurD` de la figure 3.14 est présenté sur la figure 3.15a. Dans l'exemple de la figure 3.15b, nous avons extrapolé cette règle en y ajoutant les conditions d'un surcoût d'outillage.

Du fait du regroupement par aspects, les nœuds estampillés « Règles de fabricabilité » sur le GGF sont donc apparentés à des ensembles de règles de production.

Dans le GE, les arcs sont orientés mais les transitions peuvent s'effectuer dans les deux sens. L'orientation permet d'interpréter le sens dans lequel la transition doit être effectuée en fonction de la valeur de vérité de la condition associée. Ainsi, dans l'exemple de la figure 3.15b, si

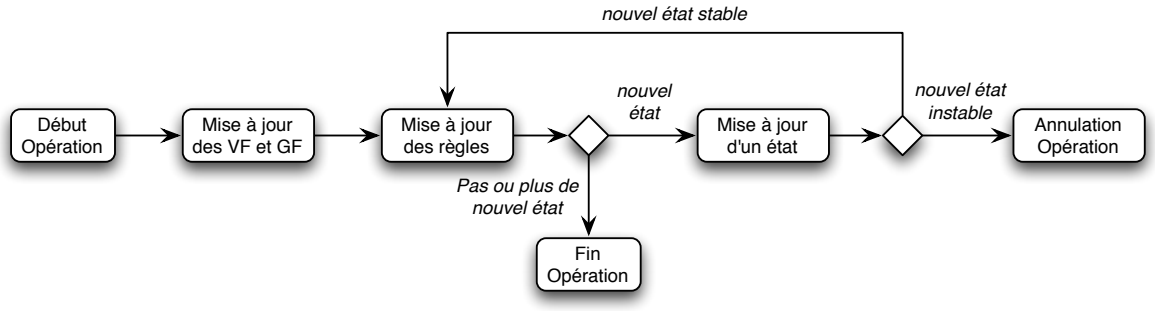


FIGURE 3.16: Processus de changement d'état.

l'état courant est « Surcoût outillage » et que la condition C_2 devient vraie, c'est que la transition t_2 doit être déclenchée. En revanche, si la transition C_1 devient fausse, c'est que la transition inverse de t_1 doit être effectuée.

Ce fonctionnement déterministe impose que l'on s'assure au moment de la spécification des règles, que plusieurs transitions concurrentes ne puissent être déclenchées simultanément (dans l'exemple précédent, nous avons implicitement $C_2 \Rightarrow C_1$). Pour les mêmes raisons, les graphes des états ne devraient normalement pas comporter de circuits. Nous ne développons pas d'avantage cette voie, et considérons les problèmes d'intégrité dans la définition des connaissances comme étant annexes [125].

La figure 3.16 présente de manière globale le processus déclenché suite à chaque opération, afin de déterminer le nouvel état de la pièce en considérant chaque règle. Il est à noter que les états correspondant à une non-fabricabilité (nœuds notés par un double-cerclage dans le GE) conduisent systématiquement à une annulation de l'opération en cours. Ils sont désignés comme étant instables sur la figure 3.16. Si plusieurs transistions (liées ou non à une même règle), sont déclenchées en même temps, elle sont examinées une à une pour vérifier qu'aucun état instable n'est atteint. Les rebouclages ne sont pas possibles du fait que par hypothèse, l'état d'une règle ne peut influer sur une autre règle.

125. Il existe p. ex. des algorithmes qui permettent d'assurer qu'il n'y a pas de cycle dans un graphe. Le problème des transitions est moins évident, mais nous considérons que si les connaissances sont intègres, elles doivent pouvoir être exprimées facilement avec ce type de formalisme. Ce type de conflit est par ailleurs simple à détecter, ce qui permettrait d'avertir l'utilisateur de l'existence du problème.

3.4.2.3 Synthèse

Dans ce paragraphe, nous avons décrit les mécanismes proposés pour prendre en compte un ensemble de règles de fabricabilité, que nous avons supposées déjà instanciées. L'impact des règles sur la conception, suite à des modifications qui ne touchent pas la topologie du modèle de dialogue a été présenté. Ce mécanisme se base sur un graphe des grandeurs de fabrication (GGF) qui permet d'optimiser le calcul des grandeurs de fabrication, et une série de graphes des états (GE) – un par aspect – qui optimisent le calcul des transitions entre états.

Dans le paragraphe suivant, nous présentons la façon dont les règles de fabricabilité évoluent suite aux modifications topologiques. En particulier, nous proposons un système de propagation, permettant aux règles d'être extrapolées après de telles modifications.

3.4.3 Prise en compte dynamique des règles

Dans ce paragraphe, nous présentons la gestion dynamique des règles de fabricabilité. Nous y étudions donc les critères qui gouvernent l'instanciation de nouvelles règles.

Pour cela, nous commençons par définir l'application d'un outil de conception DIJA en termes d'éléments de dialogue, et nous décrivons le processus de mise à jour d'un point de vue global (§ 3.4.3.1). Les possibilités de transformation structurelles du modèle de dialogue sont ensuite catégorisées à travers la définition de fonctions et d'opérateurs (§ 3.4.3.2), puis un mécanisme de propagation des règles est présenté (§ 3.4.3.3). La propagation consiste à extrapoler les règles présentes dans le modèle de fabricabilité après chaque modification de la structure du modèle de conception.

3.4.3.1 Outils de conception DIJA

De façon générique, un outil DIJA définit la mutation d'un ensemble d'ED $\mathbb{E}_1$ en un ensemble d'ED $\mathbb{E}_2$ par l'application d'un modèle de transformation éventuellement paramétré ou interagissant avec l'utilisateur (cf. figure 3.17). Une transformation peut être une déformation, une division, une transmutation, ou une succession quelconque de ces trois opérations.

Les fonctions présentées au § 3.4.3.2, notées « fontion de correspondance » et « fonction de référencement », ont pour but de définir le lien de transformation entre les ensembles $\mathbb{E}_1$ et $\mathbb{E}_2$ (à quel élément de $\mathbb{E}_1$ se rattache chaque élément de $\mathbb{E}_2$). En fonction des règles définies sur $\mathbb{E}_1$ et de la politique de propagation choisie pour chaque règle, le système peut ainsi déterminer

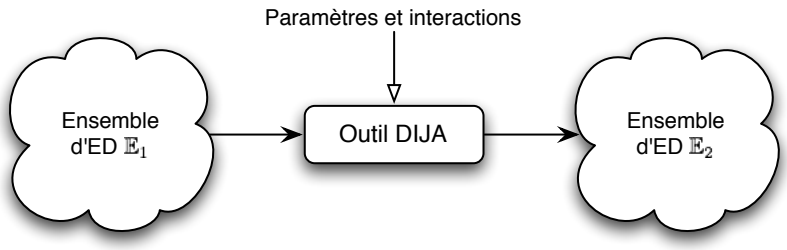


FIGURE 3.17: Application d'un outil DIJA.

les règles s'appliquant à $\mathbb{E}_2$ (mise à jour du GGF). Pour le système, la gestion des règles peut alors se décomposer en trois étapes, menées de façon automatique :

1. Détermination des fonctions de correspondance et de référencement ;
2. Propagation des règles en fonction de la politique de propagation choisie ;
3. Insertion de règles spécifiques à l'outil.

Dans la suite, nous définissons les fonctions de correspondance et de référencement, et comment les obtenir (§ 3.4.3.2), puis nous décrivons les différents types de propagation, et expliquons comment propager les règles en connaissant tous ces éléments (§ 3.4.3.3). L'insertion de règles spécifiques est abordée au § 3.4.3.4.

3.4.3.2 Transformations structurelles

3.4.3.2.1 Fonction de correspondance.

Définition. Cette thèse ne concernant pas les IHM, nous ne développons pas le comportement des outils (modèles de déformation) et nous considérons simplement un outil à travers sa fonction de correspondance :

$$f_c : \mathbb{E}_1 \rightarrow \mathbb{E}_2$$

où $\mathbb{E}_1$ est l'ensemble des ED modifiés par l'outil DIJA (de manière directe ou indirecte), et $\mathbb{E}_2$ l'ensemble des ED voués à se substituer à $\mathbb{E}_1$ après application de l'outil. La fonction f_c est utilisée par la suite afin de faire évoluer le graphe des grandeurs de fabrication (GGF) de la pièce.

La figure 3.18 présente le cas général d'application d'un outil DIJA. Les éléments de dialogue initiaux y sont représentés par l'ensemble E_i ; P est l'ensemble des paramètres de l'outil

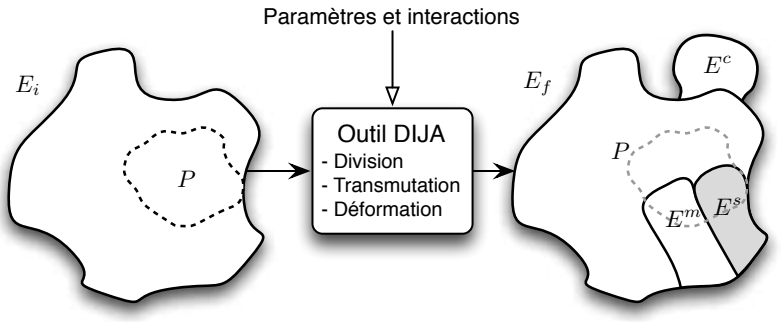


FIGURE 3.18: Cas général d'application d'un outil DIJA (la partie grisée E^s n'est pas incluse dans E_f).

($P \subseteq E_i$). E_f est l'ensemble des ED obtenus après application de l'outil. Nous avons :

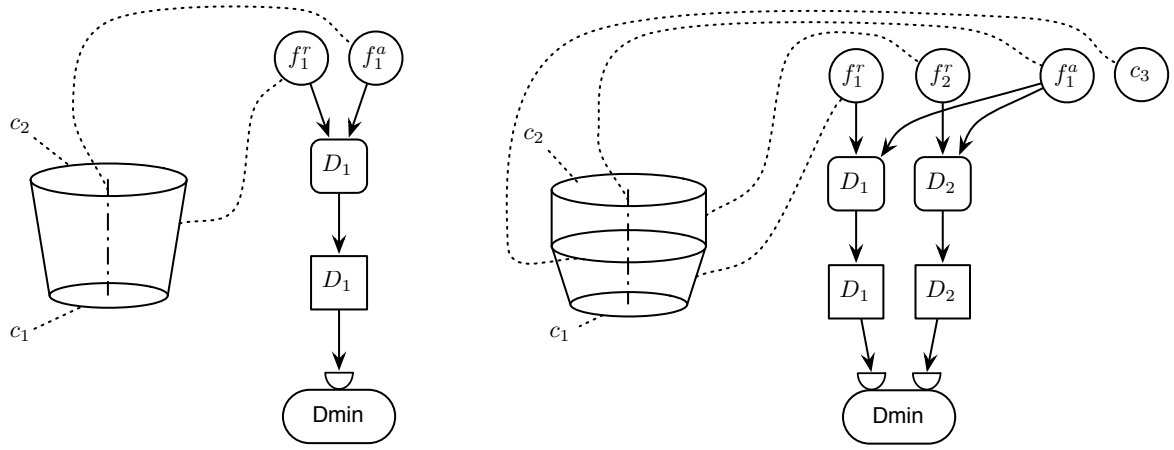
$$E_f = (E_i - E^s) \cup E^c$$

où E^s est l'ensemble des ED éliminés par l'outil et E^c l'ensemble des ED créés par l'outil. Enfin, E^m est l'ensemble des ED modifiés par l'outil. Nous avons donc :

$$\begin{aligned} \mathbb{E}_1 &= E^m \cup E^s \\ \mathbb{E}_2 &= E^m \cup E^c \end{aligned}$$

Exemple. La figure 3.19 présente un exemple d'application d'un outil DIJA. Cet outil, qui n'instancie pas directement de nouvelle règle, utilise successivement des opérations de division et de déformation sur une pièce soumise à une règle de fabrication globale (cf. figure 3.19a). Cette règle indique une valeur minimale pour le diamètre du trou.

Après que l'outil ait été appliqué, le graphe initial est censé (ce n'est qu'un exemple) se transformer en celui présenté sur la figure 3.19b. La fonction de correspondance est alors celle de la figure 3.20. La fonction f_c informe le système que la fibre de révolution f_1^r s'est dédoublée, que la fibre axiale f_1^a est inchangée (structurellement), et que le contour caractéristique c_3 n'a pas d'antécédent. De là, le bon comportement devrait pouvoir être adopté en fonction du type de propagation choisi pour les valeurs de fabrication associées à ces ED (ce point sera abordé au § 3.4.3.3.1). Ceci vaut par exemple pour la règle présentée sur la figure 3.19. Dans ce cas, le



(a) Graphe initial (b) Graphe final

FIGURE 3.19: Exemple de déformation et de GGF associés (les demi-disques indiquent les instances d'une même règle).

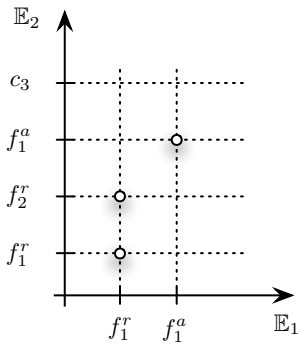
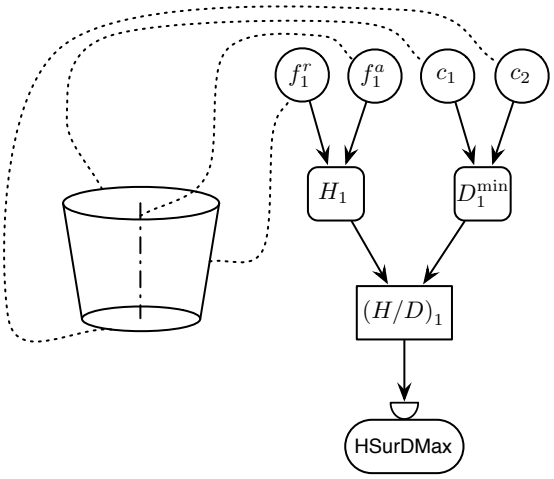


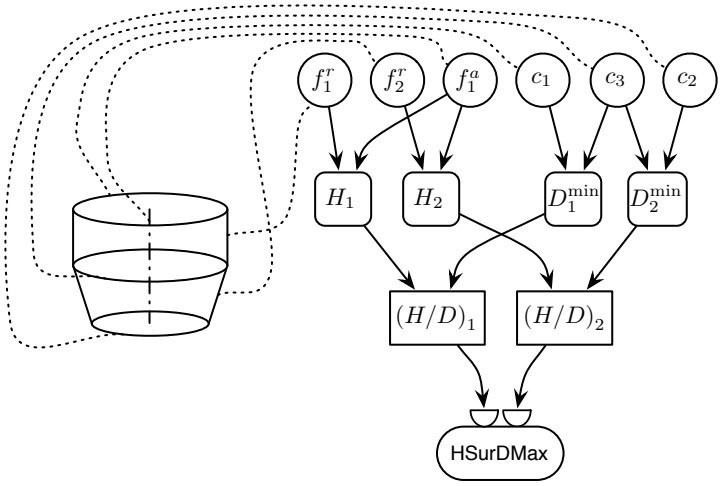
FIGURE 3.20: Fonction de correspondance f_c (seuls les fibres et un contour sont indiqués).

système est en mesure de dédoubler la partie du graphe qui est reliée à f_1^r.

Cependant, il existe des cas où la fonction de correspondance n'est pas suffisante. En effet, sur la figure 3.21, nous reprenons l'exemple précédent en changeant la règle de telle sorte qu'elle soit à présent basée sur les contours caractéristiques c_1 et c_2 (règle `HSurDMax`). Pour obtenir le graphe de la figure 3.21b, le système doit avoir connaissance que c_3 est le pendant de c_1 et de c_2 vis-à-vis de la règle `HSurDMax`. La fonction de correspondance étant inchangée par rapport à l'exemple précédent (figure 3.20), nous arrivons à la conclusion que celle-ci est nécessaire mais pas suffisante. C'est pourquoi nous introduisons au § 3.4.3.2.2 une autre fonction, appelée « fonction de référencement », servant à pallier ce manque.



(a) Graphe initial



(b) Graphe final

FIGURE 3.21: Exemple transformation de GGF où une fonction de référencement est nécessaire ($D_1^{\min}$ et $D_2^{\min}$ sont les diamètres minimaux).

Obtention de f_c. Nous expliquons ici pourquoi il est toujours possible d'obtenir la fonction f_c. La fonction de correspondance rend explicites les modifications structurelles du modèle de dialogue (B-Rep étendu). En toute logique, un peu à la manière des opérateurs d'Euler vis-à-vis de la topologie (cf. annexe A), l'ensemble des transformations possibles sont couvertes par les huit opérateurs suivants :

Effacement (noté `e`) : suppression d'un élément de dialogue : $f_c(x) = \emptyset$ avec $x \in \mathbb{E}_1$;

Ajout (noté `a`) : ajout d'un élément de dialogue : $f_c^{-1}(y) = \emptyset$ avec $y \in \mathbb{E}_2$;

Identité (noté `i`) : élément de dialogue inchangé (structurellement[126]) : $f_c(x) = \{x\} \wedge f_c^{-1}(x) = \{x\}$ avec $x \in \mathbb{E}_1 \cap \mathbb{E}_2$;

Substitution (noté `s`) : un élément de dialogue est substitué par un autre : $f_c(x) = \{y\} \wedge f_c^{-1}(y) = \{x\} \wedge x \neq y$ avec $x \in \mathbb{E}_1, y \in \mathbb{E}_2$;

Division (noté `d`) : un élément de dialogue x se divise en n parties ($n > 1$) : $\text{Card}(f_c(x)) = n \wedge x \in f_c(x) \wedge \forall y \in f_c(x), f_c^{-1}(y) = \{x\}$ avec $x \in \mathbb{E}_1$. Dans ce cas, toutes les parties sauf x seront des éléments de dialogue de l'ensemble E^a ;

Substitution et division (noté `sd`) : *idem* division mais avec $x \notin f_c(x)$;

Fusion (noté `f`) : n éléments de dialogue fusionnent en y ($n > 1$) : $\text{Card}\left(f_c^{-1}(y)\right) = n \wedge y \in f_c^{-1}(y) \wedge \forall x \in f_c^{-1}(y), f_c(x) = \{y\}$ avec $y \in \mathbb{E}_2$;

Substitution et fusion (noté `sf`) : *idem* fusion mais avec $y \notin f_c^{-1}(y)$.

Ceci garantit le fait que toute transformation structurelle effectuée par un outil DIJA pourra être décrite par l'intermédiaire d'une fonction de correspondance. Il n'y a cependant pas unicité et c'est l'outil qui définit la solution adoptée.

Propriétés de f_c. Du fait qu'il peut y avoir ajout ou effacement d'ED durant l'application de l'outil, il se peut qu'il y ait $\text{Card}(\mathbb{E}_1) \neq \text{Card}(\mathbb{E}_2)$. En outre, nous ne pouvons rien supposer de la fonction f_c dans le cas général : elle n'est pas nécessairement une application[127] ; elle n'est pas nécessairement injective[128] ; et elle n'est pas nécessairement surjective[129]. Il y a cependant

126. Ce cas ne se présente pas un grand intérêt du fait que les modifications purement géométriques n'induisent pas de modification des règles de fabricabilité.
127. Tout élément n'a pas nécessairement une et une seule image.
128. Deux éléments peuvent avoir la même image.
129. Tout élément de l'ensemble d'arrivée n'est pas nécessairement l'image d'un élément de l'ensemble de départ. Vu que f_c sert de support pour la propagation des règles du GGF, les éléments de l'ensemble d'arrivée n'ayant pas d'antécédent ne peuvent intervenir dans les nouvelles règles (du moins, si l'on ne considère que la fonction de correspondance).

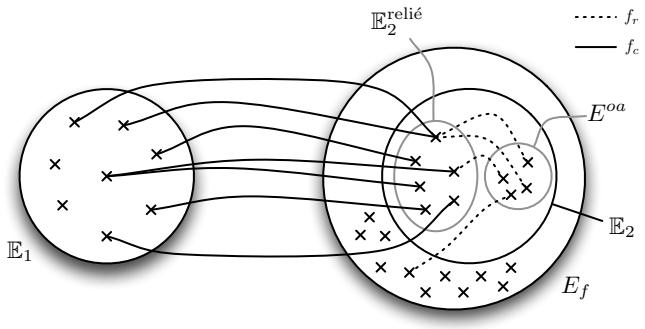


FIGURE 3.22: Ensembles impliqués dans les deux fonctions.

deux propriétés fondamentales qui peuvent être associées à la fonction f_c :

- tout élément divisé ne peut être fusionné : $\forall x \in \mathbb{E}_1, \text{Card}\left(f_c\left(x\right)\right) > 1 \Rightarrow \forall y \in f_c\left(x\right), f_c^{-1}\left(y\right) = \{x\}$;
- et réciproquement, toute fusion ne peut impliquer des éléments divisés : $\forall y \in \mathbb{E}_2, \text{Card}\left(f_c^{-1}\left(y\right)\right) > 1 \Rightarrow \forall x \in f_c^{-1}\left(y\right), f_c\left(x\right) = \{y\}$.

En plus de ces deux propriétés, la fonction f_c doit également vérifier une concordance au niveau des types. Par exemple, le fruit de la fusion de plusieurs fibres ne pourra être qu'une fibre. Pour une division, nous avons $\forall y \in f_c\left(x\right), \text{TypeED}\left(y\right) = \text{TypeED}\left(x\right)$, et pour une fusion, $\forall x \in f_c^{-1}\left(y\right), \text{TypeED}\left(x\right) = \text{TypeED}\left(y\right)$, l'application *TypeED* étant définie ainsi :

$$\text{TypeED} : X^{ED} \rightarrow \{\texttt{fibreRev}, \texttt{fibreAx}, \texttt{contourC}, \texttt{ligneC}, \texttt{face}\}$$

3.4.3.2.2 Fonction de référencement

Définition. Ce que ne décrit pas la fonction f_c est la provenance ou le rattachement de certains des éléments de dialogue ajoutés par l'outil. Plus précisément, le rôle de la fonction de référencement, notée f_r, est de déterminer ce à quoi rattacher les ED issus de l'opérateur `a` (opérateur d'ajout, cf. paragraphe précédent), dans le but de pouvoir faire évoluer le GGF de façon cohérente. Soit E^{oa} l'ensemble des ED issus de cet opérateur ($E^{oa} \subseteq E^c$). La figure 3.22 illustre ceci. Nous définissons la fonction de référencement comme suit :

$$f_r : E^{oa} \rightarrow \left(E^f - E^{oa}\right)$$

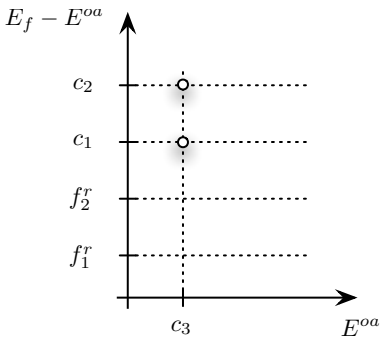


FIGURE 3.23: Exemple de fonction de référencement (représentation partielle).

Exemple. La figure 3.23 présente la fonction de référencement correspondant à l'exemple de la figure 3.21. Elle indique au système que c_3 est « issu » de c_1 et c_2. Autrement dit, c_3 s'apparente à c_1 et c_2 (même type d'ED), et s'y rattache topologiquement (face commune). Dans cet exemple, les fonctions de correspondance et de référencement sont alors suffisantes pour faire évoluer le graphe de la figure 3.21a en celui de la figure 3.21b. Il n'est cependant pas garanti, a priori, que tel sera toujours le cas. Notamment, nous nous basons ici sur l'hypothèse que les fonctions f_c et f_r définissent tous les liens nécessaires existant entre les ensembles $\mathbb{E}_1$ et $\mathbb{E}_2$, et que ceux-ci sont fidèles à la transformation réellement opérée.

Obtention de f_r. La fonction de référencement relie à des éléments connus les ED qui ne le sont pas encore. Le critère principal pour qu'un ED se réfère à un autre est qu'ils soient en relation du point de vue structurel dans le modèle de représentation. Dans l'exemple présenté plus haut, c_3 est rattaché à c_1 et c_2 car il se trouve que c_3 constitue une jonction entre deux faces dont l'une est reliée à c_1 et l'autre à c_2. C'est donc par un parcours du modèle de dialogue guidé par un certain nombre de règles pré-établies que le système pourrait être en mesure de définir la fonction f_r. Dans le cadre de cette thèse nous ne poussons pas l'investigation plus en avant dans cette direction. Cette question relève des domaines de la topologie et de la géométrie (approches d'extraction localisée). Nous considérons pour l'heure que la fonction f_r est, tout comme f_c, donnée par l'outil.

Propriétés de f_r. Les propriétés de la fonction f_r sont analogues à celles de la fonction f_c, exception faite de l'impossibilité qu'un ED puisse à la fois intervenir dans une division et une fusion : ce type de configuration est possible pour f_r.

TABLE 3.1: Types de propagation possibles pour chaque opérateur.

Opérateurs de Op	Types de propagation correspondants T_{Op}^{propa}
`i`	{`conservation`}
`a`	$\emptyset$
`e`, `s`	{`effacement`}
`d`, `f`, `sf`	{`effacement`, `conservation`, `adaptation`}
`sd`	{`effacement`, `adaptation`}

Dans le paragraphe suivant, le processus de propagation est décrit en se basant sur les fonctions f_c et f_r.

3.4.3.3 Propagation

Dans ce paragraphe nous abordons la façon dont les règles évoluent en nous basant sur les fonctions f_c et f_r introduites au paragraphe précédent.

3.4.3.3.1 Paramétrage de la propagation. Dans le GGF, à chaque ED, un certain nombre de VF sont reliées. Suivant l'opérateur appliqué à un ED, plusieurs choix d'évolution sont possibles pour chacune des VF initialement connectées (voir p. ex. les figures 3.19 et 3.21) : suppression, duplication, adaptation... Certains opérateurs n'offrent qu'une seule alternative alors que d'autres en offrent plusieurs. Ces dernières sont définies ci-après. Dans la mesure où il doit être possible de choisir le type d'évolution pour chaque arc existant entre les ED et les VF du GGF, nous associons à chaque triplet (e, v, o) valide [130] ($e \in X^{ED}$, $v \in X^{VF}$ et $o \in Op$), un type de propagation $t \in T_{Op}^{\text{propa}}$. Soit f_p la fonction qui fournit cette information. Nous avons :

$$f_p : X^{ED} \times X^{VF} \times Op \rightarrow T_{Op}^{\text{propa}}$$

avec $T_{Op}^{\text{propa}} \subseteq \{\texttt{effacement}, \texttt{conservation}, \texttt{adaptation}\}$ et $Op \in \{\texttt{e}, \texttt{a}, \texttt{i}, \texttt{s}, \texttt{d}, \texttt{sd}, \texttt{f}, \texttt{sf}\}$. La table 3.1 recense les différents types de propagation pour chaque opérateur. Les opérateurs pour lesquels il est nécessaire de conserver le type de propagation sont les opérateurs pour lesquels $\text{Card}\left(T_{Op}^{\text{propa}}\right) > 1$, c'est-à-dire `d`, `sd`, `f` et `sf`. Les paragraphes suivants expliquent plus en détail les différentes possibilités d'évolution du GGF pour chaque opérateur.

130. Par valide, nous entendons que pour chaque triplet, la VF v doit être reliée à l'ED e dans le GGF, c.-à-d. que $\forall (e, v, o) \in f_p, v \in \text{vfd}(e)$ (la fonction *vfd* renvoie les VF reliées à l'élément de dialogue e, cf. annexe C).

Propagation avec l'opérateur `i`. Dans le cas de `i`, aucune modification n'est nécessaire ; les règles sont donc laissées telles quelles.

Propagation avec l'opérateur `a`. Lorsque la fonction f_c indique qu'un ED de $\mathbb{E}_2$ n'a pas d'antécédent, aucune propagation n'est requise car l'élément ajouté n'étant pas présent dans le GGF initial, aucune règle ne peut lui être associée, sauf en tant que référant via f_r, mais dans ce cas, ce n'est pas cet ED qui déclenche la propagation (ceci est explicité dans la description de l'algorithme de propagation, en annexe C).

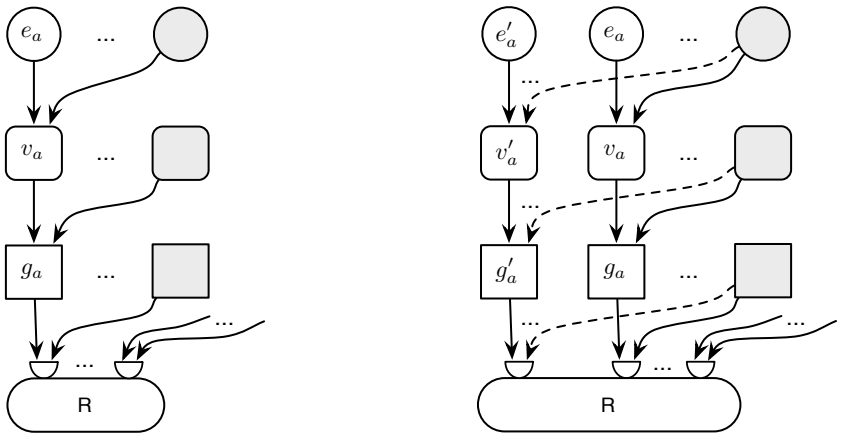
Propagation avec les opérateurs `e` et `s`. Suite à une suppression d'ED (lorsque la fonction f_c indique qu'un élément n'a pas d'image), aucune propagation n'est possible. Les instances de règles ainsi déconnectées de X^{ED} sont alors systématiquement supprimées. La substitution d'un ED par un autre ne peut, à notre sens, avoir pour effet que la suppression de l'ensemble des instances de règles qui lui sont associées.

Propagation avec les opérateurs `d` et `sd`. Suite à une division d'ED (lorsque la fonction f_c indique que plusieurs images différentes ont un même antécédent), les trois types de propagation existants sont possibles :

1. `effacement` : dans ce cas, la division a pour conséquence la suppression des règles connectées ;
2. `conservation` : dans ce cas, il n'y a qu'à ajouter les nouveaux ED dans le graphe et à conserver telles quelles les règles associées à l'ED divisé (les nouveaux ED n'y sont pas connectés) ;
3. `adaptation` (cf. figure 3.24) : les instances des règles associées à l'ED divisé sont alors dupliquées, ce qui induit également la recopie des VF et des GF associées.

Remarquons que pour le cas n° 3, les duplicata utilisent des références sur les autres ED utilisés par la règle avant division (voir les arcs en pointillés sur la figure 3.24). Ceci doit en réalité être mis en corrélation avec les autres divisions d'ED pouvant survenir en parallèle, ainsi que les relations données par la fonction f_r. Cette question est traitée dans le § 3.4.3.3.2.

Le cas de l'opérateur `sd` se rapporte à celui de `d` à ceci près que dans le cas n° 2, les règles associées ne sont pas conservées (il y a alors élimination de la règle, ce qui équivaut au cas n° 1).



(a) Graphe avant division

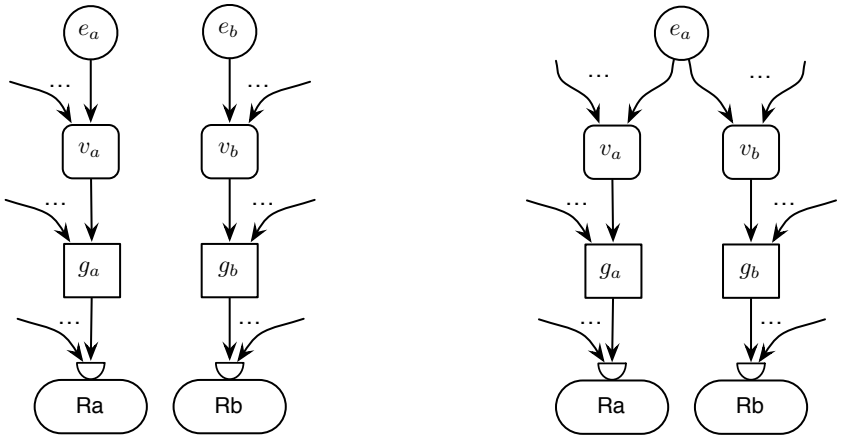
(b) Graphe après division (adaptation)

FIGURE 3.24: Adaptation lors de la division d'un ED (e_a se divise en e_a et e'_a). Le raisonnement est similaire pour la division d'un ED en plus de deux parties.

Propagation avec les opérateurs `f` et `sf`. Suite à une fusion d'ED (lorsque la fonction f_c indique que plusieurs éléments ont la même image), les trois types de propagation sont possibles :

1. `effacement` : dans ce cas, la fusion a pour conséquence la suppression des règles connectées ;
2. `conservation` : dans ce cas, les arcs anciennement rattachés à chaque ED sont rattachés au nouvel ED (le fruit de la fusion) ;
3. `adaptation` : dans ce cas, il existe deux configurations non exclusives. En effet, les ED à fusionner peuvent se rattacher à des règles distinctes (il s'agit du cas de la figure 3.25), ou bien à une même règle (il s'agit des quatre cas de la figure 3.26). Pour les parties menant à des règles distinctes, il y a conservation (cf. figure 3.25b). En revanche, pour les parties menant à une même règle, seule la jonction de la figure 3.26d peut être réellement adaptée (voir figure 3.27), à condition toutefois, que l'analogie soit possible entre les différentes instances à fusionner [131]. Dans les autres cas, la fusion aboutira donc nécessairement à une conservation. Ceci peut mener à des situations avec des arcs redondants dans les cas

131. Plus précisément, en reprenant l'exemple de la figure 3.26d, il est nécessaire de vérifier que les sous-graphes correspondant aux chemins $e_a, v_a, g_a, \mathtt{R}$ et $e_b, v_b, g_b, \mathtt{R}$ sont bien compatibles ; c'est-à-dire qu'ils comportent des connexions orientées vers les sommets $e_a, v_a, g_a, \mathtt{R}$ et $e_b, v_b, g_b, \mathtt{R}$ identiques en genre et en nombre ; les connexions orientées vers d'autres sommets ne sont pas gênantes.



(a) Pas de jonction (b) Fusion de e_a et e_b

FIGURE 3.25: L'adaptation consiste en une conservation si les règles sont disjointes.

des figures 3.26a, 3.26b et 3.26c mais il n'existe pas de meilleure solution du fait que le raisonnement doit s'effectuer dans le cas général.

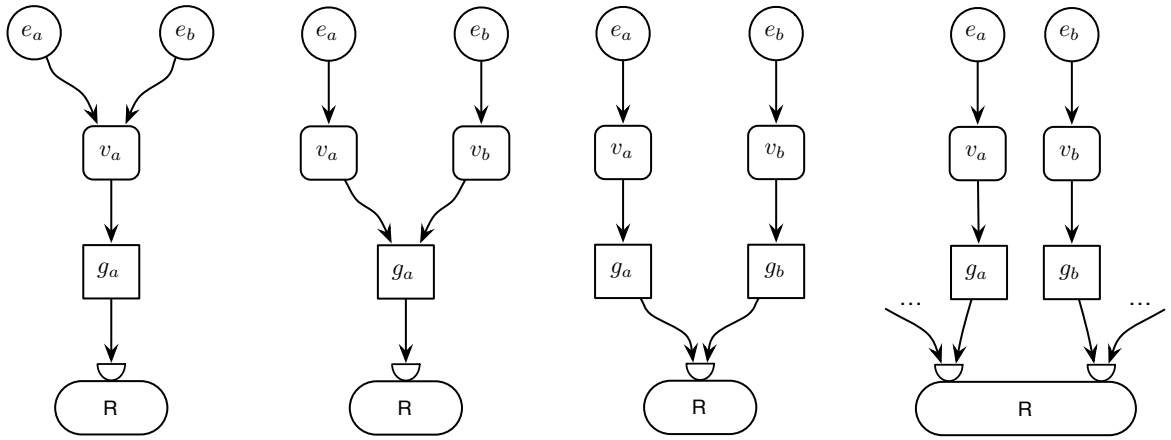
Le cas de l'opérateur `sf` est analogue à celui de `f`. Notons tout de même que dans le cas n° 3 les VF et les GF conservées ne pourront être choisies qu'arbitrairement.

Dans le paragraphe suivant, l'algorithme utilisé pour la propagation des règles est présenté.

3.4.3.3.2 Algorithme de propagation. Dans ce paragraphe, nous décrivons le processus de mise à jour des règles suite à une modification de la structure du modèle de conception. Nous ne considérons donc que l'évolution (nouvelle instanciation, destruction, modification) de règles déjà instanciées. L'instanciation « directe » de règles peut, quant à elle, être effectuée directement sur la pièce initiale (préalablement pour les pièces de la librairie, ou via un scénario pour une esquisse), ainsi que de façon spécifique via l'outil (cf. § 3.4.3.4).

Propagation. L'algorithme 3.2 présente d'un point de vue global les actions entreprises suite à l'application d'un outil. Il est tout d'abord effectué une copie $\mathbb{G}_2$ du GGF initial $\mathbb{G}_1$. $\mathbb{G}_1$ est utilisé pour garder une trace des connexions existant initialement entre les ED de $\mathbb{E}_1$, les VF, les GF et les règles. Ceci est nécessaire car ces ensembles et leurs connexions sont altérés dans $\mathbb{G}_2$ au fur et à mesure que l'algorithme s'exécute. Celui-ci se constitue de différentes étapes :

1. Suppression des ED de E^s ;



(a) Jonction sur une VF (b) Jonction sur une GF (c) Jonction sur une instance de règle (d) Jonction sur une règle abstraite

FIGURE 3.26: Différents types de jonctions possibles avant fusion.

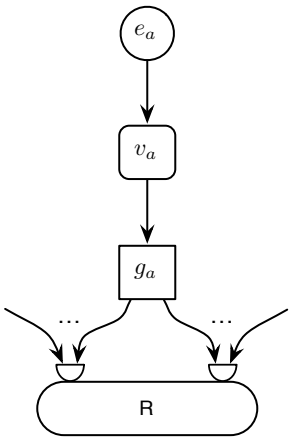


FIGURE 3.27: Graphe après fusion (adaptation) de e_a et e_b sur la figure 3.26d (en considérant qu'e_a doit être le fruit de la fusion).

2. Insertion des ED de E^a ;
3. Substitution des ED issus de l'opérateur `s` ;
4. Fusion des ED issus des opérateurs `f` et `sf` et répercussion sur les règles correspondantes ;
5. Division des ED issus des opérateurs `d` et `sd` et répercussion sur les règles correspondantes.

Les étapes n° 3 et 4 sont en réalité menées en parallèle afin d'économiser une boucle (cf. algorithme 3.2). Le détail des algorithmes *Fusion* et *Division* est disponible en annexe C, page 185. Ils sont décrits de manière informelle dans les deux paragraphes suivants.

Division. Pour diviser un ED e_a, l'idée générale consiste à parcourir chaque VF connectée v_a et d'adapter le traitement en fonction du type de propagation enregistré pour chaque triplet $(e_a, v_a, \texttt{d})$ ou $(e_a, v_a, \texttt{sd})$ selon le cas. Pour les cas de l'effacement et de la conservation, le traitement est simple : il suffit d'effacer les règles ou de les laisser telles quelles. Le cas de l'adaptation est quant à lui plus complexe : il faut dupliquer les règles de façon cohérente. En particulier, des corrélations doivent être effectuées afin de ne pas dupliquer plusieurs fois la même règle, par exemple dans le cas où deux ED utilisés par une règle sont divisés en parallèle. Ce type de cas est détecté en conservant jusqu'à la fin de l'algorithme l'ensemble des ED, VF et GF déjà divisés. Si une division compatible a déjà été opérée sur un élément connecté à un ED en cours de division, alors les duplicata sont mis en correspondance et réutilisés. Dans cette corrélation, intervient également la fonction f_r.

Fusion. Pour fusionner un ensemble d'ED, le fonctionnement est similaire : le traitement s'adapte en fonction des types de propagation conservés pour chaque triplet $(e_x, v_x, \texttt{f})$ ou $(e_x, v_x, \texttt{sf})$. Les cas de l'effacement et de la conservation sont simples pour les mêmes raisons. Le cas de l'adaptation se révèle être mené à l'inverse du cas de la division où les règles sont dupliquées en premier : ici, les règles ne sont totalement regroupées qu'au moment où les derniers éléments sont sur le point d'être fusionnés.

3.4.3.4 Insertion de règles spécifiques

Chaque outil peut bien sûr insérer des règles s'appliquant aux ED présents à l'issue de la transformation ; il s'agit principalement d'éléments de $\mathbb{E}_2$. L'insertion des règles spécifiques se fait après que la propagation ait eu lieu, de sorte que les VF et les GF insérées dans le GGF

ALGORITHME 3.2: Propagation.

procédure Propagation $(\mathbb{E}_1, \mathbb{E}_2, f_c, f_r)$

Recopier le graphe des grandeurs initial $\mathbb{G}_1$ en $\mathbb{G}_2$
{Suppression des ED de E^s}
pour tout $x \in \mathbb{E}_1$ **faire**
  si $f_c(x) = \emptyset$ **alors**
    Supprimer x et les instances des règles qui lui sont associées de $\mathbb{G}_2$
  fin si
fin pour
{Insertion des ED de E^a}
pour tout $x \in \mathbb{E}_2$ **faire**
  si $f_c^{-1}(x) = \emptyset$ **alors**
    Insérer x dans $\mathbb{G}_2$
  fin si
fin pour
{Fusions et substitutions}
pour tout $x \in \mathbb{E}_1$ **faire**
  si Card $(f_c(x)) = 1$ **alors**
    Soit $i \mid i \in f_c(x)$
    si Card $\left(f_c^{-1}(i)\right) = 1$ **alors**
      si $i \neq x$ **alors**
        Substituer x par i dans $\mathbb{G}_2$
      fin si
    sinon, si $i \notin f_c^{-1}(i)$ **alors**
      fusion $\left(f_c^{-1}(i), f_c(x)\right)$
    fin si
  fin si
fin pour
{Divisions}
pour tout $x \in \mathbb{E}_1$ **faire**
  si Card $(f_c(x)) > 1$ **alors**
    division $(x, f_c(x))$
  fin si
fin pour

lors de la propagation puissent être réutilisées. Nous ne décrivons pas davantage ce processus car il consiste simplement à instancier des règles en les rattachant à des ED situés dans des zones spécifiques (zones que l'outil est normalement apte à localiser par l'intermédiaire de ses paramètres). Naturellement, de telles règles peuvent par la suite être propagées.

3.4.4 Synthèse

Dans cette section, nous avons présenté les processus de gestion de la fabricabilité dans le cadre de la modélisation synthétique. Nous avons dissocié le cas où les modifications ne touchent que la géométrie du modèle, du cas où elles s'appliquent également à la topologie (niveau dialogue).

En premier lieu, il a été proposé de réaliser le calcul des expressions formant les règles de fabricabilité par l'intermédiaire d'un graphe des grandeurs de fabrication (GGF) qui permet d'optimiser (dans une certaine mesure) le calcul des grandeurs de fabrication au cours de la conception. Nous avons également adapté la représentation des connaissances présentée dans la section 3.3 (règles de production), afin de mieux prendre en compte les aspects, caractérisés par des ensembles d'états exclusifs (graphes des états).

En second lieu, nous avons proposé les fonctions de correspondance f_c et de référencement f_r afin de prendre en compte l'évolution topologique du modèle de dialogue. Les différentes évolutions topologiques ont été classifiées par l'intermédiaire d'opérateurs intervenant dans la fonction f_c. Le GGF a été complété afin de conserver les différentes stratégies d'évolution possibles, suite aux modifications topologiques. Enfin, nous avons présenté un algorithme permettant la propagation des règles, en se basant sur ces éléments.

La section 3.5 est un bilan de ce qui a été présenté dans ce chapitre, et recense les apports ainsi que quelques limites de notre approche.

3.5 Bilan

Dans ce chapitre, nous avons proposé une approche de gestion de la fabricabilité basée sur l'approche synthétique du projet DIJA, une approche de modélisation descendante qui fonctionne par raffinements successifs. Nous introduisons un module de fabricabilité pour chacun des trois premiers niveaux d'abstraction de DIJA, la jonction entre les modèles de conception et de fabrication s'effectuant au niveau dialogue. L'approche proposée permet de stipuler des

règles génériques, a priori applicables à tout type de procédé. Cette approche s'effectue en temps réel, au fur et à mesure de la conception, ce qui a l'avantage d'éviter de trop longs rebouclages, et permet de sensibiliser le concepteur aux problèmes de fabrication. Les informations prises en compte peuvent être de natures variées du fait qu'elles sont susceptibles de provenir du module de connaissances (à plusieurs niveaux d'abstraction). En outre, l'utilisation du niveau dialogue comme base (modèle canonique), apporte une garantie vis-à-vis de la confidentialité.

Cette approche montre néanmoins quelques limites, dès lors que l'on considère l'évolution des règles. En premier lieu, le GGF, les GE, ainsi que les fonctions de correspondance et de référencement doivent être spécifiés de manière cohérente par un ingénieur en gestion des connaissances, ce qui demande un certain investissement au départ. En outre, le fait de propager les règles peut, dans le cas où des interférences géométriques apparaissent du fait de l'application de plusieurs outils, mener à des situations complexes qu'il est difficile de prévoir à l'avance. Cela étant dit, de telles interférences ne sont jamais simples à gérer, y-compris avec des approches différentes (approches d'extraction, entre autres).

La spécification des règles de fabricabilité s'effectue à différents stades : au sein de la librairie de pièces initiales, et au sein des outils. Ceci complique la définition des interfaces de spécification, un certain nombre de vérifications devant être effectuées ; mais cette contrainte est imposée par l'approche synthétique. Dans le cas d'une esquisse, c'est là encore l'ingénieur en gestion des connaissances qui est sollicité. Des recherches supplémentaires seraient nécessaires afin d'améliorer cette phase, p. ex. concevoir des interfaces la mettant à la portée du fabricant.

Enfin, dans le cas où des recoupements ne pourraient être effectués (fonctions f_r et f_c incomplètes), le système aboutirait à une accumulation inutile d'éléments dans le GGF, ce qui alourdirait ce dernier, ou au contraire, à une perte d'informations, ce qui est plus ennuyeux encore. Nous reviendrons sur les limites de l'approche au chapitre *Conclusion*.

Le chapitre 4 présente l'implémentation qui a été faite de l'approche présentée dans ce chapitre.

Chapitre IV

Mise en œuvre informatique

Sommaire

4.1 Introduction 144

4.2 Contexte d'implémentation 144

4.2.1 Environnement DIJA 144

4.2.2 Limites de la maquette 145

4.3 Étude de l'implémentation 146

4.3.1 Architecture logicielle 146

4.3.2 Éléments relatifs à la représentation des connaissances 148

4.3.3 Répercussion des règles de fabricabilité 157

4.3.4 Mécanismes mis en œuvre pour l'évolution des connaissances 158

4.4 Présentation de la maquette 160

4.4.1 Interface 160

4.4.2 Étude de cas 161

4.5 Bilan 169

4.1 Introduction

Dans les chapitres précédents, nous avons présenté une approche de modélisation et d'exploitation de règles de fabricabilité au sein de l'approche DIJA. Cette approche permet de prendre en compte des règles de fabricabilité exprimées à travers deux types de graphes servant à représenter à la fois la fabricabilité (métiers de la fabrication), et ses relations avec la géométrie particulière de la pièce en cours de conception, là encore selon une approche métier (métiers de la conception). Après cette étude théorique, nous souhaitons présenter dans ce chapitre une application de quelques-uns de ces concepts à travers la réalisation d'une maquette écrite en langage Java.

Dans la suite, nous commençons par définir le contexte et le périmètre de cette implémentation (sect. 4.2). Nous présentons dans la section 4.3 une étude portant sur quelques aspects qui nous semblent centraux, avant de présenter la maquette à travers un scénario de conception typique accompagné de captures d'écran (sect. 4.4). La section 4.5 est un bilan d'implémentation où les principaux points abordés dans ce chapitre sont repris et critiqués.

4.2 Contexte d'implémentation

4.2.1 Environnement DIJA

L'implémentation présentée dans ce chapitre se base sur la librairie DIJA. Celle-ci est constituée de différents modules qui sont développés au cours des travaux des membres de l'équipe CMCAO. Ces travaux de recherche ont concerné les IHM [Danesi, 2002 ; Malik et coll., 2002], les aspects liés au réseau et à l'historique de conception [Denis, 2003], la définition d'une base de connaissances orientée métier [Gardan et Gardan, 2003], la sécurisation des échanges de données géométriques [Minich et coll., 2004], la modélisation du processus de conception dans un contexte d'entreprise étendue [Gardan, 2005] ou l'auto-adaptation de maillages [Heschung, 2005]. D'autres travaux sont actuellement en cours sur les thèmes de la prise en compte des fonctions dans la définition des formes [Brosse et coll., 2006 ; Brosse et coll., 2007], de l'optimisation des liens entre conception et simulation [Danesi et coll., 2007 ; Gardan et coll., 2007] ou de la déformation de surfaces [Ciuciu et coll., 2008].

Le fait que DIJA utilise l'Internet implique la prise en compte de contraintes portant sur

la nature du matériel utilisé ainsi que sur le niveau de compétence en informatique des utilisateurs. L'environnement DIJA se veut accessible à tout type d'utilisateur (novice ou expert) disposant d'un matériel standard avec un accès au réseau. Ces éléments ont conduit à choisir le langage objet Java [www-Java] pour l'implémentation de la librairie. La couche médiatrice de Java (dénommée *bytecode*) offre une certaine indépendance vis-à-vis du matériel et évite d'avoir à décliner la compilation pour chaque plateforme. En outre, un atout indéniable de ce langage est le nombre de bibliothèques directement accessibles : Java3D ou JOGL [132] pour la représentation de scènes tridimensionnelles [www-Java3D ; www-JOGL], JDBC [133] pour l'accès aux bases de données [www-JDBC], SAX [134] [www-SAX] et DOM [135] [DOM, 2004] pour l'utilisation du format d'échange XML *(eXtensible Markup Language)* [XML, 2006], etc.

Le cœur de DIJA est donc une API [136] comportant tous les éléments nécessaires à la programmation d'un logiciel de CAO de base. Le développement de modules spécifiques permet ensuite d'étendre ses fonctionnalités et de spécialiser le logiciel dans un ou plusieurs domaines particuliers. Ceci est l'approche générale que nous avons adoptée pour la réalisation de la maquette présentée dans la suite.

4.2.2 Limites de la maquette

À travers cette implémentation, nous souhaitons mettre en œuvre les concepts que nous avons présentés dans les chapitres précédents, afin de prouver leur viabilité. Néanmoins, l'objectif n'est pas de développer un système de CAO complet. Nous imposons ainsi certaines limites à cette implémentation et axons le développement sur les points qui paraissent les plus importants. L'accent a été mis sur la méthode de représentation des règles et l'évolution des graphes. En revanche, l'interface a été réduite à son minimum. En particulier, la spécification des règles de fabricabilité se fait uniquement par l'intermédiaire de fichiers, ce qui suffit à étudier la viabilité de l'approche, même si l'idéal, dans le cadre d'une utilisation industrielle, serait bien sûr de développer des outils à interfaces évoluées.

Par conséquent, nous commençons l'étude de l'implémentation par la présentation de l'architecture logicielle utilisée pour organiser les différents modules Java. Nous décrivons ensuite l'implémentation des concepts centraux dans le cadre de notre approche de gestion de la fa-

132. *Java Open Graphics Library*.
133. *Java DataBase Connectivity*.
134. *Simple API for XML*.
135. *Document Object Model*.
136. *Application Programming Interface*.

bricabilité. Nous détaillons ainsi la modélisation informatique des graphes des grandeurs et des graphes des états, et en particulier le double niveau de représentation employé pour les règles (notion de règle abstraite). Nous présentons également les mécanismes permettant l'exploitation de ces graphes, puis finalement, les mécanismes mis en œuvre dans le cadre de leur évolution.

4.3 Étude de l'implémentation

Cette étude s'organise comme suit. Le § 4.3.1 présente l'architecture générale de l'application. Le § 4.3.2 présente les éléments relatifs à la représentation des règles de fabricabilité. Le § 4.3.3 présente l'implémentation du mécanisme de répercussion de règles de fabricabilité déjà instanciées. Enfin, le § 4.3.4 présente l'implémentation des mécanismes d'évolution des règles.

4.3.1 Architecture logicielle

Nous abordons l'architecture de l'application à travers le diagramme de paquets présenté sur la figure 4.1. Ce type de diagramme, intégré à la norme UML *(Unified Modeling Language)* [ISO/IEC 19501, 2005], permet de représenter l'organisation des différents constituants logiciels d'une application, ainsi que les liens de dépendance et d'imbrication existant entre eux.

La figure 4.1 reprend l'architecture modulaire présentée au § 3.2.1.2 du chapitre précédent, en la précisant, en particulier au niveau du superviseur (paquet `dija.superviseur`) et de la base de connaissances (paquet `dija.connaissances`).

Le paquet `dija.superviseur` permet de gérer l'aiguillage des instructions dans les différents niveaux d'abstraction (application, métier, commun, dialogue et géométrique) ainsi que sur le réseau. Les niveaux d'abstraction ne sont pas représentés sur le diagramme. Comme ils fonctionnent sur des modes analogues, nous regroupons les structures et traitements qui les concernent dans des paquets identiques, mais dans des classes différentes (le niveau d'abstraction est donc considéré à l'exécution, par instanciation d'une classe particulière).

Le paquet `dija.connaissances` est découpé en deux sous-paquets correspondant aux connaissances des domaines de la conception et de la fabrication. Le paquet `dija.connaissances.conception` comprend les entités qui forment la base de connaissances décrite au § 3.2.1.3 (règles, contraintes, expressions grapho-numériques…). Le paquet

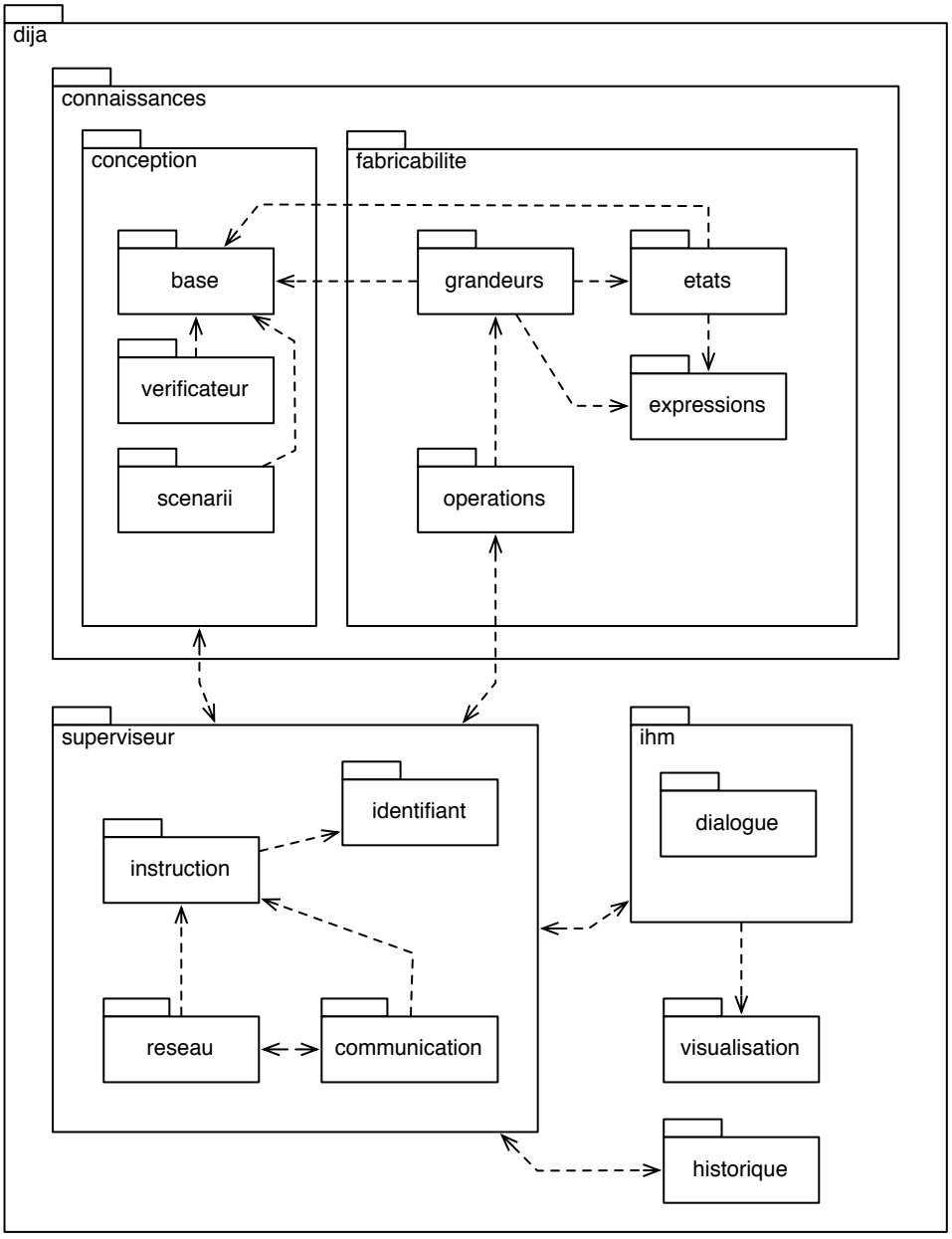


FIGURE 4.1: Architecture générale de l'application (les flèches correspondent à la notion d'importation).

`dija.connaissances.fabricabilite` englobe, quant à lui, les entités relatives à la fabricabilité (règles, états...). Ce dernier est à nouveau découpé en quatre sous-paquets :

le paquet `dija.connaissances.fabricabilite.grandeurs` comprend les classes utilisées pour représenter le graphe des grandeurs et donc les notions de règle de fabricabilité, de grandeur et de valeur de fabrication. Il est en lien avec la base de connaissances de conception, en particulier pour prendre en compte les paramètres grapho-numériques (PGN) ;

le paquet `dija.connaissances.fabricabilite.etats` comprend les classes utilisées pour représenter les graphes des états ;

le paquet `dija.connaissances.fabricabilite.operations` comprend des classes dédiées au traitement des opérations s'appliquant aux graphes des grandeurs de fabrication (fonctions de correspondance et de référencement). Il assure l'interface avec le superviseur ;

le paquet `dija.connaissances.fabricabilite.expressions` englobe des classes utilisées pour modéliser les expressions logiques et arithmétiques. Celles-ci sont utiles pour les graphes des grandeurs et des états.

C'est le superviseur qui est chargé de répercuter les modifications intervenant au niveau de la conception (éléments de dialogue ou bien PGN) dans la partie de la base de connaissances dédiée à la fabricabilité. Ainsi, à chaque opération menée sur un élément de dialogue ou un paramètre grapho-numérique, le superviseur vérifie que l'élément n'est pas référencé dans le graphe des grandeurs de fabrication de la pièce et, le cas échéant, transmet la modification au module de fabricabilité.

Dans le paragraphe suivant, nous décrivons plus en détail les éléments constitutifs des paquets `dija.connaissances.fabricabilite.grandeurs`, `dija.connaissances.fabricabilite.etats` et `dija.connaissances.fabricabilite.expressions`, qui permettent la représentation des connaissances.

4.3.2 Éléments relatifs à la représentation des connaissances

Dans cette section, nous mettons l'accent sur certains points du développement liés à la représentation des règles de fabricabilité : le modèle de conception, les expressions logiques et arithmétiques et le modèle de fabrication.

4.3.2.1 Modèle de conception

Le modèle de conception est un B-Rep étendu dont la structure simplifiée est présentée sur le diagramme de classes UML de la figure 4.2. Les classes `Solide`, `Face`, `Triangle`, `Arete` et `Sommet` sont des constituants habituels pour ce type de représentation (modèles de type FAS, cf. § 1.3.1.1). La classe `ED` regroupe les fibres, faces, contours et lignes caractéristiques, conformément à ce qui est indiqué au § 3.2.1, où ces concepts ont été introduits.

Dans le cadre du développement de cette maquette, cette structure a été complétée en y intégrant l'interface `Mesurable`, constituée d'une unique méthode, nommée `mesure()`, vouée à réaliser les mesures entre éléments géométriques (cf. figure 4.2). Les classes relatives aux éléments de dialogue, ainsi qu'aux éléments géométriques l'implémentent. Ceci permet d'effectuer des mesures n-aires ($n \geq 0$), en les définissant de manière récursive (p. ex. : la longueur d'une fibre, l'aire d'une face, la distance entre deux points, deux contours, etc.). L'arité des opérateurs de mesure n'est pas fixée car la méthode `mesure()` est dotée d'une liste d'arguments variable [137]. Ces arguments étant eux aussi définis comme des implémentations de l'interface `Mesurable`, la nature des opérandes n'est pas nécessairement homogène, ce qui permet par exemple de définir la distance entre un point et une fibre. Les opérateurs de mesure pourraient en outre être chargés à l'exécution, similairement aux règles de fabricabilité. Mais le périmètre de la maquette ne nécessitant pas ce degré de complexité, nous nous limitons à des mesures unaires ou binaires et à quelques opérateurs de mesure géométriques simples (longueur, distance, distance minimale…), qui sont donc codés de manière statique (l'annexe D indique les calculs utilisés pour ces opérateurs).

Si des anomalies surviennent lors d'une mesure (une incohérence de calcul, p. ex. liée à une division par zéro), une exception de mesure est déclenchée. En capturant cette exception, le système est en mesure d'adopter un comportement adéquat : ignorer les règles affectées par le calcul, laisser leur état inchangé, informer l'utilisateur en les plaçant dans un état indéfini… Là encore, ce comportement pourrait être déterminé par configuration, mais il a été fait le choix de signaler l'erreur à l'utilisateur par un simple message afin de ne pas alourdir inutilement le développement. Un appel à un opérateur non disponible (non codé) serait quant à lui traité via une exception d'opérateur (cf. figure 4.2).

137. Ce raccourci syntaxique est possible avec le langage Java depuis la version 5.0 (version 1.5 dans l'ancien système de numérotation). Une telle liste est assimilée à un paramètre formel de type tableau, et à une liste de paramètres d'appel homogènes, ou bien un paramètre d'appel de type tableau (mode d'appel polymorphe) [www-Java].

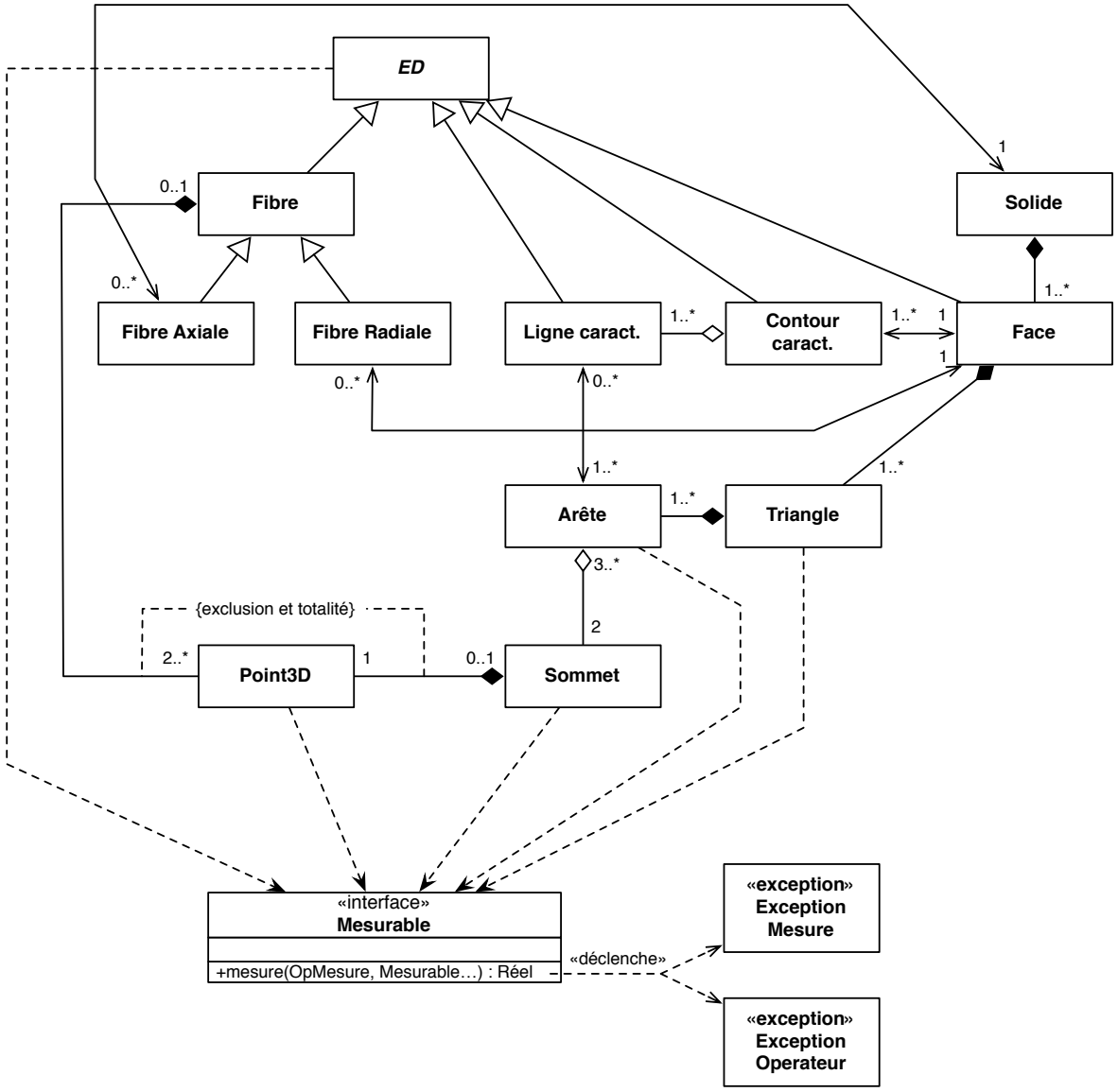


FIGURE 4.2: Modèle de représentation par les frontières (B-Rep étendu inspiré de [Danesi, 2002]). Les points de suspension (interface `Mesurable`) désignent une liste d'arguments variable (notation non standard).

Dans la suite, nous présentons l'implémentation des expressions arithmétiques et logiques, nécessaires à l'implémentation des graphes.

4.3.2.2 Expressions

La figure 4.3 est le diagramme de classes UML représentant les classes utilisées pour modéliser les expressions. Les expressions logiques sont utilisées au niveau des transitions dans les graphe des états (GE). Les expressions arithmétiques sont utilisées afin de spécifier les grandeur de fabrication (GF) en fonction des valeur de fabrication (VF), et les VF en fonction des mesures dans le graphe des grandeurs de fabrication (cf. ci-après).

Le diagramme de classes présenté sur la figure 4.3 décrit une version des expressions logiques (associées aux transitions des GE) moins restrictive que celle que nous avons adoptée au chapitre précédent (cf. § 3.4.2.2). Notamment, nous supposons qu'elles sont d'emblée en forme normale disjonctive. Les expressions logiques sont représentées par des arbres formés à partir des opérateurs logiques de conjonction, disjonction et négation appliqués à des expressions de comparaison, elles-mêmes basées sur des expressions arithmétiques. Les expressions arithmétiques sont elles aussi assimilées à des arbres dont les feuilles sont soit des constantes numériques, soit des références à des variables qui, selon le cas, peuvent être des VF ou des GF (cf. plus loin). Le type de références qu'une expression arithmétique contient est spécifié via le paramétrage de la classe abstraite `ExprArithmetique`. Implicitement, l'évaluation d'une expression s'effectue de manière récursive, jusqu'à aboutir aux feuilles de l'arbre qui la décrit. Notons que le diagramme de classes, tel qu'il est présenté sur la figure 4.3, ne garantit pas qu'une expression est un arbre (il peut s'agir d'un graphe). Cette propriété est en réalité assurée par le fait que les graphes, qui font référence aux expressions, sont stockés dans des fichiers qui sont décrits par une structure arborescente (cf. ci-après). En revanche, il est garanti que les feuilles de l'arbre peuvent être évaluées du fait qu'elles implémentent nécessairement l'interface `Evaluable` (cf. figures 4.3 et 4.4).

La figure 4.4 est un diagramme de classes UML représentant l'interface d'évaluation des règles et expressions. Le résultat d'une évaluation pouvant être booléen ou réel, l'interface `Evaluable` est paramétrée [138] par le type `T` de ce résultat. Les GF, VF, PGN, mesures, références et expressions arithmétiques sont évaluées par des réels, tandis que les expressions logiques

138. Les classes paramétrées, introduites dans la version 5.0 du langage Java, sont l'équivalent des patrons du C++ (concept de généricité).

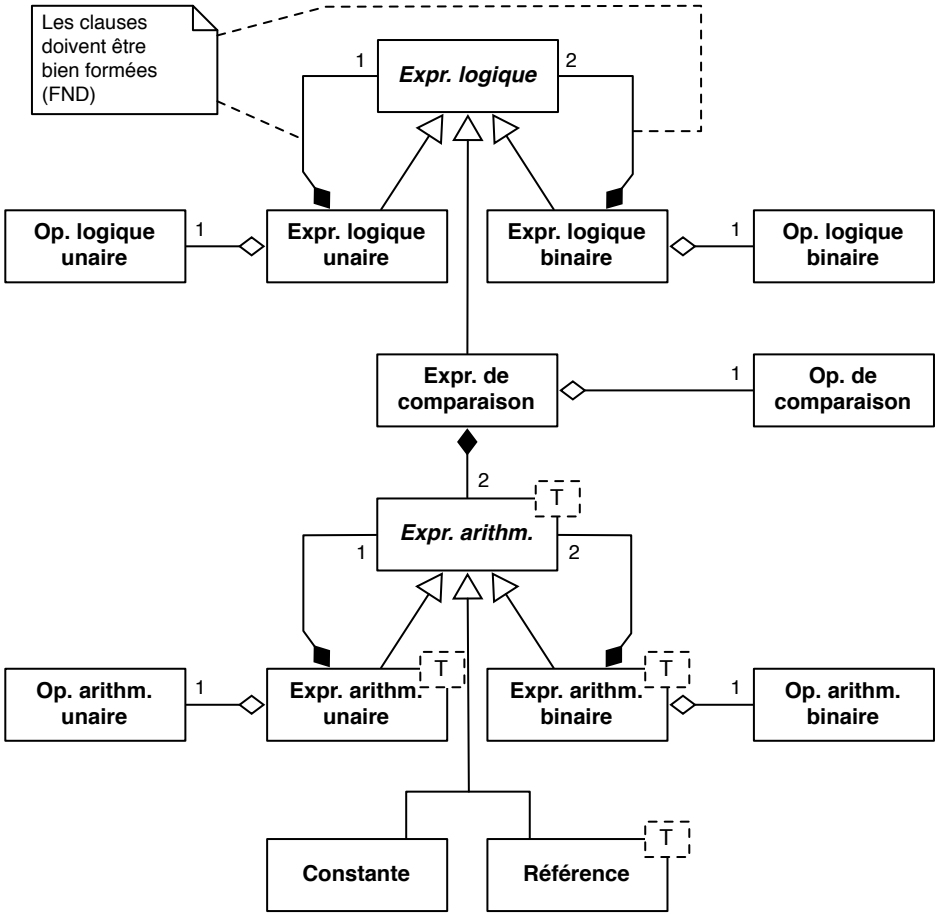


FIGURE 4.3: Diagramme de classes UML représentant les expressions logiques et arithmétiques (les noms de classes sont abrégés pour une question de clarté).

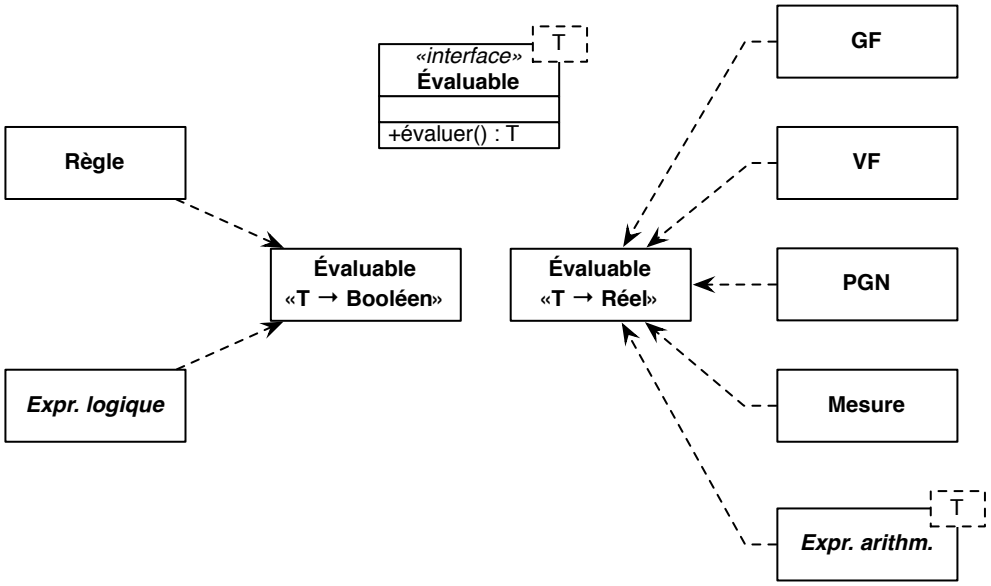


FIGURE 4.4: Diagramme de classes UML représentant l'interface `Evaluable`.

sont évaluées par des booléens. Par commodité, les règles peuvent également être évaluées par un booléen indiquant si un changement d'état est survenu suite à l'évaluation.

Les expressions, telles que nous venons de les présenter, sont utilisées au sein des graphes des grandeurs et des graphes des états, dont l'implémentation est présentée dans le paragraphe suivant.

4.3.2.3 Graphes

La figure 4.5 est un diagramme de classes UML qui représente les classes relatives aux graphes des grandeurs de fabrication (GGF). Ainsi, nous observons qu'un GGF est associé à un solide, des ED et PGN, et qu'il comprend un ensemble de règles dont on connaît l'état courant, ainsi qu'un ensemble de GF et de VF. En outre, comme indiqué au § 3.3.2, nous caractérisons deux niveaux de représentation :

un niveau abstrait dans lequel les règles de fabricabilité abstraites sont représentées. Les instances des classes formant ce niveau correspondent aux spécifications des différents types de règles qui sont susceptibles d'être appliquées à une pièce. Elles sont instanciées au chargement d'une librairie métier (ensemble de pièces initiales et leurs outils de conception associés).

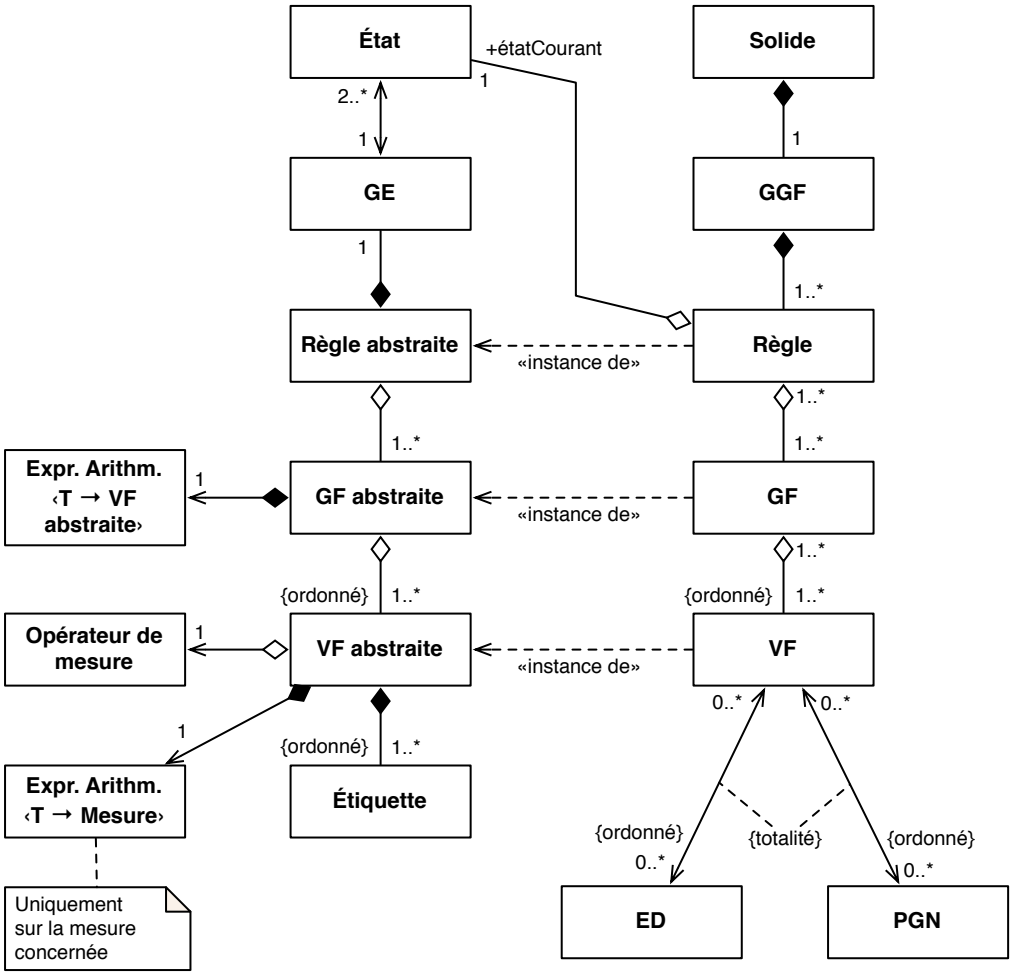


FIGURE 4.5: Diagramme de classes UML représentant les graphes des grandeurs.

un niveau instance qui correspond aux instances des règles abstraites. Celles-ci sont instanciées ou détruites lors de la création d'une nouvelle pièce (choix d'une forme initiale) ou lors de l'application d'un outil de conception.

Le lien d'instanciation qui relie ces deux niveaux, bien qu'il soit représenté sur la figure 4.5 comme l'instanciation d'une classe, n'en est pas réellement un. En effet, les règles, abstraites ou instanciées, doivent co-exister à l'exécution en tant qu'objets (niveau `M0` du MOF [139] [MOF, 2006]), car la description permettant d'évaluer les éléments instanciés est détenue par les éléments abstraits correspondants. Les classes `Regle` et `RegleAbstraite` se trouvent par conséquent au niveau `M1`. L'instanciation du niveau abstrait n'est donc pas implémentée en

139. *Meta-Object Facility*.

tant que telle au sens de la syntaxe du langage Java. Le lien existant entre une règle abstraite et ses instances (concept de réflexion) est représenté artificiellement par des associations standard, bien qu'il soit interprété de manière particulière (notamment, l'instanciation se fait en passant par les classes du niveau abstrait). Naturellement, nous n'occultons pas le fait que ceci pourrait aussi être obtenu par l'exploitation des mécanismes de réification du langage Java (utilisation de méta-classes du niveau `M2`). Mais il faudrait alors trouver un moyen de relier les classes `RegleAbstraite` et `GE`, qui se trouveraient alors à deux niveaux d'abstraction différents [140]. En outre, le code produit au final serait inutilement lourd, le périmètre de l'utilisation de ces mécanismes réflexifs étant ici connu, restreint et borné.

La mise à jour du GGF donne lieu à la réévaluation d'une ou de plusieurs VF, selon les ED et/ou les PGN qui ont été modifiés. Ceci déclenche ainsi en cascade la réévaluation d'une ou de plusieurs GF, puis d'une ou de plusieurs règles. Les VF sont valuées en deux temps. L'opérateur de mesure de la VF abstraite correspondante est dans un premier temps appliqué aux ED et/ou aux PGN référencés par la VF. Le résultat de ce calcul, appelé « mesure », est ensuite exploité par l'intermédiaire de l'expression arithmétique liée à la VF abstraite, ce qui fournit une nouvelle valeur à la VF considérée. Les étiquettes associées à une VF servent à identifier les paramètres des règles au moment de l'instanciation. Il ne s'agit que de simples chaînes de caractères. Les GF sont quant à elles simplement valuées en passant par la GF abstraite correspondante, en substituant les VF abstraites de l'expression arithmétique associée par la valeur des VF correspondantes. Similairement, les règles sont évaluées en passant par le GE de la règle abstraite correspondante, en substituant les GF abstraites référencées dans les expressions logiques des transitions par la valeur des GF correspondantes. Notons que cette façon de procéder n'est pas optimale en termes de rapidité d'exécution. Les performances pourraient être améliorées (aux dépens de l'utilisation de la mémoire), par la duplication des informations pour chaque instance de règle, de GF et de VF (économie des substitutions).

Les graphes des états se conforment au diagramme de classes UML présenté sur la figure 4.6. Les transitions entre états se font par l'intermédiaire de l'association `Transition`, étant convenu qu'une valuation positive de l'expression logique associée se traduit par une transition de l'état nommé `orig` vers l'état nommé `dest`. Les VF et GF référencées dans les expressions logiques ou les expressions grapho-numériques (EGN) d'un graphe des états donné

140. La solution la plus élégante consisterait probablement à dissocier les classes `GE` et `Etat` dans les deux niveaux d'abstraction. Ceci n'est pas indispensable car les niveaux demeureraient isomorphes (la structure des GE n'est jamais modifiée).

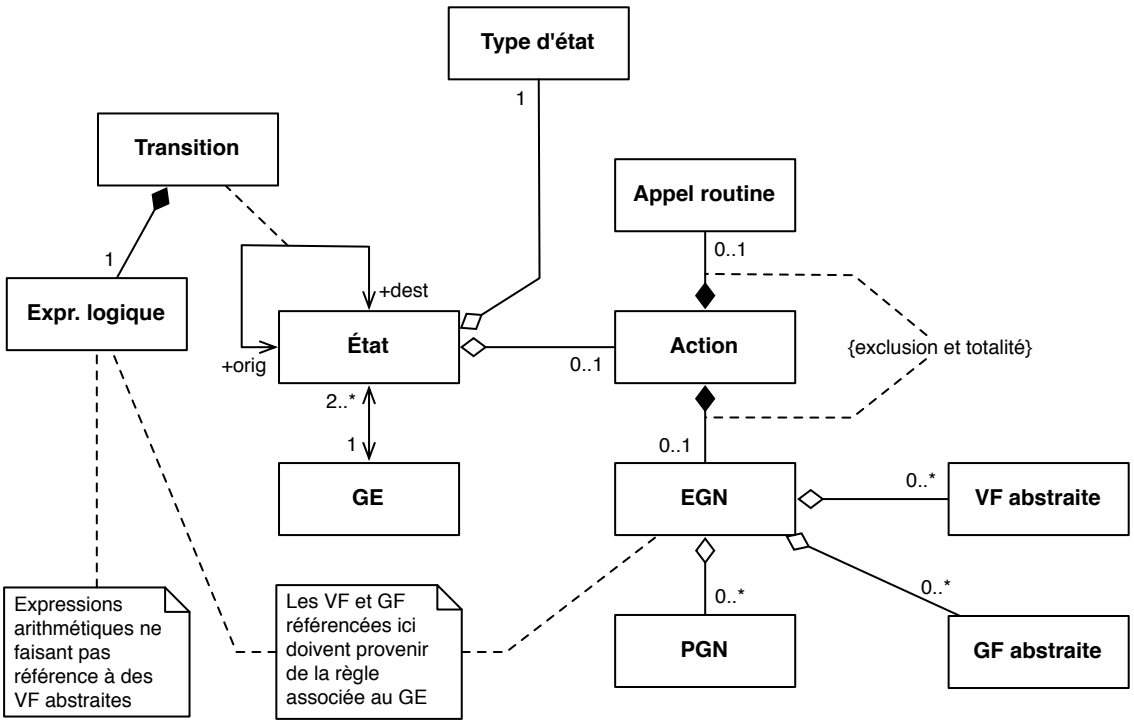


FIGURE 4.6: Diagramme de classes UML représentant les graphes des états.

sont supposées se rapporter à la règle correspondante. Cette contrainte d'intégrité est vérifiée par l'intermédiaire du format de représentation des règles (cf. § 4.3.2.4). Ce dernier est présenté dans le paragraphe suivant.

4.3.2.4 Enregistrement des règles

Similairement aux informations de conception (historique des instructions, cf. [Denis, 2003]), les règles de fabricabilité doivent pouvoir être enregistrées par le système. Nous proposons de les stocker sous la forme de fichiers au format XML [141]. Le format XML a l'avantage d'être aisément lisible par un humain (fichier texte), et d'être chargeable à l'aide des bibliothèques Java SAX et DOM. La vérification syntaxique, qui, dans le cas de l'utilisation d'un format autre (binaire ou texte), devrait entièrement être codée, peut en outre être en grande partie déportée sur les modules de validation prévus à cet effet [142] et déjà éprouvés.

Ces fichiers XML pourraient être intégrés à la base de données de l'historique afin de conser-

141. Cette solution a déjà été retenue pour représenter les instructions [Denis, 2003].
142. Il s'agit de la classe `DocumentBuilderFactory` du paquet `javax.xml.parsers`.

ver une trace des règles de fabricabilité obsolètes (cette fonctionnalité n'a pas été développée). Ils adoptent une certaine syntaxe, en rapport étroit avec les diagrammes de classe présentés ci-avant (cf. § 4.3.2.3). Le respect de cette syntaxe est garanti par conformation à des fichiers au format *XML Schema* [XSD, 2004] (p. ex. le fichier `ge.xsd` pour les GE). Un fichier de description *XML Schema* est un fichier XML particulier [143] (généralement d'extension `.xsd` ou `.xs`), qui permet de définir une structure de document XML et, par la même occasion, d'en vérifier la validité. Ce format remplit le même rôle que les DTD [144] du format XML, mais permet en plus de spécifier le domaine de validité des champs, ainsi que des contraintes d'intégrité plus fines entre les balises.

La vérification de conformité et la gestion des erreurs de syntaxe sont directement effectuées via les bibliothèques SAX et DOM. Les schémas XML définis dans le cadre de cette maquette ne garantissent que partiellement la validité des graphes, mais quelques vérifications supplémentaires sont effectuées lors du chargement (il est par exemple vérifié que les VF et GF référencées dans l'expression logique d'une transition sont bien définies dans la règle associée). Pour de plus amples détails, le lecteur pourra se référer à l'annexe **E,** où les schémas XML conçus ainsi que des exemples de fichiers de graphes XML sont présentés.

4.3.3 Répercussion des règles de fabricabilité

La répercussion des règles de fabricabilité s'effectue selon le processus décrit par le driagramme d'activité UML de la figure 4.7. Ainsi, à chaque modification d'un ED ou d'un PGN, le système réévalue chaque règle affectée grâce au mécanisme expliqué précédemment (cf. § 4.3.2.3) et, selon que l'état a changé ou non, propose à l'utilisateur d'accepter ou de rejeter l'opération en cours, par exemple via une boîte de dialogue, le mode d'interaction précis étant déterminé par l'action associée au nouvel état. Si plusieurs transitions d'état ont lieu simultanément, le système les considère une à une, en opérant les interactions nécessaires.

Lors de la conception, les retours en arrière sont fréquents. Afin de ne pas poser de nombreuses fois les mêmes questions à l'utilisateur, les changements d'état sont mémorisés (aspect discuté au § 3.4.1). Cette mémorisation est opérée au niveau de la classe `Regle` (cf. figure 4.5) afin qu'elle soit propre à chaque instance de règle [145]. Les états déjà validés par l'utilisateur

143. Notamment, la structure des fichiers *XML Schema* est auto-représentée.
144. *Document Type Definition*.
145. Notons qu'il serait aisé de définir un deuxième niveau de mémorisation au niveau de la classe `RegleAbstraite`, et p. ex. de l'associer à une case à cocher intitulée « Ne plus poser la question pour ce type de règle ».

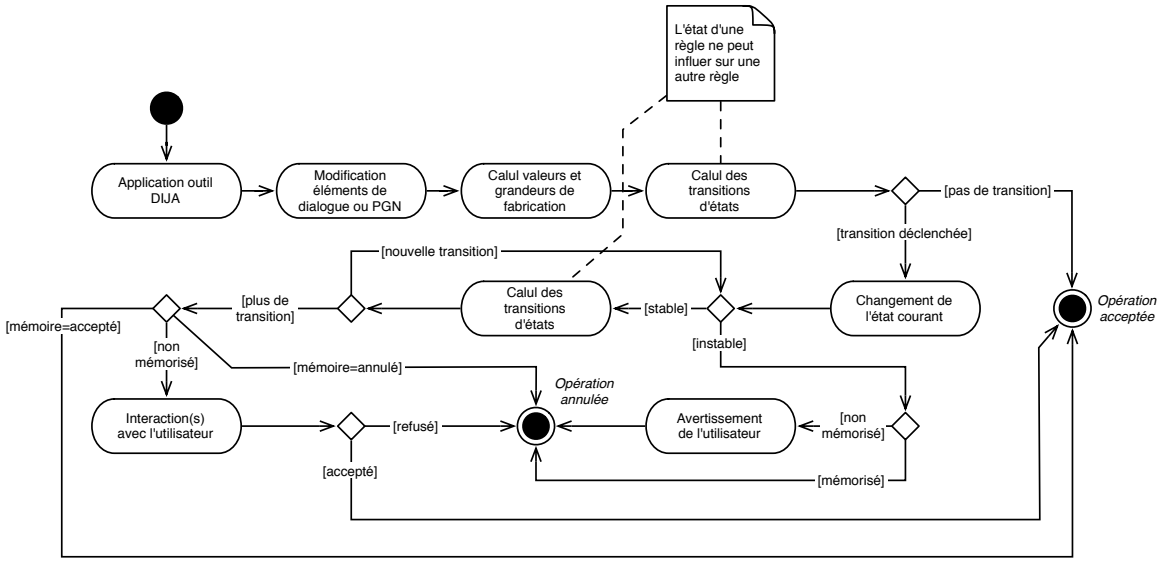


FIGURE 4.7: Diagramme d'activité UML décrivant le processus de changement d'état.

y sont référencés dans un objet de la classe paramétrée `TreeSet<Etat>` [146] (paquet `java.util`). Les informations de mémorisation sont utilisées pour éviter l'affichage intempestif de boîtes de dialogue, mais également pour proposer la réinitialisation des règles déjà mémorisées (et uniquement de celles-ci) à travers des contrôles qui apparaîtront par exemple dans un panneau dédié en temps voulu.

4.3.4 Mécanismes mis en œuvre pour l'évolution des connaissances

Les algorithmes relatifs à la propagation des règles (propagation, division, fusion...) sont enregistrés dans la classe `GGF` du paquet `dija.fabricabilite.grandeurs`. Un certain nombre de méthodes utilitaires pour manipuler les graphes y sont également présentes avec une visibilité privée (p. ex. : accès aux VF pointées par un ED, accès aux VF dont dépend une GF, dédoublement d'une VF, etc.).

La classe `Outil` du paquet `dija.fabricabilite.operations` permet de répercuter les transformations du modèle géométrique communiquées par le superviseur sur le modèle de fabricabilité, et d'assurer la liaison avec l'IHM, notamment pour signaler la fabricabilité (cf. figure 4.8). Les fonctions de correspondance f_c et de référencement f_r sont enregistrées dans cette classe. Ceci permet d'isoler les instructions gérées par le superviseur et la manipulation des graphes

146. Représentation d'un ensemble d'états sous forme d'arbre binaire de recherche.

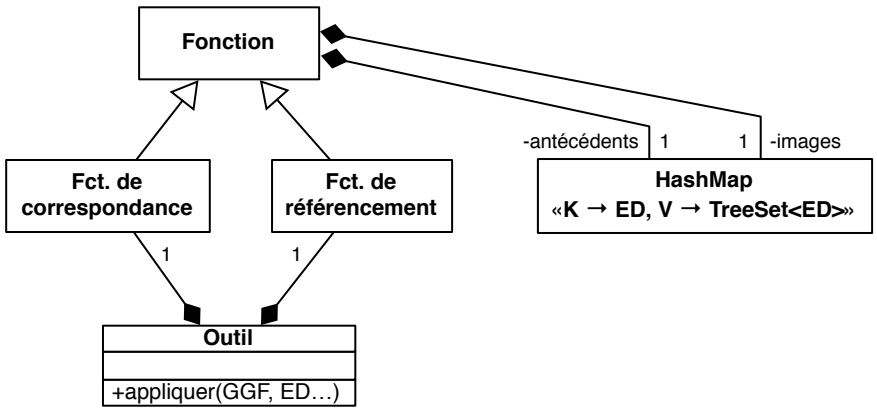


FIGURE 4.8: Diagramme de classes UML représentant les fonctions f_c et f_r.

et des fonctions f_c et f_r. Cette séparation vaut également pour les choix d'implémentation qui pourraient être conduits à évoluer distinctement dans l'avenir.

Leurs modèles de représentation étant similaires, les classes `FctCorrespondance` et `FctReferencement` sont dérivées d'une même classe, nommée `Fonction`. Cette dernière est basée sur la classe paramétrée `HashMap<ED,TreeSet<ED>>` (paquet `java.util`) pour enregistrer les relations entre éléments de dialogue. La classe `FctCorrespondance` offre en particulier une interface basée sur les huit opérateurs définis au § 3.4.3.2. La classe `FctReferencement` dispose quant à elle d'une interface extensible permettant d'effectuer des traitements géométriques visant à compléter les informations fournies par f_c en fonction du contexte d'exécution. Avec un seul objet `HashMap`, l'implémentation favorise la lecture des images, au détriment de celle des antécédents (ou bien l'inverse). Si le nombre d'éléments de dialogue venait à croître, la recherche des antécédents d'un ED pourrait devenir lourde. Nous proposons donc de maintenir en parallèle deux représentations correspondant respectivement à une fonction et sa fonction inverse. Ceci alourdit légèrement la construction d'un objet de la classe `Fonction`, mais les lectures étant plus nombreuses que les écritures pour cette structure [147], les gains compensent la perte.

Les informations nécessaires au paramétrage de l'algorithme de propagation sont enregistrées dans la classe `VFAbstraite` (cf. § 3.4.3.3.1). Ces informations, au départ chargées avec la pièce initiale et les outils associés, évoluent au gré de l'application de chacun des outils, et

147. Ceci est dû au fait que les fonctions d'accès aux images et antécédents sont beaucoup sollicitées par l'algorithme de propagation, lequel est appelé à chaque construction d'un exemplaire de f_c et de f_r.

doivent donc être considérées à chaque fois qu'un outil est appliqué. C'est pourquoi l'instance de la classe `GGF` qui correspond à la pièce en cours de conception est passé en paramètre à chaque appel de la méthode `appliquer()` (classe `Outil`, cf. figure 4.8).

Si, suite à l'application d'un outil qui conduirait à des modifications structurelles du modèle de conception, l'opération en cours doit être annulée, un mécanisme de retour en arrière intervient. Ce mécanisme fait appel au module d'historique pour reconstruire la pièce telle qu'elle était avant l'opération initiée. Les instructions de chargement des graphes, ainsi que les paramètres qui les accompagnent (p. ex. les choix de l'utilisateur), sont donc des instructions « historisables », de même que celles permettant de construire la géométrie. Le système de graphe d'historique déjà développé intègre sans modification notable cette évolution.

4.4 Présentation de la maquette

Dans cette section, nous présentons tout d'abord l'interface de l'application qui a été développée (§ 4.4.1). Dans un second temps, l'utilisation de la maquette réalisée est présentée à travers un scénario de conception minimal (§ 4.4.2).

4.4.1 Interface

L'interface de l'application est présentée sur la figure 4.9. La partie gauche de l'écran est une vue tridimensionnelle de la pièce en cours de conception. Les éléments de dialogue définis pour la pièce y apparaissent distinctement : faces en jaune translucide, fibres en rouge, lignes caractéristiques en mauve [148] et contours caractéristiques en vert. La pièce est manipulée dans cette zone à l'aide de la souris. La partie droite de la fenêtre comprend :

- une rubrique *affichage* qui offre quelques options de présentation ;
- une rubrique *forme initiale* pour choisir une forme de départ ;
- une rubrique *outils* qui liste les outils disponibles pour modifier la pièce en cours de conception ;
- une rubrique *fabricabilité* dans laquelle sont affichés les indicateurs de fabricabilité en cours de validité (états courants des règles, hormis ceux qui correspondent à une fabricabilité normale) ;

148. Sur la figure 4.9, les lignes caractéristiques n'apparaissent pas car elles sont occultées par les contours (dans cet exemple, lignes et contours caractéristiques sont confondus).

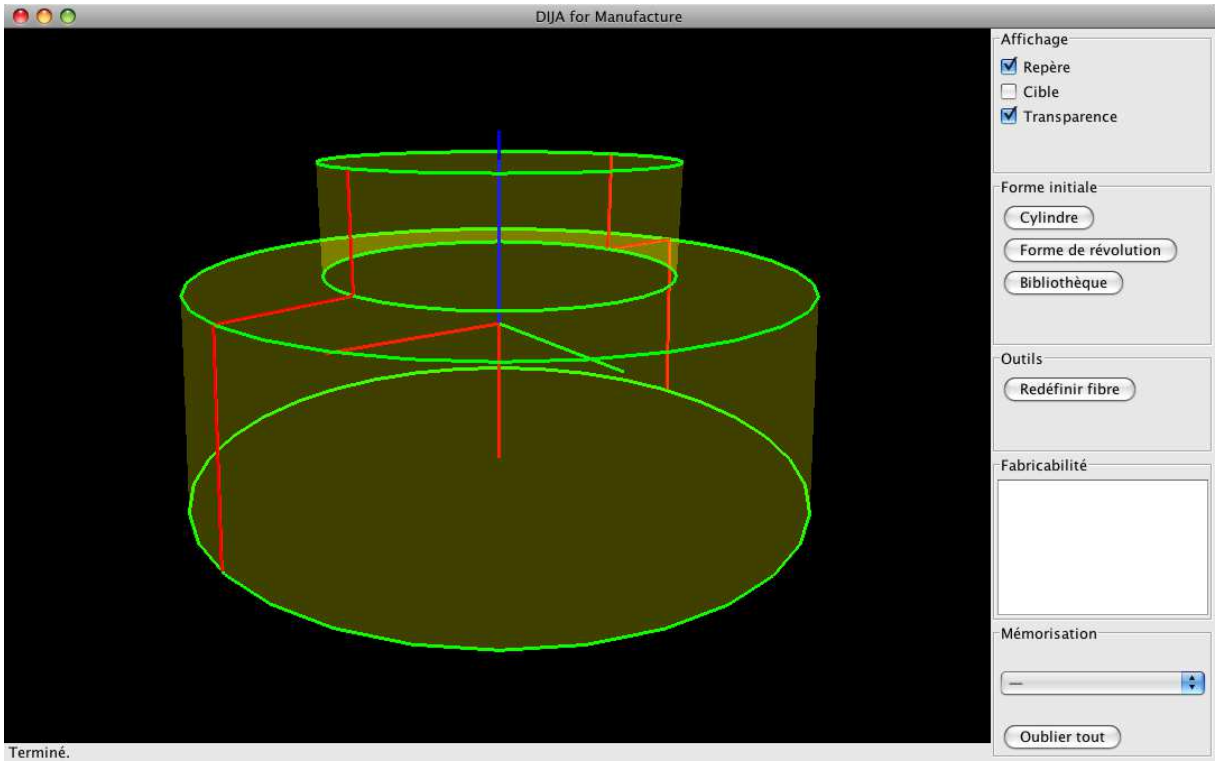


FIGURE 4.9: Interface graphique de l'application.

– une rubrique *mémorisation* qui rescence les états mémorisés (ceux dont l'utilisateur a déjà été informé une fois).

L'application dispose en outre d'une barre de menu, non visible sur la figure 4.9 car affichée en haut de l'écran (capture réalisée sous le système Mac OS), proposant les fonctionnalités d'entrée/sortie courantes (ouverture, enregistrement...).

Dans le paragraphe suivant, nous présentons l'application développée à travers une petite étude de cas.

4.4.2 Étude de cas

Dans cette section, nous présentons l'application développée en nous appuyant sur un scénario de conception typique (au sens des interactions relevant de la fabricabilité). Nous choisissons volontairement une pièce simple, afin que les graphes, exposés conjointement aux captures d'écran, ne soient pas surchargés.

Dans le § 4.4.2.1, nous présentons la pièce ainsi que le procédé de fabrication pris pour exemple dans cette étude de cas. Nous découpons l'étude selon deux axes : la capitalisation

des connaissances qui consiste à entrer les règles de fabricabilité dans le système (§ 4.4.2.2) ; et la réutilisation des connaissances qui correspond à la conception d'une pièce guidée par les règles de fabricabilité entrées (§ 4.4.2.3).

4.4.2.1 Exemple illustratif

L'exemple que nous présentons dans la suite consiste en la conception d'une pièce de révolution, devant être réalisée par un procédé de fabrication par frittage de poudre métallique. Les procédés de fabrication par frittage de poudre métallique sont divers [Eudier, 1994]. Celui que nous prenons en exemple est le frittage au lazer d'une poudre métallique enrobée de polymère [149]. Une étude a montré que ce type de procédé requiert du produit certaines propriétés de formes [Agnez, 2002]. Pour les besoins de l'étude nous considérons les critères de fabricabilité suivants (cf. figure 4.10) :

1. La section correspondant à la base de la pièce doit être plus large que les autres sections paralèlles ;
2. Le diamètre de la pièce ne peut être inférieur à un seuil s_1 ou supérieur à un seuil s_2 ;
3. Si le diamètre de la pièce est inférieur à un seuil s_3 ($s_1 < s_3 < s_2$), la pièce est fabriquée moins rapidement, ce qui provoque un surcoût de fabrication ;
4. Le rapport du diamètre de la base à la hauteur totale de la pièce $\frac{h}{d}$ doit être inférieur à un seuil s_4 ;
5. Si le rapport $\frac{h}{d}$ est inférieur à un seuil s_5, alors la pièce est fabricable, mais il sera plus difficile à l'opérateur de la nettoyer. Ceci induit un surcoût de fabrication lié à la main d'œuvre.

Comme nous le verrons au § 4.4.2.3, ces critères peuvent être traduits sous forme d'un GGF, en n'employant que trois règles, les critères 2 et 3 d'une part et 4 et 5 d'autre part, étant chacun regroupés sous un même aspect.

4.4.2.2 Phase de capitalisation

Dans le contexte de notre approche, le savoir-faire de fabricabilité se répartit en :

149. Ce procédé de fabrication par ajout de matière comprend cinq phases [Agnez, 2002] : polymérisation au lazer ; nettoyage de la pièce obtenue, dénommée « pièce verte » ; déliantage (brûlage des polymères résiduels) ; infiltration au bronze par capillarité ; et enfin polissage.

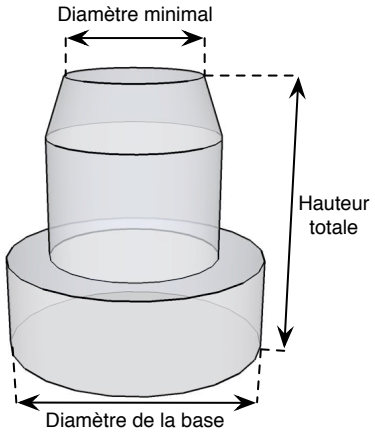


FIGURE 4.10: Exemple de pièce de révolution.

– des règles associées aux pièces initiales ;
– des règles associées aux outils.

Pour construire un nouvel environnement de CAO, il faut donc procéder à une capitalisation des règles selon ces deux axes.

Par manque de temps, il n'a pas été développé d'interface dédiée à la spécification des règles de fabricabilité. Celles-ci sont introduites dans le système par de simples fichiers au format XML (cf. § 4.3.2.4). Que les règles soient rattachées à une pièce initiale ou à un outil, le mécanisme est assez similaire. Le lien avec les ED est effectué par l'intermédiaire des étiquettes évoquées au § 4.3.2.3. Une étiquette représente le nom qui est donné à un ED significatif du point de vue de la fabricabilité. Dans le cas d'une pièce initiale, le rattachement se fait directement par le fabricant. Les étiquettes peuvent alors correspondre aux identifiants des instructions de construction des éléments de dialogue (cf. § 3.2.1.2). Pour les règles associées à un outil, les étiquettes doivent être rattachées à des ED connus par l'outil, directement (paramètres, nouveaux ED…) ou indirectement (parcours basé sur les paramètres).

Ainsi, le rattachement des ED aux étiquettes se fait, dans la maquette développée, de manière statique, même si les règles sont chargées à l'exécution depuis des fichiers XML.

Dans la suite, nous considérons que les graphes nécessaires ont été entrés dans le système, et qu'ils sont consistants. Nous présentons dans ce qui suit un exemple d'utilisation de tels graphes dans le cadre de la conception d'une pièce réalisée par frittage de poudre métallique.

4.4.2.3 Phase de réutilisation

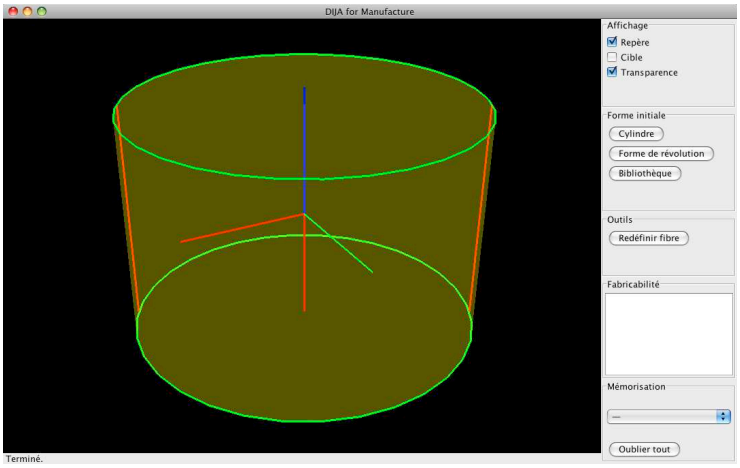
La figure 4.11 présente un scénario de conception dans lequel l'utilisateur crée une pièce de révolution destinée à être réalisée par frittage de poudre.

La conception se déroule comme suit. L'utilisateur sélectionne dans un premier temps un cylindre comme forme de départ (cf. figure 4.11a). Il décide ensuite de diminuer une première fois le diamètre de la pièce, ce qui consiste à modifier sa fibre de révolution (cf. figures 4.11b et 4.11c). Mais après qu'il ait réduit à nouveau le diamètre de la pièce (cf. figure 4.11d), le système l'avertit que la modification opérée conduit à un surcoût de fabrication du fait du franchissement du seuil s_3, conformément à ce qui est indiqué au § 4.4.2.1 (cf. figure 4.11e). Il décide alors d'annuler la modification, ce qui replace la pièce dans son état précédent (cf. figure 4.11f). À ce stade, l'utilisateur décide de créer un chanfrein sur la face supérieure du cylindre (cf. figures 4.11g et 4.11h). Ceci correspond à une division de la fibre de révolution (opérateur `d` dans la fonction de correspondance). La modification est validée par le système. Mais alors que l'utilisateur tente d'ajuster les dimensions du chanfrein (cf. figures 4.11i et 4.11j), le système l'avertit qu'un nouveau surcoût est provoqué par cette opération. Ce surcoût est lié à la même règle (dénommée `RegleD`, cf. ci-après) que pour le premier surcoût rencontré, celle-ci ayant été propagée (« divisée ») au cours de la division de la fibre de révolution. Cette fois, l'utilisateur accepte la condition de fabricabilité, ce qui a pour effet de faire apparaître l'avertissement dans la rubrique *fabricabilité* du panneau latéral de l'application (cf. figures 4.11k et 4.11l).

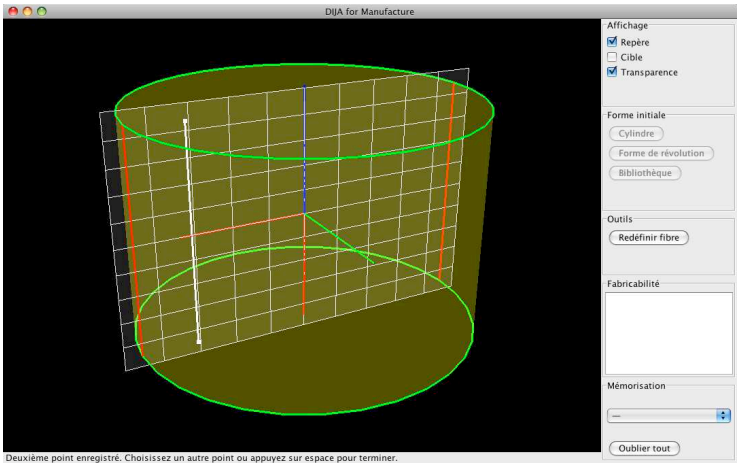
Afin de faciliter la lecture des informations, les règles et états affichés dans cette rubrique sont accompagnés d'une icône qui est différente selon le type de critère de fabricabilité (surcoût de main d'œuvre, difficulté d'assemblage, surcoût lié à l'outillage…). Cependant, notons qu'avec des pièces complexes, le nombre d'avertissements pourrait croître jusqu'à rendre la lecture de cette rubrique difficile. En outre, il pourrait arriver que plusieurs instances d'une même règle donnent lieu à des messages identiques. Étant donné que chaque message est relié à un ensemble d'ED connu par le système, une amélioration de l'application pourrait consister à mettre en emphase les ED qui contribuent au critère sélectionné dans la liste [150]. Ces éléments n'étant pas fondamentaux, ils n'ont pas été développés.

Pour finir, la mémorisation des interactions est gérée dans la rubrique *mémorisation*, située en bas du panneau latéral. Dès que l'utilisateur valide la transition d'une règle vers un nouvel

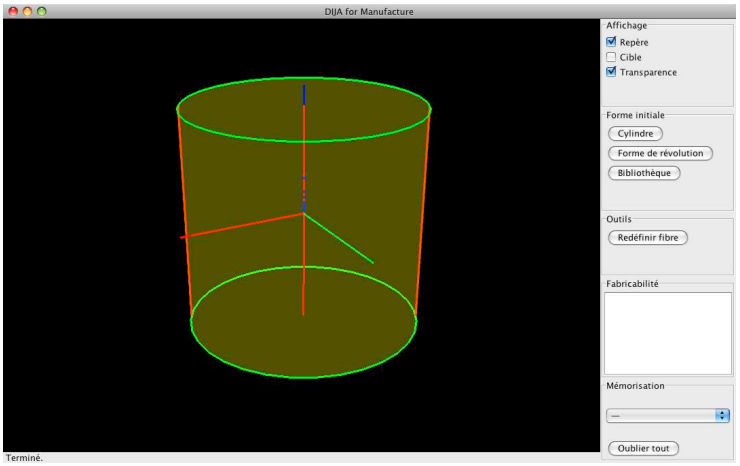
150. L'opération inverse serait d'ailleurs tout aussi possible (mise en emphase des avertissements relatifs à l'ED sélectionné).



(a)

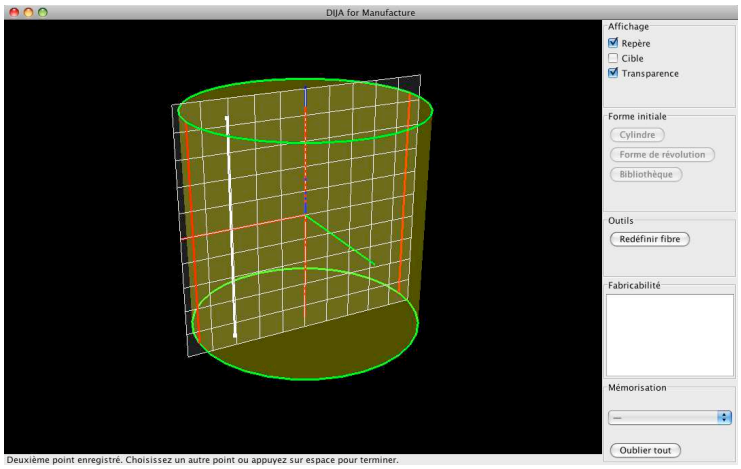


(b)

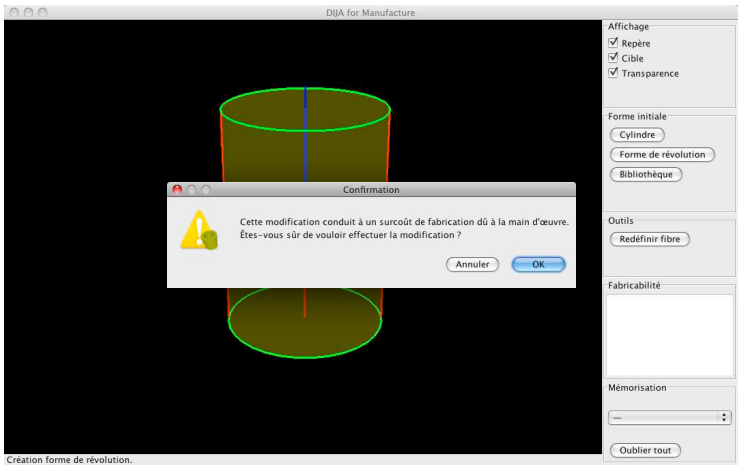


(c)

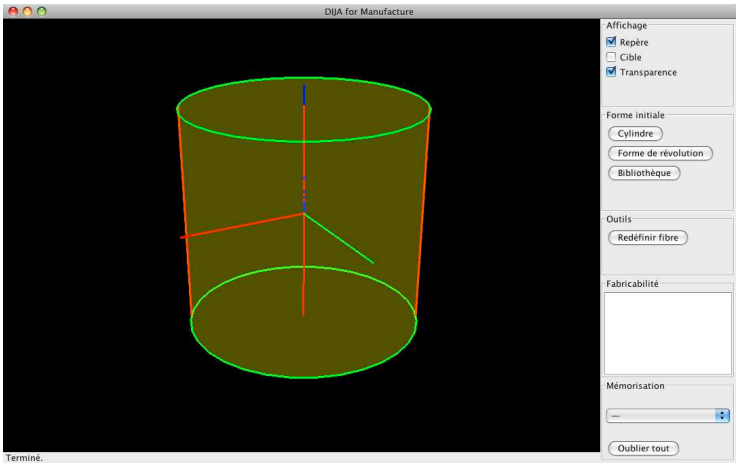
FIGURE 4.11: Exemple de scénario de conception.



(d)

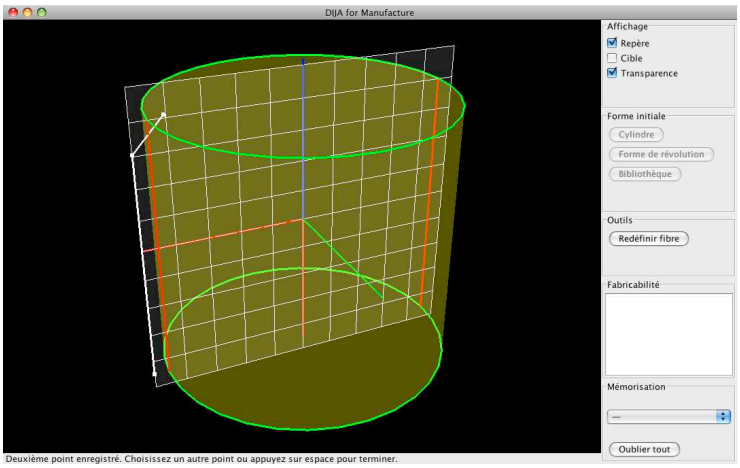


(e)

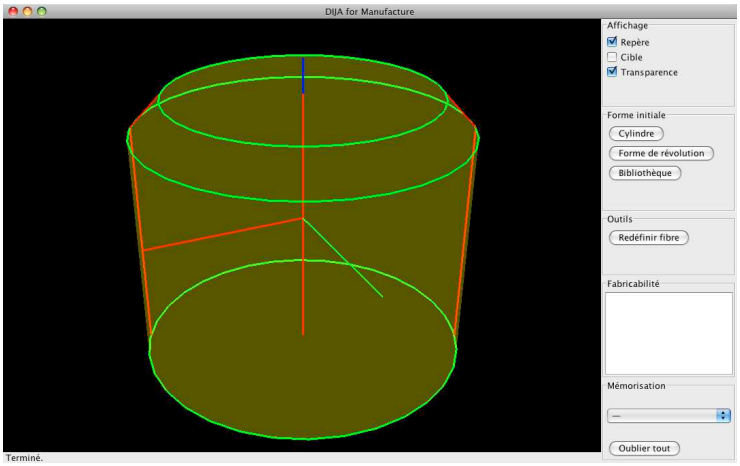


(f)

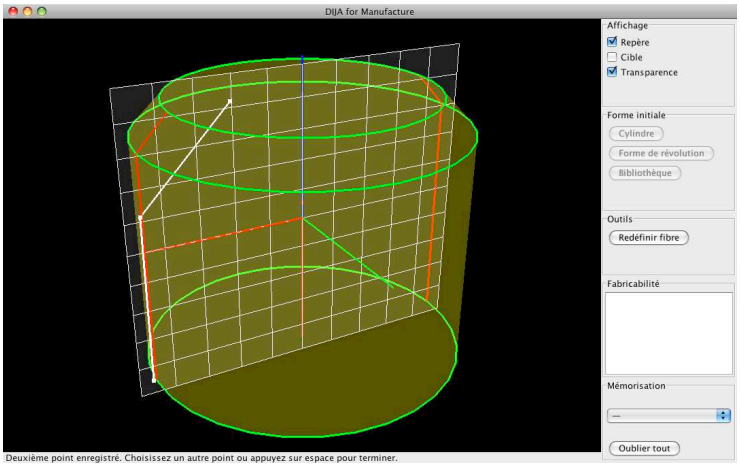
FIGURE 4.11: Exemple de scénario de conception (suite).



(g)

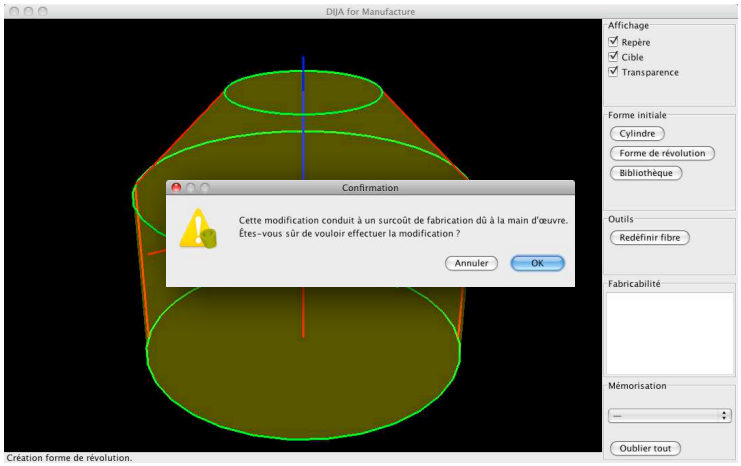


(h)

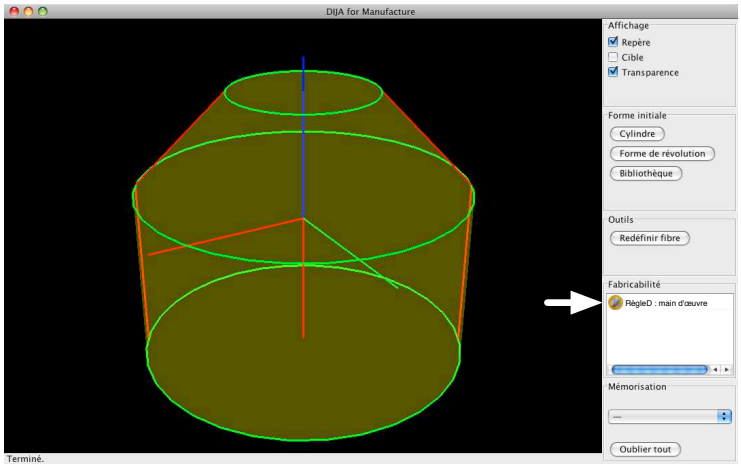


(i)

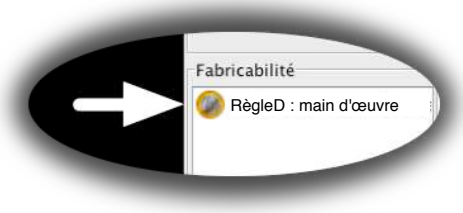
FIGURE 4.11: Exemple de scénario de conception (suite).



(j)



(k)



(l)

FIGURE 4.11: Exemple de scénario de conception (suite).

état, celui-ci est enregistré et apparaît dans la liste déroulante. Il suffit alors de sélectionner l'entrée voulue pour que le système « oublie » cette information et procède de nouveau aux interactions correspondantes au moment opportun, comme si elles n'avaient jamais eu lieu.

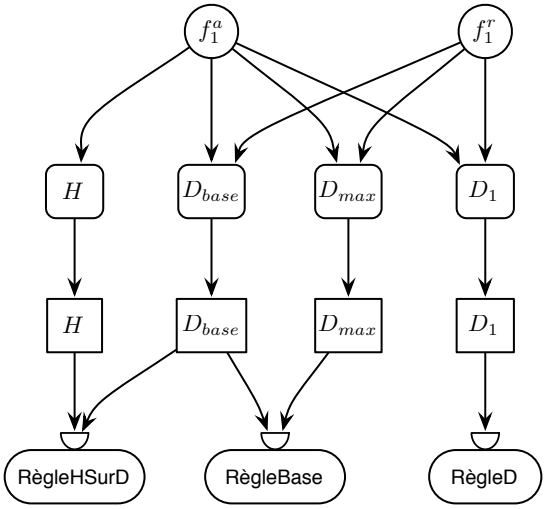
La figure 4.12 présente les deux GGF relatifs au scénario de la figure 4.11 (figure 4.12a avant division de la fibre de révolution, et figure 4.12b après division). Sur ces deux graphes, f_1^a est la fibre axiale de la pièce. Celle-ci n'est pas affectée par la division. f_1^r et f_2^r sont les deux fragments de la fibre de révolution. Les éléments de dialogue non utilisés ne sont pas représentés.

La règle dénommée `RègleHSurD` permet de prendre en compte les critères 4 et 5 indiqués au § 4.4.2.1. Elle dépend de la hauteur de la fibre et du diamètre de la base. Ces éléments ne sont pas modifiés par la division ; la structure de la règle ne change donc pas. En revanche, la division de la fibre f_1^r se propage aux règles dénommées `RègleBase` et `RegleD` qui correspondent respectivement au critère 1 et aux critères 2 et 3 (cf. § 4.4.2.1). La VF marquée D_{max} calcule la distance maximale entre la fibre axiale f_1^a et les fibres de révolution qui y sont connectées. Le type de propagation choisi pour l'arc reliant f_1^r à D_{max} est `conservation` (cf. § 3.4.3.3.1, page 134), ce qui a pour effet de ne pas dupliquer la règle (ajout d'un arc entre f_2^r et D_{max}). Au contraire, l'arc reliant f_1^r à D_1 a pour type de propagation `adaptation`, ce qui, comme indiqué plus haut, induit une duplication de la règle permettant de vérifier le diamètre pour chaque tronçon de la pièce.

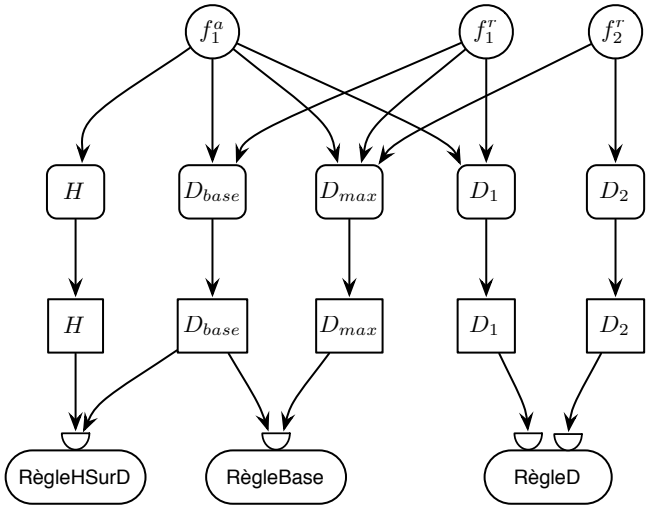
4.5 Bilan

Dans ce chapitre, nous avons exposé une implémentation de l'approche qui a été présentée au fil de ce manuscrit. Nous avons tout d'abord défini le contexte général des développements en abordant les travaux antérieurs et les choix généraux, puis en délimitant le périmètre de la maquette. Nous avons ensuite présenté l'architecture logicielle choisie, puis avons décrit les concepts centraux de l'implémentation, en particulier la représentation des connaissances sur deux niveaux, et les structures de données et mécanismes permettant l'évaluation, la répercussion et l'évolution des règles de fabricabilité.

La seconde partie de ce chapitre a permis d'observer la maquette en action, à travers le scénario de conception d'une pièce de révolution guidée par des règles de fabricabilité relatives au frittage de poudre métallique. Les captures d'écran et les graphes correspondants ont permis d'appréhender l'approche sur un cas réel, bien que relativement simple. Cependant, cet



(a) Avant division



(b) Après division

FIGURE 4.12: Graphes des grandeurs avant et après division de la fibre de révolution (cf. figure 4.11).

exemple a permis de constater que l'exploitation de l'approche nécessite une phase de spécification qui n'est pas évidente à mener. Notamment, le paramétrage de la propagation (arcs reliant les ED et les VF) a une influence déterminante sur l'évolution des règles, ce qui le rend délicat. Une erreur de paramétrage, même minime, peut en effet mener à l'une des deux situations suivantes :

1. Une règle ne s'est pas propagée alors qu'elle aurait dû l'être. Dans ce cas, l'utilisateur peut concevoir une pièce ayant une mauvaise fabricabilité ou invalide sans en être informé. Ce type de problème peut cependant être détecté avant la fabrication ;
2. Une règle s'est propagée alors qu'elle n'aurait pas dû l'être. Cette situation est en revanche plus difficile à identifier, car le concepteur dispose alors d'informations non fondées qui ne mènent pas nécessairement à des incohérences visibles avant la fabrication [151].

À l'issue de ce chapitre, nous pouvons considérer que l'implémentation réalisée révèle la viabilité générale de l'approche, à savoir : un mécanisme de vérification de règles de fabricabilité agissant en temps réel, directement à partir du logiciel de CAO ; une mise à jour incrémentale de ces règles ; une séparation stricte entre les connaissances des domaines de la conception et de la fabrication (respect de la confidentialité) ; et la possibilité de modifier automatiquement la pièce afin de la rendre conforme à des critères de fabricabilité ou d'établir des interactions directes entre concepteur et fabricant si la situation le requiert. Ainsi, la maquette valide tous ces concepts fondamentaux présentés au chapitre 3.

151. Par exemple une règle infondée peut laisser penser au concepteur qu'une pièce valide n'est pas fabricable. Dans ce cas, seul le concepteur est à même de détecter le problème, et de le vérifier auprès du fabricant.

Conclusion

Nous concluons ce manuscrit par un bilan général qui rappelle les points abordés, les propositions, leurs apports et leurs limites, puis nous abordons des directions de recherche possibles dans le cadre de la conception guidée par la fabricabilité.

Le chapitre 1 a présenté la conception à travers plusieurs aspects. Nous avons parcouru un certain nombre de travaux permettant de passer du besoin au fonctions, puis des fonctions à la forme du produit fini, ainsi qu'à sa validation. Nous avons observé l'enrichissement progressif des modèles de représentation du produit, permettant de mieux prendre en compte le savoir-faire, en notant les difficultés inhérentes au maintient de la cohérence entre métiers, notamment de conception et de fabrication. Enfin, nous avons passé en revue les tendances d'évolution actuelles à travers l'intégration des phases du cycle de vie du produit, l'ingénierie simultanée et le travail collaboratif. Il apparaît que les outils de CAO doivent évoluer pour prendre en compte ces nouvelles préoccupations et ouvrir ainsi la voie à l'optimisation du cycle de vie.

Le chapitre 2 a concerné la conception guidée par la fabricabilité. Nous avons décrit le concept de fabricabilité et différents indicateurs de fabricabilité et référenciels, et nous avons observé qu'il était important d'intégrer la notion de règle de fabricabilité pour prendre en compte le contexte d'application des indicateurs. L'analyse de fabricabilité, qu'elle soit qualitative ou quantitative, suscite des échanges entre les acteurs des domaines de la conception et de la fabrication. Une automatisation totale de ces flux n'étant pas réaliste, il convient de proposer des indicateurs permettant de décider et de collaborer lorsque des compromis doivent être trouvés, ce qui peut s'opérer de manière interactive. Nous avons parcouru les approches existantes et noté leurs limites, notamment vis-à-vis du caractère automatique de la spécification des règles de fabrication, du manque de généricité ou à cause d'ambiguïtés (interprétations multiples, conciliation de contraintes). Néanmoins, nous avons noté que l'utilisation d'un modèle canonique permettait en principe d'intégrer plusieurs métiers, et que l'utilisation d'ap-

proches incrémentales ouvrait la voie à des interactions synchrones, les approches synchrones et asynchrones étant complémentaires.

Dans le chapitre 3 nous avons proposé une approche pour la représentation et la prise en compte de règles de fabricabilité en nous basant sur l'approche de modélisation synthétique du projet DIJA. Les informations relatives aux processus définissant le déroulement de la conception sont enregistrées à différents niveaux d'abstraction, ce qui permet de structurer la conception tout en laissant la possibilité d'y incorporer des phases de conception libre (ordre d'application des outils non déterminé à l'avance). Ainsi, le maintien de la cohérence des modèles de conception et de fabricabilité est assuré à l'aide des éléments de dialogue et des paramètres grapho-numériques qui constituent un modèle canonique de représentation de la géométrie et des autres paramètres déterminant la fabricabilité. Nous pouvons au passage noter que le modèle de représentation des connaissances de fabrication adopté permet des estimations basées sur des heuristiques (cf. § 2.3.4.1), des méthodes paramétriques, et, dans une certaine mesure, également des approches par variation (cf. § 2.3.4.2) de par la nature de l'approche synthétique. Nous avons proposé de représenter la notion de fabricabilité à travers deux aspects : le calcul d'indicateurs réalisé à l'aide de graphes associés à des expressions arithmétiques et logiques ; et le concept d'état pour représenter l'appréciation de la fabricabilité avec, le plus souvent, un traitement associé (avertissement, confirmation, scénario collaboratif…). Cette approche permet de guider et de sensibiliser le concepteur de manière synchrone dès les premières phases de la conception. Les règles de fabricabilité sont en outre propagées de façon incrémentale, à chaque fois que la géométrie est modifiée. Cette approche a l'avantage d'être générique, et permet d'intervenir directement par l'intermédiaire de la CAO, ce qui évite de trop longs rebouclages. Par opposition aux approches multi-vues pour lesquelles les règles de fabricabilité doivent être entrées directement par le fabricant, les règles sont ici instanciées directement par le système. En contrepartie, elles doivent être stipulées au préalable par un ingénieur en gestion des connaissances. Néanmoins, nous avons noté certaines limites au caractère automatique de cette instanciation. En premier lieu, il est naturellement impossible de tout prévoir. Le système peut ainsi se trouver dans une situation pour laquelle les fonctions de correspondance et de référencement ne sont pas adaptées ou sont incomplètes. Ceci peut alors aboutir à une accumulation inutile de règles, ou à une perte de règle. En outre, la spécification des règles n'est pas nécessairement évidente, notamment vis-à-vis du paramétrage qui détermine les possibilités d'évolution des règles.

Enfin, le chapitre 4 a révélé la viabilité de l'approche par la présentation d'une maquette développée en langage Java. L'architecture, les principaux composants et les mécanismes d'évaluation et de propagation ont été présentés à travers des diagrammes UML. Les règles abstraites et instanciées ont été représentées au même niveau, sous forme d'objets Java, grâce à un mode d'instanciation particulier. Elles sont enregistrées par l'intermédiaire de fichiers au format XML. La maquette a enfin été présentée à travers un scénario de conception typique, qui a mis en relief l'importance de la phase de spécification des règles de fabricabilité, et en particulier du paramétrage de la propagation. Toutefois, les concepts fondamentaux présentés au chapitre 3, ont été validés par l'implémentation.

Les perspectives d'évolution ne manquent pas. Par exemple, la spécification des règles de fabricabilité pourrait être améliorée sensiblement par l'étude d'une interface adaptée. Celle-ci pourrait comprendre les trois constituants suivants :

1. Une interface pour la modélisation des graphes (GGF [152] et GE [153]), dans laquelle l'utilisateur pourrait notamment visualiser et construire les graphes (ajout/suppression de nœuds et d'arêtes, paramétrage…). Certains archétypes de règles pourraient être enregistrés dans une librairie. Une qualité de cette interface serait qu'elle garantisse les propriétés de bonne formation des graphes (absence de cycles, comportement déterministe, cf. § 3.4.2) ;
2. Une interface pour associer des graphes à une pièce initiale. L'utilisateur pourrait ici choisir une étiquette du GGF, et l'associer à un élément de dialogue de la pièce, par exemple par pointage à l'aide de la souris ;
3. Une interface pour associer des graphes à un outil. Cette interface reprendrait le même principe que la précédente, mais comprendrait en plus des zones dans lesquelles des instructions de parcours de la géométrie pourraient être tapées au clavier. Par conséquent, un langage devrait être défini à cet effet.

Ce langage de parcours de la géométrie offrirait notamment des primitives de navigation s'appliquant au niveau dialogue, et basées sur une combinaison quelconque de critères tels que les trois suivants :

- le type des éléments de dialogue ;
- l'outil qui les a engendrés (information disponible dans l'historique) ;

152. Graphe des Grandeurs de Fabrication.
153. Graphe des États.

– la notion de voisinage géométrique.

Ceci permettrait de faciliter la définition des règles, à condition d'y associer un retour visuel adéquat. En outre, les opérateurs de mesure géométriques pourraient être étendus à des calculs différents (aire, volume, distance selon une direction…), appliqués à des entités plus complexes (courbes et surfaces gauches), ou encore à des calculs basés sur des opérations d'estimation globales (évaluation de la fabricabilité par des méthodes paramétriques, cf. § 2.3.4.2). En outre, dans le cas des assemblages, qui n'ont pas été considérés dans l'approche, il serait nécessaire de prendre en compte la nature des liaisons mécaniques entre pièces. En particulier, ceci nécessiterait de pouvoir stipuler des règles impliquées dans plusieurs GGF.

D'autres qualités d'un outil de spécification de règles de fabricabilité seraient qu'il permette d'éviter l'apparition de règles redondantes, ou de savoir si un graphe des grandeurs de fabrication évolue de manière déterministe après retour à une situation déjà rencontrée [154].

Enfin, nous ne pouvons clôturer ce mémoire sans évoquer la perspective, pour un outil, de générer même partiellement, la fonction de référencement f_r de façon automatique. L'exploitation de la fonction f_c conjointement à un langage de parcours géométrique tel qu'évoqué ci-avant pourrait constituer une base intéressante pour la recherche de solutions à cette problématique.

154. Nous pensons à des cas où, partant d'une topologie t_1 au niveau dialogue, l'utilisateur applique une succession d'outils menant à une topologie t_2 ($t_1 \neq t_2$), puis à nouveau à la topologie t_1, sans que ces opérations ne fassent appel à l'historique. Ainsi, en toute logique, les graphes associés à chaque situation où la topologie est t_1 devraient être équivalents (en considérant que les PGN influents soient également inchangés).

Annexe A

Représentation par les frontières et topologie

Sommaire

A.1 Relation d'Euler-Poincaré 178

A.2 Opérateurs d'Euler 178

Nous rappelons ici quelques points relatifs à la topologie dans le cadre des représentations par les frontières. La section A.1 rappelle la relation d'Euler-Poincaré qui permet de vérifier la validité topologique d'un modèle de représentation par les frontières, tandis que la section A.2 présente les opérateurs d'Euler qui sont des fonctions canoniques permettant de construire un modèle de représentation par les frontières de manière incrémentale.

A.1 Relation d'Euler-Poincaré

La validité topologique d'un modèle de représentation par les frontières peut être vérifiée grâce à la relation d'Euler-Poincaré (théorème de Descartes-Euler généralisé) :

$$\chi = V - E + F = 2\,(S - H) + R$$

où χ est la caractéristique d'Euler, V le nombre de sommets *(vertices)*, E le nombre d'arêtes *(edges)*, F le nombre de faces *(faces)*, S le nombre de composantes connexes *(shells)*, H le nombre de trous *(holes)* et R le nombre d'anneaux dans les faces *(rings)*. Un modeleur respectant cette relation est dit eulérien. Pour un solide, nous aurons toujours $\chi = 2$. Cette relation ne garantit cependant pas la validité géométrique.

Plus formellement, la relation d'Euler peut être définie comme l'équation d'un hyperplan h dans un espace de dimension 6, dont les axes sont $\langle V, E, F, S, H, R \rangle$ [Gardan, 2002]. Elle a été à l'origine de la définition des opérateurs d'Euler.

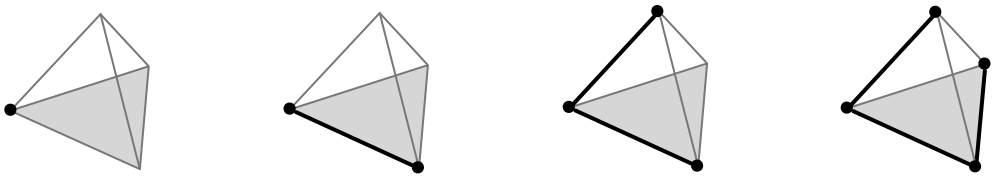
A.2 Opérateurs d'Euler

Les opérateurs d'Euler permettent de transformer un solide structurellement, tout en conservant la valeur de χ (ce qui les rend utiles dans la preuve du théorème de Descartes-Euler). Par définition, un opérateur d'Euler est donc un vecteur non nul dans h. La table A.1 recense les principaux opérateurs d'Euler. Cependant, les cinq opérateurs MVFS, MEV, MEF, KEMR et KFMRH suffisent pour réaliser l'ensemble des opérations [Gardan, 2002]. La figure A.1 présente un exemple d'utilisation de ces opérateurs pour la construction d'un tétraèdre.

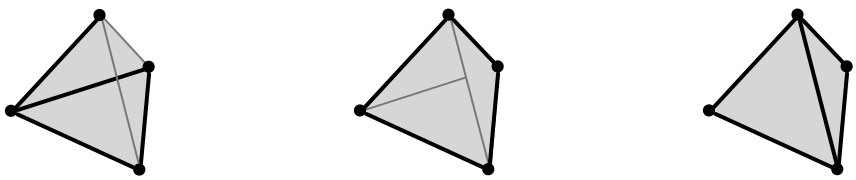
À l'image de la relation d'Euler-Poincaré, l'utilisation des opérateurs d'Euler ne garantit pas la validité géométrique, mais uniquement la satisfaction des conditions de validité topolo-

TABLE A.1: Principaux opérateurs d'Euler.

Nom de l'opérateur	Signification	*V*	*E*	*F*	*S*	*H*	*R*
Opérateurs de création							
MVFS	*Make vertex, face, and shell*	+1		+1	+1		
MEV	*Make edge and vertex*	+1	+1				
MEF	*Make edge and face*		+1	+1			
MEKR	*Make edge, kill ring*		+1				-1
MFKRH	*Make face, kill ring and hole*			+1		-1	-1
Opérateurs de destruction							
KVFS	*Kill vertex, face, and shell*	-1		-1	-1		
KEV	*Kill edge and vertex*	-1	-1				
KEF	*Kill edge and face*		-1	-1			
KEMR	*Kill edge, make ring*		-1				+1
KFMRH	*Kill face, make ring and hole*			-1		+1	+1



(a) MVFS (b) MEV (c) MEV (d) MEV



(e) MEF (f) MEF (g) MEF

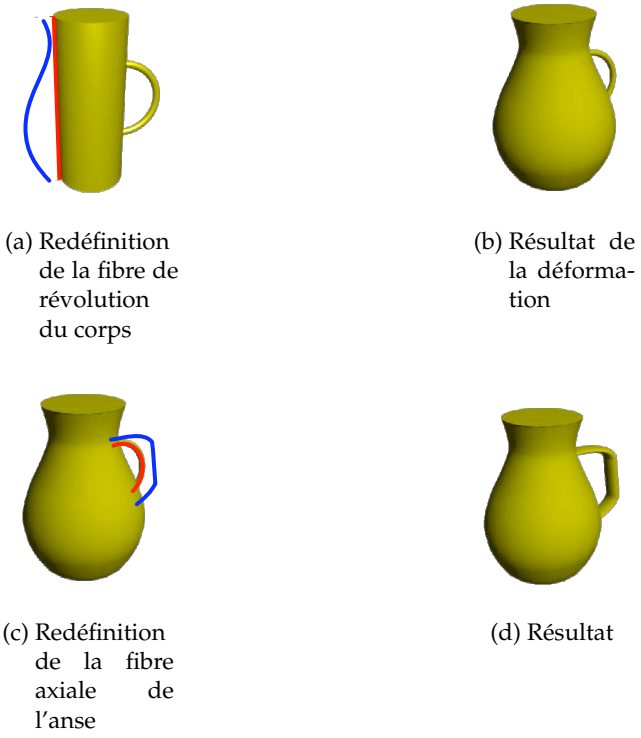
FIGURE A.1: Construction d'un tétraèdre à l'aide des opérateurs d'Euler. Le nom de l'opérateur appliqué est indiqué à chaque étape.

gique. En effet, si les configurations des figures A.1a à A.1f ne sont pas valides, c'est uniquement du fait de leur géométrie (qui est d'ailleurs volontairement faussée par souci de clarté de l'illustration). Ainsi, elles vérifient bel et bien les conditions de validité topologique [155]. Par ailleurs, Mäntylä a prouvé que les opérateurs d'Euler formaient un ensemble de primitives de modélisation complet pour les solides eulériens [Mäntylä, 1984]. Mais il n'y a cependant pas unicité de représentation : plusieurs séquences d'opérateurs différentes peuvent représenter le même objet.

155. Ce qui signifie qu'il est possible de leur associer une géométrie correspondant à un solide valide.

Annexe B

Exemple de conception utilisant l'approche synthétique



(a) Redéfinition de la fibre de révolution du corps

(b) Résultat de la déformation

(c) Redéfinition de la fibre axiale de l'anse

(d) Résultat

FIGURE B.1: Conception d'un pichet basée sur la méthode synthétique.

Nous présentons dans cette annexe un exemple de conception basé sur l'approche synthétique de DIJA. Il s'agit de la version complète de l'exemple abordé brièvement au chapitre 3 (figure 3.1, page 97), concernant la conception d'un pichet. Cet exemple est repris et développé sur la figure B.1.

La forme de départ est choisie dans une librairie métier (cf. figure B.1a). Cette forme, un corps cylindrique doté d'une anse courbe dans ce cas, répond à des fonctions liées au métier de conception concerné (contenir du liquide ; être préhensible…). Les outils de déformation associés à ce métier permettent ensuite de raffiner la forme conjointement à ses fonctions. Il est donc possible de redéfinir la forme du corps et de l'anse du pichet notamment par l'intermédiaire de ses fibres. Le corps du pichet est ainsi évasé grâce à un outil redéfinissant la fibre de révolution associée (cf. figures B.1a et B.1b). Ceci correspond au modelage que pourrait opérer un potier. L'anse est quant à elle redéfinie par l'intermédiaire de sa fibre axiale (cf. figures B.1c et B.1d).

L'application des outils peut en outre faire intervenir de nouveaux éléments de dialogue. C'est le cas lors de l'ajout d'un bec verseur au pichet. Le profil du bec est en effet défini par un

élément de dialogue, tandis que le contour caractéristique délimitant le haut du pichet est redéfini pour inclure l'échancrure du bec (différentes solutions sont présentées sur les figures B.2a à B.2f).

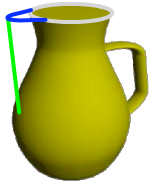
Le raisonnement du concepteur s'appuie ainsi sur les éléments de dialogue qui sont utilisés comme support tangible pour le raffinement des fonctions du produit. Ce rapprochement entre raisonnement sur les fonctions et transformation de la géométrie rend cette approche plus intuitive pour le concepteur en comparaison aux techniques de conception classiques (notamment constructives). En outre, le lien entre fonctions et forme étant conservé (chaque élément de forme correspond à une ou plusieurs fonctions), l'intention du concepteur reste lisible, même une fois le produit réalisé, ce qui favorise la réutilisation.



(a) Définition du bec verseur (première solution)



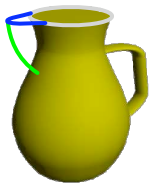
(b) Résultat



(c) Définition du bec verseur (deuxième solution)



(d) Résultat



(e) Définition du bec verseur (troisième solution)



(f) Résultat

FIGURE B.2: Différentes solutions pour le bec verseur.

Annexe C

Algorithmes utilisés pour la propagation

Sommaire

C.1 Propagation 186

C.2 Division 186

C.3 Fusion 194

Dans cette annexe nous décrivons les algorithmes utilisés dans le cadre de la propagation des règles de fabricabilité, présentée plus informellement au § 3.4.3.3.2.

C.1 Propagation

À chaque élément de dialogue e de $\mathbb{E}_2$, nous associons un numéro d'ordre, obtenu via une fonction dénommée num (e). Ce numéro est nécessaire afin de pouvoir effectuer des recoupements entre les paramètres des règles lors d'une division, de sorte que les éléments comparables (p. ex. les fragments d'une fibre de révolution et les contours caractéristiques associés) puissent être appariés correctement. Ainsi, ce numéro a une influence déterminante sur la façon dont les règles évoluent et doit donc être correctement renseigné par l'outil. Dans le cas où plusieurs sous-opérations ont lieu conjointement (p. ex. des divisions en des nombres de fragments différents), ce numéro permet également de ne pas les confondre, ce qui est obtenu en préfixant le numéro d'ordre par le numéro de l'opération à laquelle il correspond (p. ex. : 1.1, 1.2, 1.3 . . . 2.1, 2.2 . . .).

L'algorithme C.1 reprend la procédure de propagation décrite au § 3.4.3.3.2 en la complétant. Cet algorithme fait référence aux procédures de division et de fusion que nous détaillons dans cette annexe (§ C.2 et C.3). Les ensembles $\mathcal{E}^d$, $\mathcal{EE}^d$, $\mathcal{V}^d$, $\mathcal{VV}^d$, $\mathcal{G}^d$, $\mathcal{GG}^d$, $\mathcal{R}^d$, et $\mathcal{RR}^d$ d'une part, et $\mathcal{E}^f$, $\mathcal{EE}^f$, $\mathcal{VV}^f$, $\mathcal{GG}^f$, et $\mathcal{RR}^f$ d'autre part, qui apparaissent dans l'algorithme C.1 sont des variables intermédiaires décrites plus loin.

C.2 Division

Les figures C.4, C.1, C.2 et C.3 correspondent à différentes configurations rencontrées au cours de l'application de l'algorithme de division d'un ED. La division d'un ED est présentée à travers les algorithmes C.2, C.3, C.4, et C.5. Ces différents algorithmes se basent sur les variables, procédures et fonctions décrites dans la table C.1.

Le principe de l'algorithme de division est basé sur le parcours de chaque VF connectée à l'ED à diviser. Si une VF n'a pas déjà été explorée (et donc divisée) et que la propagation est paramétrée en tant qu'adaptation, elle est divisée et ses exemplaires sont raccordés un à un aux exemplaires de l'ED à diviser. Les GF et les règles connectées à une telle VF subissent un traitement analogue. Si au contraire une VF est un exemplaire d'une VF déjà divisée [156], les

156. La division doit bien sûr être compatible avant celle qui est en cours de traitement.

ALGORITHME C.1: Propagation (reformulation).

procédure Propagation $(\mathbb{E}_1, \mathbb{E}_2, f_c, f_r)$

Recopier le graphe des grandeurs initial $\mathbb{G}_1$ en $\mathbb{G}_2$
pour tout $x \in \mathbb{E}_1$ **faire** {Suppression des ED de E^s}
  si $f_c(x) = \emptyset$ **alors**
    Supprimer x et les instances des règles qui lui sont associées de $\mathbb{G}_2$
  fin si
fin pour
$E^a = \emptyset$
pour tout $x \in \mathbb{E}_2$ **faire** {Insertion des ED de E^a}
  si $f_c^{-1}(x) = \emptyset$ **alors**
    Insérer x dans $\mathbb{G}_2$
    $E^a = E^a \cup x$
  fin si
fin pour
$\mathcal{E}^f = \emptyset, \mathcal{EE}^f = \emptyset, \mathcal{V}^f = \emptyset, \mathcal{VV}^f = \emptyset, \mathcal{G}^f = \emptyset, \mathcal{GG}^f = \emptyset, \mathcal{R}^f = \emptyset, \mathcal{RR}^f = \emptyset$
pour tout $x \in \mathbb{E}_1$ **faire** {Fusions et substitutions}
  si Card $(f_c(x)) = 1$ **alors**
    Soit $i \mid i \in f_c(x)$
    si Card $\left(f_c^{-1}(i)\right) = 1$ **alors**
      si $i \neq x$ **alors**
        Substituer x par i dans $\mathbb{G}_2$
      fin si
    sinon
      mFusion $\left(f_c^{-1}(i), f_c(x)\right)$
    fin si
  fin si
fin pour
fusion $\left(\mathcal{EE}^f, \mathcal{VV}^f, \mathcal{GG}^f, \mathcal{RR}^f\right)$
$\mathcal{E}^d = \emptyset, \mathcal{EE}^d = \emptyset, \mathcal{V}^d = \emptyset, \mathcal{VV}^d = \emptyset, \mathcal{G}^d = \emptyset, \mathcal{GG}^d = \emptyset, \mathcal{R}^d = \emptyset, \mathcal{RR}^d = \emptyset$
pour tout $x \in \mathbb{E}_1$ **faire** {Divisions}
  si Card$(f_c(x)) > 1$ **alors**
    division $(x, f_c(x))$
  fin si
fin pour
divisionRef (E^a, f_r)

exemplaires de l'ED à diviser sont raccordés aux exemplaires de la VF. Il en va de même pour les GF et règles connectées à une VF venant d'être divisée.

Dans les algorithmes présentés dans cette annexe, la prise en compte de la fonction de référencement n'est pas détaillée (cf. procédure divisionRef (E^a, f_r) dans l'algorithme C.1). Mais la procédure de division par référence est similaire à celle de division explicite. La seule divergence réside dans le choix des VF à diviser. Ce choix s'effectue sur la base de la fonction de référencement. Par exemple, la division d'une fibre de révolution correspond à la création de contours caractéristiques référençant les contours caractéristiques initialement présents. Si ces derniers étaient initialement connectés à une VF, celle-ci sera divisée.

ALGORITHME C.2: Division d'un ED.

procédure division (e_{div}, E)
e_{div} : élément à diviser.
E : ensemble des éléments remplaçant e_{div} après la division.

pour tout $x^e \in E$ **faire**
  si $x^e \notin \mathbb{G}_2$ **alors**
    Insérer x^e dans $\mathbb{G}_2$
  fin si
fin pour
pour tout $x^v \in \text{vfd}(e_{div})$ **faire**
  si $x^v \notin \mathcal{V}^t$ **alors**
    si $f_p(e_{div}, x^v, \texttt{d}) = \texttt{adaptation}$ **alors**
      divisionVF (E, x^v)
    sinon
      si $f_p(e_{div}, x^v, \texttt{d}) = \texttt{effacement} \vee e_{div} \notin E$ **alors**
        Délier e_{div} et x^v
      sinon {cas de la conservation}
        Relier x^v à chaque élément de E
      fin si
    fin si
  fin si
fin pour
si $e_{div} \notin E$ **alors**
  Supprimer e_{div} de $\mathbb{G}_2$ {en toute logique, e_{div} est isolé}
fin si
$\mathcal{E}^d = \mathcal{E}^d \cup e_{div}$
$\mathcal{EE}^d = \mathcal{EE}^d \cup E$

TABLE C.1: Variables et procédures définies globalement.

Nom	Description
Variables	
$\mathbb{G}_1$	graphe des grandeurs originel (avant application de l'outil).
$\mathbb{G}_2$	graphe des grandeurs modifié (pendant la propagation).
$\mathcal{E}^d$	ensemble des ED déjà divisés (issus d'une division).
$\mathcal{EE}^d$	ensemble des ensembles d'ED déjà divisés.
$\mathcal{V}^d$	ensemble des VF déjà divisées.
$\mathcal{VV}^d$	ensemble des ensembles de VF déjà divisées.
$\mathcal{G}^d$	ensemble des GF déjà divisées.
$\mathcal{GG}^d$	ensemble des ensembles de GF déjà divisées.
$\mathcal{R}^d$	ensemble des règles déjà divisées.
$\mathcal{RR}^d$	ensemble des ensembles de règles déjà divisées.
$\mathcal{E}^f$	ensemble des ED issus d'une fusion.
$\mathcal{EE}^f$	ensemble des ensembles d'ED à fusionner.
$\mathcal{VV}^f$	ensemble des ensembles de VF candidates à la fusion.
$\mathcal{GG}^f$	ensemble des ensembles de GF candidates à la fusion.
$\mathcal{RR}^f$	ensemble des ensembles de règles candidates à la fusion.
Procédures et fonctions	
$f_c\,(e)$	fonction de correspondance.
$f_r\,(e)$	fonction de référencement.
$f_p\,(e, v, o)$	fonction de paramétrage de la propagation sur l'arc reliant l'ED e à la VF v, pour l'opérateur o.
vfd $(e_1 \dots e_n)$	renvoie les VF dépendantes du ou des ED $e_1 \dots e_n$.
eda $(v_1 \dots v_n)$	renvoie les ED antécédents de la ou des VF $v_1 \dots v_n$.
gfd $(v_1 \dots v_n)$	renvoie les GF dépendantes de la ou des VF $v_1 \dots v_n$.
vfa $(g_1 \dots g_n)$	renvoie les VF antécédentes de la ou des GF $g_1 \dots g_n$.
rd $(g_1 \dots g_n)$	renvoie les règles dépendantes de la ou des GF $g_1 \dots g_n$.
gfa $(r_1 \dots r_n)$	renvoie les GF antécédentes de la ou des règles $r_1 \dots r_n$.
num (e)	renvoie le numéro d'ordre de l'ED e.
compat (E_1, E_2)	renvoie `vrai` si les deux ensembles d'ED sont compatibles (nombres d'éléments identiques et numéros d'ordre indexés sur la même opération, cf. ci-avant), ou `faux` sinon.

ALGORITHME C.3: Division d'une VF.

procédure divisionVF (E, e_{div}, v_{div})

E : ensemble d'éléments connectés à v_{div} après la division.

e_{div} : élément connecté à v_{div}.

v_{div} : VF à diviser.

---

si $v_{div} \in \mathcal{V}^d$ **alors** {la VF a déjà été divisée (cf. figure C.1)}
  Soit $G^v \in \mathcal{VV}^d \mid v_{div} \in G^v \wedge \left(\exists G^e \in \mathcal{EE}^d \mid G^e \subseteq \text{eda}\left(G^v\right) \wedge \text{compat}\left(G^e, E\right) = \texttt{vrai}\right)$
  si $G^v \neq \emptyset$ **alors**
    Délier v_{div} et e_{div}
    pour tout $x^e \in E$ **faire**
      Soit $x^v \in G^v \mid (\exists y^e \in G^e \mid y^e \in \text{eda}\left(x^v\right) \wedge \text{num}\left(y^e\right) = \text{num}\left(x^e\right))$
      Relier x^e et x^v similairement à e_{div} et v_{div} dans $\mathbb{G}_1$
    fin pour
    si $e_{div} \notin E$ **alors** {cas de la substitution}
      Délier e_{div} et v_{div} dans $\mathbb{G}_2$
    fin si
    $\mathcal{V}^t = \mathcal{V}^t \cup G^v$
    sortie procédure
  fin si
fin si
{la VF n'a pas encore été divisée ou la division n'est pas compatible}
Dupliquer v_{div} ainsi que les liens qui pointent sur elle
Soit V l'ensemble des VF ainsi obtenues
Substituer les liens entre e_{div} et V par un nouveau lien entre chaque élément de E et un élément de V
si $e_{div} \notin E$ **alors** {cas de la substitution}
  Délier e_{div} et v_{div} dans $\mathbb{G}_2$
fin si
pour tout $x^g \in \text{gfd}\left(v_{div}\right)$ **faire**
  si $x^g \notin \mathcal{G}^t$ **alors**
    divisionGF (E, V, v_{div}, x^g)
  fin si
fin pour
$\mathcal{VV}^d = \mathcal{VV}^d \cup V$
$\mathcal{V}^d = \mathcal{V}^d \cup v_{div}$
$\mathcal{V}^t = \mathcal{V}^t \cup v_{div}$

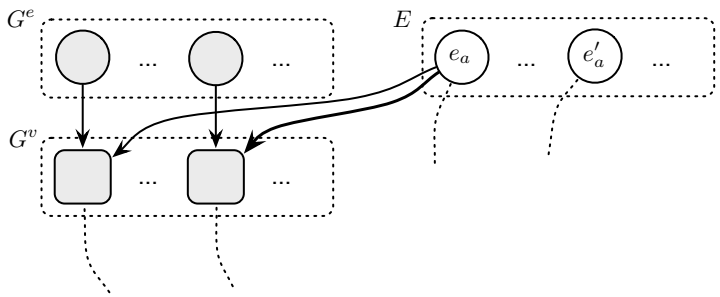


FIGURE C.1: Valeur de fabrication déjà divisée.

ALGORITHME C.4: Division d'une GF.

procédure divisionGF (E, V, v_{div}, g_{div})

E : ensemble d'éléments connectés à V.

V : ensemble de VF connectées à g_{div} après la division.

v_{div} : VF connectée à g_{div}.

g_{div} : GF à diviser.

---

si $g_{div} \in \mathcal{G}^d$ **alors** {la GF a déjà été divisée (cf. figure C.2)}
  Soit $G^g \in \mathcal{GG}^d \mid g_{div} \in G^g \wedge \Big(\exists G^v \in \mathcal{VV}^d \mid G^v \subseteq \text{vfa}\,(G^g) \wedge$
    $\Big(\exists G^e \in \mathcal{EE}^d \mid G^e \subseteq \text{eda}\,(G^v) \wedge \text{compat}\,(G^e, E) = \texttt{vrai}\Big)\Big)$
  si $G^g \neq \emptyset$ **alors**
    Délier g_{div} et v_{div}
    pour tout $x^e \in E$ **faire**
      Soit $x^g \in G^g \mid (\exists x^v \in G^v \mid x^v \in \text{vfa}\,(x^g) \wedge$
        $(\exists y^e \in G^e \mid y^e \in \text{eda}\,(x^v) \wedge \text{num}\,(y^e) = \text{num}\,(x^e)))$
      Soit $y^v \in V \mid y^v \in \text{vfd}\,(x^e)$
      Relier y^v et x^g similairement à v_{div} et g_{div} dans $\mathbb{G}_1$
    fin pour
    $\mathcal{G}^t = \mathcal{G}^t \cup G^g$
    sortie procédure
  fin si
fin si
{la GF n'a pas encore été divisée ou la division n'est pas compatible}
Dupliquer g_{div} ainsi que les liens qui pointent sur elle
Soit G l'ensemble des GF ainsi obtenues
Substituer les liens entre v_{div} et G par un nouveau lien entre chaque élément de V et un élément de G
pour tout $x^r \in \text{rd}\,(g_{div})$ **faire**
  si $x^r \notin \mathcal{R}^t$ **alors**
    divisionRegle (E, V, G, g_{div}, x^r)
  fin si
fin pour
$\mathcal{GG}^d = \mathcal{GG}^d \cup G$
$\mathcal{G}^d = \mathcal{G}^d \cup g_{div}$
$\mathcal{G}^t = \mathcal{G}^t \cup g_{div}$

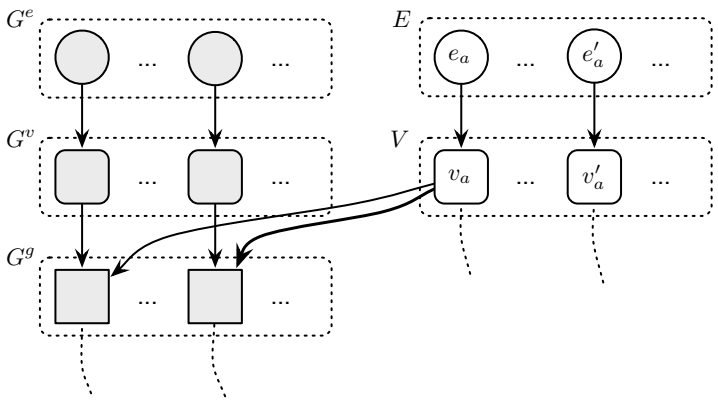


FIGURE C.2: Grandeur de fabrication déjà divisée.

ALGORITHME C.5: Division d'une règle.

procédure divisionRegle $(E, V, G, g_{div}, r_{div})$

E : ensemble d'éléments connectés à V.

V : ensemble de VF connectées à G.

G : ensemble de GF connectées à r_{div} après la division.

g_{div} : GF connectée à r_{div}.

r_{div} : règle à diviser.

si $r_{div} \in \mathcal{R}^d$ **alors** {la règle a déjà été divisée (cf. figure C.3)}
  Soit $G^r \in \mathcal{RR}^d \mid r_{div} \in G^r \wedge \Big(\exists G^g \in \mathcal{GG}^d \mid G^g \subseteq \text{gfa}\,(G^r) \wedge \Big(\exists G^v \in \mathcal{VV}^d \mid G^v \subseteq \text{vfa}\,(G^g) \wedge \Big(\exists G^e \in \mathcal{EE}^d \mid G^e \subseteq \text{eda}\,(G^v) \wedge \text{compat}\,(G^e, E) = \texttt{vrai}\Big)\Big)\Big)$
  si $G^r \neq \emptyset$ **alors**
    Délier r_{div} et g_{div}
    pour tout $x^e \in E$ **faire**
      Soit $x^r \in G^r \mid (\exists x^g \in G^g \mid x^g \in \text{gfa}\,(x^r) \wedge (\exists x^v \in G^v \mid x^v \in \text{vfa}\,(x^g) \wedge (\exists y^e \in G^e \mid y^e \in \text{eda}\,(x^v) \wedge \text{num}\,(y^e) = \text{num}\,(x^e))))$
      Soit $(y^v, y^g) \in (V, G) \mid y^v \in \text{vfd}\,(x^e) \wedge y^g \in \text{gfd}\,(x^v)$
      Relier y^g et x^r similairement à g_{div} et r_{div} dans $\mathbb{G}_1$
    fin pour
    $\mathcal{R}^t = \mathcal{R}^t \cup G^r$
    sortie procédure
  fin si
fin si
{la règle n'a pas encore été divisée (cf. figure C.4) ou la division n'est pas compatible}
Dupliquer r_{div} ainsi que les liens qui pointent sur elle
Soit R l'ensemble des règles ainsi obtenues
Substituer les liens entre g_{div} et R par un nouveau lien entre chaque élément de G et un élément de R
$\mathcal{RR}^d = \mathcal{RR}^d \cup R$
$\mathcal{R}^d = \mathcal{R}^d \cup r_{div}$
$\mathcal{R}^t = \mathcal{R}^t \cup r_{div}$

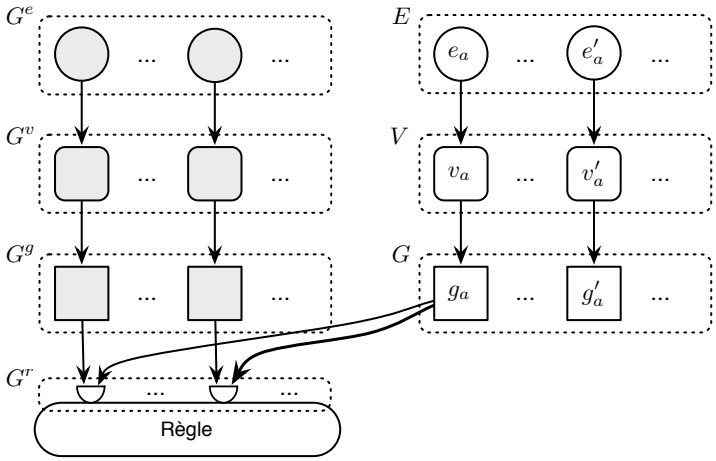


FIGURE C.3: Règle déjà « divisée » (instance clonée).

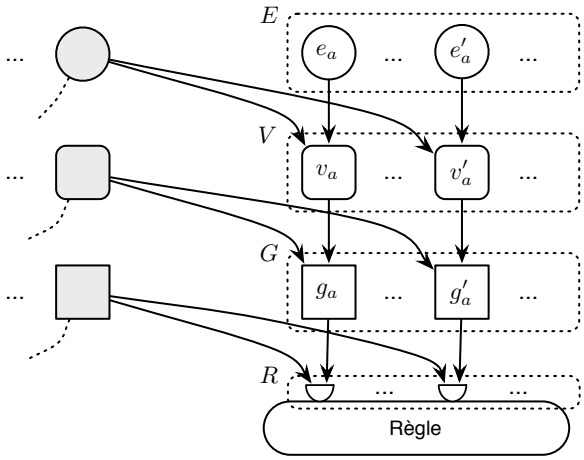


FIGURE C.4: VF, GF et règle non encore divisées.

C.3 Fusion

Le principe de la fusion est comparable à celui de la division, à ceci près qu'il est mené à l'inverse. Une fusion ne survient qu'une fois que tous les éléments pouvant être fusionnés ont été explorés, ce qui signifie que, contrairement aux ensembles $\mathcal{E}^d$, $\mathcal{EE}^d$, $\mathcal{V}^d$, $\mathcal{VV}^d$, $\mathcal{G}^d$, $\mathcal{GG}^d$, $\mathcal{R}^d$, et $\mathcal{RR}^d$, les ensembles $\mathcal{E}^f$, $\mathcal{EE}^f$, $\mathcal{VV}^f$, $\mathcal{GG}^f$, et $\mathcal{RR}^f$ sont remplis avant que la fusion ne soit opérée. Il serait en effet impossible de ne fusionner que partiellement des règles car les parties non encore fusionnées conduiraient à des incohérences au niveau des arcs incidents aux VF, GF ou règles déjà fusionnées. Les ensembles intermédiaires sont ainsi entièrement remplis avant qu'une fusion ne soit opérée. L'algorithme de fusion opère ainsi en deux temps :

1. Marquage des éléments candidats à la fusion. Avant le marquage, les arcs devant être supprimés sont traités. La phase de marquage est un parcours à chaque niveau du GGF s'assurant des compatibilités de type abstrait entre éléments d'un même ensemble candidat à la fusion.

2. Fusion des éléments marqués à l'étape précédente, sous condition de compatibilité des signatures (genre et nombre des paramètres des VF, GF et règles). À l'issue de cette étape, seuls les nœuds issus de la fusion sont conservés dans $\mathbb{G}_2$, les arcs devant être conservés étant rattachés à l'ED fruit de la fusion.

La première phase est décrite à travers les algorithmes C.6, C.7, C.8, et C.9, tandis que la seconde phase correspond à l'algorithme C.10. Les figures C.5, C.6, C.7 et C.8 correspondent à différentes configurations rencontrées au cours du marquage des éléments candidats à la fusion.

ALGORITHME C.6: Marquage de la fusion pour un ensemble d'ED.

procédure mFusion $(e_{fus,}, E)$

e_{fus} : élément issu de la fusion.

E : ensemble des éléments fusionnés en e_{fus} après la fusion.

si $e_{fus} \in \mathcal{E}^f$ **alors**
  sortie procédure {cette fusion a déjà été traitée}
fin si
si $e_{fus} \notin \mathbb{G}_2$ **alors**
  Insérer e_{fus} dans $\mathbb{G}_2$
fin si
pour tout $x^e \in E$ **faire** {effacement}
  pour tout $x^v \in \text{vfd}(x^e)$ **faire**
    si $f_p(x^e, x^v, \texttt{f}) = \texttt{effacement}$ **alors**
      Délier x^v, puis effacer x^v en cascade dans $\mathbb{G}_2$
    fin si
  fin pour
fin pour
$\mathcal{VV}^c = \emptyset$
pour tout $x^e \in E$ **faire** {recherche des ensembles de VF candidates à la fusion}
  pour tout $x^v \in \text{vfd}(x^e)$ **faire**
    si $f_p(x^e, x^v, \texttt{f}) = \texttt{adaptation}$ **alors**
      si $\exists X^v \in \mathcal{VV}^c$ comprenant des VF compatibles avec x^v (même VF abstraite) **alors**
        $X^v = X^v \cup x^v$
      sinon
        $\mathcal{VV}^c = \mathcal{VV}^c \cup \{x^v\}$
      fin si
    fin si
  fin pour
fin pour
pour tout $X^v \in \mathcal{VV}^c$ **faire**
  si $\text{Card}(X^v) \neq \text{Card}(E)$ **alors**
    $\mathcal{VV}^c = \mathcal{VV}^c \setminus X^v$
  sinon
    mFusionVF (X^v)
  fin si
fin pour
$\mathcal{E}^f = \mathcal{E}^f \cup e^f$
$\mathcal{EE}^f = \mathcal{EE}^f \cup E$

ALGORITHME C.7: Marquage de la fusion pour un ensemble de VF.

procédure mFusionVF (V)

V : ensemble d'éléments candidats à la fusion.

si $V \in \mathcal{VV}^f$ **alors**
  sortie procédure {cette fusion a déjà été traitée}
fin si
$\mathcal{GG}^c = \emptyset$
pour tout $x^v \in V$ **faire** {recherche des ensembles de GF candidates à la fusion}
  pour tout $x^g \in \text{gfd}\,(x^v)$ **faire**
    si $\exists X^g \in \mathcal{GG}^c$ comprenant des GF compatibles avec x^g (même GF abstraite) **alors**
      $X^g = X^g \cup x^g$
    sinon
      $\mathcal{GG}^c = \mathcal{GG}^c \cup \{x^g\}$
    fin si
  fin pour
fin pour
pour tout $X^g \in \mathcal{GG}^c$ **faire**
  si $\text{Card}\,(X^g) \neq \text{Card}\,(V)$ **alors**
    $\mathcal{GG}^c = \mathcal{GG}^c \setminus X^g$
  sinon
    mFusionGF (X^g)
  fin si
fin pour
$\mathcal{VV}^f = \mathcal{VV}^f \cup V$

ALGORITHME C.8: Marquage de la fusion pour un ensemble de GF.

procédure mFusionGF (G)
G : ensemble d'éléments candidats à la fusion.

si $G \in \mathcal{GG}^f$ **alors**
  sortie procédure {cette fusion a déjà été traitée}
fin si
$\mathcal{RR}^c = \emptyset$
pour tout $x^g \in V$ **faire** {recherche des ensembles de règles candidates à la fusion}
  pour tout $x^r \in \text{rd}(x^g)$ **faire**
    si $\exists X^r \in \mathcal{RR}^c$ comprenant des règles compatibles avec x^r (même règle abstraite) **alors**
      $X^r = X^r \cup x^r$
    sinon
      $\mathcal{RR}^c = \mathcal{RR}^c \cup \{x^r\}$
    fin si
  fin pour
fin pour
pour tout $X^r \in \mathcal{RR}^c$ **faire**
  si $\text{Card}(X^r) \neq \text{Card}(G)$ **alors**
    $\mathcal{RR}^c = \mathcal{RR}^c \setminus X^r$
  sinon
    mFusionRegles (X^r)
  fin si
fin pour
$\mathcal{GG}^f = \mathcal{GG}^f \cup G$

ALGORITHME C.9: Marquage de la fusion pour un ensemble de règles.

procédure mFusionRegles (R)
R : ensemble d'éléments candidats à la fusion.

si $R \in \mathcal{RR}^f$ **alors**
  sortie procédure {cette fusion a déjà été traitée}
fin si
$\mathcal{RR}^f = \mathcal{RR}^f \cup R$

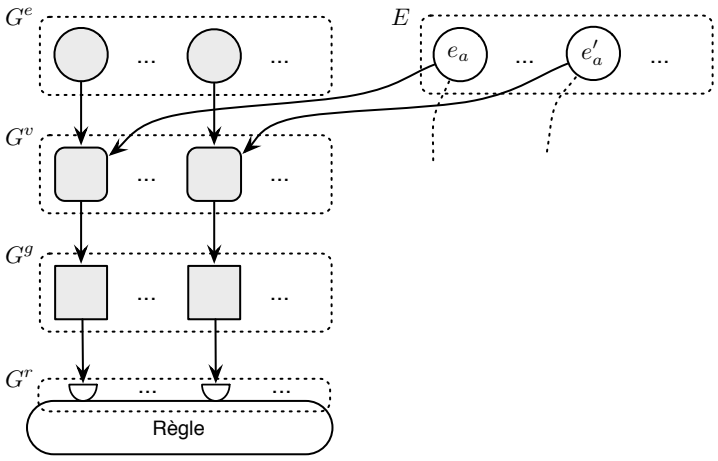


FIGURE C.5: Une fusion peut encore survenir au niveau des VF.

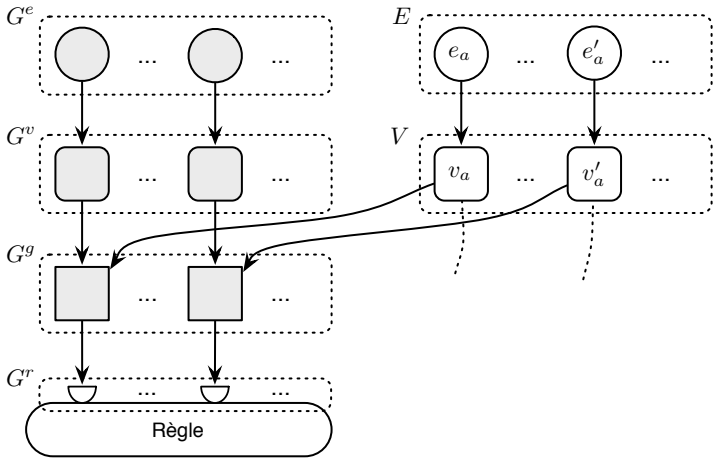


FIGURE C.6: Une fusion peut encore survenir au niveau des GF.

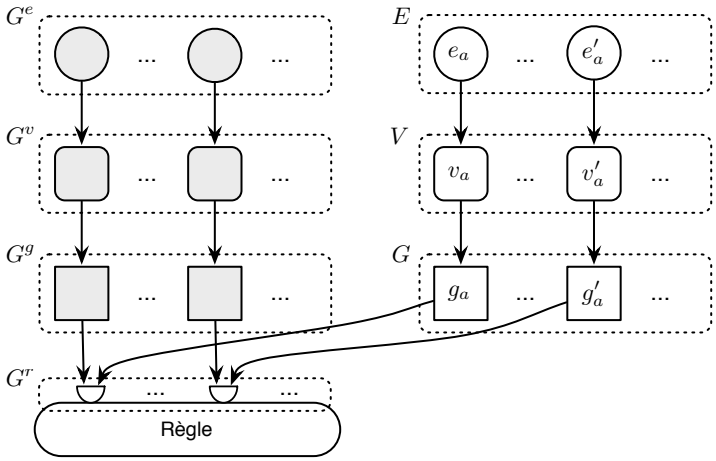


FIGURE C.7: Une fusion peut encore survenir au niveau des instances de la règle.

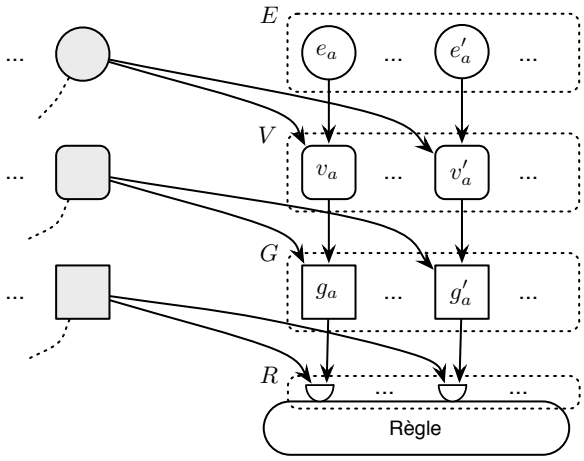


FIGURE C.8: Dernière fusion possible d'un ensemble homogène.

ALGORITHME C.10: Fusion des ensembles marqués.

procédure fusion $\left(\mathcal{EE}^f, \mathcal{VV}^f, \mathcal{GG}^f, \mathcal{RR}^f\right)$

$\mathcal{EE}^f$: ensemble d'éléments candidats à la fusion.
$\mathcal{VV}^f$: ensemble des VF candidates à la fusion.
$\mathcal{GG}^f$: ensemble des GF candidates à la fusion.
$\mathcal{RR}^f$: ensemble des règles candidates à la fusion.

pour tout $X^e \in \mathcal{EE}^f$ **faire**
  Délier chaque VF reliée à X^e et la relier à l'élément de $\mathcal{E}^f$ correspondant
  Supprimer les autres ED de X^e
fin pour
pour tout $X^v \in \mathcal{VV}^f$ **faire** {traiter les ensembles par cardinalités décroissantes}
  si les « signatures » des VF le permettent **alors**
    Délier chaque GF reliée à X^v et la relier à l'élément fusionné
    Délier chaque ED relié à X^v et le relier à l'élément fusionné
    Supprimer les autres VF de X^v
  fin si
fin pour
pour tout $X^g \in \mathcal{GG}^f$ **faire** {traiter les ensembles par cardinalités décroissantes}
  si les « signatures » des GF le permettent **alors**
    Délier chaque règle reliée à X^g et la relier à l'élément fusionné
    Délier chaque VF reliée à X^g et la relier à l'élément fusionné
    Supprimer les autres GF de X^g
  fin si
fin pour
pour tout $X^r \in \mathcal{RR}^f$ **faire** {traiter les ensembles par cardinalités décroissantes}
  si les « signatures » des règles le permettent **alors**
    Délier chaque GF reliée à X^r et la relier à l'élément fusionné
    Supprimer les autres règles de X^r
  fin si
fin pour

Annexe D

Calcul de distances dans l'espace

Sommaire

D.1 Mesures devant être effectuées 202

D.2 Calculs sous-jacents 203

D.2.1 Distance entre deux points dans l'espace 203

D.2.2 Distance entre deux segments coplanaires 203

D.2.3 Distance entre deux contours polygonaux 204

D.2.4 Coûts 208

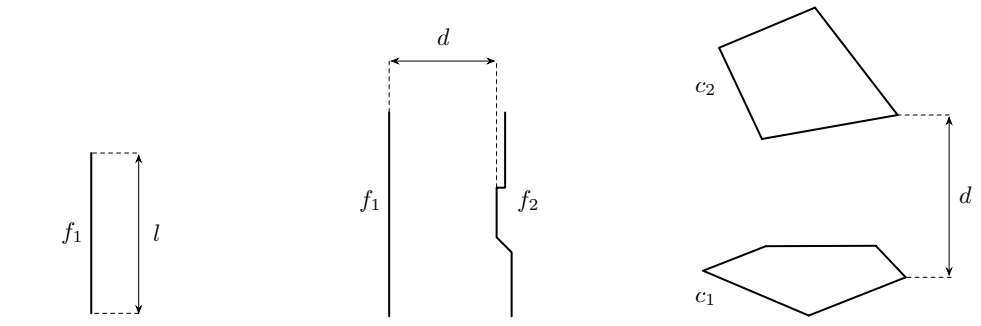
D.3 Remarques 208

Nous présentons dans cette annexe quelques calculs de distance entre éléments de dialogue dans l'espace. Ces calculs sont utilisés dans la maquette présentée au chapitre 4, dans le cadre de la définition des opérateurs de mesure. Nous ne décrivons que des calculs entre éléments formés de lignes brisées (les courbes et surfaces libres ne sont pas considérées).

D.1 Mesures devant être effectuées

L'approche proposée dans ce manuscrit fait apparaître la nécessité d'effectuer des mesures entre des éléments de dialogue. Le plus fréquemment, il s'agit de calculer des distances entre des entités géométriques. Ces mesures de distance sont entre autres :

1. Distance intra-fibre (cf. figure D.1a) : il s'agit de la distance entre les deux extrémités d'une fibre, ou de la longueur d'une fibre ;
2. Distance inter-fibres (cf. figure D.1b) : la distance entre deux fibres coplanaires (qui correspondent le plus souvent à une même révolution). L'intérêt de mesurer la distance entre deux fibres non coplanaires semble limité dans le cadre de DIJA ;
3. Distance inter-contours caractéristiques (cf. figure D.1c) : la distance entre deux contours caractéristiques. Les deux contours délimitent des surfaces planes dont l'intersection est vide, mais qui ne sont pas nécessairement parallèles.



(a) Distance intra-fibre (b) Distance inter-fibres (c) Distance inter-contours

FIGURE D.1: Différents calculs de distance.

D'autres types de mesures sont possibles (p. ex. les calculs d'aire) mais nous ne les avons pas considérés du fait que les calculs de distances étaient suffisants pour démontrer l'intérêt de l'approche.

D.2 Calculs sous-jacents

Les calculs de base nécessaires pour réaliser les mesures décrites ci-avant sont :

1. Le calcul de la distance entre deux points dans l'espace ;
2. Le calcul de la distance entre deux ensembles de segments coplanaires, ramené au calcul de la distance entre deux segments coplanaires ;
3. Le calcul de la distance entre deux contours polygonaux plans délimitant des surfaces qui ne s'intersectent pas.

Chacun de ces calculs est détaillé dans les paragraphes suivants.

D.2.1 Distance entre deux points dans l'espace

Ce calcul est simple. La distance entre deux points $A(x_A, y_A, z_A)$ et $B(x_B, y_B, z_B)$ est donnée par la formule suivante :

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

D.2.2 Distance entre deux segments coplanaires

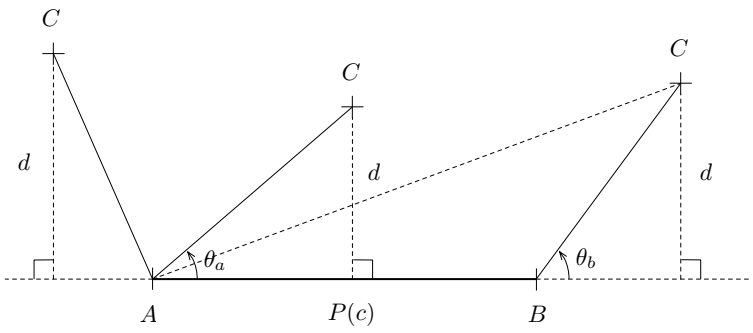


FIGURE D.2: Distance entre un point et un segment dans l'espace.

Les deux segments étant supposés coplanaires, ce calcul se ramène à celui de la distance entre un point C et un segment $[AB]$ dans l'espace [157] (cf. figure D.2). Les points A, B et C sont par hypothèse distincts.

157. La distance entre deux segments $[AB]$ et $[CD]$ peut en effet être obtenue par comparaison des distances entre le point x et le segment $[yz]$, avec $(x, y, z) \in \{(A, C, D), (B, C, D), (C, A, B), (D, A, B)\}$.

La droite porteuse du segment est définie par son équation paramétrique :

$$\begin{cases} P_x(t) &= x_A + t(x_B - x_A) \\ P_y(t) &= y_A + t(y_B - y_A) \\ P_z(t) &= z_A + t(z_B - z_A) \end{cases} \quad \text{avec } P(0) = A \text{ et } P(1) = B$$

La coordonnée paramétrique c du projeté du point C sur la droite (AB) est définie par :

$$c = \frac{d(A, P(c))}{d(A, B)} = \frac{\left\|\overrightarrow{AC}\right\| \times \cos\theta_A}{\left\|\overrightarrow{AB}\right\|} = \frac{\overrightarrow{AC}.\overrightarrow{AB}}{\left\|\overrightarrow{AB}\right\|^2} = \frac{\overrightarrow{AC}.\overrightarrow{AB}}{\overrightarrow{AB}.\overrightarrow{AB}}$$

De là, la distance d entre un point C et un segment $[AB]$ est définie par :

$$d = \begin{cases} d = \left\|\overrightarrow{AC}\right\| \text{ si } \overrightarrow{AB}.\overrightarrow{AC} \leq 0 \\ d = \left\|\overrightarrow{BC}\right\| \text{ si } \overrightarrow{AB}.\overrightarrow{AC} \geq \overrightarrow{AB}.\overrightarrow{AB} \\ d = \sqrt{\alpha^2 + \beta^2 + \gamma^2} \text{ avec } \begin{cases} \alpha &= x_A + c(x_B - x_A) - x_C \\ \beta &= y_A + c(y_B - y_A) - y_C \\ \gamma &= z_A + c(z_B - z_A) - z_C \end{cases} \text{ sinon} \end{cases}$$

D.2.3 Distance entre deux contours polygonaux

Il s'agit de trouver la distance entre deux ensembles de segments dans l'espace. Nous donnons dans un premier temps le calcul de la distance entre deux droites, puis entre deux segments non coplanaires (cf. figure D.3), et enfin nous envisageons le calcul de la distance entre deux contours polygonaux en nous basant sur le calcul de la distance entre deux segments.

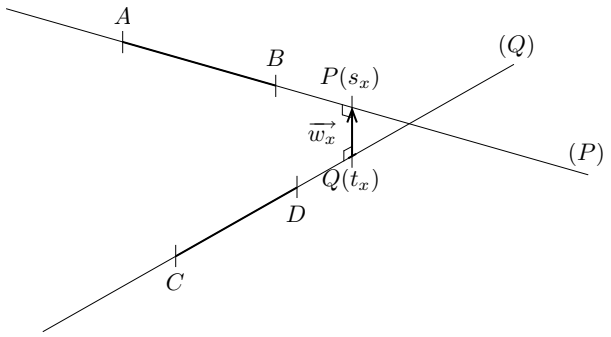


FIGURE D.3: Distance entre deux segments dans l'espace.

D.2.3.1 Distance entre deux droites

Là encore, nous définissons les droites porteuses des segments $[AB]$ et $[CD]$ par leurs équations paramétriques [158] :

$$\begin{cases} P(s) &= P(0) + s(P(1) - P(0)) &= A + s.\overrightarrow{AB} \\ Q(t) &= Q(0) + t(Q(1) - Q(0)) &= C + t.\overrightarrow{CD} \end{cases}$$

Soit le vecteur $\overrightarrow{w(s,t)} = P(s) - Q(t)$. La distance minimale entre les deux droites est la norme de $\overrightarrow{w_x} = \overrightarrow{w(s_x, t_x)}$. Deux cas sont possibles :

1. (P) et (Q) ne sont pas parallèles. Nous avons donc :

$$\begin{aligned} \overrightarrow{AB}.\overrightarrow{w_x} &= 0 \\ &= \overrightarrow{AB}.\left(\overrightarrow{CA} + s_x.\overrightarrow{AB} - t_x.\overrightarrow{CD}\right) \\ &= s_x\left(\overrightarrow{AB}.\overrightarrow{AB}\right) - t_x\left(\overrightarrow{AB}.\overrightarrow{CD}\right) + \overrightarrow{AB}.\overrightarrow{CA} \end{aligned}$$

et

$$\begin{aligned} \overrightarrow{CD}.\overrightarrow{w_x} &= 0 \\ &= \overrightarrow{CD}.\left(\overrightarrow{CA} + s_x.\overrightarrow{AB} - t_x.\overrightarrow{CD}\right) \\ &= s_x\left(\overrightarrow{CD}.\overrightarrow{AB}\right) - t_x\left(\overrightarrow{CD}.\overrightarrow{CD}\right) + \overrightarrow{CD}.\overrightarrow{CA} \end{aligned}$$

158. Là aussi, A, B, C et D sont par hypothèse distincts.

Nous avons alors le système suivant :

$$\begin{cases} \alpha s_x - \beta t_x + \delta &= 0 \\ \beta s_x - \gamma t_x + \epsilon &= 0 \end{cases}$$

avec

$$\begin{aligned} \alpha &= \overrightarrow{AB}.\overrightarrow{AB} \\ \beta &= \overrightarrow{AB}.\overrightarrow{CD} \\ \gamma &= \overrightarrow{CD}.\overrightarrow{CD} \\ \delta &= \overrightarrow{AB}.\overrightarrow{CA} \\ \epsilon &= \overrightarrow{CD}.\overrightarrow{CA} \end{aligned}$$

La solution à ce système est :

$$\begin{aligned} s_x &= \frac{\beta\epsilon - \delta\gamma}{\alpha\gamma - \beta^2} \\ t_x &= \frac{\alpha\epsilon - \beta\gamma}{\alpha\gamma - \beta^2} \end{aligned}$$

avec $\alpha\gamma - \beta^2 \neq 0$. Par ailleurs,

$$\begin{aligned} \alpha\gamma - \beta^2 &= \left(\overrightarrow{AB}.\overrightarrow{AB}\right) \times \left(\overrightarrow{CD}.\overrightarrow{CD}\right) - \left(\overrightarrow{CD}.\overrightarrow{AB}\right)^2 \\ &= \left\|\overrightarrow{AB}\right\|^2 \times \left\|\overrightarrow{CD}\right\|^2 \times \left(1 - (\cos\tau)^2\right) \\ &= \left\|\overrightarrow{AB}\right\|^2 \times \left\|\overrightarrow{CD}\right\|^2 \times (\sin\tau)^2 \end{aligned}$$

τ étant l'angle entre $\overrightarrow{CD}$ et $\overrightarrow{AB}$. Donc $\forall\tau, \alpha\gamma - \beta^2 \geq 0$. Or $\alpha\gamma - \beta^2 = 0 \iff \tau = 0 \vee \tau = \pi$, ce qui est exclu car par hypothèse, (P) et (Q) ne sont pas parallèles et A, B, C et D sont distincts. Nous pouvons donc écrire $\forall\tau, \alpha\gamma - \beta^2 > 0$.
Connaissant s_x et t_x, la distance entre (P) et (Q) est :

$$\begin{aligned} \|\overrightarrow{w_x}\| &= \left\|\overrightarrow{CA} + s_x.\overrightarrow{AB} - t_x.\overrightarrow{CD}\right\| \\ &= \left\|\overrightarrow{CA} + \frac{\beta\epsilon - \delta\gamma}{\alpha\gamma - \beta^2}.\overrightarrow{AB} - \frac{\alpha\epsilon - \beta\gamma}{\alpha\gamma - \beta^2}.\overrightarrow{CD}\right\| \end{aligned}$$

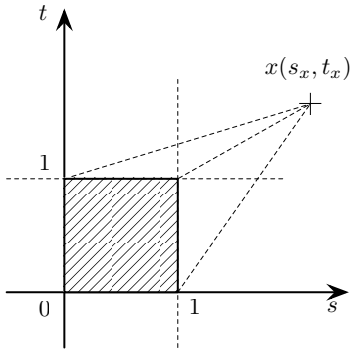


FIGURE D.4: Domaine de validité des paramètres s et t.

2. (P) et (Q) sont parallèles. Dans ce cas, $\alpha\gamma - \beta^2 = 0$ (cf. plus haut). On fixe alors par exemple $s_x = 0$ et on en déduit :

$$t_x = \frac{\delta}{\beta} = \frac{\epsilon}{\gamma}$$

La distance entre les deux droites est alors :

$$\begin{aligned} \|\overrightarrow{w_x}\| &= \left\|\overrightarrow{CA} - t_x.\overrightarrow{CD}\right\| \\ &= \left\|\overrightarrow{CA} - \frac{\delta}{\beta}.\overrightarrow{CD}\right\| \end{aligned}$$

D.2.3.2 Distance entre deux segments non coplanaires

Dans le cas où $(s_x, t_x) \in [0,1]^2$, la distance entre $[AB]$ et $[CD]$ est égale à la distance entre (P) et (Q). Dans les autres cas, il est possible d'obtenir les valeurs de s et t correspondant aux points les plus proches de $\overrightarrow{w_x}$ sur les deux segments, en minimisant la distance de $x\,(s_x, t_x)$ par rapport à l'espace paramétrique délimitant les deux segments, à savoir le carré unitaire hachuré sur la figure D.4. Ces valeurs sont obtenues par application de l'algorithme D.1.

D.2.3.3 Distance entre deux contours polygonaux

De façon naïve, on peut effectuer ce calcul par comparaison des distances entre chacun des segments formant les deux contours polygonaux.

ALGORITHME D.1: Prise en compte du cas où x est en dehors de l'espace paramétrique des deux segments pour le calcul de la distance.

procédure correction_distance(s_x, t_x)

si $s_x < 0$ **alors**
  $s_x \leftarrow 0$
sinon, si $s_x > 1$ **alors**
  $s_x \leftarrow 1$
fin si
si $t_x < 0$ **alors**
  $t_x \leftarrow 0$
sinon, si $t_x > 1$ **alors**
  $t_x \leftarrow 1$
fin si

D.2.4 Coûts

Pris indépendamment, le coût de ces calculs semble assez élevé pour du temps réel, mais pas rédhibitoire :

- la racine carrée peut être calculée en $O(\log(n))$, mais pour la recherche des minima, il est possible d'effectuer les comparaisons uniquement sur les carrés des distances ;
- les autres opérations sont en $O(n)$.

Cependant, pour le dernier type de mesure abordé (distance entre deux contours), les comparaisons doivent être effectuées sur chaque paire de segments constituant les contours. L'opération globale serait donc en $O(n^2)$.

D.3 Remarques

Quelques remarques découlent de cette annexe. Tout d'abord, le calcul de la distance entre deux contours polygonaux pourrait être compromis par l'explosion combinatoire engendrée par le nombre de comparaisons devant être effectuées. Cependant, nous notons que le nombre d'éléments géométriques mis en œuvre sera dans la majorité des cas relativement restreint. Nous avons également noté que les calculs des minima peuvent être effectués sur la base du carré des distances, ce qui allège les algorithmes. Enfin, si l'on veut définir l'épaisseur d'une zone de matière, il faut également tenir compte de la direction dans laquelle la mesure est réalisée. La notion de distance pourrait ainsi être paramétrée par une direction spécifiée relati-

vement à une ou plusieurs entités existantes (p. ex. une fibre axiale).

Annexe E

Format de représentation XML

Sommaire

E.1 Schémas XML 212

E.1.1 Représentation des GGF 212

E.1.2 Représentation des GE 216

E.1.3 Représentation des expressions 218

E.2 Exemple d'utilisation 220

E.2.1 Exemple de GGF 220

E.2.2 Exemple de GE 222

Nous présentons dans cette annexe une description de structure *XML Schema* [XSD, 2004] permettant la représentation des connaissances de fabricabilité par l'intermédiaire de fichiers XML. Les schémas de description sont eux-mêmes spécifiés par des fichiers XML respectant une grammaire particulière. Afin de faciliter la lecture du code listé, des commentaires délimitent les différentes parties de chaque fichier. La section E.1 présente les fichiers XSD [159], alors que la section E.2 décrit un exemple d'utilisation de ces structures.

E.1 Schémas XML

Afin de séparer les différents aspects du format de représentation, nous éclatons la définition du schéma dans trois fichiers distincts :

le fichier `ggf.xsd` décrit les graphes des grandeurs de fabrication ;

le fichier `ge.xsd` décrit les graphes des états. Ces derniers sont référencés dans les fichiers de définition des GGF ;

le fichier `expressions.xsd` décrit les règles de formation des expressions logiques et arithmétiques.

Les liens de dépendance existant entre ces trois fichiers sont indiqués sur la figure E.1. Les paragraphes qui suivent sont les listings respectifs de ces trois fichiers. Notons que le format est défini de sorte qu'un GGF est défini dans un fichier XML, et que chaque GE doit également être décrit dans un fichier XML distinct.

E.1.1 Représentation des GGF

Voici le listing du fichier `ggf.xsd` :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:annotation>
  <xsd:documentation xml:lang="fr">
   Schéma XML pour les graphes des grandeurs de fabrication.
  </xsd:documentation>
 </xsd:annotation>

 <!-- Importations -->
 <xsd:include schemaLocation="expressions.xsd"/>
```

159. *XML Schema Description*.

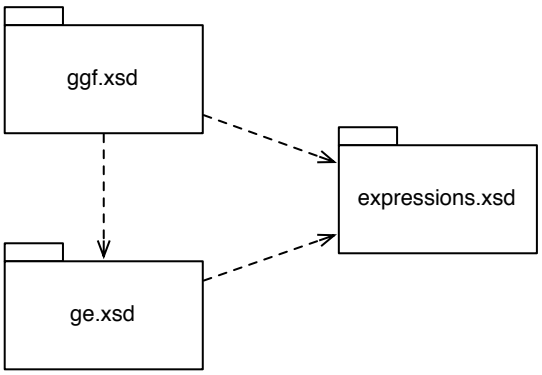


FIGURE E.1: Diagramme de paquets UML représentant les dépendances entre les trois fichiers de description *XML Schema*.

```
<xsd:include schemaLocation="ge.xsd"/>

<xsd:element name="ggf" type="typeGGF">
 <!-- Contraintes -->
 <xsd:key name="clefRegle">
  <xsd:selector xpath="./regle"/>
  <xsd:field xpath="@nom"/>
 </xsd:key>
 <xsd:key name="clefGF">
  <xsd:selector xpath="./gf"/>
  <xsd:field xpath="@nom"/>
 </xsd:key>
 <xsd:keyref name="refGF" refer="clefGF">
  <xsd:selector xpath="./regle/gf/gf"/>
  <xsd:field xpath="@nom"/>
 </xsd:keyref>
 <xsd:key name="clefVF">
  <xsd:selector xpath="./vf"/>
  <xsd:field xpath="@nom"/>
 </xsd:key>
 <xsd:keyref name="refVF" refer="clefVF">
  <xsd:selector xpath="./regle/gf/vf"/>
  <xsd:field xpath="@nom"/>
 </xsd:keyref>
 <xsd:key name="clefEtiquette">
  <xsd:selector xpath="./etiquette"/>
  <xsd:field xpath="@nom"/>
 </xsd:key>
 <xsd:keyref name="refEtiquette" refer="clefEtiquette">
  <xsd:selector xpath="./regle/gf/vf/reference"/>
  <xsd:field xpath="@etiquette"/>
 </xsd:keyref>
```

```
</xsd:element>

<!-- GGF -->
<xsd:complexType name="typeGGF">
 <xsd:sequence>
  <xsd:element name="regle" type="typeRegleAbs" minOccurs="1"
      maxOccurs="unbounded"/>
  <xsd:element name="gf" type="typeGFAbs" minOccurs="1"
      maxOccurs="unbounded"/>
  <xsd:element name="vf" type="typeVFAbs" minOccurs="1"
      maxOccurs="unbounded"/>
  <xsd:element name="etiquette" minOccurs="1" maxOccurs="unbounded">
   <xsd:complexType>
    <xsd:attribute name="nom" type="xsd:string" use="required"/>
    <xsd:attribute name="rattachement" type="typeED" use="required"/>
   </xsd:complexType>
  </xsd:element>
 </xsd:sequence>
 <xsd:attribute name="nom" type="xsd:string" use="optional"/>
</xsd:complexType>

<!-- Règle abstraite -->
<xsd:complexType name="typeRegleAbs">
 <xsd:sequence>
  <xsd:element name="gf" type="videAvecNom" minOccurs="1"
      maxOccurs="unbounded"/>
  <xsd:element name="ge" type="videAvecNom"/>
 </xsd:sequence>
 <xsd:attribute name="nom" type="xsd:string" use="required"/>
</xsd:complexType>

<!-- GF abstraite -->
<xsd:complexType name="typeGFAbs">
 <xsd:sequence>
  <xsd:element name="vf" type="videAvecNom" minOccurs="1"
      maxOccurs="unbounded"/>
  <xsd:element name="calcul" type="typeExprArithm"/>
 </xsd:sequence>
 <xsd:attribute name="nom" type="xsd:string" use="required"/>
</xsd:complexType>

<!-- VF abstraite -->
<xsd:complexType name="typeVFAbs">
 <xsd:sequence>
  <xsd:element name="reference" minOccurs="1" maxOccurs="unbounded">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="configuration" type="typePropagation"
         minOccurs="4" maxOccurs="4"/>
```

```
      </xsd:sequence>
      <xsd:attribute name="etiquette" type="xsd:string"/>
     </xsd:complexType>
     <!-- contraintes -->
     <xsd:unique name="uniqOperateur"> <!-- chaque opérateur n'apparaît
        qu'une seule fois -->
      <xsd:selector xpath="./configuration"/>
      <xsd:field xpath="@operateur"/>
     </xsd:unique>
    </xsd:element>
    <xsd:element name="expression" type="typeExprArithm"/>
   </xsd:sequence>
   <xsd:attribute name="nom" type="xsd:string" use="required"/>
   <xsd:attribute name="mesure" type="typeMesure"/>
</xsd:complexType>

<!-- typeED -->
<xsd:simpleType name="typeED">
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="fibre_axiale"/>
  <xsd:enumeration value="fibre_radiale"/>
  <xsd:enumeration value="contour_caracteristique"/>
  <xsd:enumeration value="ligne_caracteristique"/>
  <xsd:enumeration value="face"/>
 </xsd:restriction>
</xsd:simpleType>

<!-- typePropagation -->
<xsd:complexType name="typePropagation">
 <xsd:attribute name="operateur" type="typeOperateur" use="required"/>
 <xsd:attribute name="propagation" type="typeActPropa" use="required"/>
</xsd:complexType>

<!-- typeOperateur -->
<xsd:simpleType name="typeOperateur">
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="d"/>
  <xsd:enumeration value="sd"/>
  <xsd:enumeration value="f"/>
  <xsd:enumeration value="sf"/>
 </xsd:restriction>
</xsd:simpleType>

<!-- typeActPropa -->
<xsd:simpleType name="typeActPropa">
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="effacement"/>
  <xsd:enumeration value="conservation"/>
  <xsd:enumeration value="adaptation"/>
```

```
  </xsd:restriction>
 </xsd:simpleType>

 <!-- typeMesure -->
 <!-- Remarque : le nbre et le type des paramètres auxquels s'appliquent
     les mesures sont ici implicites -->
 <xsd:simpleType name="typeMesure">
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="distance"/>
   <xsd:enumeration value="longueur"/>
   <xsd:enumeration value="surface"/>
   <xsd:enumeration value="courbure"/>
   <xsd:enumeration value="angle"/>
   <xsd:enumeration value="distance_min"/>
   <xsd:enumeration value="distance_max"/>
  </xsd:restriction>
 </xsd:simpleType>

</xsd:schema>
```

E.1.2 Représentation des GE

Voici le listing du fichier `ge.xsd` :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:annotation>
  <xsd:documentation xml:lang="fr">
   Schéma XML pour les graphes des états.
  </xsd:documentation>
 </xsd:annotation>

 <!-- Importations -->
 <xsd:include schemaLocation="expressions.xsd"/>

 <xsd:element name="ge" type="typeGE">
  <!-- Contraintes référentielles -->
  <xsd:key name="clefEtat">
   <xsd:selector xpath="./etat"/>
   <xsd:field xpath="@nom"/>
  </xsd:key>
  <xsd:keyref name="refEtatOrig" refer="clefEtat">
   <xsd:selector xpath="./transition"/>
   <xsd:field xpath="@orig"/>
  </xsd:keyref>
  <xsd:keyref name="refEtatDest" refer="clefEtat">
   <xsd:selector xpath="./transition"/>
```

```
    <xsd:field xpath="@dest"/>
   </xsd:keyref>
</xsd:element>

<!-- GE -->
<xsd:complexType name="typeGE">
 <xsd:sequence>
  <xsd:element name="etat" type="typeEtat" minOccurs="2"
      maxOccurs="unbounded"/>
  <xsd:element name="transition" type="typeTransition" minOccurs="1"
      maxOccurs="unbounded"/>
 </xsd:sequence>
 <xsd:attribute name="nom" type="xsd:string" use="optional"/>
</xsd:complexType>

<!-- États -->
<xsd:complexType name="typeEtat">
 <xsd:sequence>
  <xsd:element name="description" minOccurs="0">
   <xsd:complexType mixed="true">
    <xsd:all>
     <xsd:element name="vf" type="xsd:string"/>
     <xsd:element name="gf" type="xsd:string"/>
    </xsd:all>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="action" type="typeAction" minOccurs="0"/>
 </xsd:sequence>
 <xsd:attribute name="nom" type="xsd:string" use="required"/>
 <xsd:attribute name="type" use="required">
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="non_fabricable"/>
    <xsd:enumeration value="difficilement_fabricable"/>
    <xsd:enumeration value="fabricable"/>
    <xsd:enumeration value="surcoût"/>
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:attribute>
</xsd:complexType>

<!-- Action -->
<xsd:complexType name="typeAction">
 <xsd:attribute name="nom" type="xsd:string"/>
 <xsd:attribute name="libelle" type="xsd:string"/>
 <xsd:attribute name="type" use="required">
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="appelRoutine"/>
```

```
      <xsd:enumeration value="EGN"/>
     </xsd:restriction>
    </xsd:simpleType>
   </xsd:attribute>
  </xsd:complexType>

  <!-- Transitions -->
  <xsd:complexType name="typeTransition">
   <xsd:all>
    <xsd:element name="expression" type="typeExprLog"/>
   </xsd:all>
   <xsd:attribute name="nom" type="xsd:string" use="optional"/>
   <xsd:attribute name="orig" type="xsd:string" use="required"/>
   <xsd:attribute name="dest" type="xsd:string" use="required"/>
  </xsd:complexType>

</xsd:schema>
```

E.1.3 Représentation des expressions

Voici le listing du fichier `expressions.xsd` :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xhtml="http://www.w3.org/1999/xhtml">

 <xsd:annotation>
  <xsd:documentation xml:lang="fr">
   Schéma XML pour les expressions logiques et arithmétiques.
  </xsd:documentation>
 </xsd:annotation>

 <xsd:element name="exprLog" type="typeExprLog"/>

 <!-- Expressions logiques -->
 <xsd:complexType name="typeExprLog">
  <xsd:choice>
   <xsd:element name="neg" type="typeExprLog"/>
   <xsd:element name="et" type="typeExprLogBin"/>
   <xsd:element name="ou" type="typeExprLogBin"/>
   <xsd:element name="cmp" type="typeExprComp"/>
  </xsd:choice>
 </xsd:complexType>
 <xsd:complexType name="typeExprLogBin">
  <xsd:sequence>
   <xsd:element name="gauche" type="typeExprLog"/>
   <xsd:element name="droite" type="typeExprLog"/>
  </xsd:sequence>
```

```
</xsd:complexType>

<!-- Expressions de comparaison -->
<xsd:complexType name="typeExprComp">
 <xsd:sequence>
  <xsd:element name="gauche" type="typeExprArithm"/>
  <xsd:element name="droite" type="typeExprArithm"/>
 </xsd:sequence>
 <xsd:attribute name="op" type="typeOpComp"/>
</xsd:complexType>

<!-- Expressions arithmétiques -->
<xsd:complexType name="typeExprArithm">
 <xsd:choice>
  <xsd:element name="opUn" type="typeExprArithmUn"/>
  <xsd:element name="opBin" type="typeExprArithmBin"/>
  <xsd:element name="cst">
   <xsd:complexType>
    <xsd:attribute name="valeur" type="xsd:decimal" use="required"/>
    <xsd:attribute name="unite" type="xsd:string" use="optional"/>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="var" type="videAvecNom"/> <!-- utilisé uniquement
     pour les expressions liées aux mesures -->
  <xsd:element name="gf" type="videAvecNom"/>
  <xsd:element name="vf" type="videAvecNom"/>
 </xsd:choice>
</xsd:complexType>
<xsd:complexType name="typeExprArithmUn">
 <xsd:complexContent>
  <xsd:extension base="typeExprArithm">
   <xsd:attribute name="op" type="typeOpArithmUn"/>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="typeExprArithmBin">
 <xsd:sequence>
  <xsd:element name="gauche" type="typeExprArithm"/>
  <xsd:element name="droite" type="typeExprArithm"/>
 </xsd:sequence>
 <xsd:attribute name="op" type="typeOpArithmBin"/>
</xsd:complexType>

<!-- Opérateurs -->
<xsd:simpleType name="typeOpComp">
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="egal"/>
  <xsd:enumeration value="diff"/>
  <xsd:enumeration value="inf"/>
```

```
    <xsd:enumeration value="inf_egal"/>
    <xsd:enumeration value="sup_egal"/>
    <xsd:enumeration value="sup"/>
   </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="typeOpArithmUn">
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="racine"/>
    <xsd:enumeration value="ln"/>
   </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="typeOpArithmBin">
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="+"/>
    <xsd:enumeration value="-"/>
    <xsd:enumeration value="*"/>
    <xsd:enumeration value="/"/>
   </xsd:restriction>
  </xsd:simpleType>

  <!-- Balises ayant un unique attribut "nom" -->
  <xsd:complexType name="videAvecNom">
   <xsd:attribute name="nom" type="xsd:string"/>
  </xsd:complexType>

</xsd:schema>
```

E.2 Exemple d'utilisation

Dans cette section, nous présentons un exemple d'utilisation des fichiers *XML Schema* présentés en section E.1. Cet exemple consiste en deux fichiers :

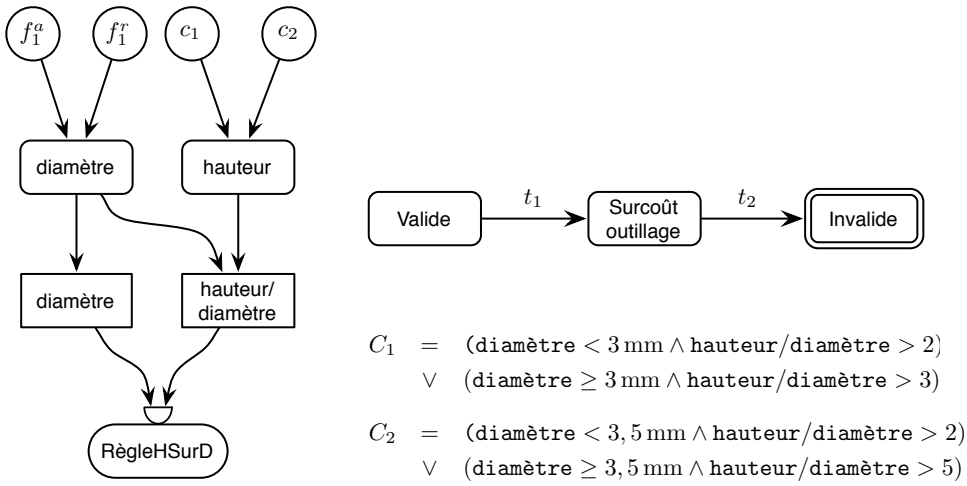
- un premier, nommé `ggf1.xml`, pour la description du GGF de la pièce. Afin de faciliter la lecture, ce graphe ne contient qu'une unique règle, nommée `RègleHSurD` ;
- un second, nommé `ge1.xml`, pour la définition d'un GE associé à la règle `RègleHSurD`.

La figure E.2 représente les deux graphes en utilisant le formalisme graphique introduit au chapitre 3. Les paragraphes suivants sont les listings des deux fichiers correspondants.

E.2.1 Exemple de GGF

Voici le listing du fichier `ggf1.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>
```



(a) Graphe des grandeurs (b) Graphe des états

FIGURE E.2: Représentation graphique des graphes de l'exemple.

```
<ggf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="ggf.xsd" nom="Exemple de GGF">

 <!-- Règles -->
 <regle nom="regleHSurD">
  <gf nom="diamètre"/>
  <gf nom="hauteur/diamètre"/>
  <ge nom="ge1.xml"/>
 </regle>

 <!-- GF -->
 <gf nom="diamètre">
  <vf nom="diamètre"/>
  <calcul><vf nom="diamètre"/></calcul>
 </gf>
 <gf nom="hauteur/diamètre">
  <vf nom="hauteur"/>
  <vf nom="diamètre"/>
  <calcul><opBin op="/"><gauche><vf nom="hauteur"/></gauche><droite><vf
      nom="diamètre"/></droite></opBin></calcul>
 </gf>

 <!-- VF -->
 <vf nom="hauteur" mesure="distance_min">
  <reference etiquette="c1">
    <configuration operateur="d" propagation="conservation"/>
    <configuration operateur="sd" propagation="adaptation"/>
    <configuration operateur="f" propagation="adaptation"/>
    <configuration operateur="sf" propagation="adaptation"/>
```

```
  </reference>
  <reference etiquette="c2">
    <configuration operateur="d" propagation="conservation"/>
    <configuration operateur="sd" propagation="adaptation"/>
    <configuration operateur="f" propagation="adaptation"/>
    <configuration operateur="sf" propagation="adaptation"/>
  </reference>
  <expression><var nom="mesure"/></expression>
 </vf>
 <vf nom="diamètre" mesure="distance_min">
  <reference etiquette="fa1">
    <configuration operateur="d" propagation="conservation"/>
    <configuration operateur="sd" propagation="adaptation"/>
    <configuration operateur="f" propagation="adaptation"/>
    <configuration operateur="sf" propagation="adaptation"/>
  </reference>
  <reference etiquette="fr1">
    <configuration operateur="d" propagation="conservation"/>
    <configuration operateur="sd" propagation="adaptation"/>
    <configuration operateur="f" propagation="adaptation"/>
    <configuration operateur="sf" propagation="adaptation"/>
  </reference>
  <expression>
   <opBin op="*"><gauche><cst valeur="2"/></gauche><droite><var
       nom="mesure"/></droite></opBin>
  </expression>
 </vf>

 <!-- Étiquettes -->
 <etiquette nom="fa1" rattachement="fibre_axiale"/>
 <etiquette nom="fr1" rattachement="fibre_radiale"/>
 <etiquette nom="c1" rattachement="contour_caracteristique_haut"/>
 <etiquette nom="c2" rattachement="contour_caracteristique_bas"/>

</ggf>
```

E.2.2 Exemple de GE

Voici le listing du fichier `ge1.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>

<ge xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="ge.xsd" nom="ge1.xml">

 <!-- États -->
 <etat nom="valide" type="fabricable"/>
 <etat nom="surcoût␣outillage" type="surcoût">
```

```
 <action nom="Confirmation" libelle="Cette␣modification␣conduit␣à␣un␣
     surcoût␣de␣fabrication␣dû␣à␣l'outillage.\nÊtes-vous␣sûr␣de␣vouloir␣
     effectuer␣la␣modification␣?" type="appelRoutine"/>
</etat>
<etat nom="invalide" type="non_fabricable">
 <action nom="Erreur" libelle="Cette␣modification␣rend␣la␣pièce␣non␣
     fabricable␣avec␣le␣procédé.\nL'opération␣va␣être␣annulée."
     type="appelRoutine"></action>
</etat>

<!-- Transitions -->
<transition nom="t1" orig="valide" dest="surcoût␣outillage">
 <expression>
  <ou>
   <gauche>
    <et>
     <gauche>
      <cmp op="inf">
       <gauche><gf nom="diamètre"/></gauche>
       <droite><cst valeur="3" unite="mm"/></droite>
      </cmp>
     </gauche>
     <droite>
      <cmp op="sup">
       <gauche><gf nom="hauteur/diamètre"/></gauche>
       <droite><cst valeur="2"/></droite>
      </cmp>
     </droite>
    </et>
   </gauche>
   <droite>
    <et>
     <gauche>
      <cmp op="sup_egal">
       <gauche><gf nom="diamètre"/></gauche>
       <droite><cst valeur="3" unite="mm"/></droite>
      </cmp>
     </gauche>
     <droite>
      <cmp op="sup">
       <gauche><gf nom="hauteur/diamètre"/></gauche>
       <droite><cst valeur="3"/></droite>
      </cmp>
     </droite>
    </et>
   </droite>
  </ou>
 </expression>
</transition>
```

```
<transition nom="t2" orig="surcoût␣outillage" dest="invalide">
 <expression>
  <ou>
   <gauche>
    <et>
     <gauche>
      <cmp op="inf">
       <gauche><gf nom="diamètre"/></gauche>
       <droite><cst valeur="3.5" unite="mm"/></droite>
      </cmp>
     </gauche>
     <droite>
      <cmp op="sup">
       <gauche><gf nom="hauteur/diamètre"/></gauche>
       <droite><cst valeur="3"/></droite>
      </cmp>
     </droite>
    </et>
   </gauche>
   <droite>
    <et>
     <gauche>
      <cmp op="sup_egal">
       <gauche><gf nom="diamètre"/></gauche>
       <droite><cst valeur="3.5" unite="mm"/></droite>
      </cmp>
     </gauche>
     <droite>
      <cmp op="sup">
       <gauche><gf nom="hauteur/diamètre"/></gauche>
       <droite><cst valeur="5"/></droite>
      </cmp>
     </droite>
    </et>
   </droite>
  </ou>
 </expression>
</transition>

</ge>
```

Sigles et acronymes

Afnor : Association française de normalisation.

AHP : *Analytic Hierarchy Process* – processus de la hiérarchie analytique.

ANN : *Artificial Neural Network* – réseau neuronal artificiel.

ANP : *Analytic Network Process* – processus du réseau analytique.

API : *Application Programming Interface* – interface de programmation d'applications.

AV : Analyse de la Valeur *(value analysis)*.

BCA : *Benefit-Cost Analysis* – analyse « bénéfice-coût ».

BNF : *Backus-Naur Form* – forme de Backus-Naur.

B-Rep : *Boundary Representation* – représentation par les limites.

B-spline : *Basis spline* – spline basique.

CAM : *Computer-Aided Manufacturing*.

CAMM : *Computer-Aided Management and Manufacturing*.

CAD : *Computer-Aided Design*.

CAD/CAM : *Computer-Aided Design and Manufacturing*.

CAO : Conception Assistée par Ordinateur (en anglais, CAD).

CAPP : *Computer-Aided Process Planning* – génération automatique de gammes de fabrication.

CBR : *Case-Based Reasoning* – raisonnement à base de cas.

CCO : Conception à Coût Objectif ou COD.

CFAO : Conception et Fabrication Assistées par Ordinateur (en anglais, CAD/CAM).

COD : Conception à Objectif Désigné ou CCO.

CQD : Coûts, Qualité, Délais (*idem* en anglais : *Cost, Quality, Delay*).

CRM : *Customer Relationship Management*.

CSCW : *Computer-Supported Cooperative Work* – travail collaboratif assité par ordinateur.

CSG : *Constructive Solid Geometry* – géométrie solide constructive.

CSP : *Constraint Satisfaction Problem* – problème de satisfaction de contraintes.

DAO : Dessin Assisté par Ordinateur (en anglais, CAD).

DBD : *Decision-Based Design* – conception basée sur la décision.

DFA : *Design for Assembly* – conception en vue de l'assemblage.

DFC : *Design for cost* – conception guidée par le coût.

DFE : *Design for Environment* – écoconception.

DFM : *Design for Manufacture* – conception pour la fabrication.

DFMA : *Design for Manufacture and Assembly* – conception en vue de la fabrication et de l'assemblage.

DFQ : *Design for Quality* – conception orientée vers la qualité.

DFX : *Design for 'X'* – conception orientée ou guidée.

DSG : *Destructive Solid Geometry* – géométrie solide destructive.

DTD : *Document Type Definition* – définition de type de document ; défini par le standard [XML, 2006].

DOM : *Document Object Model* – définition de type de document ; défini par le standard [DOM, 2004].

ED : Élément de Dialogue.

EGN : Expression Grapho-Numérique.

ERP : *Enterprise Ressource Planning*.

FAO : Fabrication Assistée par Ordinateur (en anglais, CAM).

FAS : Face, Arrête, Sommet.

FAST : *Functional Analysis System Technique* – technique d'analyse fonctionnelle systématique.

FBS : *Function, Behaviour, Structure* – fonction, comportement, structure.

FEP : *Functional Evolution Process* – processus d'évolution fonctionnel.

GCSP : *Geometrical Constraint Satisfaction Problem* – problème de satisfaction de contraintes géométriques.

GDT : *General Design Theory* – théorie générale de la conception.

GE : Graphe des États.

GF : Grandeur de Fabrication.

GGF : Graphe des Grandeurs de Fabrication.

GPAO : Gestion de Production Assistée par Ordinateur (en anglais, CAMM ou MRP).

GRAI : Groupe de Recherche en Automatique Intégrée.

GRC : Gestion de la Relation avec la Clientèle (en anglais, CRM).

IA : Intelligence Artificielle.

IHM : Interface Homme-Machine.

ISO : *International Organization for Standardization*.

JDBC : *Java DataBase Connectivity*.

JOGL : *Java Open Graphics Library*.

KBE : *Knowledge-Based Engineering* – ingénierie à base de connaissances.

MCOES : *Manufacturing Cell Operator's Expert System* – système expert de l'opérateur en cellule de fabrication.

MDC : Module De Connaissances dans l'architecture DIJA.

MH : Module d'Historique dans l'architecture DIJA.

MIHM : Module d'Interaction Homme-Machine dans l'architecture DIJA.

MOCN : Machine-Outil à Commande Numérique.

MOF : *Meta-Object Facility* – fonction méta-objet ; défini par le standard [MOF, 2006].

MRP : *Manufacturing Ressource Planning* – planification des ressources de fabrication.

MRSEV : *Material Removal Shape Element Volume* – caractéristique d'usinage par soustraction de volume.

MS : Module Superviseur dans l'architecture DIJA.

MV : Module de Visualisation dans l'architecture DIJA.

NTIC : Nouvelles Technologies de l'Information et de la Communication.

NURBS : *Non-Uniform Rational B-splines* – B-splines rationnelles non-uniformes.

PGI : Progiciel de Gestion Intégré (en anglais, ERP).

PGN : Paramètre Grapho-Numérique.

PLM : *Product Lifecycle Management* – gestion du cycle de vie du produit.

Pro^3 : Produit, Procédé, Processus.

QFD : *Quality Function Deployment* ou *house of quality* – maison de la qualité.

SADT : *Structured Analysis and Design Technique* – technique d'analyse et de conception structurées [160].

SAFE : *Sequential Analysis of Functional Elements* – analyse séquentielle des éléments fonctionnels.

SART : *Structured Analysis for Real Time* – analyse structurée pour le temps réel.

SATT : Surfaces Associées Technologiquement et Topologiquement.

SAX : *Simple API for XML* – interface de programmation simple pour XML [www-SAX].

SBC : Système à Base de Connaissances.

SCM : *Supply Chain Management* – gestion de la chaîne logistique.

SGDT : Système de Gestion de Données Techniques.

SMA : Système Multi-Agents.

SYSFUND : *SYStematization tool of FUNctional knowledge for Design* – outil de systématisation de la connaissance fonctionnelle pour la conception.

TC : Travail Collaboratif (en anglais, CSCW).

TIC : Technologies de l'Information et de la Communication.

TQM : *Total Quality Management* – gestion totale de la qualité.

UDT : *Universal Design Theory* – théorie universelle de la conception.

UML : *Unified Modeling Language* – langage de modélisation objet unifié ; défini par la norme [ISO/IEC 19501, 2005].

VF : Valeur de Fabrication.

160. Connue aussi sous le nom IDEF∅ (*Integration DEfinition for Function modeling*), méthode qui en est en réalité dérivée.

XML : *eXtensible Markup Language* – langage de balisage extensible ; défini par le standard [XML, 2006].

XSD : *XML Schema Description* – description de schéma XML ; défini par le standard [XSD, 2004].

Bibliographie

A

[Aamodt et Plaza, 1994] Agnar AAMODT et Enric PLAZA, « Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches », *AI Communications*, vol. 7, nº 1, p. 39–59, mars 1994. *Cité page 22.*

[Adams, 2006] Vince ADAMS, *How to Manage Finite Element Analysis in the Design Process*, National Agency for Finite Element Methods and Standards (NAFEMS), ISBN 1-874376-12-3, adresse : `www.nafems.org`, avr. 2006. *Cité pages 28 et 57.*

[Agnez, 2002] Yannick AGNEZ, *Caractérisation des limites du frittage de poudre*, Rapport de DESS, IFTS, UTT, 2002. *Cité page 162.*

[Allen, 1984] Thomas J. ALLEN, *Managing the Flow of Technology*, MIT Press, ISBN 0-262-51027-8, jan. 1984. *Cité page 47.*

[Alliot et Schiex, 2002] Jean-Marc ALLIOT et Thomas SCHIEX, Calcul propositionnel et résolution, dans *Intelligence artificielle et informatique théorique*, Jean-Marc ALLIOT, Thomas SCHIEX, Pascal BRISSET et Frédérick GARCIA (coordinateurs), chapitre 6, p. 93–104, Cépaduès, seconde édition, avr. 2002, ISBN 2-85428-578-6. *Cité page 112.*

[Altshuller, 1992] Genrich S. ALTSHULLER, « The History of ARIZ Development », *Journal of TRIZ*, vol. 3, nº 1, 1992. *Cité page 19.*

[Altshuller et Shapiro, 1956] Genrich S. ALTSHULLER et Raphael V. SHAPIRO, « About a Technology of Creativity », *Questions of Psychology*, nº 6, p. 37–49, 1956. *Cité page 19.*

[Anderl et Mendgen, 1996] Reiner ANDERL et Ralf MENDGEN, « Modelling with constraints : theoretical foundation and application », *Computer-Aided Design*, vol. 28, nº 3, p. 155–168, mars 1996. *Cité pages 84 et 85.*

B

[Barnabé et coll., 2003] Franz BARNABÉ, Michel GANIER, Bernard LAFLEUR, René MALOSSE, Vincent MOULIN et Philippe SCHIESSER, *L'éco-conception pour les mécaniciens. Comment concilier conception et environnement pour un développement durable*, ISBN 2-85400-546-5, 2003. *Cité page 41.*

[Beiter et coll., 2000] Kurt A. BEITER, Brent CHELDELIN et Kosuke ISHII, « Assembly quality method : A tool in aid of product strategy, design, and process improvements », dans *Design Engineering Technical Conferences – Design for Manufacturing*, nº DETC2000/DFM-14020, Baltimore, USA, ASME, sept. 2000. *Cité page 58.*

[Bellinger, 2004] Gene BELLINGER, Knowledge management—emerging perspectives, adresse : `http://www.systems-thinking.org/kmgmt/kmgmt.htm`, 2004. *Cité page 51.*

[Ben Amara et coll., 1996] Abdelmajid BEN AMARA, Dominique DENEUX, René SOËNEN et Abdelwaheb DOGUI, « Pré-dimensionnement en conception fonctionnelle », *Revue internationale de CFAO et d'informatique graphique*, vol. 11, nº 1-2, p. 133–147, 1996. *Cité page 15.*

[Blain, 1990] Patrick BLAIN, « CAO et méthodologie de conception », *Techniques de l'ingénieur, traité Génie mécanique*, nº B2810, août 1990. *Cité page 6.*

[Boothroyd et coll., 2001] Geoffrey BOOTHROYD, Peter DEWHURST et Winston KNIGHT, *Product Design for Manufacture & Assembly, Revised & Expanded*, Marcel Dekker, second édition, ISBN 0-8247-0584-X, sept. 2001. *Cité pages 57, 60, 61, 62, 65, 67 et 75.*

[Borg et MacCallum, 1995] Jonathan BORG et Ken J. MACCALLUM, « A HyperCAD Expert System for Plastic Product design », dans *International Conference on Computer Integrated Manufacturing*, vol. 1, p. 295–302, Singapore, juil. 1995. *Cité page 21.*

[Bouleau, 2002] Nicolas BOULEAU, *Enquête sur le concept de modèle*, chap. La modélisation et les sciences de l'ingénieur, p. 101–119, collection Science, histoire et société, Presses Universitaires de France, ISBN 2-13-052619-5, avr. 2002. *Cité page 29.*

[Boyle et coll., 2004] Iain M. BOYLE, Kevin RONG et David C. BROWN, « CAFixD : A case-based reasoning fixture design method. Framework and indexing mechanisms. », dans *Design Engineering Technical Conferences – Computers and Information in Engineering*, nº DETC2004/CIE-57689, Salt Lake City, USA, ASME, sept. 2004. *Cité page 20.*

[Braha et Maimon, 1998] Dan BRAHA et Oded MAIMON, *A Mathematical Theory of Design : Foundations, Algorithms and Applications*, collection Applied Optimization, Kluwer Academic Publishers, ISBN 0-7923-5079-0, juin 1998. *Cité page 17.*

[Bralla, 1999] James G. BRALLA, *Design for Manufacturability Handbook*, McGraw-Hill, second édition, ISBN 0-07-007139-X, 1999. *Cité pages 67 et 75.*

[Brazier et coll., 2001] Frances M. T. BRAZIER, Lilia V. MOSHKINA et Niek J.E. WIJNGAARDS, « Knowledge level model of an individual designer as an agent in collaborative distributed design », *Artificial Intelligence in Engineering*, vol. 15, n° 2, p. 137–152, avr. 2001. *Cité page 24.*

[Bronsvoort et Noort, 2004] Willem F. BRONSVOORT et Alexander NOORT, « Multiple-view feature modelling for integral product development », *Computer-Aided Design*, vol. 36, n° 10, p. 929–946, sept. 2004. *Cité page 88.*

[Brosse et coll., 2006] Étienne BROSSE, Yvon GARDAN et Estelle PERRIN, « DIJA project : Forging preforms design using trade knowledge », dans *International Conference on Geometric Modeling and Imaging (GMAI'06)*, p. 244–249, London, UK, IEEE Computer Society, 2006, ISBN 0-7695-2604-7. *Cité page 144.*

[Brosse et coll., 2007] Étienne BROSSE, Yvon GARDAN et Estelle PERRIN, « Forging-dedicated cad : how getting forging preform by using trade knowledge and feature reorganization », dans *4th International Conference on Computer Graphics, Imaging and Visualization (CGIV'07)*, p. 211–216, Bangkok, Thailand, IEEE Computer Society, août 2007. *Cité page 144.*

[Brown, 1998] David C. BROWN, Revision of 1993 Article on Intelligent Computer-Aided Design, dans *Encyclopedia of Computer Science and Technology*, James G. WILLIAMS et Allen KENT (coordinateurs), vol. 28, p. 153–166, Marcel Dekker, sept. 1998. *Cité pages 9 et 24.*

[Bureau International des Poids et Mesures, 2006] BUREAU INTERNATIONAL DES POIDS ET MESURES (coordinateur), *Le système international d'unités*, STEDI Media, huitième édition, ISBN 92-822-2213-6, adresse : `http://www.bipm.org`, 2006. *Cité pages 63, 64 et 113.*

[Bytheway, 1965] Charles W. BYTHEWAY, « Basic Function Determination Technique », dans *Fifth National Meeting, Society Of American Value Engineers*, p. 21–23, avr. 1965. *Cité page 13.*

C

[Chan et Salustri, 2003] Vincent CHAN et Filippo A. SALUSTRI, Product cost estimation, adresse : `http://deed.ryerson.ca/~fil/t/dfmcost.html`, 2003, `http://deed.ryerson.ca/~fil/t/dfmcost.html`. *Cité page 62.*

[Chau et Albermani, 2003] Kwok-Wing CHAU et Faris ALBERMANI, « A coupled knowledge-based expert system for design of liquid-retaining structures », *Automation in Construction*, vol. 12, n° 5, p. 589–602, sept. 2003. *Cité page 21.*

[Chen et Doumeingts, 2000] David CHEN et Guy DOUMEINGTS, « Developing a theory of design – a tentative approach using mathematics », *Studies in Informatics and Control*, vol. 11, n° 1, mars 2000. *Cité page 14.*

[Chen et coll., 2002] Li CHEN, Pu JIAN et Xiankui WANG, « A general model for machinable features and its application to machinability evaluation of mechanical parts », *Computer-Aided Design*, vol. 34, n° 3, p. 239–249, mars 2002. *Cité page 74.*

[Chen, 2004] Xiao Chuan CHEN, « The relationship between design for environment (dfe) and design for cost (dfc) », dans *World Engineers' Convention*, vol. G, p. 293–296, Shangai, China, nov. 2004. *Cité page 58.*

[Chen et coll., 2005] Zheng Ming CHEN, Shu Ming GAO et Wei Dong LI, « An approach to incremental feature model conversion », *International Journal of Advanced Manufacturing Technology*, août 2005. *Cité page 83.*

[Chou, 2004] Chi-Wu CHOU, *Evolutionary Cluster Costing for Weapon System Early Design*, Thèse de doctorat, School of Information Technology and Electrical Engineering, University of New South Wales, Australia, oct. 2004. *Cité page 77.*

[Chung et coll., 2000] Jack C.H. CHUNG, Teng-Shang HWANG, Chien-Tai WU, Yu JIANG, Jia-Yi WANG, Yong BAI et Hongliu ZOU, « Framework for integrated mechanical design automation », *Computer-Aided Design*, vol. 32, n° 5-6, p. 355–365, avr. 2000. *Cité pages 23, 39 et 84.*

[CIMdata, Inc., 2002] CIMdata, Inc., Product Lifecycle Management – Empowering the Future of Business, 2002, `http://www.cimdata.com`. *Cité page 43.*

[Ciuciu et coll., 2008] Ioana-Georgiana CIUCIU, Frédéric DANESI, Yvon GARDAN et Estelle

PERRIN, « Surface deformation on a discrete model for a cad system », dans *24th European Workshop on Computational Geometry*, p. 185–188, mars 2008. *Cité page 144.*

[Clément et coll., 1996] André CLÉMENT, Alain RIVIÈRE et Philippe SERRÉ, « Les SATT : un modèle déclaratif commun pour le paramétrage, le tolérancement et l'assemblage », *Revue internationale de CFAO et d'informatique graphique*, vol. 11, n° 1-2, p. 149–164, 1996. *Cité page 15.*

[Cochran et Reynal, 1996] David S. COCHRAN et Vicente A. REYNAL, « Axiomatic design of manufacturing systems », Rapport technique RP96-05-14, MIT, Cambridge, USA, nov. 1996. *Cité page 19.*

[Craveur et Marceau, 2001] Jean-Charles CRAVEUR et Dominique MARCEAU, *De la CAO au calcul*, collection L'usine nouvelle, Conception, Dunod, ISBN 2-10-005220-9, nov. 2001. *Cité page 57.*

[Cutkosky et Tenenbaum, 1990] Mark R. CUTKOSKY et Jay M. TENENBAUM, « A methodology and computational framework for concurrent product and process design », *Mechanism and Machine Theory*, vol. 25, n° 3, p. 365–381, 1990. *Cité pages 72 et 81.*

D

[Dacia, 2004] vol. Dacia, Logan, un programme de conquête internationale pour la croissance rentable du groupe Renault, Communiqué de presse de DACIA, groupe Renault, juin 2004. *Cité page 26.*

[Danesi, 2002] Frédéric DANESI, *Contribution à la définition et à l'intégration de notions intuitives dans le dialogue Homme-Machine en CAO*, Thèse de doctorat, Institut de Formation Technique Supérieur, Université de Reims Champagne-Ardenne, nov. 2002. *Cité pages 37, 38, 39, 96, 144 et 150.*

[Danesi et coll., 2002] Frédéric DANESI, Laurent DENIS, Yvon GARDAN et Estelle PERRIN, « Basic Components of the DIJA Project », dans *Solid Modeling*, AMERICAN COMPUTER MACHINERY (coordinateur), p. 234–240, juin 2002. *Cité page 38.*

[Danesi et coll., 2003] Frédéric DANESI, Nicolas GARDAN, Yvon GARDAN et Estelle PERRIN, « Synthetic Modelling System Architecture », dans *International Conference on Geometric Modeling and Graphics, GMAG*, p. 174–179, Londres, juil. 2003. *Cité pages 99 et 100.*

[Danesi et coll., 2007] Frédéric DANESI, Nicolas GARDAN, Yvon GARDAN et Mickaël REIMERINGER, « P^4LM : A methodology for product lifecycle management », *Computers in Industry*, vol. 59, n° 2-3, p. 304–317, mars 2007, ISSN 0166-3615. *Cité pages 104 et 144.*

[Decreuse et Feschotte, 1998] Christophe DECREUSE et Daniel FESCHOTTE, « Ingénierie simultannée », *Techniques de l'ingénieur, traité Génie industriel*, n° A5310, jan. 1998. *Cité pages 42, 44 et 45.*

[Denis, 2003] Laurent DENIS, *Vers une architecture adaptée à un système de CAO utilisable à travers Internet*, Thèse de doctorat, Institut de Formation Technique Supérieur, Université de Reims Champagne-Ardenne, nov. 2003. *Cité pages 96, 100, 101, 103, 118, 144 et 156.*

[Denis et coll., 2004] Laurent DENIS, Yvon GARDAN et Estelle PERRIN, « A framework for a distributed CAD system », *Computer-Aided Design*, vol. 9, n° 36, p. 761–773, août 2004. *Cité page 38.*

[Desai et Mital, 2006] Anoop DESAI et Anil MITAL, « Design for maintenance : basic concepts and review of literature », *International Journal of Product Development*, vol. 3, n° 1, p. 77–121, 2006. *Cité page 58.*

[Dissinger et Magrab, 1996] Thomas E. DISSINGER et Edward B. MAGRAB, « Geometric reasoning for manufacturability evaluation – application to powder metallurgy », *Computer-Aided Design*, vol. 28, n° 10, p. 783–794, 1996. *Cité page 74.*

[Drogoul, 2005] Alexis DROGOUL, Les systèmes multi-agents, dans *L'intelligence*, Jacques LAUTREY et Jean-François RICHARD (coordinateurs), collection Traité des sciences cognitives, chapitre 7, Hermes science publications, mars 2005, ISBN 2-7462-1088-6. *Cité page 24.*

[Druon, 2001] Maurice DRUON (coordinateur), *Dictionnaire de l'Académie française*, vol. 1, Librairie Arthème Fayard, neuvième édition, 2001. *Cité pages 10, 18 et 35.*

[Duverlie et coll., 1999] Pascal DUVERLIE, Jean-Marie CASTELAIN et Thibault FARINEAU, « Estimation des coûts en production mécanique », *Techniques de l'ingénieur, traité Génie mécanique*, n° BM7076, juil. 1999. *Cité page 71.*

E

[Economist Intelligence Unit, 2005] ECONOMIST INTELLIGENCE UNIT, Know how managing knowledge for competitive advantage, juin 2005. *Cité page 51.*

[Edwards, 2002] K. L. EDWARDS, « Towards more strategic product design for manufacture and assembly : priorities for concurrent engineering », *Materials and Design*, n° 23, p. 651–656, mai 2002. *Cité page 57.*

[Engelbart, 1962] Douglas C. ENGELBART, « Augmenting Human Intellect : A Conceptual Framework », Summary Report AFOSR-3233, SRI Project No. 3578, Stanford Research Institute, Menlo Park, USA, oct. 1962. *Cité page 46.*

[Esposito, 2002] Nicolas ESPOSITO, *Cooper : une architecture d'égal à égal pour la conception collaborative*, Thèse de doctorat, INPG, 9nov. 2002. *Cité page 49.*

[Eudier, 1994] Michel EUDIER, « Fabrication des produits frittés », *Techniques de l'ingénieur, traité Matériaux métalliques*, avr. 1994. *Cité page 162.*

[Eynard, 2005] Benoît EYNARD, *Gestion du cycle de vie des produits et dynamique des connaissances industrielles en conception intégrée*, Mémoire d'habilitation à diriger les recherches, Université de Technologie de Troyes, nov. 2005. *Cité page 53.*

F

[Faltings et Sun, 1996] Boi FALTINGS et Kun SUN, « FAMING : supporting innovative mechanism shape design », *Computer-Aided Design*, vol. 28, n° 3, p. 207–216, mars 1996. *Cité page 22.*

[Farineau et coll., 2001] Thibault FARINEAU, Besoa RABENASOLO, Jean-Marie CASTELAIN, Yves MEYER et Pascal DUVERLIE, « Use of parametric models in an economic evaluation step during the design phase », *International Journal of Advanced Manufacturing Technology*, vol. 17, n° 2, p. 79–86, jan. 2001. *Cité pages 77 et 78.*

[Feng et Kusiak, 1995] Chang-Xue FENG et Andrew KUSIAK, « Constraint-based design of parts », *Computer-Aided Design*, vol. 27, n° 5, p. 343–352, avr. 1995. *Cité page 85.*

[Feng et coll., 1996] Chang-Xue (Jack) FENG, Chun-Che HUANG, Andrew KUSIAK et Pei-Gen LI, « Representation of functions and features in detail design », *Computer-Aided Design*, vol. 28, n° 12, p. 961–971, 1996. *Cité page 35.*

[Ficko et coll., 2005] Mirko FICKO, Igor DRSTVENŠEK, Miran BREZOČNIK, Jože BALIČ et Boštjan VAUPOTIČ, « Prediction of total manufacturing costs for stamping tool on the basis of cad-model of finished product », *Journal of Materials Processing Technology*, vol. 164-165, p. 1327–1335, mai 2005. *Cité page 77.*

[Fortin et Huet, 2007] Clement FORTIN et Gregory HUET, « Manufacturing process management : iterative synchronisation of engineering data with manufacturing realities », *International Journal of Product Development*, vol. 4, nº 3-4, p. 280–295, 2007. *Cité page 61.*

[Fuh et Li, 2005] Jerry Ying Hsi FUH et Wei Dong LI, « Advances in collaborative CAD : the state-of-the-art », *Computer-Aided Design*, vol. 37, nº 5, p. 571–581, avr. 2005. *Cité page 50.*

G

[Gardan, 2005] Nicolas GARDAN, *Proposition d'une méthodologie de travail collaboratif : concepts et applications*, Thèse de doctorat, IFTS, Charleville-Mézières, France, 23nov. 2005. *Cité pages 28, 48, 91, 96, 104, 106 et 144.*

[Gardan et Gardan, 2003] Nicolas GARDAN et Yvon GARDAN, « An application of knowledge based modelling using scripts », *Expert Systems with Applications*, vol. 25, nº 4, p. 555–568, 2003. *Cité pages 105 et 144.*

[Gardan et coll., 2007] Nicolas GARDAN, Mickaël REIMERINGER, Frédéric DANESI et Yvon GARDAN, « Graphonumerical parameters based on scenarios and ontologies », dans *International Conference on Product Lifecycle Management PLM'07*, Inderscience publishers, 2007. *Cité page 144.*

[Gardan, 2002] Yvon GARDAN, « CAO : modélisation géométrique », *Techniques de l'ingénieur, traité Informatique*, nº H3750, août 2002. *Cité pages 30, 32, 33 et 178.*

[Gardan, 2003] Yvon GARDAN, « CAO : vers la modélisation fonctionnelle », *Techniques de l'ingénieur, traité Informatique*, nº H3752, août 2003. *Cité pages 16, 17, 26, 34, 81 et 84.*

[Gardan, 1982] Yvon GARDAN, *Éléments méthodologiques pour la réalisation de systèmes de CFAO et leur introduction dans les entreprises*, Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, 1982. *Cité page 104.*

[Gardan, 1991] Yvon GARDAN, *La CFAO – introduction, techniques, et mise en œuvre*, collection Traité des nouvelles technologies – Assistance par ordinateur (XAO), Hermès, troisième édition, ISBN 2-86601-115-5, déc. 1991. *Cité pages 29, 31, 33 et 34.*

[Gero et Maher, 1990] John S. GERO et Mary Lou MAHER, « Guest editorial – artificial intelligence in design », *AI magazine, Special issue on AI-based design systems*, vol. 11, nº 4, p. 7, 1990. *Cité page 20.*

[Girard, 2001] Philippe GIRARD, « Analyse des décisions en conception », *Techniques de l'ingénieur, traité L'entreprise industrielle*, nº AG2220, oct. 2001. *Cité pages 8, 9 et 23.*

[Gomes et coll., 2005] Samuel GOMES, Patrick SERRAFERO, Davy MONTICOLO et Benoît EYNARD, « Extracting engineering knowledge from plm systems : An experimental approach », dans *International Conference on Product Lifecycle Management PLM'05. Emerging solutions and challenges for Global Networked Enterprise*, ABDELAZIZ BOURAS, BALAN GURUMOORTHY et RACHURI SUDARSAN (coordinateurs), p. 33–43, Lyon, France, Inderscience publishers, 2005. *Cité page 43.*

[Grayer, 1976] Alan R. GRAYER, *A computer Link Between Design and Manufacturing*, Thèse de doctorat, University of Cambridge, sept. 1976. *Cité page 35.*

[Grecu et Brown, 1996] Dan L. GRECU et David C. BROWN, « Learning by single function agents during spring Design », dans *Artificial Intelligence in Design*, John S. GERO et Fay SUDWEEKS (coordinateurs), p. 409–428, Boston, USA, Kluwer Academic Publishers, juin 1996. *Cité page 24.*

[Gupta et coll., 1995] Satyandra K. GUPTA, William C. REGLI, Diganta DAS et Dana S. NAU, « Current Trends and Future Challenges in Automated Manufacturability Analysis », Technical Research Report TR-95-16, Institute for Systems Research, 1995. *Cité pages 59, 65 et 68.*

[Gupta et coll., 1997] Satyandra K. GUPTA, William C. REGLI, Diganta DAS et Dana S. NAU, « Automated Manufacturability Analysis : A Survey », *Research in Engineering Design*, vol. 9, nº 3, p. 168–190, sept. 1997. *Cité pages 60, 71 et 72.*

H

[Han, 1996] Jung Hyun HAN, « Survey of Feature Research », Rapport technique IRIS-96-346, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, USA, 1996. *Cité page 36.*

[Han et Requicha, 1997] Jung Hyun HAN et Aristides A. G. REQUICHA, « Integration of feature based design and feature recognition », *Computer-Aided Design*, vol. 29, nº 5, p. 393–403, mai 1997. *Cité page 82.*

[Han et coll., 2000] Jung Hyun HAN, Michael PRATT et William C. REGLI, « Manufacturing

Feature Recognition from Solid Models : A Status Report », *IEEE Transactions on Robotics and Automation*, vol. 16, nº 6, p. 782–796, déc. 2000. *Cité pages 81, 82 et 84.*

[Haton et Haton, 2000] Jean-Paul HATON et Marie-Christine HATON, « Systèmes à bases de connaissances », *Techniques de l'ingénieur, traité Informatique*, nº H3740, fév. 2000. *Cité pages 20, 21, 24 et 65.*

[Hazelrigg, 1998] George A. HAZELRIGG, « A framework for decision-based engineering design », *Journal of Mechanical Design*, vol. 120, p. 653–658, 1998. *Cité pages 70 et 71.*

[Heschung, 2005] Frédéric HESCHUNG, *Vers une construction et une adaptation automatiques des maillages*, Thèse de doctorat, Institut de Formation Technique Supérieur, Université de Reims Champagne-Ardenne, mai 2005. *Cité pages 96 et 144.*

[Hoffmann et Joan-Arinyo, 2000] Christoph M. HOFFMANN et Robert JOAN-ARINYO, « Distributed maintenance of multiple product views », *Computer-Aided Design*, vol. 32, nº 7, p. 421–431, juin 2000. *Cité page 88.*

[Hoffmann et Joan-Arinyo, 2002] Christoph M. HOFFMANN et Robert JOAN-ARINYO, Parametric Modeling, dans *Handbook of Computer-Aided Geometric Design*, Gerald FARIN, Josef HOSCHEK et Myung-Soo KIM (coordinateurs), chap. 21, Elsevier Science, North Holland, juin 2002, ISBN 0-444-51104-0. *Cité page 39.*

[Hoffmann et Joan-Arinyo, 1998] Christoph M. HOFFMANN et Robert JOAN-ARINYO, « Cad and the product master model », *Computer-Aided Design*, vol. 30, nº 11, p. 905–918, sept. 1998. *Cité page 87.*

[Huang et Mak, 1999] George Q. HUANG et Kai Ling MAK, « Design for manufacture and assembly on the Internet », *Computers in Industry*, nº 38, p. 17–30, 1999. *Cité page 58.*

[Hégron, 2002] Gérard HÉGRON, « Conception architecturale et modélisation déclarative », dans *EVEN workshop – Ingénierie Virtuelle : Outils et Méthodes*, Nantes, France, IRCCyN, nov. 2002. *Cité page 23.*

J

[Jakiela et coll., 2000] Mark J. JAKIELA, Colin CHAPMAN, James DUDA, Adenike ADEWUYA et Kazuhiro SAITOU, « Continuum structural topology design with genetic algorithms », *Computer Methods in Applied Mechanical and Engineering*, vol. 186, nº 2-4, p. 339–356, juin 2000. *Cité page 25.*

[Johansen et coll., 1988] Robert JOHANSEN, Jeff CHARLES, Robert MITTMAN et Paul SAFFO, *Groupware : Computer Support for Business Teams*, collection Communication Technology and Society, The Free Press, ISBN 0-02-916491-5, oct. 1988. *Cité page 47.*

[Johnson-Lenz et Johnson-Lenz, 1980] Peter JOHNSON-LENZ et Trudy JOHNSON-LENZ, « Groupware : the emerging art of orchestrating collective intelligence », dans *World Future Society's First Global Conference on the Future*, Toronto, Canada, juil. 1980. *Cité page 46.*

[Johnson-Lenz et Johnson-Lenz, 1989] Peter JOHNSON-LENZ et Trudy JOHNSON-LENZ, « Humanizing Hyperspace », *In Context, The Ecology Of Media*, nº 23, p. 52–57, 1989. *Cité page 46.*

K

[Keeney et Raiffa, 1976] Ralph L. KEENEY et Howard RAIFFA, *Decisions with Multiple Objectives*, John Wiley & Sons, ISBN 0-471-46510-0, jan. 1976. *Cité page 70.*

[Kesavadas et coll., 2005] Mini P. KESAVADAS, Amit PAYGUDE, Karthik BANDI et T. KESAVADAS, « Development of formal ontology for product design lifecycle », dans *International Conference on Product Lifecycle Management PLM'05. Emerging solutions and challenges for Global Networked Enterprise*, ABDELAZIZ BOURAS, BALAN GURUMOORTHY et RACHURI SUDARSAN (coordinateurs), p. 3–9, Lyon, France, Inderscience publishers, 2005. *Cité page 43.*

[Kim et coll., 2006] Kyoung-Yun KIM, David G. MANLEY et Hyungjeong YANG, « Ontology-based assembly design and information sharing for collaborative product development », *Computer-Aided Design*, vol. 38, nº 12, p. 1233–1250, déc. 2006. *Cité page 43.*

[de Kraker, 1998] Klaas Jannis DE KRAKER, *Feature Conversion for Concurrent Engineering*, Thèse de doctorat, Advanced School for Computing and Imaging, Delft University, Netherlands, jan. 1998. *Cité pages 87 et 88.*

[de Kraker et coll., 1997] Klaas Jannis DE KRAKER, Maurice DOHMEN et Willem F. BRONSVOORT, « Maintaining multiple views in feature modeling », dans *Proceedings of the fourth ACM symposium on Solid modeling and applications (SMA'97)*, ACM SIGGRAPH (coordinateur), p. 123–130, Atlanta, Georgia, USA, ACM Press, 1997, ISBN 0-89791-946-7. *Cité pages 86 et 87.*

[Kuo et coll., 2001] Tsai-C. KUO, Samuel H. HUANG et Hong-C. ZHANG, « Design for manufacture and design for 'X' : concepts, applications and perspectives », *Computers and Industrial Engineering*, n° 41, p. 241–260, 10mai 2001. *Cité pages 56, 57 et 58.*

[Kwaiter et Gaildrat, 1998] Ghassan KWAITER et Véronique GAILDRAT, « Modelling with Constraints : A Bibliographical Survey », dans *International Conference on Information Visualisation*, INFODESIGN (coordinateur), p. 211–220, London, UK, juil. 1998. *Cité page 23.*

L

[Laakko et Mäntylä, 1993] Timo LAAKKO et Martti MÄNTYLÄ, « Feature modelling by incremental feature recognition », *Computer-Aided Design*, vol. 25, n° 8, p. 479–492, août 1993. *Cité page 83.*

[Le Roux et coll., 2001] Olivier LE ROUX, Véronique GAILDRAT et René CAUBET, « Using Constraint Propagation and Domain Reduction for the Generation Phase in Declarative modeling », dans *International Conference on Information VIsualisation*, IEEE COMPUTER SOCIETY PRESS (coordinateur), p. 117–123, London, UK, The printing house, USA, juil. 2001. *Cité page 23.*

[Lee et Kazuhiro, 2002] Byungwoo LEE et Saitou KAZUHIRO, « Design of part family robust-to-production plan variations based on quantitative manufacturability evaluation », *Research in Engineering Design*, vol. 13, n° 4, p. 199–212, 2002. *Cité page 77.*

[Lee et Kim, 1998] Jae Yeol LEE et Kwangsoo KIM, « A feature-based approach to extracting machining features », *Computer-Aided Design*, vol. 30, n° 13, p. 1019–1035, nov. 1998. *Cité page 83.*

[Lee et coll., 2004] Kwang-Rak LEE, Ji-Hyung PARK et Jahng-Hyon PARK, « The analysis, comparison, and redesign of cd-case using the principles of axiomatic design », dans *International Conference on Axiomatic Design*, Seoul, juin 2004. *Cité page 19.*

[Lee, 2005] Sang Hun LEE, « A CAD–CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques », *Computer-Aided Design*, vol. 37, n° 9, p. 941–955, août 2005. *Cité page 87.*

[Lesage, 2002] David LESAGE, *Un modèle dynamique de spécifications d'ingénierie basé sur une approche de géométrie variationnelle*, Thèse de doctorat, Institut National Polytechnique de Grenoble, nov. 2002. *Cité page 39.*

[Levan, 2004] Serge K. LEVAN, *Travail collaboratif sur Internet – concepts, méthodes et pratiques des plateaux projets*, collection Entreprendre Informatique, Vuibert, ISBN 2-7117-4828-6, mai 2004. *Cité pages 45, 46, 47 et 48.*

[Li et coll., 2002] Wei Dong LI, Soh Khim ONG et Andrew Yeh Ching NEE, « Recognizing manufacturing features from a design-by-feature model », *Computer-Aided Design*, vol. 34, nº 11, p. 849–868, sept. 2002. *Cité page 82.*

[Li et coll., 2004] Wei Dong LI, Soh Khim ONG, Jerry Ying Hsi FUH, Yoke San WONG, Yi Qiang LU et Andrew Yeh Ching NEE, « Feature-based design in a distributed and collaborative environment », *Computer-Aided Design*, vol. 36, nº 9, p. 775–797, août 2004. *Cité page 50.*

[Li et coll., 2005] Wei Dong LI, Wen Feng LU, Jerry Ying Hsi FUH et Yoke San WONG, « Collaborative computer-aided design—research and development status », *Computer-Aided Design*, vol. 37, nº 9, p. 931–940, août 2005. *Cité page 50.*

[Liao, 2005] Shu-Hsien LIAO, « Expert system methodologies and applications—a decade review from 1995 to 2004 », *Expert Systems with Applications*, vol. 28, nº 1, p. 93–103, jan. 2005. *Cité page 21.*

[Lin et Shen, 1995] Zone-Ching LIN et Chi-Chih SHEN, « An investigation of an expert system for hydraulic circuit design with learning », *Artificial Intelligence in Engineering*, vol. 9, nº 3, p. 153–165, 1995. *Cité page 21.*

[Liu et coll., 2002] Hong LIU, Mingxi TANG et John Hamilton FRAZER, « Supporting evolution in a multi-agent cooperative design environment », *Advances in Engineering Software*, vol. 33, p. 319–328, 2002. *Cité page 25.*

[Lonchampt, 2004] Pierre LONCHAMPT, *Co-évolution et processus de conception intégrée de produits : Modèle et support de l'activité de conception*, Thèse de doctorat, Institut National Polytechnique de Grenoble, juin 2004. *Cité page 17.*

[Lossack et Grabowski, 2000] Ralf LOSSACK et Hans GRABOWSKI, « The Axiomatic Approach in the Universal Design Theory », dans *International Conference on Axiomatic Design (ICAD2000)*, Cambridge, USA, juin 2000. *Cité page 18.*

[Lucas et coll., 1990] Michel LUCAS, Dominique MARTIN, Philippe MARTIN et Dimitri PLEMENOS, « Le projet ExploFormes : quelques pas vers la modélisation déclarative de formes », *BIGRE*, nº 67, p. 35–49, jan. 1990. *Cité page 23.*

M

[Maculet et Daniel, 2003] Robert MACULET et Marc DANIEL, « Conception, modélisation géométrique et contraintes en CAO : une synthèse », Rapport de Recherche LSIS-2003-005, Laboratoire des Sciences de l'Information et des Systèmes, UMR CNRS 6168, Marseille, France, 2003. *Cité pages 36, 38, 39, 40, 85 et 110.*

[Madhusudan, 2005] Therani MADHUSUDAN, « An agent-based approach for coordinating product design workflows », *Computers in Industry*, vol. 56, nº 3, p. 235–259, avr. 2005. *Cité page 24.*

[Malik et coll., 2002] E. MALIK, Y. GARDAN et E. PERRIN, « Vers une approche plus intuitive des systèmes de C.A.O. », dans *AFIG*, p. 79–88, déc. 2002. *Cité page 144.*

[Martin et Ishii, 1997] Mark V. MARTIN et Kosuke ISHII, « Design for variety : Development of complexity indices and design charts », dans *Design Engineering Technical Conferences – Design for Manufacturing*, nº DETC97/DFM-4359, Sacramento, USA, ASME, sept. 1997. *Cité page 57.*

[Martin et Martin, 1988] Philippe MARTIN et Dominique MARTIN, « An expert system for polyhedra modelling », dans *Eurographics conference*, David A. DUCE et Pierre JANCENE (coordinateurs), Elsevier Science, 1988. *Cité page 23.*

[McAlinden et coll., 1998] Leo P. MCALINDEN, Barry O. FLORIDA-JAMES, Kuo-Ming CHAO, Peter NORMAN, William HILLS et Peter SMITH, « Information and Knowledge Sharing for Distributed Design Agents », dans *Artificial Intelligence in Design*, John S. GERO et Fay SUDWEEKS (coordinateurs), p. 537–556, Lisbon, Portugal, Kluwer Academic Publishers, juil. 1998. *Cité page 24.*

[Medani et Ratchev, 2006] O. MEDANI et S. M. RATCHEV, « A step ap224 agent-based early manufacturability assessment environment using xml », *International Journal of Advanced Manufacturing Technology*, vol. 27, nº 9-10, p. 854–864, fév. 2006. *Cité page 86.*

[Medjdoub et Yannou, 2000] Benachir MEDJDOUB et Bernard YANNOU, « Separating topology and geometry in space planning », *Computer-Aided Design*, vol. 32, nº 1, p. 39–61, jan. 2000. *Cité page 23.*

[Mervyn et coll., 2004] Fathianathan MERVYN, A. Senthil KUMAR, Shung Hwee BOK et Andrew Yeh Ching NEE, « Developing distributed applications for integrated product and process design », *Computer-Aided Design*, vol. 36, nº 8, p. 679–689, juil. 2004. *Cité page 43.*

[Minich et Pallez, 1999] Christian MINICH et Denis PALLEZ, « Vers des outils informatiques d'assistance aux phases amont de la conception – État de l'art », *Revue internationale de CFAO et d'informatique graphique*, vol. 14, n° 2, p. 177–206, déc. 1999. *Cité pages 10 et 11.*

[Minich et coll., 2004] Christian MINICH, Karine MONNIER et Yvon GARDAN, « Vers une sécurisation accrue des échanges cao », dans *Conférence internationale MICAD*, HERMÈS (coordinateur), Paris, France, 2004. *Cité page 144.*

[Minsky, 1988] Marvin MINSKY, *The Society of Mind*, Simon & Schuster, New York, USA, ISBN 0-671-65713-5, mars 1988. *Cité page 23.*

[Morgenstern et von Neumann, 1947] Oskar MORGENSTERN et John VON NEUMANN, *Theory of Games and Economic Behavior*, Princeton University Press, ISBN 0-691-00362-9, 1947. *Cité page 70.*

[Mori et Cutkosky, 1998] Toshiki MORI et Mark R. CUTKOSKY, « Agent-Based Collaborative Design of Part in Assembly », dans *Design Engineering Technical Conferences – Computers and Information in Engineering*, n° DETC98/CIE-5697, Atlanta, USA, ASME, sept. 1998. *Cité page 50.*

[Myung et Han, 2001] Sehyun MYUNG et Soonhung HAN, « Knowledge-based parametric design of mechanical products based on configuration design method », *Expert Systems with Applications*, vol. 21, n° 2, p. 99–107, août 2001. *Cité page 21.*

[Mäntylä, 1984] Martti MÄNTYLÄ, « A note on the modeling space of Euler operators », *Computer Vision, Graphics, and Image Processing*, vol. 26, n° 1, p. 45–60, avr. 1984. *Cité page 180.*

N

[Nedeß et Jacob, 1997] Christian NEDESS et Uwe JACOB, « A case-based reasoning approach towards learning from experience connecting design and shop floor », *Computers in Industry*, vol. 33, n° 1, p. 127–137, août 1997. *Cité page 77.*

[Nouvel, 2002] Pascal NOUVEL (coordinateur), *Enquête sur le concept de modèle*, collection Science, histoire et société, Presses Universitaires de France, ISBN 2-13-052619-5, avr. 2002, Préface de Dominique LECOURT. *Cité page 29.*

O

[O'Driscoll, 2001] Martin O'DRISCOLL, « Design for manufacture », dans *Journal of Materials Processing Technology*, n° 122, p. 318–321, Elsevier Science, sept. 2001. *Cité page 57.*

[Ong et coll., 2003] Soh Khim ONG, Ming Ju SUN et Andrew Yeh Ching NEE, « A fuzzy set AHP-based DFM tool for rotational parts », dans *International Journal of Materials Processing Technology*, vol. 138, p. 223–230, Elsevier Science, 2003. *Cité page 76.*

P

[Pahl et Beitz, 1999] Gerhard PAHL et Wolfgang BEITZ, *Engineering Design – A Systematic Approach*, Springer-Verlag, second édition, ISBN 3-540-19917-9, juin 1999. *Cité page 15.*

[Pallez, 2000] Denis PALLEZ, *Une nouvelle approche fonctionnelle pour une assistance géométrique pendant les premières phases de conception de produits*, Thèse de doctorat, Université de Metz, jan. 2000. *Cité page 28.*

[Perrin, 2005] Jacques PERRIN, « Remettre la valeur au cœur des débats en sciences économiques », dans *Political and Ethical Knowledge regarding Economic Activities conference (PEKEA 2005)*, Rennes, France, nov. 2005. *Cité page 12.*

[Pettichord, 2002] Bret PETTICHORD, « Design for testability », dans *Pacific Northwest Software Quality Conference*, Portland, USA, oct. 2002. *Cité page 58.*

[Pontacq, 2000] Michel PONTACQ, « Gestion des flux d'informations : workflow », *Techniques de l'ingénieur, traité Informatique*, n° H3518, août 2000. *Cité page 48.*

[Pourroy, 1999] Franck POURROY, « Apport des éléments finis à la conception mécanique », *Techniques de l'ingénieur, traité L'entreprise industrielle*, n° AG2530, avr. 1999. *Cité page 27.*

[Poveda, 2001] Olivier POVEDA, *Pilotage technique des projets d'ingénierie simultannée, modélisation des processus, analyse et instrumentation*, Thèse de doctorat, Institut National Polytechnique de Grenoble, déc. 2001. *Cité pages 45 et 46.*

[Pugh, 1991] Stuart PUGH, *Total Design : Integrated Methods for Successful Product Engineering*, Addison-Wesley, ISBN 0-201-41639-5, fév. 1991. *Cité page 16.*

Q

[Qin et Regli, 2000] Xiaoli QIN et William C. REGLI, « Applying case-based reasonning to mechanical bearing design », dans *Design Engineering Technical Conferences – Design for Manufacturing*, n° DETC2000/DFM-14011, Baltimore, USA, ASME, sept. 2000. *Cité page 22.*

R

[Rachuri et coll., 2005] Sudarsan RACHURI, Steven J. FENVES, Ram D. SRIRAM et Fujun WANG, « A product information modeling framework for product lifecycle management », *Computer-Aided Design*, vol. 37, nº 13, p. 1399–1411, nov. 2005. *Cité page 43.*

[Ramana et coll., 2006] Kolla V. RAMANA, Manvinder SINGH, Abhishek GUPTA, Rohit DEY, Ankur KAPOOR et P. V. Madhusudhan RAO, « A manufacturability advisor for spun and rollformed sheet metal components », *International Journal of Advanced Manufacturing Technology*, vol. 28, nº 3-4, p. 249–254, mars 2006. *Cité pages 66 et 75.*

[Ranta et coll., 1996] Mervi RANTA, Martti MÄNTYLÄ, Yasushi UMEDA et Tetsuo TOMIYAMA, « Integration of functional and feature-based product modelling – the IMS/GNOSIS experience », *Computer-Aided Design*, vol. 28, nº 5, p. 371–381, 1996. *Cité page 16.*

[Reich, 1995] Yoram REICH, « A critical review of General Design Theory », *Research in Engineering Design*, vol. 7, nº 1, p. 1–18, 1995. *Cité page 17.*

[Rosenman, 1997] Mike A. ROSENMAN, The generation of form using an evolutionary approach, dans *Evolutionary Algorithms in Engineering Applications*, Dipankar DASGUPTA et Zbigniew MICHALEWICZ (coordinateurs), p. 69–86, Springer-Verlag, mars 1997, ISBN 3-540-62021-4. *Cité page 25.*

[Rosenman et Gero, 1996] Mike A. ROSENMAN et John S. GERO, « Modelling multiple views of design' objects in a collaborative cad environment », *Computer-Aided Design*, vol. 28, nº 3, p. 193–205, mars 1996. *Cité page 86.*

[Ross, 1977] Douglas T. ROSS, « Structured Analysis (SA) : A Language for Communicating Ideas », *IEEE Transactions on Software Engineering*, vol. 3, nº 1, p. 16–34, 1977. *Cité page 13.*

[Rossignol, 2001] Vincent ROSSIGNOL, *Approche déclarative de la modélisation de courbes*, Thèse de doctorat, Université de Nantes, oct. 2001. *Cité page 23.*

[Roucoules, 1999] Lionel ROUCOULES, *Méthodes et connaissances : contribution au développement d'un environnement de conception intégrée*, Thèse de doctorat, Institut National Polytechnique de Grenoble, sept. 1999. *Cité pages 51 et 52.*

[Rush et Roy, 2001] Christopher RUSH et Rajkumar ROY, « Expert judgement in cost estimating : Modelling the reasoning process », *Concurrent Engineering : Research and Applications*, vol. 9, nº 4, p. 271–284, déc. 2001. *Cité page 79.*

[Russell et Sayers, 1986] Gordon RUSSELL et Ian L. SAYERS, « Design for testability – a review of advanced methods », *Microprocessors and Microsystems*, vol. 10, n° 10, p. 531–539, déc. 1986. *Cité page 58.*

S

[Saaty, 1980] Thomas L. SAATY, *The Analytic Hierarchy Process*, McGraw-Hill, New York, USA, 1980. *Cité pages 70 et 76.*

[Schank et Abelson, 1977] Roger C. SCHANK et Robert P. ABELSON, *Scripts, Plans, Goals and Understanding : an Inquiry into Human Knowledge Structures*, chap. 1-3, Lawrence Erlbaum Associates, Hillsdale, N. J., ISBN 0-89859-138-4, 1977. *Cité page 21.*

[Serenko et Bontis, 2004] Alexander SERENKO et Nick BONTIS, « Meta-review of knowledge management and intellectual capital literature : Citation impact and research productivity rankings », *Knowledge and Process Management*, vol. 11, n° 3, p. 185–198, 2004. *Cité page 50.*

[Shah, 1991] Jami J. SHAH, « Assessment of features technology », *Computer-Aided Design*, vol. 23, n° 5, p. 331–343, juin 1991. *Cité page 35.*

[Shah et Wright, 2000] Jami J. SHAH et Paul K. WRIGHT, « Developing theoretical foundations of DFM », dans *Design Engineering Technical Conferences – Design for Manufacturing*, n° DETC2000/DFM-14015, Baltimore, USA, ASME, sept. 2000. *Cité page 77.*

[Shakeri et Brown, 2004] Cirrus SHAKERI et David C. BROWN, « Constructing Design Methodologies Using Multi-Agent Systems », *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, special issue on Intelligent Agents in Design*, vol. 18, n° 2, p. 115–134, mai 2004. *Cité page 24.*

[Shapiro, 2001] Vadim SHAPIRO, « Solid Modeling », Rapport technique SAL 2001-2, University of Wisconsin Madison, oct. 2001, Spatial Automotion Laboratory. *Cité pages 30 et 31.*

[Shimomura et coll., 1996] Yoshiki SHIMOMURA, Sadao TANIGAWA, Hideaki TAKEDA, Yasushi UMEDA et Tetsuo TOMIYAMA, « Functional evaluation based on function content », dans *Design Engineering Technical Conferences – Computers and Information in Engineering*, n° DETC96/DTM-1412 de « Design Theory and Methodology », Irvine, California, USA, ASME, août 1996. *Cité page 15.*

[Simon, 1969] Herbert A. SIMON, *The sciences of the artificial*, MIT Press, third édition, 1969, Traduction et postface par Jean-Louis textscLe Moigne en 1974. Réédité en 1991, puis en 2004 sous le nom textitLes sciences de l'artificiel aux Éditions Gallimard (ISBN 2-07-030152-4). *Cité page 22.*

[Slavila et coll., 2005] Corneliu-Alexandru SLAVILA, Christophe DECREUSE et Michel FERNEY, « Fuzzy approach for maintainability evaluation in the design process », *Concurrent Engineering : Research and Applications*, vol. 13, nº 4, p. 291–300, déc. 2005. *Cité page 58.*

[Sly, 2004] David SLY, « Manufacturing process management (mpm), technology trends in plm », Rapport technique, Collaborative Product Development Associates, LLC, juin 2004. *Cité page 61.*

[Sobek et Ward, 1996] Durward K. SOBEK et Allen C. WARD, « Principles from toyota's set-based concurrent engineering process », dans *Design Engineering Technical Conferences – Computers in Engineering*, nº DETC96/DTM-1510, Irvine, California, USA, ASME, août 1996. *Cité page 69.*

[Subrahmanian et coll., 2005] Eswaran SUBRAHMANIAN, Sudarsan RACHURI, Steven J. FENVES, Sebti FOUFOU et Ram D. SRIRAM, « Product Lifecycle Management support : a challenge in supporting product design and manufacturing in a networked economy », *International Journal on Product Lifecycle Management*, vol. 1, nº 1, p. 4–25, 2005. *Cité page 43.*

[Suh, 1990] Nam P. SUH, *The principles of design*, nº 6 de « Oxford series on advanced manufacturing », Oxford University Press, New York, USA, ISBN 0-19-504345-6, fév. 1990. *Cité page 18.*

[Sánchez et coll., 1997] José M. SÁNCHEZ, John W. PRIEST et Rogelio SOTO, « Intelligent reasoning assistant for incorporating manufacturability issues into the design process », dans *Expert Systems with Applications*, vol. 12, p. 81–88, Elsevier Science, 1997. *Cité pages 74 et 75.*

T

[Takeda et coll., 1997] Hideaki TAKEDA, Masaharu YOSHIOKA, Tetsuo TOMIYAMA et Yoshiki SHIMOMURA, Analysis of Design Processes by Function, Behaviour and Structure, dans

Analysing Design Activity, Nigel CROSS, Henri CHRISTIAANS et Kees DORST (coordinateurs), p. 187–209, John Wiley & Sons, Chichester, UK, jan. 1997, ISBN 0-471-96060-8. *Cité page 16.*

[Tay et Roy, 2003] Francis Eng Hock TAY et Avijit ROY, « CyberCAD : a collaborative approach in 3D-CAD technology in a multimedia-supported environment », *Computers in Industry*, vol. 52, n° 2, p. 127–145, oct. 2003. *Cité page 50.*

[Thurston, 1991] Deborah L. THURSTON, « A formal method for subjective design evaluation with multiple attributes », *Research in Engineering Design*, vol. 3, n° 2, p. 105–122, juin 1991. *Cité page 70.*

U

[Ullman, 2002] David G. ULLMAN, *The mechanical design process*, collection Mechanical engineering, McGraw-Hill, third edition édition, ISBN 0-07-237338-5, 2002. *Cité pages 8, 16, 40 et 52.*

[Ulrich et Eppinger, 2003] Karl T. ULRICH et Steven D. EPPINGER, *Product Design and Development*, collection Higher Education, McGraw-Hill, third édition, ISBN 0-07-123273-7, sept. 2003. *Cité page 16.*

[Umeda, 1997] Yasushi UMEDA, « Functional Reasoning in Design », *IEEE Expert*, vol. 12, n° 2, p. 42–48, 1997. *Cité page 10.*

[Umeda et coll., 1990] Yasushi UMEDA, Hideaki TAKEDA, Tetsuo TOMIYAMA et Hiroyuki YOSHIKAWA, « Function, Behaviour, and Structure », dans *Applications of Artificial Intelligence in Engineering V*, John S. GERO (coordinateur), vol. 1, p. 177–193, Boston, USA, Computational Mechanics Publications and Springer-Verlag, juil. 1990. *Cité page 15.*

V

[van Vliet et coll., 1999] J. W. (Hans-Willem) VAN VLIET, C. A. (Kees) VAN LUTTERVELT et Hubert Jan Jozef KALS, « A requirement analysis for dfm design support in a dfx context », dans *CIRP International Design Seminar, Integration of process knowledge into design support systems*, Hubert Jan Jozef KALS et Fred J. A. M. VAN HOUTEN (coordinateurs), p. 269–278, Enschede, Netherlands, Kluwer Academic Publishers, mars 1999. *Cité page 60.*

[Váncza, 1999] József VÁNCZA, « Artificial intelligence support in design : A survey », dans *Keynote paper of the CIRP International Design Seminar*, Enschede, Pays-Bas, Kluwer Academic Publishers, mars 1999. *Cité pages 22, 23 et 53.*

W

[Wassenaar et Chen, 2001] Henk Jan WASSENAAR et Wei CHEN, « An approach to decision-based design », dans *Design Engineering Technical Conferences – Computers and Information in Engineering*, nº DETC2001/DTM-21683, Pittsburgh, USA, ASME, sept. 2001. *Cité page 70.*

X

[Xue et Yang, 2004] Deyi XUE et Haoguang YANG, « A concurrent engineering-oriented design database representation model », *Computer-Aided Design*, vol. 36, nº 10, p. 947–965, sept. 2004. *Cité page 43.*

Y

[Yoshikawa, 1981] Hiroyuki YOSHIKAWA, « General Design Theory and a CAD system », dans *Man-Machine Communication in CAD/CAM, Proceedings of the IFIP WG5.2-5.3 Working Conference 1980 (Tokyo)*, Toshio SATA et Ernest A. WARMAN (coordinateurs), p. 35–57, Amsterdam, North Holland, 1981. *Cité page 17.*

[Yoshikawa, 1989] Hiroyuki YOSHIKAWA, « Design philosophy : the state of the art », dans *39th CIRP General Assembly, Annuals of the CIRP*, vol. 38, p. 579–586, Trondheim, Norway, 1989. *Cité page 20.*

Z

[Zakarian et Kaiser, 1999] Vaagn L. ZAKARIAN et Mark J. KAISER, « An embedded hybrid neural network and expert system in a computer-aided design system », *Expert Systems with Applications*, vol. 16, nº 2, p. 233–243, fév. 1999. *Cité pages 20 et 25.*

[Zhao et Shah, 2002] Zuozhi ZHAO et Jami J. SHAH, « A normative dfm framework based on benefit-cost analysis », dans *Design Engineering Technical Conferences – Design for Manufacturing*, nº DETC2002/DFM-34176, Montreal, Canada, ASME, oct. 2002. *Cité pages 60, 67, 69 et 70.*

[Zhao et Shah, 2004] Zuozhi ZHAO et Jami J. SHAH, « Modeling and representation of manufacturing knowledge for dfm systems », dans *Design Engineering Technical Conferences – Computers and Information in Engineering*, n° DETC2004/CIE-57724, ASME, oct. 2004. *Cité pages 65 et 66.*

Normes et standards

[DOM, 2004] DOM (2004). « Document Object Model, version 1.0 ». World Wide Web Consortium Recommendation. *Cité pages 145 et 226.*

[ISO/IEC 19501, 2005] ISO/IEC 19501 (2005). « Unified Modeling Language, Version 1.4.2 ». International Organization for Standardization. Objet Management Group. *Cité pages 146 et 228.*

[MOF, 2006] MOF (2006). « Meta-Object Facility, version 2.0 ». Object Management Group. *Cité pages 154 et 227.*

[NF X 50-100, 1996] NF X 50-100 (1996). « Analyse fonctionnelle – caractéristiques fondamentales ». Association française de normalisation (Afnor). *Cité page 11.*

[NF X 50-150, 1990] NF X 50-150 (1990). « Analyse de la valeur, analyse fonctionnelle ». Association française de normalisation (Afnor). *Cité pages 7, 10, 11 et 12.*

[NF X 50-156, 2003] NF X 50-156 (2003). « Management par la valeur – conception à objectif désigné ou à coût objectif (COD/CCO) – exigences pour un pilotage concerté de la conception ». Association française de normalisation (Afnor). *Cité page 12.*

[XML, 2006] XML (2006). « eXtensible Markup Language, version 1.1 ». World Wide Web Consortium Recommendation. *Cité pages 145, 226 et 229.*

[XSD, 2004] XSD (2004). « XML Schema Description, version 1.1 ». World Wide Web Consortium Recommendation. *Cité pages 157, 212 et 229.*

Sites Internet

[www-AFAV] « Association française pour l'analyse de la valeur (AFAV) ». `http://www.afav.asso.fr`. *Cité pages 7 et 12.*

[www-Axiomatic] « Axiomatic Design Solutions, Inc. ». http://www.axiomaticdesign.com. *Cité page 19.*

[www-CATIA] « Computer-Aided Three dimensional Interactive Application ». Logiciel édité par Dassault Systèmes. http://www.catia.com. *Cité pages 84 et 86.*

[www-Cimmetry] « Autovue Collaborative Visualization ». Cimmetry Systems Corporation. http://www.cimmetry.com/autovue.html. *Cité page 49.*

[www-CRIF-WTCM] « CRIF-WTCM ». http://www.crif.be. *Cité page 27.*

[www-DFMA] « DFMA commercial tools, Boothroyd Dewhurst, Inc. ». http://www.dfma.com. *Cité page 68.*

[www-Ecologie] « Site du ministère de l'écologie et du développement durable ». http://www.ecologie.gouv.fr. *Cité page 58.*

[www-GDT] « Grand dictionnaire terminologique ». Office québécois de la langue française. http://www.granddictionnaire.com. *Cité pages 35, 47 et 58.*

[www-Java] « Java ». Sun Microsystems. http://java.sun.com. *Cité pages 145 et 149.*

[www-Java3D] « Java 3D ». Java.net. https://java3d.dev.java.net. *Cité page 145.*

[www-JDBC] « Java database connectivity ». Sun Microsystems. http://java.sun.com/products/jdbc/overview.html. *Cité page 145.*

[www-JOGL] « Java binding for the OpenGL API (JOGL) ». Java.net. https://jogl.dev.java.net. *Cité page 145.*

[www-NX] « NX ». Logiciel édité par UGS (Unigraphics). http://www.ugs.com/products/nx/. *Cité page 84.*

[www-Pro/E] « Pro/Engineer® ». Logiciel édité par Parametric Technology Corporation (PTC). http://www.ptc.com. *Cité page 84.*

[www-SAX] « Simple API for XML ». http://www.saxproject.org. *Cité pages 145 et 228.*

Index des références bibliographiques

A

Aamodt, Agnar 22
Abdelaziz Bouras 43
Abelson, Robert P. 21
Adams, Vince 27, 57
Adewuya, Adenike 24
Agnez, Yannick 162
Albermani, Faris 21
Allen, Thomas J. 47
Alliot, Jean-Marc 112
Altshuller, Genrich S. 19
American Computer Machinery 37
Anderl, Reiner 83, 84

B

Bai, Yong 23, 38, 84
Balan Gurumoorthy 43
Balič, Jože 76
Bandi, Karthik 43
Barnabé, Franz 40
Beiter, Kurt A. 58
Beitz, Wolfgang 15
Bellinger, Gene 50
Ben Amara, Abdelmajid 15
Blain, Patrick 6
Bok, Shung Hwee 43
Bontis, Nick 50
Boothroyd, Geoffrey ... 57, 60–62, 65, 67, 75
Borg, Jonathan 21
Bouleau, Nicolas 29
Boyle, Iain M. 20
Braha, Dan 18
Bralla, James G. 67, 75
Brazier, Frances M. T. 24
Brezočnik, Miran 76
Brisset, Pascal 112
Bronsvoort, Willem F. 86, 88
Brosse, Étienne 144
Brown, David C. 9, 20, 24
Bureau International des Poids et Mesures 63, 64, 113
Bytheway, Charles W. 13

C

Castelain, Jean-Marie 71, 77
Caubet, René 23
CEntre Techniques des Industries Mécaniques 40

Chan, Vincent 62
Chao, Kuo-Ming 24
Chapman, Colin 24
Charles, Jeff 46
Chau, Kwok-Wing 21
Cheldelin, Brent 58
Chen, David 14
Chen, Li 73
Chen, Wei 70
Chen, Xiao Chuan 58
Chen, Zheng Ming 82
Chou, Chi-Wu 76
Christiaans, Henri 16
Chung, Jack C.H. 23, 38, 84
CIMdata, Inc. 42
Ciuciu, Ioana-Georgiana 144
Clément, André 15
Cochran, David S. 19
Craveur, Jean-Charles 57
Cross, Nigel 16
Cutkosky, Mark R. 49, 72, 80

D

Danesi, Frédéric . 36, 37, 39, 96, 99, 100, 104, 144, 150
Daniel, Marc 35, 38, 39, 84, 110
Das, Diganta 59, 60, 64, 67, 70–72
Dasgupta, Dipankar 24
de Kraker, Klaas Jannis 86, 87
Decreuse, Christophe 41, 43, 44, 58
Deneux, Dominique 15
Denis, Laurent 37, 96, 100, 101, 103, 118, 144, 156, 157
Desai, Anoop 58
Dewhurst, Peter 57, 60–62, 65, 67, 75
Dey, Rohit 66, 75
Dissinger, Thomas E. 73, 74
Dogui, Abdelwaheb 15
Dohmen, Maurice 86
Dorst, Kees 16
Doumeingts, Guy 14
Drogoul, Alexis 24
Drstvenšek, Igor 76
Druon, Maurice 10, 18, 34
Duce, David A. 23
Duda, James 24
Duverlie, Pascal 71, 77

E

Economist Intelligence Unit 50
Edwards, K. L. 57
Engelbart, Douglas C. 46
Eppinger, Steven D. 16
Esposito, Nicolas 49
Eudier, Michel 162
Eynard, Benoît 43, 52

F

Faltings, Boi 22
Farin, Gerald 38
Farineau, Thibault 71, 77
Feng, Chang-Xue 84

Feng, Chang-Xue (Jack) 34
Fenves, Steven J. 43
Ferney, Michel 58
Feschotte, Daniel 41, 43, 44
Ficko, Mirko 76
Florida-James, Barry O. 24
for the Engineering Analysis Community, International Association 27, 57
Fortin, Clement 60
Foufou, Sebti 43
Frazer, John Hamilton 24
Fuh, Jerry Ying Hsi 49

G

Gaildrat, Véronique 22, 23
Ganier, Michel 40
Gao, Shu Ming 82
Garcia, Frédérick 112
Gardan, Nicolas 27, 47, 90, 96, 99, 100, 104–106, 144
Gardan, Y. 144
Gardan, Yvon .. 16, 17, 25, 29, 30, 32–34, 37, 81, 83, 99, 100, 104, 105, 144, 178
Gero, John S. 15, 20, 24, 86
Girard, Philippe 8, 9, 22
Gomes, Samuel 43
Grabowski, Hans 18
Grayer, Alan R. 34
Grecu, Dan L. 24
Gupta, Abhishek 66, 75
Gupta, Satyandra K. ... 59, 60, 64, 67, 70–72

H

Han, Jung Hyun 35, 81, 83
Han, Soonhung 21
Haton, Jean-Paul 20, 21, 24, 65
Haton, Marie-Christine 20, 21, 24, 65
Hazelrigg, George A. 70
Hermès 144
Heschung, Frédéric 96, 144
Hills, William 24
Hoffmann, Christoph M. 38, 86, 87
Hoschek, Josef 38
Huang, Chun-Che 34
Huang, George Q. 58
Huang, Samuel H. 56–58
Huet, Gregory 60
Hwang, Teng-Shang 23, 38, 84
Hégron, Gérard 23

I

IEEE Computer Society Press 23
Infodesign 22
Ishii, Kosuke 57, 58

J

Jacob, Uwe 76
Jakiela, Mark J. 24
Jancene, Pierre 23
Jian, Pu 73
Jiang, Yu 23, 38, 84

Joan-Arinyo, Robert 38, 86, 87
Johansen, Robert 46
Johnson-Lenz, Peter 46
Johnson-Lenz, Trudy 46

K

Kaiser, Mark J. 20, 25
Kals, Hubert Jan Jozef 60
Kapoor, Ankur 66, 75
Kazuhiro, Saitou 76
Keeney, Ralph L. 70
Kent, Allen 9, 24
Kesavadas, Mini P. 43
Kesavadas, T. 43
Kim, Kwangsoo 82
Kim, Kyoung-Yun 43
Kim, Myung-Soo 38
Knight, Winston 57, 60–62, 65, 67, 75
Kumar, A. Senthil 43
Kuo, Tsai-C. 56–58
Kusiak, Andrew 34, 84
Kwaiter, Ghassan 22

L

Laakko, Timo 82
Lafleur, Bernard 40
Lautrey, Jacques 24
Le Roux, Olivier 23
Lee, Byungwoo 76
Lee, Jae Yeol 82
Lee, Kwang-Rak 19
Lee, Sang Hun 86
Lesage, David 38
Levan, Serge K. 44, 45, 47
Li, Pei-Gen 34
Li, Wei Dong 49, 81, 82
Liao, Shu-Hsien 20
Lin, Zone-Ching 21
Liu, Hong 24
Lonchampt, Pierre 17
Lossack, Ralf 18
Lu, Wen Feng 49
Lu, Yi Qiang 49
Lucas, Michel 23

M

MacCallum, Ken J. 21
Maculet, Robert 35, 38, 39, 84, 110
Madhusudan, Therani 24
Magrab, Edward B. 73, 74
Maher, Mary Lou 20
Maimon, Oded 18
Mak, Kai Ling 58
Malik, E. 144
Malosse, René 40
Manley, David G. 43
Marceau, Dominique 57
Martin, Dominique 23
Martin, Mark V. 57
Martin, Philippe 23
McAlinden, Leo P. 24
Medani, O. 86

Medjdoub, Benachir 23
Mendgen, Ralf 83, 84
Mervyn, Fathianathan 43
Meyer, Yves 77
Michalewicz, Zbigniew 24
Minich, Christian 10, 11, 144
Minsky, Marvin 24
Mital, Anil 58
Mittman, Robert 46
Monnier, Karine 144
Monticolo, Davy 43
Morgenstern, Oskar 70
Mori, Toshiki 49
Moshkina, Lilia V. 24
Moulin, Vincent 40
Myung, Sehyun 21
Mäntylä, Martti 16, 82, 180

N

Nau, Dana S. 59, 60, 64, 67, 70–72
Nedeß, Christian 76
Nee, Andrew Yeh Ching 43, 49, 75, 81
Noort, Alexander 88
Norman, Peter 24
Nouvel, Pascal 29

O

O'Driscoll, Martin 57
Ong, Soh Khim 49, 75, 81

P

Pahl, Gerhard 15
Pallez, Denis 10, 11, 28
Park, Jahng-Hyon 19
Park, Ji-Hyung 19
Paygude, Amit 43
Perrin, E. 144
Perrin, Estelle 37, 99, 100, 144
Perrin, Jacques 12
Pettichord, Bret 58
Plaza, Enric 22
Plemenos, Dimitri 23
Pontacq, Michel 48
Pourroy, Franck 26
Poveda, Olivier 44, 45
Pratt, Michael 81, 83
Priest, John W. 73, 74
Pugh, Stuart 16

Q

Qin, Xiaoli 22

R

Rabenasolo, Besoa 77
Rachuri Sudarsan 43
Raiffa, Howard 70
Ramana, Kolla V. 66, 75
Ranta, Mervi 16
Rao, P. V. Madhusudhan 66, 75
Ratchev, S. M. 86

Regli, William C. 22, 59, 60, 64, 67, 70–72, 81, 83
Reich, Yoram 18
Reimeringer, Mickaël 104, 144
Requicha, Aristides A. G. 81
Reynal, Vicente A. 19
Richard, Jean-François 24
Rivière, Alain 15
Rong, Kevin 20
Rosenman, Mike A. 24, 86
Ross, Douglas T. 13
Rossignol, Vincent 23
Roucoules, Lionel 50, 51
Roy, Avijit 49
Roy, Rajkumar 78
Rush, Christopher 78
Russell, Gordon 58

S

Saaty, Thomas L. 70, 75
Saffo, Paul 46
Saitou, Kazuhiro 24
Salustri, Filippo A. 62
Sata, Toshio 18
Sayers, Ian L. 58
Schank, Roger C. 21
Schiesser, Philippe 40
Schiex, Thomas 112
Serenko, Alexander 50
Serrafero, Patrick 43
Serré, Philippe 15
Shah, Jami J. 34, 60, 65–67, 69, 76
Shakeri, Cirrus 24
Shapiro, Raphael V. 19
Shapiro, Vadim 30
Shen, Chi-Chih 21
Shimomura, Yoshiki 15, 16
SIGGRAPH, ACM 86
Simon, Herbert A. 22
Singh, Manvinder 66, 75
Slavila, Corneliu-Alexandru 58
Sly, David 60
Smith, Peter 24
Sobek, Durward K. 69
Soto, Rogelio 73, 74
Soënen, René 15
Sriram, Ram D. 43
Subrahmanian, Eswaran 43
Sudweeks, Fay 24
Suh, Nam P. 18
Sun, Kun 22
Sun, Ming Ju 75
Sánchez, José M. 73, 74

T

Takeda, Hideaki 15, 16
Tang, Mingxi 24
Tanigawa, Sadao 15
Tay, Francis Eng Hock 49
Tenenbaum, Jay M. 72, 80
Thurston, Deborah L. 70
Tomiyama, Tetsuo 15, 16

U

Ullman, David G. 8, 16, 40, 51
Ulrich, Karl T. 16
Umeda, Yasushi 10, 15, 16

V

van Houten, Fred J. A. M. 60
van Luttervelt, C. A. (Kees) 60
van Vliet, J. W. (Hans-Willem) 60
Vaupotič, Boštjan 76
von Neumann, John 70
Váncza, József 21, 23, 52

W

Wang, Fujun 43
Wang, Jia-Yi 23, 38, 84
Wang, Xiankui 73
Ward, Allen C. 69
Warman, Ernest A. 18
Wassenaar, Henk Jan 70
Wijngaards, Niek J.E. 24
Williams, James G. 9, 24
Wong, Yoke San 49
Wright, Paul K. 76
Wu, Chien-Tai 23, 38, 84

X

Xue, Deyi 43

Y

Yang, Haoguang 43
Yang, Hyungjeong 43
Yannou, Bernard 23
Yoshikawa, Hiroyuki 15, 18, 20
Yoshioka, Masaharu 16

Z

Zakarian, Vaagn L. 20, 25
Zhang, Hong-C. 56–58
Zhao, Zuozhi 60, 65–67, 69
Zou, Hongliu 23, 38, 84

Résumé

Depuis une quinzaine années, de plus en plus d'entreprises tendent à passer du mode d'organisation séquentiel à un mode d'organisation simultané et distribué. Ces évolutions organisationnelles induisent logiquement la nécessité d'adapter les outils et les méthodes de travail afin de mieux gérer l'ensemble du cycle de vie du produit (notion de PLM).

Dans le cadre de cette thèse, nous nous intéressons à l'interopérabilité des connaissances des fabricants avec l'activité de conception. Ce travail s'inscrit dans une démarche de conception guidée par la fabricabilité. Un certain nombre d'outils de conception guidée existent déjà mais ils ne sont pas génériques et imposent un couplage trop fort entre les domaines.

La solution proposée consiste en une modélisation des connaissances de fabrication par l'intermédiaire de règles, et en l'utilisation et la réutilisation de ce savoir-faire au travers de l'approche de modélisation synthétique du projet DIJA. Le principe adopté consiste à assister, guider et sensibiliser le concepteur au cours de la conception, en observant l'influence de ses actes sur les possibilités et conditions de fabrication de la pièce, tout en garantissant le respect de la confidentialité.

La mise en œuvre de cette approche fait intervenir une notion générique d'état, ainsi que deux types de graphes qui servent de support aux interactions avec l'utilisateur et à la gestion des règles. Nous étudions donc l'évolution des connaissances impliquées en fonction de l'application des outils de conception, ce qui se traduit par une évolution des graphes. Une maquette adoptant cette approche a permis de valider les concepts fondamentaux.

Mots-clefs : CAO, fabricabilité, DIJA, cycle de vie produit.

Abstract

Since about fifteen years, more and more firms tend to switch from sequential process management modes to a concurrent and distributed management. These organizational evolutions induce logically the need for adapting tools and methods of work in order to better manage the product lifecycle (PLM concept).

Within the framework of this thesis, we are interested in the issue of interoperability of manufacturing knowledge with the product design activity. This is in line with an approach of design for manufacturability. Numerous design for manufacturability tools are already available, but they are not generic enough, and they lay on tightly coupling domains.

This proposition consists in modelling manufacturing knowledge as rules, and to use and reuse this know-how through synthetic design approach of the DIJA project. The choosen principle consists in assisting, guiding and making aware the designer during design, by observing influence of his/her acts over manufacturing possibilities and conditions, while respecting confidentiality.

Implementation of this approach brings into play a generic state notion, as well as two kinds of graphs which support user interactions and rules management. Thus, we study evolution of knowledge implied during design tools use, which results into transformation of the graphs. A prototype implementing this approach allowed to validate the fundamental concepts.

Keywords: CAD, manufacturability, DIJA, product lifecycle.