




UNIVERSITÉ DE MONTRÉAL

MÉTHODOLOGIE D'ÉVALUATION
DE STEP POUR SA MISE EN PLACE

MARIE-CHRISTINE CARON
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAITRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
AVRIL 1998



© Marie-Christine Caron, 1998



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-37434-3

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MÉTHODOLOGIE D'ÉVALUATION
DE STEP POUR SA MISE EN PLACE

présenté par: CARON Marie-Christine

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. GOURDEAU Richard, Ph.D., président

M. FORTIN Clément, Ph.D., membre et directeur de recherche

M. MASCLE Christian, Ph.D., membre et codirecteur de recherche

M. MONY Charles, Ph.D., membre

REMERCIEMENTS

Avant tout, il importe d'adresser mes remerciements à tous ceux et celles qui ont contribué, de près ou de loin, à la poursuite de la démarche intellectuelle que suppose la réalisation d'un mémoire.

En premier lieu, j'aimerais remercier les membres de mon comité de mémoire qui ont été à l'écoute de mes nombreux questionnements et ont su me conseiller judicieusement. Monsieur Clément Fortin, Ph. D. et Monsieur Christian Mascle, Ph. D., ont assumé respectivement les tâches de directeur et codirecteur de ce mémoire. Tout au long de mon cheminement, ils ont été à l'écoute de mes nombreuses remises en question et m'ont prodigué des conseils afin que je puisse instaurer une démarche de recherche rigoureuse. Ils ont su s'adapter à mon rythme de travail qui, parfois, s'avère bousculant pour ceux qui assument l'encadrement immédiat.

Merci à Monsieur Richard Gourdeau, Ph. D., qui a agi en tant que président de mon jury de mémoire. Je lui suis reconnaissante d'avoir fait une lecture minutieuse du manuscrit et d'avoir émis des commentaires pertinents permettant d'en améliorer la qualité.

J'aimerais aussi souligner l'intérêt manifesté par Monsieur Charles Mony, Ph. D., à l'égard de mon travail à l'intérieur du labyrinthe qu'est l'analyse de la norme ISO 10303. Il a partagé mes interrogations, mes incertitudes et m'a donné des pistes de lecture ainsi que de nombreux conseils très utiles pour éclairer ma démarche.

J'aimerais remercier le Conseil National de Recherches du Canada de m'avoir fourni les outils nécessaires à la réalisation de cette étude ainsi que pour son soutien financier tout au long de mon cheminement. Je tiens aussi à remercier l'organisme subventionnaire

C.R.S.N.G. qui m'a accordé, par l'entremise de mon directeur de mémoire, une aide financière substantielle pour la réalisation de ce projet.

Ma reconnaissance s'adresse également à Madame Pauline Minier, Ph. D., professeur au Département des sciences de l'éducation à l'Université du Québec à Chicoutimi, pour son expertise en linguistique ainsi qu'en recherche, pour ses multiples conseils et pour le support moral dont elle a fait preuve.

Ma gratitude s'adresse tout particulièrement à mes parents et à mon frère, qui ont su composer avec les contraintes diverses qu'impose ce type de démarche intellectuelle. Merci de votre présence et de votre support moral tout au long de ce parcours. À mon père, pour qui mon cheminement à la maîtrise, fut une période tout particulièrement dense en émotions. Espérant, cher père, que la joie et la fierté compenseront pour tous ces moments consacrés à veiller au maintien de mon engagement dans cette démarche. Merci à toi, chère mère, qui a su composer avec cette situation traversée d'imprévus.

J'aimerais aussi remercier mon compagnon Vincent ainsi que mes amies Sophie, Claude et Christine, qui ont su trouver les mots pour me reconforter. Merci pour votre support moral.

Reconnaissance à tous ceux et celles qui ont partagé mon enthousiasme débordant et qui ont cru en la réussite de mon entreprise.

RÉSUMÉ

Le partage d'information entre entreprises a amené l'utilisation d'échanges de données produit informatisées entre celles-ci. Ces échanges ont débuté par l'entremise d'interfaces directes et ont évolué vers des interfaces normalisées, c'est-à-dire par l'intermédiaire d'un format neutre. Ces dernières requièrent un nombre beaucoup moins important de processeurs pour réaliser les échanges.

Le concept d'interfaces normalisées a fait naître plusieurs normes telles VDA-FS, IGES et STEP (ISO 10303), mais seule cette dernière permet un échange complet de la définition du produit. Elle permet aussi l'échange de modèles solides ce qu'aucune autre norme ne permet en pratique. De plus, elle englobe plusieurs domaines tels les secteurs de l'automobile et de l'aérospatiale.

La norme STEP demeure très récente et en constante évolution puisqu'elle n'est approuvée au stade de norme internationale que depuis novembre 1994. Seulement une petite partie de cette norme est approuvée, d'autres parties, quelques unes seulement, sont prêtes pour l'approbation au stade de norme internationale et plusieurs parties sont encore en développement. Cette réalité impose la nécessité d'analyser le niveau d'avancement de la norme. Pour ce faire, une méthodologie d'évaluation est proposée.

Dans ce mémoire, une méthodologie d'évaluation de STEP est proposée. Elle exige une définition claire des objectifs et du domaine d'échange et elle couvre les échanges de données produit aux niveaux de la syntaxe, de la sémantique et des fonctions. Puisqu'elle englobe tous les niveaux du processus global de communication, même celui relié aux fonctions, cette méthodologie est novatrice car elle demeure indépendante des logiciels CAO utilisés pour les tests.

Les tests exécutés n'ont couvert qu'une infime partie des échanges de données qu'il est possible de réaliser à travers plusieurs domaines. Ils ont permis d'analyser la méthodologie afin de démontrer sa pertinence. En effet, cette analyse a démontré que chacune des parties a été examinée par le recours aux tests et chacune d'elles s'avère nécessaire. Ces diverses parties sont donc pertinentes pour la réalisation des échanges de données efficaces.

ABSTRACT

Sharing product information between companies brought computerized data exchanges. These data exchanges started with direct interfaces and evolved towards normalized interfaces (via neutral format). The latter need less processors in order to realize these exchanges.

The normalized interfaces concept triggered the development of standards like VDA-FS, IGES and STEP (ISO 10303) but only the latter allows semantic data exchange. STEP also exchange solid models which no other standard can realize in practice. Moreover, it encloses many domains like the automobile and the aerospace sectors.

Since the STEP standard has been approved to the international standard level in November 1994, it is still recent and in constant evolution. Only a small subset of the standard has been approved so far, only few other parts are now ready to be approved at the international standard level and many parts are still at the development level. Because of this, the implementation level of STEP needs to be analyze. In order to meet this goal, an evaluation methodology is proposed.

In this memoire, an evaluation methodology for STEP is proposed. It requires a clear definition of the exchange objectives and domain, and covers the syntax, the semantic and functional levels of the data exchange. Since it addresses all levels of the communication process, even at the functional level, this methodology is innovating and is independent of the CAD (Computer Aided Design) softwares used for the tests.

The tests covered only a small portion of the data exchanges that are possible across many domains. They allowed the analysis of the methodology in order to demonstrate its relevance. In fact, this analysis demonstrated that every part was examined throught the

execution of tests which has proved the necessity of each part of the methodology. Therefore, they are relevant for effective product data exchanges.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES ANNEXES	xiv
LISTE DES TABLEAUX	xv
LISTE DES FIGURES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xviii
LISTE DES EXEMPLES	xxii
1. INTRODUCTION	1
2. REVUE BIBLIOGRAPHIQUE	5
2.1. Échange de données	6
2.2. Normes existantes	7
2.3. Généralités sur STEP	10
2.3.1. Structure générale	10
2.3.2. Protocoles d'application	17

2.3.3. Méthodes de description	21
2.3.4. Méthodes de mise en œuvre	25
2.3.5. Modèles d'information de ressources intégrées	26
2.3.6. Méthodes de tests de conformité	27
2.3.7. Suites de tests abstraits	28
2.3.8. État d'avancement et perspectives de la norme STEP	29
2.4. Tests de conformité	30
2.4.1. Éléments centraux	31
2.4.2. Méthodes de tests abstraits	37
2.4.3. Suites de tests abstraits	39
2.5. Mapping	46
2.5.1. Introduction	46
2.5.2. L'étendue du domaine du "mapping"	47
2.5.3. Études sur les méthodes de "mapping"	48
2.6. Problématique dans une entreprise	53
2.7. Expériences faites avec STEP	55
2.7.1. Études menées (tests)	55
2.7.2. Projet pilote fait entre deux entreprises	58
2.7.3. Améliorations à apporter	63
3. MÉTHODOLOGIE D'ÉVALUATION DE STEP POUR SA MISE	
EN PLACE	66
3.1. Pertinence de la méthodologie	67
3.2. Procédure méthodologique	74
3.3. Définition des objectifs et du domaine d'échange (A1)	77
3.3.1. Définition des objectifs d'échanges (A11)	79
3.3.2. Sélection des études de cas (A12)	82
3.4. Traduction et réception des fichiers d'étude de cas (A2)	83

3.4.1. Traduction des fichiers (A21)	83
3.4.2. Réception et étude de l'intégrité des fichiers (A22)	93
3.4.3. Traduction des fichiers dans le format du système CAO de réception (A23)	95
3.5. Vérification des fichiers (A3)	95
3.5.1. Niveau 1 Conformité (A31)	96
3.5.2. Niveau 2 Sémantique (A32)	113
3.5.3. Niveau 3 Fonction (A33)	120
3.5.4. Actions correctives (A5)	129
3.6. Vers une procédure d'analyse	130
4. DESCRIPTION DES TESTS ET ANALYSE DE LA MÉTHODOLOGIE ...	131
4.1. Procédure d'analyse de la méthodologie	132
4.2. Présentation des tests	133
4.2.1. Première étude de cas	136
4.2.2. Deuxième étude de cas	147
4.3. Analyse de la méthodologie	156
4.3.1. Définition des objectifs et du domaine d'échange	157
4.3.2. Traduction et réception des fichiers d'étude de cas	158
4.3.3. Vérification des fichiers	162
4.3.4. Actions correctives	165
4.3.5. Discussion	166
4.4. Études futures	167
5. CONCLUSION	170
RÉFÉRENCES	174

BIBLIOGRAPHIE178

LISTE DES ANNEXES

ANNEXE A1: Diagrammes AAM, ARM et AIM	189
ANNEXE A2: Fichier STEP	195
ANNEXE A3: Paramètres d'un fichier IGES à spécifier sur CATIA	235

LISTE DES TABLEAUX

Chapitre 2

2.1: Description des classes de STEP	16
2.2: Statut des protocoles d'application	20
2.3: Tableau synthèse des langages pour faire du "mapping"	50
2.4: Comparaisons de propriétés de masse entre les systèmes	61
2.5: Comparaison aller-retour de propriétés de masse	62

Chapitre 3

3.1: Objectifs d'échanges à choix de réponse	80
3.2: Objectifs d'échanges à réponse manuscrite	81

Chapitre 4

4.1: Objectifs d'échanges à choix de réponse	138
4.2: Objectifs d'échanges à réponse manuscrite	139
4.3: Objectifs d'échanges à choix de réponse	148
4.4: Objectifs d'échanges à réponse manuscrite	150

LISTE DES FIGURES

Chapitre 2

2.1: Évolution dans le temps des normes d'échanges de données	8
2.2: STEP sur une page	13
2.3 Diagramme IDEF0	18
2.3: Exemple de diagramme EXPRESS sous forme EXPRESS-G (page1 de 2)	22
2.5: Exemple de diagramme EXPRESS sous forme EXPRESS-G (page2 de 2)	23
2.6: Type de "mapping"	48
2.7: Pompe de conduite assistée utilisée pour les transferts de données entre GM et Ford	59

Chapitre 3

3.1: Structure de communication entre deux partis	67
3.2: Niveaux de représentation d'un langage	70
3.3: Évolution des langages de modélisation	71
3.4: Méthodologie d'évaluation de STEP pour sa mise en place	75
3.5: Définition des objectifs et du domaine d'échange	78
3.6 Traduction des fichiers sous le format STEP	85
3.7: Tests Niveau 1: Conformité	98
3.8: Pré-analyse des fichiers	100
3.9: Schéma détaillé d'une face avancée avec association de lignes du fichier STEP	111
3.10: Tests Niveau 2: Sémantique	114
3.11: Solide	116
3.12: Possibilité de topologie du solide récupéré	116
3.13 Tests Niveau 3: Fonctions	121
3.14: Arbre de construction d'une vis	124

3.15: Fichier contenant une vis récupérée d'un échange via STEP	125
3.16: Superposition d'entités	127

Chapitre 4

4.1: Plate-forme de tests du CNRC	134
4.2: Assemblage 10617741 de General Motor Diesel de London (Ontario)	137
4.3: Assemblage 10617741 de GM Diesel avec erreurs de définition d'assemblage	142
4.4: Fichier avant recalcul	145
4.5: Fichier après recalcul	145
4.6: Pièce support3-modifier de l'École Polytechnique de Montréal	147
4.7: Pièce support3-modifier avec entités en trop	152
4.8: Pièce support3-modifier ayant une erreur de tolérance géométrique	154
4.9: Fichier traduit via IGES et STEP	161

Annexe A1

A1.1: Exemple d'un AAM de l'AP 203	190
A1.2: Exemple d'un AAM détaillé de l'AP 203	191
A1.3: Exemple d'un ARM de l'AP 203	192
A1.4: Première exemple d'un AIM de l'AP 203	193
A1.5: Deuxième exemple d'un AIM de l'AP 203	194

LISTE DES SIGLES ET ABRÉVIATIONS

A	Approved
AAM	Application Activity Model
AE	Application Element
AFNOR	Association Française de NORmalisation
AIAG	Automotive Industry Action Group
AIM	Application Interpreted Model
ANSI	American National Standard Institute
AP	Application Protocol
ARM	Application Reference Model
ASCII	American Standard Code Information Interchange
BREP	Boundary REPresenttion
C	Committee stage
C*	Committee distribution for comment
CAD	Computer Aided Design
CAD*I	Computer Aided Design Interfaces
CAD-LIB	Computer Aided Design Libraries
CAO	Conception Assistée par Ordinateur
CATIA	Computer Aided Three-dimensional Interactive Applications
CD	Committee Draft
CGM	Computer Graphics Metafile
CM	Configuration Management
CNRC	Conseil National de Recherches du Canada
CSG	Constructive Solid Geometry
CSTP	Centre Scientifique des Technologies du Bâtiment
CV	Computer Vision
D	Approval stage (Drafting international standard)

DIS	Drafting International Standard
DP	Draft Proposal
DPA	Digital Pre-Assembly
DXF	ASCII Drawing interchange (eXchange File)
EDIF	Electronic Design Interchange Format
EDM	Engineering Data Model
EDS	Electronic Data System
ESP	Experimental Solids Proposal
ETC	Executable Test Case
FTP	File Transfert Protocol
GE	General Electrics
GKS	Graphical Kernel System
GM	General Motors
GOSET	Groupe Opérationel SET (Standard d'Échange et de Transfert)
I	International standard
IEEE	Institute of the Electrical and Electronics Engineers
IFTI	Institut de Fabrication des Technologies Intégrées
IGES	Initial Graphics Exchange Specification
IPC	Institute for Interconnecting and Packaging of Electronic Circuits
ISO	International Standard Organization
ITFI	Institut des Technologies de Fabrication Intégrée
ITI	Industrial Technology Institute
ITW	Illinois Tool Works
IUT	Implementation Under Test
KIF	Knowledge Interchange Format
MIL-D-28000A	MILitary specification: Digital representation for communication of product data
NCMS	National Center for Manufacturing Sciences

NIST	National Institute of Standards and Technology
O	Priliminary stage
P	Proposal stage
PDDI	Product Definition Data Interface
PDES	Product Data Exchange using STEP
PDGS	Product Data Graphic System
PHIGS	Programmers' Hierarchical Interactive Graphics System
PICS	Protocole Implementation Conformance Statement
PIXIT	Protocole Implementation eXtra Information for Testing
PME	Petite et Moyenne Entreprise
PROE	PRO-Engineer
PTC	Parametric Techonology Corporation
RRM	Rapid Response Manufacturing
SC	Sub-Committee
SCRA	South Carolina Research Authority
SDAI	Standard Data Access Interface application
SDAI API	Standard Data Access Interface Application Programming Interface
SDRC	Structural Dynamics Research Corporation
SET	Standard d'Échange et de Transfert
ST	STEP Tools
STL	STéreoLithographie
STEP	STandard for Exchanging Product model data
SUT	System Under Test
SUTO	System Under Test Operator
TC	Technical Committee
TE	Test Engine
TR	Technical Report
TRW	Thomas thompson Ramo Woolridge

UG	UniGraphics
VDA-FS	Verband Der Automobilindustrie - Flächen Schnittstelle
VDA-IS	Verband Der Automobilindustrie - Inges Subset
VDA-PS	Verband Der Automobilindustrie - Programm Schnittstelle
VDM	Vienna Development Method
VDM-SL	Vienna Development Method - Specification Language
VHDL	VHSIC (Very High Speed Ics) Hardware Definition Language
VML	View Mapping Language
W	Preperatory stage (Working draft)
WB	WorkBench
XBF	Experimental Boundary File

LISTE DES EXEMPLES

Chapitre 2

2.1: Langage EXPRESS	22
2.2: Buts de tests	40
2.3: Buts de tests généraux	40
2.4: Cas de tests abstraits eb4 de la partie 304	41

Chapitre 3

3.1: Processus de traduction en format STEP pour CATIA	90
3.2: Processus de traduction en format STEP pour Pro-Engineer	90
3.3: Processus de traduction en format STEP pour Unigraphics	91
3.4: Processus de traduction en format IGES pour CATIA	93
3.5: Entête d'un fichier en format STEP	102
3.6: "Short names" et "long names"	104
3.7: Exemple d'actions correctives manuelles	130

CHAPITRE 1

INTRODUCTION

Le partage d'information entre entreprises, compagnies et sous-contractants, ainsi que la diversification des étapes de mise en marché d'un produit ont amenés l'utilisation d'échanges de données informatisées entre ces partenaires. Cette diversification est due à la spécialisation de certaines entreprises visant à produire un meilleur produit afin d'être plus compétitif sur le marché mondial.

Les échanges de données entre systèmes similaires ne causent généralement pas de problèmes, mais il y a des obstacles lorsque des échanges sont effectués entre systèmes différents. En effet, lorsque deux systèmes différents doivent échanger de l'information entre eux, des problèmes de communication surviennent puisque chacun des systèmes n'est pas structuré de la même manière. Pour résoudre ce type de problème, des interfaces directes ont été inventées. Ces dernières fonctionnent adéquatement lorsque peu de systèmes différents sont impliqués dans les échanges, mais à mesure que des systèmes distincts s'ajoutent, les problèmes se multiplient. Ce phénomène renvoie au concept des interfaces directes lequel exige que ce type d'interfaces existe entre chaque système différent. On constate alors que le nombre d'interfaces à développer augmente rapidement au fur et à mesure que le nombre de systèmes différents grossit.

Le problème du nombre important d'interfaces directes nécessaires lorsque plusieurs systèmes différents sont impliqués pour effectuer des échanges de données est résolu en partie. Depuis une vingtaine d'années plusieurs normes arborant le concept des interfaces normalisées, c'est-à-dire d'échanges de données par l'intermédiaire d'un format neutre, ont vues le jour. Toutefois, ces dernières présentent quelques carences dont, l'impossibilité de réaliser des échanges d'ordre sémantique. La nouvelle norme d'échange de données produit nommée ISO 10303 (STEP) vise à remédier à ces lacunes. Cette norme est plus complète que celles qui l'ont précédée et elle couvre plusieurs domaines tel, le secteur automobile, où il est possible d'effectuer des échanges de données produit.

Par ailleurs, la norme STEP exige l'exécution de tests afin de vérifier, où en est rendu son niveau d'implantation, ainsi que d'analyser sa fonctionnalité puisqu'elle est très récente. Elle a été approuvée au niveau de norme internationale en novembre 1994, mais elle demeure encore peu utilisée. Malheureusement, peu de tests ont été réalisés et aucune méthodologie d'analyse associée à la procédure d'exécution de ces tests existe. Pourtant, une telle méthodologie est plus efficace qu'une simple série de tests non structurés car, contrairement à ceux-ci, elle est indépendante des systèmes informatiques utilisés pour réaliser les tests.

La présente étude est principalement axée sur l'élaboration d'une méthodologie novatrice d'analyse de STEP pour sa mise en place. Elle consiste en la description de la démarche à suivre afin que l'exécution des tests soit menée adéquatement et ce, en partant de la sélection des objectifs des échanges en passant par les analyses d'informations jusqu'à la série de tests à réaliser. L'ensemble de la méthodologie renvoie essentiellement aux objectifs, définis au tout début, lesquels orientent le choix des tests à effectuer.

Au chapitre deux, il est question de la revue de la littérature concernant l'évolution dans le domaine des échanges de données produit ainsi que les normes existantes basées sur le principe d'échanges de données par des interfaces normalisées. La revue bibliographique fait aussi état de la description de la norme ISO 10303 (STEP), ainsi que de la description plus détaillée des tests de conformité faisant parties de la norme STEP. Enfin, ce chapitre se termine par un éventail de langages existants pour effectuer du "mapping", une présentation des facteurs assurant la réussite de mise en marché d'un produit pour les entreprises et, finalement, une présentation de trois plates-formes de tests qui ont été réalisées avec STEP.

Le chapitre trois présente la méthodologie d'évaluation de STEP pour sa mise en place, laquelle constitue l'apport de cette étude. Ce chapitre présente la pertinence du développement d'une méthodologie plutôt que la réalisation d'une autre plate-forme de tests. Par la suite, une description générale de la méthodologie est présentée, ainsi qu'une description détaillée de chacune des parties principales de celle-ci. Certains tests ont dû être réalisés afin de vérifier la pertinence de chacune des parties de la méthodologie et sont rapportés au chapitre suivant.

Le quatrième chapitre est consacré à la description de la plate-forme utilisée pour l'exécution des tests ainsi qu'à l'évolution de ces tests et du développement de la méthodologie. Il se termine par la présentation de la procédure suivie pour l'exécution des tests, de l'analyse de la méthodologie et des perspectives.

Finalement, la conclusion se veut être une synthèse de chacune des parties de ce mémoire et expose l'importance du développement d'une méthodologie ainsi que l'apport de cette étude au plan scientifique. Les limites et les perspectives y sont aussi présentées.

CHAPITRE 2

REVUE BIBLIOGRAPHIQUE

Ce chapitre a pour objectif de présenter les éléments qui ont contribué à la définition du problème et à la recherche de solution sur le développement d'une méthodologie d'évaluation de la norme STEP pour sa mise en place. En premier lieu, un bref historique portant sur l'évolution des échanges de données est présenté. Ensuite, un éventail des normes qui existent pour réaliser des échanges de données à l'aide d'un format neutre est exposé. Pour continuer, une explication sommaire de la norme ISO 10303 (STEP) ainsi qu'une description détaillée des tests de conformité de cette norme sont présentées. Au point suivant, on retrouve la description du domaine du "mapping" et des méthodes disponibles pour effectuer du "mapping". Ce chapitre se poursuit en exposant, la problématique d'un produit existant dans une entreprise. Finalement, les expériences réalisées en utilisant la norme STEP sont présentées.

2.1 ÉCHANGE DE DONNÉES

Il y a environ une quinzaine d'années, toutes les opérations de la mise sur le marché d'un produit, c'est-à-dire de la conception à la fabrication puis à la logistique, s'exécutaient plus souvent qu'autrement, à l'intérieur de la même compagnie. Les échanges de données informatisées s'effectuaient donc entre des postes de même type. La concurrence commerciale amena plusieurs compagnies, petites ou grandes, à diversifier leurs postes à l'intérieur de la compagnie et/ou voir même, à sous-traiter avec d'autres entreprises pour certaines phases. Ce phénomène, suscita des échanges de données entre postes hétérogènes puisque chaque entreprise utilise les systèmes adaptés à ses besoins (Xiaohu et al, 1993). Le problème fût partiellement résolu par l'emploi d'interfaces directes assurant le relai d'un poste à un autre. Rapidement, on se rendit compte que l'utilisation d'interfaces directes posaient des problèmes lorsque l'échange devait se faire entre plus de deux systèmes (Owen, 1993). C'est à ce moment que le recours aux interfaces normalisées par l'intermédiaire de formats neutres fut envisagé. Un format neutre se définit comme un format indépendant de tout logiciel, c'est-à-dire qu'il n'est pas

structuré selon le format de définition d'aucun logiciel en particulier. Grâce à cette symbolisation, on se rendit compte que les échanges de données pouvaient être réalisés plus facilement.

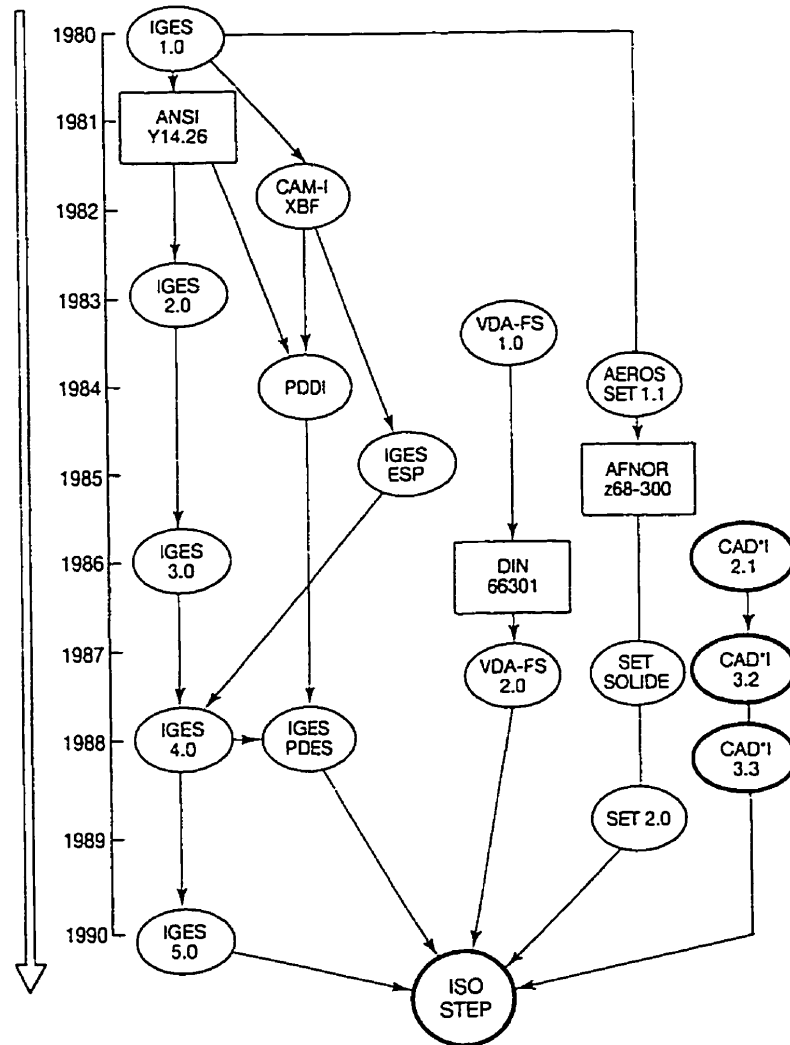
De nos jours, il est exceptionnel de voir toutes les opérations de mise en marché d'un produit s'effectuer à l'intérieur d'une même compagnie et, rarissime, de voir tout le travail de traitement de données s'effectuer à l'aide d'un réseau de postes identiques. Aussi, les échanges de données sont indispensables non seulement à l'intérieur d'une compagnie mais aussi entre elles. La qualité des échanges assure la bonne marche des projets et ce, à des fins d'efficacité eu égard au marché de la production.

2.2 NORMES EXISTANTES

Le principe de la normalisation n'est pas nouveau car il date de plusieurs décennies. Au cours des années, on a développé des normes propres aux divers champs disciplinaires. Le domaine des échanges de données techniques informatisées ne fait pas exception à la règle.

En effet, plusieurs normes ont été créés dans le domaine des échanges de données techniques. Dans l'ensemble des normes qui ont vu le jour, certaines ont eu un apport plus significatif que d'autres. Les premiers travaux effectués dans le domaine des échanges de données qui ont constitué une valeur significative au niveau de fichiers de format neutre, sont ceux qui ont mené à la formation de la norme "Initial Graphics Exchange Specification (IGES)", en 1979. Cette norme fût supportée par le "National Bureau of Standards" aux États-Unis et a été adoptée par ANSI en 1981. D'autres normes, travaux ou spécifications ont été développés par la suite dont SET, VDA-FS, CAD*I, XBF, ESP, EDIF, VHDL, IPC, VDA-PS, VDA-IS, DXF, GKS, CGM, PHIGS, PDDI, CAD-LIB et MIL-D-28000A (McMahon et Browne, 1993; Trapp, 1993; Owen,

1993). Une représentation chronologique de certains de ces projets visant l'établissement de normes est reproduite à la figure 2.1 qui suit.



Tirée de MCMAHON, C. et BROWNE, J. (1994). CAD/CAM From principles to practice, Addison-Wesley, 508 pages.

Figure 2.1 Évolution dans le temps des normes d'échanges de données

Plusieurs de ces normes, travaux ou spécifications furent conçus pour des objectifs bien

spécifiques. Parmi ces derniers, on peut penser à SET, VDA-FS, EDIF et VHDL, pour ne nommer que ceux-là.

La norme SET (McMahon et Browne, 1993) fut développée par l'industrie aérospatiale française qui cibra la résolution des difficultés de la "mise en oeuvre des traducteurs IGES par les vendeurs de systèmes CAO"¹. Elle est supportée par l'Association Française de NORmalisation soit, l'AFNOR. Quant à la norme VDA-FS, elle fut développée par l'industrie automobile allemande afin de surmonter la restriction que présentait IGES. En effet, VDA-FS se veut être une représentation de surfaces de plus haut niveau que celles cubiques de bases (McMahon et Browne, 1993). Quant à la spécification EDIF, elle fut créée par l'association des industries électroniques dans le but de faciliter l'échange d'information électronique de pièces spécifiques (puces intégrées IC). Finalement, le langage VHDL fut conçue par IEEE pour faciliter la simulation des circuits électroniques digitaux ainsi que leur conception jusqu'à un niveau très sophistiqué (Trapp, 1993).

Toutes les normes, travaux ou spécifications se rattachent à un ou plusieurs domaines et certaines de ces normes balisent un même domaine. De plus, deux normes peuvent présenter un caractère de similitude comme c'est le cas pour IGES et SET (Arbouy et al, 1994). Il arrive aussi que certains projets soient conçus pour des raisons précises et que ceux-ci se voient par la suite intégrés à d'autres projets comme c'est le cas pour XBF qui a été intégré à IGES ESP (McMahon et Browne, 1993). Enfin, chacun de ces projets présente des faiblesses plus ou moins importantes telles l'absence de représentation de surfaces complexes, les difficultés de mise en oeuvre, la taille disproportionnée des fichiers, pour ne nommer que celles-ci.

¹ Traduction libre.

2.3 GÉNÉRALITÉS SUR STEP

Cette partie contient l'ensemble du contenu de la norme STEP et la description de chacune de ses parties.

2.3.1 Structure générale

La structure générale consiste à situer la norme STEP dans ISO et à présenter les avancées qui ont contribué au développement de cette norme ainsi que ses aspects novateurs. Enfin, une description très sommaire de la norme STEP est formulée.

Situation dans ISO

STEP est élaboré sous la responsabilité du Sous-Comité 4 (Sub-Committee 4 - SC4) qui fait partie du Comité Technique 184 (Technical Committee 184 - TC184). Il est à noter que le comité technique TC184 représente l'un des quelques deux cents comités techniques de l'ISO et fut créé par ce dernier en 1983 (Arbouy et al, 1994). Le champ d'application du comité technique TC184 est relié aux systèmes d'automatisation industriels et à l'intégration «Industrial Automation Systems and Integration» (ISO/DIS 10303-1 TC184/SC4, 1993). Du côté du sous-comité SC4, son champ d'application est relié aux données industrielles et aux langages de programmation de fabrication globale "Industrial Data and Global Manufacturing Programming Languages". De plus, son objectif initial, qui demeure toujours valide, consiste en l'étude du secteur de la représentation et de l'échange des données numériques des produits.

Fondements et originalité de STEP

Plusieurs projets ont été créés avant que STEP “STandard for the Exchange of Product model data” ne prenne forme. En effet, des projets tels Computer-Aided Design Interfaces (CAD*I) et Product Data Exchange Specification (PDES) ont été élaborés vers la fin des années 80 et furent, par la suite, convertis par l’ISO en une “seule norme unifiée appelée STEP”² (McMahon et Browne, 1993). C’est à ces deux expériences que puise essentiellement le projet STEP (Richter, 1991). D’autres expériences comme celle d’IGES, de SET et de VDA-FS ont aussi contribué au développement de STEP (Vergeest, 1991). Notons que puisque c’est l’ISO qui développe STEP, cette norme porte aussi la codification de ISO 10303.

À ce sujet, Arbouy et al (1994) mentionnent que «le projet STEP est né d’un accord unanime et international de créer une norme ISO qui soit capable de représenter et d’échanger des modèles informatisés de produits sous une forme neutre». Le projet d’élaboration débuta vers le milieu des années ‘80 et, sa première présentation eu lieu en novembre 1988 par le comité TC184/SC4 comme une proposition d’essai (DP³) (Arbouy et al, 1994). En février 1993, STEP était au niveau de Standard International d’Essai, “Draft International Standard” (Evans, 1994). Ce n’est qu’en novembre 1994 que la norme STEP obtint le statut de norme internationale. Sa version initiale comprend douze parties qui sont les suivantes:

- Partie 1: Overview and fundamental principles
- Partie 11: EXPRESS language reference manual
- Partie 21: Clear text encoding of the exchange structure

² Traduction libre.

³ DP signifie Draft Proposal qui est maintenant connu sous le nom de Committee Draft, abréviation CD (Arbouy et al, 1994).

- Partie 31: Conformance testing methodology framework: General concepts
- Partie 41: Fundamentals of product description and support
- Partie 42: Geometric and topological representation
- Partie 43: Representation structures
- Partie 44: Product structure configuration
- Partie 46: Visual presentation
- Partie 101: Draughting
- Partie 201: Explicit draughting
- Partie 203: Configuration controlled design

La figure 2.2 ci-dessous présente une vue complète (STEP on a page) des parties de la norme, répertoriées par classe, accompagnées de leur nom et selon leur statut.

STEP on a Page

ISO 10303

(For an explanation of diagram, see opposite page)

APPLICATION PROTOCOLS	
<ul style="list-style-type: none"> I (101) Explicit Draughting C 202 Associative Draughting I (103) Configuration-Controlled Design C 204 Mechanical Design using Boundary Representation C 205 Mechanical Design Using Surface Representation 206 Mechanical Design Using Wireframe (discontinued) C 207 Sheet Metal Die Planning and Design W 208 Life Cycle Product Change Process W 209 Dgn Thru Anal of Composite & Metallic Structures C 210 Electronic Printed Circuit Assembly: Dgn & Mfg W 211 Electronic P C Assy: Test, Diagnostics, & Remanuf C* 212 Electrotechnical Design and Installation C 213 Num Control (NC) Process Plans for Machined Parts W 214 Core Data for Automotive Mech Design Processes W 215 Ship Arrangement 	<ul style="list-style-type: none"> W 216 Ship Moulded Forms W 217 Ship Piping W 218 Ship Structures 219 Dimension Inspection (discontinued) W 220 Printed Circuit Assembly: Mfg Planning W 221 Process Plant Functional Data & its Schematic Rep. W 222 Design-Mfg for Composite Structures W 223 Exg of Dgn and Mfg. Product Info for Cast Parts C* 224 Mech Parts Def. for P.Plg. Using Form Features W 225 Strct Bldg Elements using Explicit Shape Representation W 226 Ship's Mechanical Systems W 227 Plant Spatial Configuration W 228 Building Services: HVAC W 230 Building Structural Frame: Steelwork O Forged Parts O Multi-chip Modules O Process-Engineering Data

INTEGRATED INFORMATION RESOURCES		
INTEGRATED APPLICATION RESOURCES		
<ul style="list-style-type: none"> I (101) Draughting A 102 Ship Structures W 103 Electrical/Electronics Connectivity 	<ul style="list-style-type: none"> C 104 Finite Element Analysis C 105 Kinematics A 106 Bldg Core Constr. Model A Parametric Capability 	<ul style="list-style-type: none"> A Software Functionality Mechanical
INTEGRATED GENERIC RESOURCES		
I (41) Fundamentals of Product Description and Support		
I (42) Geometric and Topological Representations		
I (43) Representation Specialization	C 47 Tolerances	
I (44) Product Structure Configuration	48 Form Features (discontinued)	
C 45 Materials	C 49 Process Structure & Properties	
I (46) Visual Presentation		
IMPLEMENTATION METHODS		
I (21) Physical File, Exchange Structure	C 23 Early C++ (binding for #22)	
Working Format, Active Transfer	C 24 Late C (binding for #22)	
C 22 Standard Data Access Interface	C 25 Late FORTRAN (binding for #22)	
Knowledgebase		
O CORBA IDL 22 → 23 Mapping		

Legend: Part Status

- A=Approved to begin work
- O=Prelim. Stage (Proposal → approve NP ballot)
- P=Proposal Stg. (NP circ. → NP approval)
- W=Preparatory Stg. (Working Draft devel → CD reg.)
- C*=CD for Comment (WD issue prior to CD registr.)
- C=Committee Stg. (CD circ. → DIS registration)
- D=Approval Stg. (DIS circ. → Int'l Standard registr.)
- I=Publication Stg. (Int'l Standard approved & published)

Notes: STEP on a Page was conceptualized by JG Neil, National Institute of Standards and Technology, and is maintained and published by the U.S. Product Data Association. STEP on a Page may be copied and distributed freely for personal use. Any republication of STEP on a Page must contain this notice.

Origin: ISO 10303 Editing Committee

○ =STEP Initial Release

Figure 2.2 STEP sur une page

Comme il a été indiqué précédemment, STEP n'est pas la première norme conçue pour effectuer des échanges de données produit. D'autres normes sont accessibles sur le marché mais elles présentent des carences majeures. Par exemple, les normes précédentes spécifient un format de fichier et ne comprennent pas l'échange complet de la définition du produit. De plus, aucune ne permet, en pratique, l'échange de modèles solides et elles se limitent aux échanges au niveau syntaxique. La force de STEP repose sur sa capacité de palier aux lacunes des autres normes. Comme le stipule Arbouy et al (1994), cette force s'explique par le fait que STEP doit représenter toute l'information associée à la conception et la fabrication, dans toute leur diversité, afin de garantir la correspondance entre la représentation assimilable par la machine et celle interprétable par l'homme et ce, réciproquement.

Description de STEP

La norme ISO 10303 a été conçue pour répondre à un besoin spécifique soit la normalisation des données de définition de produits. Eu égard à cette mission, la norme STEP s'articule autour des 3 concepts: la normalisation, les données de définition et le produit. Son objectif, comme le mentionnent plusieurs auteurs (Brodsky, 1995; Huang et al, 1994; Zhao et al, 1996; Wang et Radack, 1995; Xiaohu et al, 1993) "est de fournir un mécanisme neutre capable de décrire les données de produit tout au long du cycle de vie d'un produit et ce, indépendamment de n'importe quel système particulier"⁴. L'atteinte de cet objectif rend non seulement possible le transfert de données neutres mais constitue aussi "le fondement de la mise en oeuvre et du partage de produit de bases de données ainsi que de l'archivage"⁴ (Xiaohu et al, 1993). En plus d'être capable de remplir ces fonctions, Owen (1993) mentionne que la norme STEP vise les objectifs suivants:

- la complétude: représentation complète;
- l'extensibilité: possibilité d'extension de la norme;

⁴ Traduction libre.

- la testabilité d'additions: revue attentive ainsi qu'une série de tests des additions possibles à la norme avant qu'elles y soient ajoutées;
- l'efficacité: efficacité de STEP relative à la taille des fichiers et aux ressources informatiques nécessaires pour l'analyser;
- la compabilité avec d'autres normes: compatibilité maximale avec d'autres normes afin de favoriser la migration;
- la redondance minimale: une seule façon de représenter un concept particulier;
- l'indépendance des environnements de calcul: indépendance des logiciels et du matériel de traitement de l'informatique;
- la classification logique d'éléments de données: définition des sous-ensembles d'implantation en regard de son angle large de saisie des domaines;
- la validation de mise en oeuvre: présence d'une structure de test de conformité afin de fournir des services pour effectuer ces tests.

Bref, la force de STEP réside dans le fait qu'elle est la résultante d'une élaboration fondée à partir des meilleurs aspects d'autres normes déjà élaborées. C'est pour cette raison que STEP a un champ très vaste d'application comme le souligne Trapp (1993). En effet, elle peut décrire des données de produits mécanique, électrique et de construction. De plus, elle présente des forces non-négligeables à l'intérieur des données de produits qui comprend l'information complète sur le produit lui-même, allant des caractéristiques géométriques à la nomenclature. On peut aussi compter parmi ses points forts, sa capacité de saisie globale des données de produits, ses fondements car elle s'appuie sur des protocoles d'applications (AP) pour son implantation, sa prise en compte de la nécessité de procéder à des tests de conformité pour les mises en oeuvre ainsi que l'inclusion d'un modèle de données sémantiques distinct des instances (data instances) (Trapp, 1993). En outre, STEP possède un langage permettant non seulement de satisfaire ces exigences mais aussi permettant à la syntaxe intégrée d'être compréhensible par l'homme et lisible par les ordinateurs. Ce langage constitue un standard et porte le

nom d'EXPRESS. Une description plus exhaustive de ce langage est présentée plus loin dans cette section.

Le survol de la norme STEP étant posée, il faut maintenant examiner le contenu informationnel de celle-ci. Cette norme qui couvre une plage assez grande d'information, c'est-à-dire plusieurs types d'information distinctes, a été divisée en différentes classes afin d'accroître et de faciliter non seulement sa compréhension mais aussi son développement. Il y a sept classes lesquelles possèdent un nom spécifique ainsi qu'une étendue bien définie des parties qu'elles englobent. Les noms des classes avec les parties couvertes par classe constituent les éléments du tableau 2.1 présenté ci-dessous.

Tableau 2.1 Description des classes de STEP

NOM DE LA CLASSE	PLAGE DES PARTIES COUVERTES
Introduction	1-9
Méthodes de description	11-19
Méthodes de mise en oeuvre	21-29
Méthodes de tests de conformité	31-39
Modèles d'information de ressources intégrées	41-99, 101-199
Protocoles d'application	201-299
Suites de tests abstraites	301-399

La première classe renvoie au survol de la norme. Les nominatifs des autres classes éclairent sur leur contenu, cependant, une brève description s'avère nécessaire à la compréhension du lecteur. Aussi est-elle présentée un peu plus loin.

2.3.2 Protocoles d'application

Vu que la norme STEP couvre plus d'un domaine, il est quasi impossible d'avoir un seul standard d'échange de données. C'est pour cette raison que les protocoles d'application (AP) ont été créés. Ces protocoles d'application, dont on fait état dans STEP, sont des parties (201 à 299) qui définissent toutes les données informatives que STEP doit contenir à propos d'un domaine d'application particulier pour effectuer des échanges de données (Brodsky, 1995; Trapp, 1993). Les protocoles d'application contiennent aussi la procédure d'utilisation de ces données informatives. Il faut mentionner que les protocoles d'application sont composés de quatre sections principales:

- 1) l'objectif et champ d'application, connu aussi sous le nom de modèle d'activité d'application (Application Activity Model - AAM)
- 2) le modèle d'application de référence (Application Reference Model - ARM)
- 3) le modèle d'application interprété (Application Implementation Model - AIM)
- 4) les exigences de conformité, les buts des tests ainsi qu'une suite de tests abstraits (Abstract Test Suite) (Owen, 1993; Trapp, 1993; Xiaohu et al, 1993).

Le AAM définit le procédé industriel et la pertinence d'un AP. Il est fourni afin d'aider à la compréhension du domaine et des exigences d'information définies dans un AP. Il est représenté comme une série de définitions pour les activités et les données sous forme de diagrammes. La représentation d'un AAM couvre des activités qui vont au-delà du sujet de l'AP. Les diagrammes utilisés pour représenter le AAM sont sous forme IDEF0 comme illustré sur la figure ci-dessous.

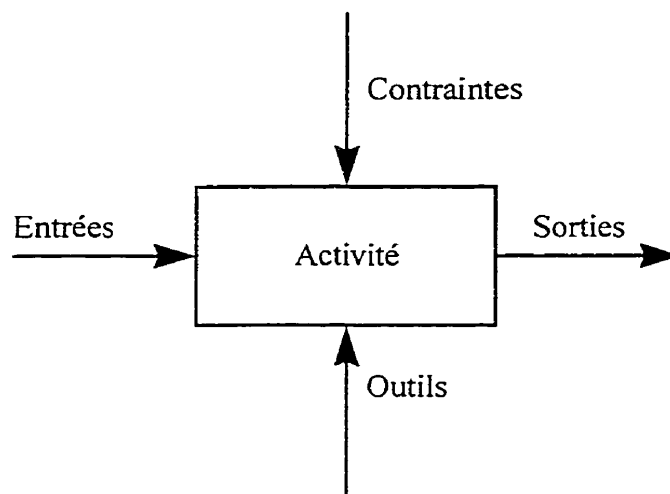


Figure 2.3 Diagramme IDEF0

La figure A1.1 de l'annexe A1 présente un exemple d'un AAM. Lorsqu'une activité doit être détaillée, elle est définie sur une autre feuille comme illustré à la figure A1.2 de l'annexe A1.

Le ARM définit les données d'un AP en termes industriels. Il est représenté sous forme de diagrammes IDEF1X qui contiennent la structure et les contraintes des objets d'application de l'AP. Un ARM contient aussi les grandes lignes de la structure de la base de données globale. La figure A1.3 de l'annexe A1 illustre un diagramme du ARM.

Le AIM transfère les données contenues dans le ARM en ressources STEP. Il contient les informations de la structure de la base de données de l'ARM en plus des informations détaillées des ressources en lien avec STEP. Le contenu d'un AIM est représenté sous forme de diagrammes EXPRESS-G comme illustré sur les figures A1.4 et A1.5 de l'annexe A1. Chaque AP doit contenir au moins un AIM. En certains cas, il y en aura plusieurs. De plus, chaque AP doit contenir une suite de tests abstraits qui, à son tour, doit contenir des cas de tests abstraits.

Actuellement, il y a une trentaine de protocoles d'application répertoriés. Leur statut va de l'approbation pour développement au stade de norme internationale. Le protocole d'application le plus utilisé actuellement est l'AP 203 car il permet l'échange d'information à propos des données de configuration de produit (Configuration Controlled Design) avec ou sans la forme de l'objet. Il y a 6 niveaux qui sont compris dans ce AP. Le premier doit être appuyé par une implantation et les suivants constituent des types de représentation de la forme d'une partie. Ces 6 niveaux sont les suivants:

1. Construit ("constructs") représentant l'information de la conception de configuration contrôlée sans forme;
2. Modèles filaires qui n'ont pas de topologie de surface;
3. Modèles filaires avec topologie;
4. Modèles de surfaces variées avec topologie;
5. Représentation par frontières (Boundary representation) à facettes;
6. Représentation par frontières (Boundary representation) avancée.

En plus de ces 6 niveaux, l'étendue de l'AP 203 englobe l'ensemble de l'information relative aux phases de la conception d'un produit telles le changement d'un plan, les données nécessaires au repérage de libération d'un plan (design's release), les données reliées aux phases de développement d'un produit, pour ne nommer que celles-là. De plus, l'ensemble des pièces et des assemblages mécaniques ainsi que l'identification des classifications de sécurité de pièces ou de ces assemblages font partie de l'étendue de l'AP 203.

Le tableau 2.2 ci-dessous présente la liste des protocoles d'application ainsi que leur statut respectif.

Tableau 2.2 Statut des protocoles d'application

# DE L'AP	NOM DU PROTOCOLE D'APPLICATION	STATUT
201	Explicit Draughting	Norme Internationale
202	Associative Draughting	Stade du Comité
203	Configuration-Controlled Design	Norme Internationale
204	Mechanical Design using Boundary Representation	Stade du Comité
205	Mechanical Design using Surface Representation	Stade du Comité
207	Sheet Metal Die Planning and Design	Stade du Comité
208	Life Cycle Product Change Process	Stade Préparatoire
209	Design Thru Anal of Composite & Metallic Structures	Stade Préparatoire
210	Electronic Printed Circuit Assembly: Design & Manufacturing	Stade du Comité
211	Electronic Printed Circuit Assembly: Test, Diagnostics & Remanufacturing	Stade Préparatoire
212	Electronical Design and Installation	Stade du Comité pour commentaires
213	Numerical Control Process Plan for Machined Parts	Stade du Comité
214	Core Data for Automotive Mechanical Design Process	Stade du Comité
215	Ship Arrangement	Stade Préparatoire
216	Ship Moulded Forms	Stade Préparatoire
217	Ship Piping	Stade Préparatoire
218	Ship Structures	Stade Préparatoire
220	Printed circuits Assembly: Manufacturing Planning	Stade Préparatoire
221	Process Plant Functional Data & its Schematic Representation	Stade Préparatoire
222	Design-Manufacturing for Composite Structures	Stade Préparatoire
223	Exchange of Design and Manufacturing Product Information for Cast Parts	Stade Préparatoire

Tableau 2.2 (suite) Statut des protocoles d'application

# DE L'AP	NOM DU PROTOCOLE D'APPLICATION	STATUT
224	Mechanical Parts Definition for Process Planning Using form Features	Stade du Comité pour commentaires
225	Structure Building Elements using Explicit Shape Representation	Stade Préparatoire
226	Ship's Mechanical Systems	Stade Préparatoire
227	Plant Spatial configuration	Stade Préparatoire
228	Building Services: HVAC	Stade Préparatoire
229	Forged Parts	Stade Préliminaire
230	Building Structural Frame: Steelwork	Stade Préparatoire
231	Process Engineering Data	Stade Préliminaire
232	Multi-Chip Modules	Stade Préliminaire

2.3.3 Méthodes de description

Les méthodes de description ont pour but d'uniformiser les méthodes utilisées pour dépeindre les éléments de bases qui permettent de représenter de l'information. De plus, elles donnent à STEP un caractère d'originalité car elles normalisent non seulement le format de données, mais aussi le langage pour décrire le format de données soit le langage nommé EXPRESS. Vu la flexibilité de la norme, il est plausible d'anticiper que celle-ci évoluera dans le temps et s'adaptera aux différentes applications. Aspect non-négligeable.

Dans cette classe, méthodes de description, on retrouve l'explication relative au langage EXPRESS. Il est important de préciser qu'EXPRESS n'est ni un langage de programmation ni une méthodologie (Owen, 1993). Il est plutôt conçu afin d'offrir une «représentation formelle des données de définition de produits» comme le rappelle Arbouy

et al (1994). EXPRESS est alors vu comme un “langage de spécification de données formelles qui fournit le mécanisme pour la description normative des données de produit pour les ressources intégrées et les protocoles d’application”⁵ (ISO/DIS 10303-1 TC184/SC4, 1993).

Afin de visualiser en quoi consiste le langage EXPRESS, voici un exemple qui est présenté à l’aide de diagrammes EXPRESS-G et d’un schéma EXPRESS.

Exemple 2.1 Langage EXPRESS:

Diagrammes EXPRESS

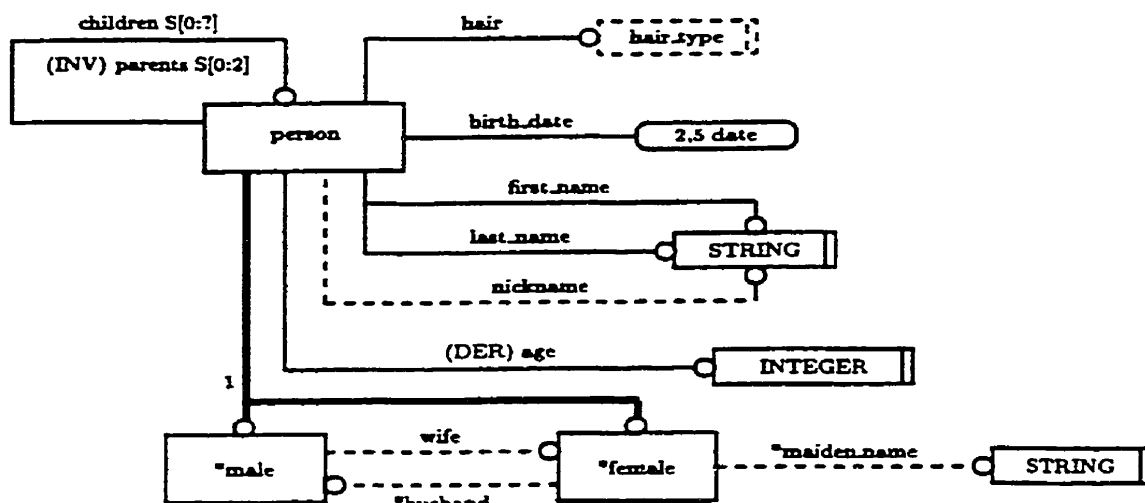


Figure tirée de ISO/DIS 10303-11 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 11: Description methods: EXPRESS language reference manual, ISO, Genève, 208 pages.

Figure 2.4 Exemple de diagramme EXPRESS sous forme EXPRESS-G (page 1 de 2)

⁵ Traduction libre

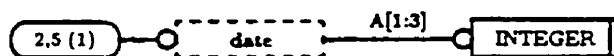


Figure tirée de ISO/DIS 10303-11 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 11: Description methods: EXPRESS language reference manual, ISO, Genève, 208 pages.

Figure 2.5 Exemple de diagramme EXPRESS sous forme EXPRESS-G (page 2 de 2)

Schéma EXPRESS;

SCHEMA exemple;

```

TYPE data = ARRAY [1:3] OF INTEGER;
END_TYPE;

```

```

TYPE hair_type = ENUMERATION OF
    (blonde, brown, black, red, white, bald);
END_TYPE;

```

ENTITY person

SUPERTYPE OF (ONEOF(female, male));

first_name : STRING;

last_name : STRIGN;

nickname : OPTIONAL STRIGN;

birth_date : date;

children : SET [0:?] OF person;

hair : hair_type;

DERIVE

age : INTEGER := years(birth_date);

INVERSE

```

    parents : SET [0:2] OF person FOR children;
END_ENTITY

```

```

ENTITY female

```

```

    SUBTYPE OF (person);
    husband      : OPTIONAL male;
    maiden_name  : OPTIONAL STRING;
WHERE
    w1: (EXISTS(maiden_name) AND EXISTS(husband)) XOR
        NOT EXISTS(maiden_name);
END_ENTITY;

```

```

ENTITY male

```

```

    SUBTYPE OF (person);
    wife : OPTIONAL female;
END_ENTITY;

```

```

RULE married FOR (female, male);

```

```

    (* checks pairwise relationship between spouses *)
WHERE
    r1 : SIZEOF(
        QUERY(tf <* female | EXISTS(tf.husband) AND
            (tf.husband.wife :<>: tf))
        ) = 0;
    r2 : SIZEOF(
        QUERY(tm <* male | EXISTS(tm.wife) AND
            (tm.wife.husband :<>: tm))
        ) = 0;

```

```
END_RULE;
```

```
FUNCTION years(past : date); INTEGER;
```

```
(* This function calculates the number of years
   between the past date and the current date *)
```

```
END_FUNCTION;
```

```
END_SCHEMA;
```

Voici des éléments concrets susceptibles d'éclaircir l'exemplaire de langage EXPRESS dont il a été question. "Une personne a certaines caractéristique définies, incluant un prénom, un nom, parfois un surnom, une date d'anniversaire et une description de sa chevelure. Une personne est du sexe féminin ou masculin. Un homme peut avoir une femme; une femme peut avoir un mari ainsi qu'un nom de jeune fille. Une personne peut avoir des enfants qui eux, sont aussi des personnes. Des contraintes sont associées au mari et au nom de jeune fille ainsi qu'à l'homme et à la femme"⁶ (ISO/DIS 10303-11 TC184/SC4, 1993).

2.3.4 Méthodes de mise en oeuvre

L'élaboration de méthodes de mise en oeuvre a été réalisée suite à des problèmes de connaissances liés à une application spécifique, survenu lors d'échange de données. Ces méthodes ont été formulées dans le but de dicter la procédure spécifique à suivre lorsque les protocoles d'applications doivent être utilisés sous forme informatisée. Elles ne requièrent, en aucun cas, des connaissances à propos de l'application en question. De plus, il existe plus d'un facteur, reconnu à ce jour, influençant les méthodes de mise en oeuvre de STEP. Parmi tous ces facteurs, deux sont facilement identifiables: la

⁶ Traduction libre.

technologie concernant les processeurs développés d'échange de données et l'architecture des systèmes de gestion des bases de données (Richter, 1991).

La classe des méthodes de mise en oeuvre (parties 21 à 29) décrit les méthodes utilisées pour représenter ou transformer en une notation formelle les données sous forme du langage EXPRESS. À l'intérieur de la norme ISO 10303, quatre méthodes de mise en oeuvre ont déjà été identifiées en ces termes:

- 1) les échanges de fichiers, c'est-à-dire les échanges de fichiers physiques;
- 2) le partage d'une information en mémoire vive d'un ordinateur;
- 3) les bases de données qui renvoient à l'accès directe aux données par des interfaces normalisées d'accès aux données (Standard Data Access Interface - SDAI);
- 4) les échanges de données fondés sur la connaissance qui vise à augmenter le niveau de sémantique afin d'atteindre l'étape du stockage de connaissances.

À ce jour, la méthode la plus utilisée est de loin celle d'échanges de fichiers, suit celle SDAI qui est parfois employée. Dans ce cas, quelques compilateurs tels C++, C et FORTRAN sont disponibles. Pour mieux comprendre ce que constitue un fichier STEP, l'annexe A2 présente le fichier en format de la partie 21 d'une vis de fixation.

2.3.5 Modèles d'information de ressources intégrées

Les modèles d'information de ressources intégrées, comme leur nom le suggère, fournissent des modèles d'information aux ressources définies de manière exhaustive et à partir desquelles sont bâtis les protocoles d'applications. L'information relative à un produit tout au long de son cycle de vie, sous forme de représentation générale, peut se décrire comme étant un modèle d'information de ressources intégrées (Xiaohu et al,

1993). Modèle qui permet l'intégration des protocoles d'application par des données de bases communes.

La classe des modèles d'information de ressources intégrées est divisée en deux principales parties: les ressources génériques et les ressources d'applications. Ces deux parties sont regroupées sous la même classe car elles sont, pour les initiateurs de la norme, au même niveau d'utilisation (Arbouy et al, 1994). Par ailleurs, il faut spécifier que les ressources génériques (parties 41 à 99) ont une application générale et sont indépendantes des applications. Quant aux ressources d'applications (parties 101 à 199), elles ont pour objet le support d'une application ou une plage d'applications similaires (Owen, 1993). De plus, elles se réfèrent et prolongent les ressources génériques puisqu'elles sont spécifiques et adaptées eu égard au domaine.

2.3.6 Méthodes de tests de conformité

Les tests de conformité ont été créés afin d'assurer une mise en oeuvre conforme aux exigences précises d'un AP. Les exigences d'un AP sont remplies par le recours à une suite de tests abstraits.

En ce qui concerne les parties des méthodes de tests de conformité (31 à 39), elles couvrent toute l'information concernant: les documents à utiliser pour effectuer les tests; les renseignements sur les laboratoires qui peuvent effectuer ces tests; la procédure à suivre pour devenir un laboratoire accrédité capable d'exécuter ces tests; et les renseignements sur les personnes susceptibles de sanctionner des accréditations.

La manière pour effectuer adéquatement des tests de conformité renvoie à trois critères essentiels. En effet, les tests doivent pouvoir se répéter, être compatibles et fiables, c'est-

à-dire que les tests doivent être comparables peu importe quand et où ils ont été effectués et que la procédure a été suivie de manière rigoureuse (Trapp, 1993).

2.3.7 Suites de tests abstraits

La classe dont il est question ici a été créée en regard des protocoles d'application. En effet, chaque protocole d'application doit posséder des données de test. Ces dernières constituent les suites de tests abstraits et sont utilisées afin de mettre à l'essai le degré de conformité de produits développés associés à un AP particulier (Nell, 1995). En somme, ces suites de tests ont été élaborées afin de vérifier les produits développés (Arbouy et al, 1994) et se définissent ainsi.

Une suite de tests abstraits se définit comme étant "l'ensemble complet des cas de tests abstraits qui inclut tous les buts nécessaires à l'implantation d'un protocole d'application"⁷ (ISO/CD 10303-33 TC184/SC4, 1995). Une suite de tests abstraits est composée de plusieurs éléments dont les cas de tests abstraits. Ces cas sont, à leur tour, composés de plusieurs éléments dont les spécifications d'entrées et les critères de verdicts qui sont cernés aux paragraphes suivants.

Les spécifications d'entrée renvoient à "une brève description des exigences de données de tests pour les pré-processeurs et les post-processeurs pertinents pour les cas abstraits d'évaluation"⁸ (ISO/CD 10303-33 TC184/SC4, 1995). Ce sont ces spécifications qui, sans attribuer des valeurs spécifiques, permettent de définir les concepts ainsi que de choisir les relations à évaluer. Il faut spécifier que chaque spécification doit être complète, limpide et indépendante du système sous évaluation.

⁷ Traduction libre.

Quant aux critères de verdict, ils doivent être écrits dans un langage naturel et une forme conventionnelle peut leur être associée pour fins de clarification.

En plus de l'ensemble de l'information qui est contenue dans les cas de tests abstraits, on retrouve parfois de l'information additionnelle. Cette information additionnelle est présente afin d'assurer la clarté des cas de tests. C'est dans ce but qu'elle contient des détails à propos de l'ordre d'exécution de l'évaluation et de la séquence de construction.

La classe des suites de tests abstraits couvre les parties 301 à 399. Il faut signaler que les chiffres des parties couvertes dans cette classe ont une signification particulière. En effet, puisque chaque protocole d'application doit posséder une suite de tests abstraits, les deux derniers numéros de la suite sont les mêmes que ceux du protocole d'application auquel est relié cette suite. Il faut mentionner qu'en ajoutant la somme de cent à un AP, on obtient le numéro de la partie de la suite de tests abstraits qui lui est associé.

2.3.8 État d'avancement et perspectives de la norme STEP

Comme mentionné précédemment aux pages 11 et 12, douze parties sont officiellement reconnues au stade de norme internationale. Toutefois, avant d'être reconnu comme tel, chaque partie doit être soumise à un processus d'approbation. À cet égard, Nell (1995), propose huit phases: l'approbation pour commencer le travail (A), le stade préliminaire (O), le stade de proposition (P), le stade préparatoire (W), le stade de comité (C), le stade de comité pour commentaires (C*), le stade d'approbation, c'est-à-dire de circulation DIS (D), et finalement, le stade de publication où une partie devient norme internationale approuvée (I). La figure 2.2, STEP sur une page, située à la section 2.3.1 présente une vue d'ensemble des parties de STEP, divisées par classes, identifiées (nom) et avec leur statut.

Comme on peut le remarquer sur la figure 2.2, il reste encore beaucoup de travail à effectuer avant que la majorité des parties de STEP atteignent le niveau de norme internationale. De plus, au fil des années, il y aura sûrement d'autres parties qui viendront s'y greffer et d'autres, qui disparaîtront puisque STEP est une norme en constante évolution. Il faudra être vigilant concernant la progression de la norme afin que son développement s'effectue de façon productive (Laurance, 1994).

D'après Crume (1993), «STEP lancera la prochaine révolution industrielle». Cette affirmation est peu réaliste et trop générale car la norme évolue à un rythme très rapide et des changements ne peuvent y être apportés au même rythme puisqu'une norme internationale ne peut être modifiée au cours des cinq années suivant sa date d'approbation. Cependant, même si aucune modification ne peut être apportée, des ajouts (ex: parties additionnelles) peuvent être insérés à la norme. De plus, la migration de l'utilisation d'autres normes existantes vers STEP, peut se faire de manière progressive et pondérée. Par ailleurs, à l'instar de Thu-Hua et Fisher (1995), il est plausible d'anticiper que STEP contribuera grandement à l'opérationnalisation de la technique de l'ingénierie simultanée, qui on le sait, s'avère de plus en plus populaire

2.4 TESTS DE CONFORMITÉ

Cette partie consiste à décrire de façon détaillée le contenu des tests de conformité de la norme STEP. Ces tests constituent la classe garantissant des bases valides pour réaliser l'évaluation de conformité. Ils se définissent comme une méthodologie, une procédure générale permettant d'exécuter la vérification d'une mise en oeuvre de la norme ISO 10303 (ISO/DIS 10303-31 TC184/SC4, 1993). À l'intérieur de ces tests, il y a différentes phases permettant d'effectuer la vérification lesquelles sont constituées des trois tronçons majeurs dont nous faisons état: la définition des suites de tests abstraits (abstract test suites) associés aux protocoles d'applications respectifs; la description des méthodes de

tests abstraits (abstract test methods) associés aux méthodes de mise en oeuvre respectifs; et finalement, la procédure d'évaluation de conformité effectuée par les gens d'un laboratoire pour un client spécifique.

Il faut spécifier que pour réaliser les tests de conformité, trois critères doivent être respectés: il faut qu'à l'intérieur des tests on atteigne un niveau de conformité convenable, c'est-à-dire que peu importe l'endroit ou l'heure à laquelle les tests sont effectués, les résultats doivent être comparables et la communication entre les parties responsables des activités mentionnées ci-haut doit être facilitée.

Cette description commence par une présentation des éléments centraux des tests de conformité, se poursuit avec des spécifications sur les méthodes de tests abstraits et se termine par des spécifications sur les suites de test abstraits.

2.4.1 Éléments centraux

La signification de la conformité selon la norme STEP, l'évaluation ainsi que les procédés d'analyses de la conformité sont abordés dans la présente section.

Les tests de conformité ont été instaurés dans le but de vérifier si une implantation, reliée à un AP, est conforme aux exigences de la norme 10303. Une implantation est "conforme" lorsqu'elle «répond aux exigences d'une règle, d'une norme» selon le Petit Larousse illustré (1998).

Reconnaissant cette nécessité, il faut examiner la signification de la conformité dans le cadre de la norme STEP.

La signification de la conformité dans ISO 10303

Pour qu'une mise en oeuvre soit conforme, il faut qu'elle ait été faite en référence à un protocole d'application. De plus, chaque mise en oeuvre doit présenter une méthode spécifique afin de fournir un procédé par lequel les implantations peuvent être observées et contrôlées lors de l'exécution des tests.

Il convient aussi de mentionner que lorsque l'on veut effectuer la procédure de conformité, on se retrouve face à des exigences. Cependant, parmi ces exigences, certaines sont obligatoires (à respecter en tout temps), d'autres sont conditionnelles (à respecter si des conditions sont imposées) et certaines sont optionnelles (choisies pour accommoder la mise en oeuvre). Il y a donc une hiérarchie. De plus, ces exigences peuvent être décrites de façon positive ou négative (négative en raison de l'interdiction). On entend par positive, une exigence qui doit être réalisée et par négative, une exigence de non réalisation.

Afin de procéder à l'évaluation de conformité d'une implantation précise, par rapport à des exigences, un document comportant l'ensemble des options implantées doit être fourni. Ce document porte le nom de rapport de conformité d'implantation de protocole (PICS \Rightarrow Protocol Implementation Conformance Statement). Le PICS est utilisé par les gens du laboratoire évaluateur et ce, dans le but de mieux comprendre le système sous évaluation (SUT \Rightarrow System Under Test) et d'aider à établir les limites du domaine testé. Il faut spécifier que le SUT se définit comme étant le matériel informatique et du réseau de communication nécessaire au support de l'implantation sous test (IUT \Rightarrow Implementation Under Test).

Comme mentionné précédemment, une implantation est dite conforme si elle satisfait aux exigences de conformité du protocole d'application implanté. Outre ce critère général,

elle doit, de plus, respecter le PICS, c'est-à-dire qu'elle doit répondre aux tests exécutés pour vérifier la conformité d'un protocole d'application.

Évaluation de conformité

L'évaluation de conformité est la procédure par laquelle un produit doit passer afin que soit vérifié si certaines caractéristiques requises par une norme sont présentes et ce, dans le but de déterminer si son implantation est conforme. Plusieurs éléments constituent cette partie de la procédure: les types de tests de conformité, le rapport d'évaluation d'information supplémentaire pour l'implantation du protocole (PIXIT \Rightarrow Protocol Implementation eXtra Information for Testing), lequel est défini plus loin dans cette section, et finalement, le procédé d'analyse de conformité.

Le procédé d'évaluation de conformité doit être élaboré de façon à ce que les résultats de l'évaluation soient répétables, c'est-à-dire que les résultats sont indépendants du temps, comparables dans le sens que les résultats sont indépendants du laboratoire évaluateur (résultats pareils ou semblables), conformes, c'est-à-dire que les procédures ont été suivies rigoureusement.

Il existe deux types de test de conformité: les tests de bases et les tests de capacité dont il est question dans les paragraphes qui suivent.

Tests de bases

Ces tests présentent l'affirmation initiale que IUT est conforme. Ils sont optionnels et peuvent être utilisés durant les préparations de l'évaluation. Ces tests sont appropriés pour détecter la non-conformité de cas sévères (ex: défaillance d'un logiciel) ainsi que pour déterminer la pertinence du recours ou non aux tests de capacité (comme étape

préliminaire) et pour déterminer si l'échange de communication entre la mise en oeuvre à l'étude et d'autres mises en oeuvre conformes peut être possible. Toutefois, ces tests ne sont pas appropriés pour évaluer si l'implantation est conforme et pour détecter la cause de défaillance d'inter-opérabilité.

Tests de capacité

Ces tests permettent de vérifier si les capacités observables d'une IUT sont conformes avec celles exigées dans le PICS. L'utilisation des tests de capacité est appropriée pour vérifier la cohérence des capacités d'une IUT par rapport aux exigences de conformité, pour évaluer si la mise en oeuvre est identifier et identifier les causes de défaillance d'inter-opérabilité. Toutefois, ces tests présentent des limites car ils ne sont pas appropriés pour l'évaluation exhaustive du comportement de chaque exigence implantée ainsi que pour garantir l'inter-opérabilité.

Afin de pouvoir évaluer une implantation, c'est-à-dire d'effectuer les tests nécessaires, certaines informations supplémentaires vis-à-vis de l'IUT et de son environnement sont requises. Ces informations sont inscrites dans le rapport d'évaluation des informations supplémentaires utiles à l'implantation du protocole (PIXIT \Rightarrow Protocol Implementation eXtra Information for Testing). Pour fins de compréhension, il convient de mentionner que le PICS définit l'information pour le domaine sous évaluation en laboratoire tandis que le PIXIT fournit l'information concernant la procédure de l'évaluation. Une précision s'impose, le PIXIT ne doit pas entrer en conflit avec le PICS qui lui est associé.

Procédé d'analyse de conformité

Le procédé d'analyse de conformité consiste en quatre phases: la préparation pour évaluation, la réalisation de l'évaluation, l'analyse des résultats et la production du rapport d'évaluation de conformité.

Préparation pour l'évaluation

La préparation pour l'évaluation est constituée de huit phases lesquelles sont présentées ci-dessous:

- ♦ produire l'information administrative;
- ♦ produire le PICS et le PIXIT;
- ♦ identifier la méthode de tests abstraits et la suite de tests abstraits;
- ♦ réviser le PICS pour voir s'il respecte les exigences de conformité applicables;
- ♦ réviser le PIXIT par une vérification consistante vis-à-vis du PICS;
- ♦ sélectionner la première version des cas de tests abstraits et des valeurs de paramètre basés sur le PICS et PIXIT;
- ♦ préparer le SUT;
- ♦ sélectionner la version finale des cas de tests abstraits et des valeurs de paramètre basés sur le PICS et PIXIT.

Une fois franchis les huit phases ci-haut présentées, le but du procédé d'évaluation de conformité ainsi que l'IUT sont fixés de manière définitive.

Réalisation de l'évaluation

La réalisation de l'évaluation renvoie au procédé selon lequel la suite de tests exécutables est réalisée. Cette suite de tests exécutables consiste en un ensemble complet de cas de

tests exécutables qui doivent subir des tests de conformité pour une ou plusieurs normes. Ce cas de tests exécutables est une instantiation, c'est-à-dire une version plus définie d'un cas de tests abstraits avec des valeurs.

C'est aussi à cette étape de l'évaluation que les résultats ainsi que toutes les informations pertinentes sont inscrites dans le registre de conformité.

Analyse des résultats

L'analyse des résultats est effectuée selon une procédure de comparaison entre les résultats de sortie et les critères de verdict. À la suite de cette analyse, un verdict d'évaluation doit être associé. Ce verdict renvoie à l'une des trois options suivantes:

- ♦ réussi: tous les résultats sont conformes aux exigences;
- ♦ échoué: un ou plusieurs résultats ne sont pas conformes aux exigences;
- ♦ non-concluant: aucun verdict de réussite ou d'échec n'a pu être donné.

En ce qui concerne les verdicts échoués et non-concluants, une justification doit être associée audit verdict.

Production du rapport d'évaluation de conformité

Les résultats des tests de conformité sont colligés dans un rapport. Ce rapport doit être divisé en deux parties un résumé qui consiste en un sommaire général relatif au statut de conformité de l'IUT, une partie informative détaillée laquelle consiste en la description des verdicts liés aux cas de tests traités lors du procédé d'évaluation de conformité.

2.4.2 Méthodes de tests abstraits

Dans les méthodes de tests abstraits on peut trouver la description des différentes étapes à suivre, allant de la sélection des cas de tests abstraits à la production de leur rapport, utiles pour effectuer la suite de tests abstraits. Comme mentionné précédemment, ces méthodes servent de cadre de référence pour l'exécution des tests de conformité d'une implantation.

Deux méthodes sont présentées dans cette partie car il y a deux implantations qui, lors de cette étude, sont sous évaluation: les méthodes qui génèrent des structures d'échange, aussi connu sous le nom de pré-processeurs, et les méthodes qui interprètent les structures d'échange, aussi connu sous le nom de post-processeurs. Ces deux méthodes ont un même point commun: elles sont toutes les deux des méthodes de structure d'échange.

Les méthodes en question sont basées sur trois principales composantes qu'il faut présenter:

- ♦ l'implantation sous évaluation (IUT): l'implantation à être évaluée;
- ♦ la machine de test (TE \Rightarrow Test Engine): les fonctions du système pour l'évaluation;
- ♦ l'opérateur du système sous évaluation (SUTO \Rightarrow Système Under Test Operator): un humain ou une composante animée (automated component) entre l'IUT et le TE.

De plus, elles constituent la procédure générale qui présente les grandes lignes à suivre de la démarche d'évaluation pour les pré-processeurs et les post-processeurs. Les quatre procédés qui sont présentés sont les mêmes que ceux retenus pour l'analyse de conformité dont il a été question dans la partie traitant des éléments centraux:

- la préparation pour l'évaluation: identification et sélection de variables et de valeurs paramétriques;

- la réalisation de l'évaluation: exécution séquentielle des cas de tests exécutables visant qu'à chaque but de test un verdict lui soit associé;
- l'analyse des résultats: attribution de verdicts en regard des résultats de sortie;
- la production du rapport: rédaction du rapport en adéquation avec l'évaluation effectuée.

Il importe maintenant de présenter la procédure utilisée pour réaliser les implantations en question.

Pré-processeurs

La procédure des pré-processeurs commence par la création du modèle constitué d'une description textuelle et/ou graphique du modèle exemplaire provenant d'une part, d'un ETC et d'autre part, du système évaluateur. La procédure se poursuit par la génération de la structure d'échange qui consiste en la génération, par l'IUT, du modèle d'entrée en format ISO 10303-21. La dernière phase de cette procédure, soit l'analyse, est divisée en trois parties:

- ◆ syntaxe: analyse des verdicts échoués ou non-concluants;
- ◆ structure: analyse des éléments-exemples et de leur structure à savoir s'ils sont conformes à ceux permis tel que définis dans l'AIM et les clauses de conformité du AP;
- ◆ sémantique⁸: vérification de la sémantique supportée par le AP en question, à savoir si elle est correctement transmise dans le modèle exemple de la structure d'échange de sortie observée.

⁸ La différence qui existe au niveau de l'analyse de la sémantique entre les pré-processeurs et les post-processeurs est que les post-processeurs ont besoin de l'assistance du SUTO.

Les deux dernières phases de la procédure consistent en l'allocation des verdicts et la production du rapport des résultats des cas de tests. Ce dernier est, dans les faits, le rapport des verdicts des cas de tests incluant l'information détaillée ainsi que les résultats des phases d'analyses.

Post-processeurs

Quant à la procédure des post-processeurs, elle comprend quatre phases. Elle commence par l'interprétation de la structure d'échange qui consiste en l'application des paramètres identifiés lors de la préparation pour l'évaluation et par celle de la structure d'échange d'entrée et ce, par rapport à l'IUT qui à son tour, doit répondre à des questions via le SUTO. La procédure se poursuit par l'analyse qui est divisée en deux parties:

- ♦ traitement d'entrée: vérification de la validité du fichier d'entrée;
- ♦ sémantique³: vérification de la sémantique, à savoir si elle est conforme à celle permise par la clause des exigences d'information de chaque AP.

L'allocation des verdicts et le rapport des résultats des cas de tests sont les deux dernières phases de cette procédure et renvoient au même contenu que ceux des pré-processeurs.

2.4.3 Suites de tests abstraits

Une description explicite des suites de tests abstraits est déjà énoncée dans ce chapitre à la partie des généralités sur STEP. Toutefois, certains points n'ont pas été traités et il faut le faire à cette section.

Pour une meilleure compréhension de la nature d'une suite de tests abstraits, voici un exemple. Une suite de tests abstraits est composée de plusieurs éléments: l'étendue d'application, les références à d'autres normes, les définitions, les buts des tests et les cas

de tests abstraits. Seulement deux de ces éléments, les buts des tests et les cas de tests abstraits, seront mis en évidence car ils sont représentatifs des suites de tests abstraits et non à d'autres parties de la norme, comme le sont les autres éléments.

Quant aux buts des tests, ils se regroupent en deux volets: les buts de tests à proprement dit, c'est-à-dire la définition de tous les buts de tests qui seront vérifier à travers les différents cas de tests abstraits; les buts de tests généraux, c'est-à-dire les " énoncés d'exigences qui s'appliquent à tous les cas de tests abstraits, tous les cas de tests abstraits des pré-processeurs ou tous les cas de tests abstraits des post-processeurs"⁹ (ISO/TR 10303-304 TC184/SC4, 1996). Il est à noter que parmi les buts de tests généraux, il y a ceux relatifs aux éléments d'application (Application Element - AE) et ceux relatifs aux modèles intégrés d'application (Application Interpreted Model - AIM).

Exemple 2.2 Buts de tests

- ae12 Conique comme un cercle (conic as circle);
- ae200 Représentation par frontières comme une représentation par frontières à facettes (B-Rep as a Facetted_B-Rep);
- aim70 boucle comme une boucle d'arête (loop as edge_loop); et
- aim83 arête enlignée avec **orientation** = vrai (oriented_edge with **orientation** = true).

Exemple 2.3 Buts de tests généraux

- g1: La sortie d'une implantation sous test (Implementation Under Test - IUT) préserve l'ensemble de la sémantique définie par le modèle d'entrée;
- gvc1: La sémantique du modèle d'entrée sont préservés dans la sortie du IUT; et
- gvc5: La sortie d'un pré-processeur est conforme à la méthode de mise en oeuvre à qui le IUT se veut d'être conforme.

⁹ Traduction libre.

Les cas de tests abstraits contiennent cinq éléments: un identificateur unique, un sommaire du cas, une liste de buts de tests (définis précédemment dans la documentation de la suite de tests abstraits), des spécifications d'entrées et des critères de verdict applicables à chacun des buts de test. Il faut mentionner que les cas de tests abstraits servent à vérifier les buts des tests.

Exemple 2.4 Cas de tests abstraits eb4 de la partie 304

Sommaire du cas de test:

Le cas de test eb4 est un cas de test qui consiste des faces nécessaires pour définir un seul solide résultant d'une union de deux cylindres de rayons différents et qui ont des axes orthogonaux. La courbe d'intersection est une courbe en trois dimensions fermée, représentée par une "polyline". La définition de l'enveloppe est fournie par le contexte de la `polyligne_d'union_cylindre` (`cylinder_union_polyline`) et faite en référence des paramètres originaux.

Couverture d'autre but de test:

autre3: La polyligne devrait être le seul sous-type de courbe restreinte permise et pouvant être utilisée pour définir une représentation de frontières élémentaires.

PRÉ-PROCESSEUR

Spécification d'entrée:

Id	V	Application Element	Valeur	Req
@101	*	elementary_B_rep	#ebsr	M
@73	*	B-rep	#cylxcyl_solid	M
@104.1	*	cylindrical_surface	#cyl_1	M
@21.1	*	cylindrical_surface.axis	#pos_z	C1
@22.1	*	cylindrical_surface.radius	50.0	C1
@104.2		cylindrical_surface	#cyl_2	C1
@21.2		cylindrical_surface.axis	#pos_y	C1
@22.2		cylindrical_surface.radius	20.0	C1
@24	*	edge_curve	#edgemo	M
@107.1	*	face	#cyl_1f	C1
@64		face.bounds	[#bcyltop, #bcylm, #bcylb]	C1
@65	*	face.	#cyl_1	C1
@107.2		face	#cyl_2f	C1
@34	aim9	location	#at	M
@17	*	curve as bounded_curve	#poly	M
@117	*	polyline	#poly	M
@9	*	shell	#shell_object	C1

Contraintes sur les valeurs:

C1: Les valeurs de dimensions géométriques sont interdépendantes et peuvent seulement être variées selon un mode: un mode qui assure l'intégrité géométrique et topologique de la représentation par frontières déjà définie. En général ceci est réalisable en faisant varier les paramètres utilisés dans la définition du test post-processeur dans le contexte EXPRESS-I.

Critère de verdict spécifique:

aim9: Le modèle devrait contenir un exemple d'une **surface_élémentaire** ayant comme position l'**axe2_placement_3d** et la **réf_direction** absente.

aim11: Le modèle devrait contenir un exemple d'une **surface_élémentaire** ayant comme position l'**axe2_placement_3d** ainsi que l'**axe** absent.

Aim12: Le modèle devrait contenir un exemple de **courbe_arrête** ayant une **géométrie_arête** comme "polyline"(3D fermé).

Détails en extra:

Créer une **représentation_forme_représentation** de **frontière_élémentaires** consistant en un simple **représentation** de **frontière_solide_"manifold"**. La **représentation** de **frontière_solide_"manifold"** devrait avoir la forme de deux cylindres perpendiculaires qui se croisent. Les défauts devraient être utilisés pour définir l'**axe2_placement_3d** pour localiser un de ces cylindres. La courbe d'intersection devrait être représentée par une "polyline". Un ensemble de dimensions convenables est défini dans la spécification d'EXPRESS-I ci-dessous présentée.

POST-PROCESSEURÉtendue des buts des tests AIM:

aim9, aim11, aim12

Spécification d'entrée:

NOTES

1 - Dans la spécification ci-dessous, le contexte de la **structure_produit_base** est utilisé pour définir le contexte du produit utile aux définitions géométriques.

2 - Le contexte de la “**polyline**”_d’**union_cylindre** est utilisé, avec des paramètres par défaut, pour la définition des faces et de l’ensemble de la géométrie ainsi que la topologie de la représentation de frontières.

*)

```
TEST_CASE example_ebrep_4; WITH part_204_brep_product_schema;
```

```
REALIZATION
```

```
LOCAL
```

```
shell_objet : closed_shell ;
cylxcyl_solid : manifold_solid_brep ;
ebsr : elementary_brep_shape_representation ;
its_units : named_unit ;
len_exp : dimensional_exponents :=
    dimensional_exposant(1, 0, 0, 0, 0, 0, 0) ;
its_context ; geometric_representation_context ;
prod1_name ; STRING := ‘prodi_name’;
shape_1_def : product_definition_shape ;
shape1_def_rep : shape_definition_representation;
```

```
END_LOCAL;
```

```
CALL basic_product_structure ;
    IMPORT (shape_1_def := @prod_def_shape; );
    WITH (prod_name := @prodi_name; );
END_CALL;
```

```
CALL cylinder_union_polyline ; -- uses default values, so no WITH
```

```

        IMPORT (shell_object := @cxcshell );
    END_CALL;
    cylxcyl_colid := manifold_solid_brep ('cylxcyl_solid',
shell_object);

    its_units := named_unit(len-exp) || length_unit() ||
        si_unit ('milli', 'metre');

    its_context := geometric_representation_context
        ('its_context', 'context'for'cylxcyl', 3) ||
        global_unit_assigned_context ([its_units]);

    ebse := elementary_brep_shape_representation
        ('ebsr', [cylxcyl_solid], its_context);
    shape1_def_rep := shape_definition_representation
        (shape_1_def, ebse);

    END_REALIZATION
END_TEST_CASE;
(*

```

Critère de verdict spécifique:

aim9: Valeur par défaut de l'attribut de la **direction_réf** devrait être fournie correctement.

aim11: Valeur par défaut de l'attribut de l'axe devrait être fournie avec exactitude.

aim12: Tous les points "polyline" devraient se trouver sur les DEUX **surfaces_cylindriques** avec une tolérance de moins de 0.000001.

L'exemple étant présenté, il est pertinent d'élaborer autour d'un autre élément relié aux échanges de données.

2.5 MAPPING

Le domaine du "mapping" demeure peu connu, d'où la nécessité d'amorcer cette partie en introduisant le sujet. Ensuite, l'étendue de ce domaine est présentée et enfin, quelques études sur les méthodes du "mapping" sont soumises à l'attention du lecteur.

2.5.1 Introduction

Il faut rappeler brièvement que les échanges de données, peu importe les formes qu'ils prennent (ex: électroniques, orales etc...), sont, de nos jours, indispensables dans tous les domaines. Le domaine de l'ingénierie ne fait pas exception à la règle car les échanges de données peuvent se faire entre deux ou plusieurs parties. Naturellement, un modèle commun est plus avantageux puisque les informations peuvent être partagées par plus d'une personne. Selon de récentes recherches, ce modèle commun ne résout pas tous les problèmes concernant les échanges de données. Des problèmes types d'intégration tels que l'incompatibilité sémantique entre les modèles surviennent lorsque ceux-ci sont extraits de leur contexte de conception pour être utilisés en milieu de pratique (Verhoef et al, 1995). À cet égard, on propose une solution qui consiste à supporter des vues multiples du modèle à l'intérieur d'un système intégré. De plus, plusieurs savent qu'un modèle a un caractère dynamique et non statique. Ce phénomène s'explique par le fait qu'un modèle a besoin d'être fréquemment ajusté entre le moment où il est conçu et le moment où il est mis en marché.

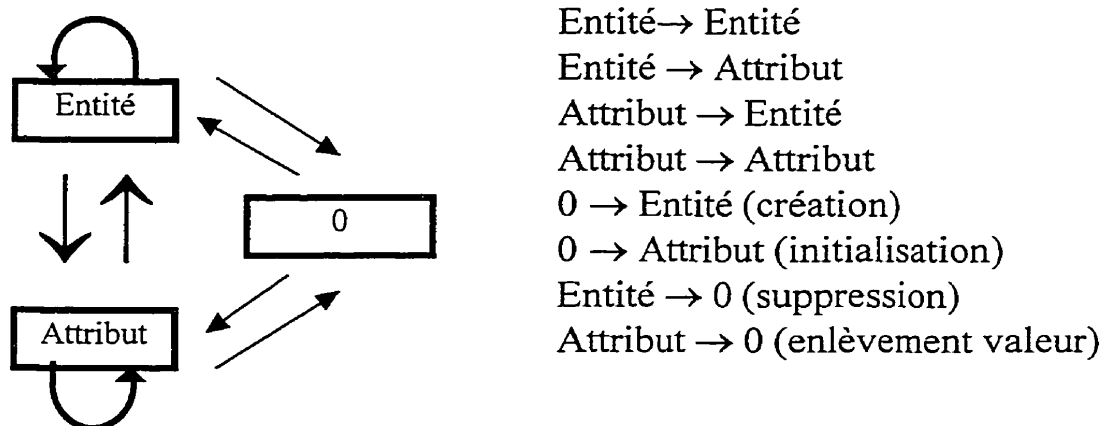
Le domaine du “mapping” existe afin de faciliter la résolution des problèmes identifiés ci-haut. Il peut se définir comme un modèle de résolution des problèmes d’échange de données d’une syntaxe x, point de départ, vers une syntaxe y, point d’arrivée. Toutefois, il n’existe pas de solution unique pour résoudre tous les problèmes. En effet, la recherche et l’industrie en utilisent plus d’une comme par exemple, EXPRESS-M, KIF et VML, pour ne nommer que celles-là, et elles ont leurs points forts comme leurs limites. Une description plus exhaustive des solutions proposées est présentée dans les sous-sections qui suivent.

2.5.2 L’étendue du domaine du “mapping”

Bien que très performant eu égard à l’alternative qu’il offre aux problèmes d’échanges, le domaine du “mapping” éprouve tout de même quelques problèmes. Ces problèmes se retrouvent dans plusieurs champs comme les bases de données relationnelles. Il faut avant tout, présenter quelques exemples de problèmes possibles: transformations exécutées à l’intérieur d’un schéma simple (ex: additions d’entités), transformations exécutées sur deux schémas dont la sémantique est peu différente (ex: migration de version) et transformations exécutées sur deux schémas dont la sémantique diffère largement. Toutefois une étude, «A multi-paradigm mapping method survey» (Verhoef et al, 1995), dirigée vers des méthodes de “mapping” dans le domaine de la technologie de données de produit a été effectuée. C’est à partir de cette étude que les informations présentées ci-haut prennent leur racine.

À la lecture de l’étude mentionnée ci-haut, on décèle différentes méthodes pour résoudre les problèmes de “mapping”. Ces méthodes sont reliés à des langages de modélisation sous forme EXPRESS directe (ex: EXPRESS-C, SDAI) ou indirecte (ex: VDM, KIF) et sont présentées au point suivant de cette section.

Il existe différents types de “mappings” qui peuvent survenir à l’intérieur d’un problème de “mapping” typique. La forme la plus simple réside en celle d’équations reliant les attributs. Quant aux problèmes plus compliqués, ils demandent des procédures plus lourdes. La figure 2.4 qui suit illustre quelques types de “mapping” possibles.



Tirée de VERHOEF, M., LIEBICH, T. et AMOR, R. (1995). A multi-paradigm mapping method survey. *Modeling of Buildings through their Life-Cycle*, 233-247.

Figure 2.6 Types de “mapping”

2.5.3 Études sur les méthodes de “mapping”

Comme nous l’avons déjà annoncé, il existe plus d’une méthode disponible dans l’arène de la technologie de produit de données. Ces méthodes se basent sur des paradigmes de spécification et de programmation différents. Parmi ces derniers, il y a le “style déclaratif (VML), le style fonctionnel (VDM-SL), ceux basés sur la connaissance (règles XP, KIF) et ceux basés sur les styles de programmation impératifs (EXPRESS-V, EXPRESS-C, EXPRESS-M, SDAI et C++)¹⁰ (Verhoef et al, 1995). Pour fins de clarifications, il faut décrire brièvement de ces styles.

¹⁰ Traduction libre.

Le style déclaratif «énonce de façon explicite qu'elles propriétés le résultat désiré doit avoir mais il n'énonce pas comment le résultat désiré doit être obtenu» (Hermann et Lavoisier, 1991). Le style fonctionnel ressemble énormément au style déclaratif puisqu'il est une sous-classe de celui-ci. Toutefois, le style fonctionnel est basé sur le lambda-calcul, «formalisme qui permet de représenter des fonctions et les façons de combiner des fonctions» (Hermann et Lavoisier, 1991) et sur les équations récursives. Quant au style de programmation impératif, il ressemble lui aussi au style déclaratif puisqu'ils sont compatibles l'un par rapport à l'autre. En effet, le style de programmation impératif énonce la procédure pour obtenir le résultat désiré mais il n'énonce pas explicitement qu'elles sont les propriétés de ce résultat désiré. Enfin, le style basé sur les connaissances est en fait une forme de langage qui s'exprime à partir de règles.

L'étude menée sur les méthodes du "mapping" s'est effectuée à partir de plusieurs critères. Ces derniers sont au nombre de sept: le champ d'application, c'est-à-dire quels sortes de problèmes l'approche du "mapping" peut-il supporter; l'adaptabilité qui renvoie à la facilité d'adapter le "mapping" afin d'accomoder de nouveaux aspects; la clarté de spécification, c'est-à-dire la disponibilité des différentes sortes de mécanismes abstraits; la clarté de lecture qui consiste en la compréhension des "mappings" pour une tierce personne; la détermination du support d'outil, à savoir si on parle de support de développement incrémental et/ou de support graphique; la facilité d'intégration, à savoir si la méthode peut être reliée à des systèmes d'échange de données; et finalement, la granularité des "mappings", c'est-à-dire s'il existe des possibilités de mises à jour incrémentales et/ou des transferts complets d'un modèle.

À partir des résultats de l'étude faite par Verhoef et al (1995), le tableau 2.3 présente une synthèse des caractéristiques principales de chaque langage ainsi que l'endroit où ils ont été développés. Avant de présenter ce tableau, voici les huit critères sur lesquels la synthèse est élaborée:

1. Une ligne d'introduction présentant l'emphase principale du langage;
2. Le schéma sur lequel le langage se base;
3. La présence d'une notation textuelle ou non;
4. La présence d'une notation graphique ou non;
5. Le langage est procédural, déclaratif ou fonctionnel;
6. Les "mappings" sont partiels ou entier;
7. Le "mapping" peut être utilisé pour des transferts bi-directionnels ou uni-directionnels;
8. Le système de "mapping" est un outil de recherche, un produit commercial ou une théorie.

Tableau 2.3 Tableau synthèse des langages pour faire du "mapping"

LANGAGE	DÉVELOPPÉ À	CARACTÉRISTIQUES PRINCIPALES
EXPRESS-M	CIMIO Ltd., Surrey, Nations Unis	<ul style="list-style-type: none"> • Adresse les problèmes d'inter-opérabilité des AP; • S'intègre dans EXPRESS; • Notation textuelle; • Pas de notation graphique; • En partie déclaratif, en partie procédural; • Conversion totale; • "Mapping" uni-directionnel; et • Exécution commerciale.
EXPRESS-V	Rensselaer Polytechnic Institute	<ul style="list-style-type: none"> • Additionne des vues aux schémas EXPRESS; • S'intègre dans EXPRESS; • Notation textuelle; • Pas de notation graphique; • En partie déclaratif, en partie procédural; • Conversion totale; • "Mapping" bi-directionnel; et • Exécution pour recherche en plus de produit commercial.

Tableau 2.3 (suite) Tableau synthèse des langages pour faire du "mapping"

LANGAGE	DÉVELOPPÉ À	CARACTÉRISTIQUES PRINCIPALES
EXPRESS-C	Projet ESPRIT, PISA	<ul style="list-style-type: none"> • Rehausse les capacités de création d'EXPRESS; • S'intègre dans EXPRESS; • Notation textuelle; • Pas de notation graphique; • Procédural; • Conversion totale; • "Mapping" uni-directionnel; et • Exécution à titre de recherche.
Règles XP	CSTB, Sophia Antipolis, France	<ul style="list-style-type: none"> • Adresse le "mapping" dans un environnement AI (Artificial Intelligence); • Compatible à EXPRESS, "Lisp late binding of SDAI"; • Notation textuelle; • Pas de notation graphique; • Déclaratif; • Conversion totale et partielle; • "Mapping" uni-directionnel; et • Exécution à titre de recherche.
VML (View Mapping Langage)	University of Auckland, Nouvelle Zélande	<ul style="list-style-type: none"> • Langage de "mapping" générique; • Langage indépendant (liens à EXPRESS); • Notation textuelle; • Notation graphique; • Déclaratif; • Conversion totale et partielle; • "Mapping" bi-directionnel; et • Exécution à titre de recherche.
EDM (Engineering Data Base)	Center for Design and Computation, UCLA	<ul style="list-style-type: none"> • Environnement et langage de base de données dymanique; • Intégré dans le langage du schéma EDM; • Notation textuelle; • Notation graphique; • Déclaratif; • Conversion totale et partielle; • "Mapping" uni-directionnel; et • Exécution à titre de recherche.

Tableau 2.3 (suite) Tableau synthèse des langages pour faire du "mapping"

LANGAGE	DÉVELOPPÉ À	CARACTÉRISTIQUES PRINCIPALES
KIF (Knowledge Interchange Format)	DARPA, Knowledge Sharing Initiative, État- Unis,	<ul style="list-style-type: none"> • Adresse la collaboration d'agent; • Langage indépendant (pas de liens avec EXPRESS); • Notation textuelle; • Pas de notation graphique; • Déclaratif; • Conversion sur demande; • "Mapping" uni-directionnel; et • Exécution à titre de recherche.

Tableau tiré de VERHOEF, M., LIEBICH, T. et AMOR, R. (1995). A multi-paradigm mapping method survey. Modeling of Buildings through their Life-Cycle, Fisher, Law and Luiten, CIB/W78-TG10 publication 180, Stanford University, 233-247.

Quelques langages dont, VDM-SL et SDAI API, ne sont pas présentés dans le tableau synthèse ci-dessus mais il faut les décrire brièvement.

VDM-SL est un langage de spécification relié à la méthode VDM et consiste en "une collection de techniques pour la spécification et le développement formel de systèmes de calculs"¹¹ (Verhoef et al, 1995). Le langage VDM-SL est, d'une part, constitué de règles pour le raffinement de données et d'opérations qui permettent d'établir le niveau de code et d'autre part, d'une théorie de la preuve. Il faut spécifier que ce langage présente deux inconvénients: l'impossibilité de traduire directement les entités EXPRESS en équivalent VDM-SL et la nécessité d'effectuer une opération d'intégration de la spécification résultante dans un système d'échange de données avant qu'un système exécutable puisse être livré.

¹¹ Traduction libre.

Quant au langage SDAI API, au lieu d'offrir une intelligence conceptuelle comme bien des méthodes spécialisées le font, il fournit une interface de programmation d'application, ce qui facilite les transformations de modèle.

La conclusion que l'on peut tirer de l'étude faite sur les langages pour effectuer du "mapping" est qu'il n'existe pas un langage qui permette d'effectuer l'ensemble des opérations d'une façon illimitée. Chaque langage couvrant un domaine particulier. Il peut arriver que dans un domaine il existe plus d'un langage qui permette d'effectuer le "mapping" mais il y en a toujours un qui s'avère être meilleur que les autres et ce, tout dépendamment des exigences qui sont fixées.

2.6 PROBLÉMATIQUE DANS UNE ENTREPRISE

Dans une entreprise, il existe plusieurs conditions indispensables à la réussite de la mise en oeuvre d'un produit quelconque et il en existe d'autres qui ne sont pas nécessairement indispensables, mais qui constituent un atout de premier ordre. Les facteurs jugés indispensables varient beaucoup d'une entreprise à une autre, car les points de vue diffèrent d'un milieu à l'autre. En dépit des différences d'opinions, on peut facilement affirmer que tous les dirigeants d'entreprises s'entendent sur deux aspects importants pour la réussite d'une entreprise: premièrement, une bonne organisation de l'entreprise avec une bonne interaction entre les individus de cette entreprise sont indispensables et, deuxièmement, la qualité d'un produit, quel qu'il soit, est elle aussi indispensable. Un fait demeure cependant: la qualité d'un produit peut être difficilement atteinte si une entreprise est malade c'est-à-dire, s'il existe des problèmes majeures tel, le manque de communication entre départements à l'intérieur de cette entreprise. Dans le cas qui nous intéresse, ce qui prime est de savoir quel sera l'influence des échanges de données sur la qualité du produit lui-même. En d'autres termes plus précis, la qualité des échanges de

données est un facteur central via l'atteinte de la qualité elle-même du produit car ils influencent la marche à suivre pour atteindre les standards de qualité désirés.

Le phénomène de la diversité des opinions à propos des facteurs nécessaires à la réussite d'un produit est le même pour les facteurs qui contribuent au développement d'un meilleur produit. À cet égard, seulement quelques facteurs peuvent être clairement ciblés car, encore une fois, un bon nombre d'individus ne partagent pas la même ligne de pensée à ce sujet. Bref, il n'y a pas de consensus. Parmi ces facteurs on peut compter le temps de modélisation et de modification d'un modèle. Un autre facteur très important est celui de la qualité de conception du modèle en vue d'utilisations ultérieures comme lors de processus d'analyses ou encore de fabrication.

Le temps de modélisation est considéré comme un facteur central: la rapidité d'exécution s'avérant très avantageuse au plan de la productivité. En effet, si le temps de modélisation est court ou très raisonnable, il permettra un délai plus court de mise en marché d'un produit. Toutefois, il faut que la qualité de modélisation soit aussi respectée même si un temps de modélisation court est souhaitable. De plus, l'adaptabilité du modèle en vue d'un objectif spécifique est important car il sera plus facilement utilisable lors d'opérations subséquentes. Il faut mentionner ici que le modèle ne doit pas être modifié par ces opérations subséquentes. C'est à ce niveau-ci qu'entre en jeu la qualité de conception d'un modèle. En effet, un modèle bien modélisé et aisément modifiable est préférable à un modèle où l'apport de modifications devient une tâche ardue. En bout de ligne, il vaut mieux qu'un modèle exige un peu moins de temps de modélisation et soit facilement modifiable, qu'il soit modélisé parfaitement et très complexe à modifier. Ce temps de modification est très important car, à l'intérieur de la période de temps de la conception du modèle à sa mise en marché, ce même modèle peut subir plusieurs ajustements et ce, dans le but de mettre sur le marché un produit de meilleure qualité. De

plus, la manière dont un modèle est construit est très importante car elle influence les résultats obtenus lorsque ce modèle est échangé entre systèmes CAO.

2.7 EXPÉRIENCES FAITES AVEC STEP

Cette partie présente deux études réalisées avec la norme STEP ainsi qu'un projet pilote réalisé avec cette même norme entre les entreprises GM et Ford. Cette partie de la revue bibliographique se termine par des propositions d'améliorations qui pourraient être apportées à la norme.

2.7.1 Études menées (tests)

La norme STEP est opérationnelle que depuis trois ou quatre années et il a déjà des propositions de projets pour tester cette norme. Parmi ces projets, il en existe deux dont l'un du nom d'AutoSTEP et l'autre du nom d'AEROSTEP.

AutoSTEP

AutoSTEP est un projet pilote qui a débuté à la fin de l'année 1994 sous la gouverne du Groupe d'Action de l'Industrie Automobile "Automotive Industry Action Group - AIAG". Comme le stipule Krishnaswami (1995) "ce projet pilote a pour but d'initier les échanges de données, sous forme d'activités pilotes, entre fournisseurs et clients dans le domaine de l'industrie de la chaîne d'approvisionnement automobile"¹². Plusieurs compagnies ainsi que des fournisseurs font partie des participants de ce projet. En plus des participants, il y a quelques organisations de support qui sont reliés au projet AutoSTEP. Les compagnies participantes sont General Motors, Ford, Chrysler, General Dynamics Land Systems et United Defense limited Partners. Les fournisseurs

¹² Traduction libre.

participants sont Eaton Corporation, Allied Signal Automotive, Kearfott, Korry Electronics, Delco Electronics, TRW Automotive, Delphi Saginaw, Dana Corporation, Nypro, ITW Deltar, Breacher, Shellcast, Ganton Technologies, Zovamax et Sealed Power Technologies. Parmi les organisations de support, il y a des vendeurs de conception assistée par ordinateur (CAO) ainsi que des fournisseurs d'outils STEP. Les vendeurs de CAO sont EDS (Unigraphics - UG), Dassault (Catia), PTC (Pro-Engineer - ProE), CV et SDRC alors que les fournisseurs d'outils STEP sont STEP Tools Inc., ITI Ohio et Michigan, PDES Inc., l'Institut National des Standards et Technologie "National Institute of Standards and Technology (NIST)", SCRA, Wayne State University, Oakland Community College et plusieurs autres (Krishnaswami, 1995).

Ce projet pilote repose sur le principe qu'un échange actif entre compagnies et fournisseurs, dans le domaine de l'industrie automobile, s'avère de plus en plus indispensable. AutoSTEP effectue une démonstration de l'efficacité des procédés de communication de données de produit. De plus, toujours selon Krishnaswami (1995), on s'attend à ce qu'AutoSTEP "apporte un avantage réel en démontrant comment les chaînes d'approvisionnement automobile peuvent mieux intégrer leur produit ainsi que leur procédé d'activités de conception"¹³. Ainsi on pourra raccourcir la période nécessaire pour procéder à la mise en marché d'un produit.

Le projet AutoSTEP est divisé en trois phases s'échelonnant sur une durée de trois à quatre années. Toutes les phases de ce projet sont élaborées à partir des principes de bases suivants: les phases doivent, à travers l'échange de données de test, démontrer une aptitude particulière de STEP; il faut une documentation concernant les processus administratifs qui sont reliés à l'échange des données d'un produit et du procédé de conception; le troisième principe concerne les fournisseurs et leurs clients car il s'agit d'identifier les occasions qui offrent la possibilité à ces individus d'avoir de meilleurs

¹³ Traduction libre.

échanges de données de produit, ce qui peut être réalisé par l'amélioration des processus administratifs qui supportent le lancement de produit et de procédé de conception; le dernier principe est lié de près au premier puisqu'il doit l'utiliser, ainsi que réferer et améliorer les procédés administratifs et ce, dans le but de soutenir les nouveaux projets de conception lors d'échanges de données de produit.

Chacunes des trois phases renvoient à un but précis: la première a pour effet de briser la glace; la seconde, d'élargir la nature de l'information échangée; et la dernière, d'élargir l'information ainsi que la profondeur de la chaîne d'alimentation impliquée. Chacune de ces phases se divise deux parties: la première consiste à démontrer, en utilisant les logiciels disponibles, les échanges effectués entre partenaires et a pour but d'évaluer l'avancement des produits commerciaux et de déterminer les standards; la seconde partie consiste à évaluer l'utilité des échanges mentionnés à la première étape et elle doit aussi permettre d'évaluer la vicacité des traducteurs.

Le projet AutoSTEP apporte plusieurs avantages dont, entre autres, la familiarisation des fournisseurs avec la norme STEP et du même coup, l'utilisation élargie dans l'industrie automobile. Le projet contribue à réduire le nombre d'informations manquantes dans les divers systèmes CAO, le temps de mise en marché, le coût du produit et augmente la qualité, finalement, il offre la possibilité d'intégrer des données mathématiques et administratives sur le marché.

Des informations supplémentaires sur le projet AutoSTEP peuvent être examinées sur le réseau Internet à l'adresse www.iti.org/cec/autostep.htm.

AEROSTEP

Quant au projet AEROSTEP, il a été conçu par le groupe PDES Inc. et son but consiste à présenter la viabilité de l'utilisation de STEP, comme appareil d'échange utilisant le pré-assemblage digitalisé "Digital Pre-Assembly (DPA)", tout en supportant les exigences de données d'un fuselage/moteur. Le projet pilote a débuté au milieu de l'année 1993 et s'est terminé dans la première moitié de l'année 1994. Plusieurs compagnies telles Boeing, GE Aircraft Engines, Pratt & Whitney et Rolls-Royce participent au projet AEROSTEP et sont des membres du groupe de mise en oeuvre de PDES Inc. "PDES Inc. Implementation Group". Certains vendeurs tels Computervision, Dassault Systems, EDS, International TechneGroup Inc., STEP Tools Inc. et Theorem Solutions Consultancy fournissent des produits de conception assistée par ordinateur ainsi que des produits de support pour STEP. AEROSTEP met principalement l'accent sur l'utilisation d'un protocole d'application en particulier soit, l'AP 203, pour les échanges de données à l'intérieur d'une compagnie travaillant en aéronautique.

2.7.2 Projet pilote fait entre deux entreprises

La norme STEP peut être considérée comme étant assez récente. C'est pour cette raison qu'il existe peu¹⁴ d'expériences réalisées pour vérifier la pertinence et l'efficacité de cette norme. Toutefois, il y a certaines entreprises et/ou organismes qui ont fait des essais pour vérifier l'efficacité de celle-ci. En effet, un projet pilote commandité par le Consortium Industriel de Réponse Rapide "Rapid Response Manufacturing Consortium" a été réalisé par deux compagnies. Ce consortium est un projet piloté par le Centre national pour les sciences manufacturières "National Center for Manufacturing Sciences - NCMS". Le projet pilote fût réalisé par Ford et General Motors de la division Saginaw afin de

¹⁴ Le terme "peu" signifie qu'il y a pas beaucoup (moins de 5) expériences réalisées au moment de la recherche bibliographique de la présente étude.

démontrer la faisabilité d'échanges de données de définition et ce, par l'intermédiaire de STEP (Clark et Staley, 1995). Quelques critères tels la faisabilité, l'automatisation et la robustesse de STEP, constituent des exigences à combler. Une partie bien précise de STEP a été retenue pour réaliser ce projet qui eu lieu en mars 1994. En effet, les échanges de données ont été réalisés à l'aide de l'AP 203. De plus, les échanges de données étaient principalement concentrés sur des solides à représentation par frontières (Boundary Representation - B-Rep) ainsi que sur des données de gestion de configuration (Configuration Management - CM).

Une pompe de conduite assistée "power steering pump", présentée sur la figure 2.7 ci-dessous, est employée pour les échanges entre Ford et GM.

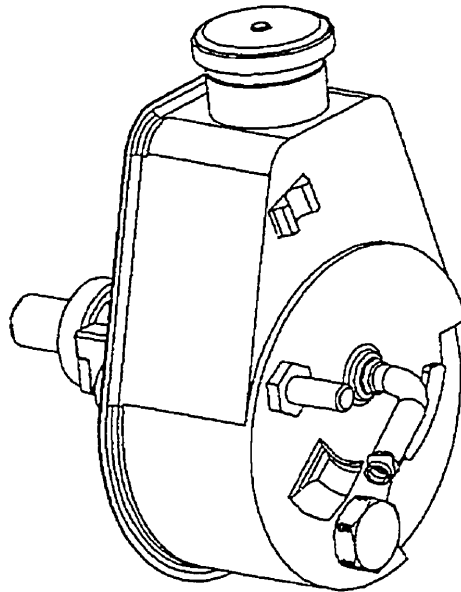


Figure 2.7 Pompe de conduite assistée employée pour les transferts de données entre GM et Ford

L'utilisation de cette pompe est un exemple de problèmes importants auxquels font face les compagnies américaines et leurs fournisseurs. Les transferts des données de la pompe se font de la Division GM Saginaw, qui utilise le logiciel de modélisation Unigraphics,

vers Ford, qui utilise ses propres logiciels de modélisation tels, PDGS, ARIES et ICAD, pour différentes activités d'ingénierie. Entre les deux compagnies, un système central du nom de "Rapid Respond Manufacturing - RRM" est utilisé comme intermédiaire. Ce système a pour but d'automatiser les transferts de données entre les deux parties impliquées par le recours au réseau Internet, et ce, en utilisant le protocole de transfert de fichier d'Internet soit le "file transfer protocol - ftp". Afin de rendre plus facile les échanges par ce site pour les utilisateurs, trois programmes Unix ont été réalisés.

La procédure de transfert entre Ford et GM s'effectue par l'intermédiaire du système central, comprenant quatre étapes: la première consiste à transférer le modèle modélisé par GM en format STEP, par l'entremise du traducteur AERO-STEP de EDS; la deuxième étape est d'envoyer le fichier sur le système central RRM; la troisième consiste à récupérer le fichier STEP chez Ford et à le traduire en format ACIS par le recours au traducteur ACIS de STEP Tools Inc.; enfin, la dernière étape consiste à transférer le fichier ACIS dans les systèmes PDGS, ARIES et ICAD de Ford. Cette procédure a aussi été suivie mais à l'inverse après quelques modifications effectuées sur la pompe par les ingénieurs de Ford.

Durant tous les transferts effectués, les critères de faisabilité, d'automatisation et de robustesse de STEP n'ont pas été respectés. Le premier critère, la faisabilité, c'est-à-dire la possibilité d'effectuer des transferts de données, a été respecté qu'à moitié car il fût très difficile à démontrer en raison de l'élimination des erreurs des logiciels de traduction STEP qui a dû être faite. Le deuxième critère, l'automatisation, c'est-à-dire les transferts s'effectuant avec l'intervention minimale d'une personne et ne demandant pas de connaissances sur STEP, n'a pas été respecté non plus puisqu'il a fallu préparer à la main le fichier physique "physical file" de STEP afin que le transfert se produise¹⁵. Cette

¹⁵ Exemple d'actions correctives manuelles. Se référer à la partie 3.6.4 du chapitre 3 pour des spécifications sur ce qu'est une action corrective.

intervention n'a pas été nécessaire lors du transfert de retour des données de la pompe. Enfin, le troisième critère, la robustesse de STEP, c'est-à-dire la possibilité de transférer n'importe quelle pièce désirée ainsi qu'une grande variété de géométries et de ne rencontrer aucune difficultés majeures lors du procédé, n'a pas été respectée en raison de l'échec du transfert lors de l'utilisation d'une autre pompe de même complexité que la première.

Bref, plusieurs problèmes ont fait surface. Suite à ce constat, le projet pilote réalisé entre la division GM-Saginaw et la compagnie Ford a donné lieu à plusieurs conclusions et recommandations. En effet, on réalise que le protocole d'application 203 ne peut pas, du moins pas pour le moment, remplacer les besoins du logiciel de modélisation interne de Ford, soit PDGS. À ceci s'ajoute un besoin de disposer d'un plus grand nombre de données d'analyse ainsi qu'un besoin d'échange de spécifications d'ingénierie. Enfin, l'information de gestion de configuration de l'AP 203 est très complète mais, il y a nécessité manifeste d'effectuer des études plus approfondies sur les protocoles d'application 201 et 202.

Afin de mieux cerner le projet pilote entre Ford et GM, les tableaux 2.4 et 2.5 présentent respectivement des comparaisons de propriétés de masse entre systèmes et des comparaisons de propriétés de masse aller-retour entre GM et Ford. Ces données ont été relevées dans le but de s'assurer de la validité des données échangées.

Tableau 2.4 Comparaisons de propriétés de masse entre les systèmes

Propriétés	Unigraphics	PDGS	Aries	ICAD
Volume	77.5646	77.5372	77.582	77.5259
Masse (lb)	21.9430	21.9350 ¹⁶	21.8477	21.9318

¹⁶ PDGS n'inclue pas un support de densité alors pour des fins de comparaisons, les moments et produits de masse ont été calculés avec une densité de 0.0282897 lb/po³ (densité utilisée par les autres systèmes).

Tableau 2.4 (suite) Comparaisons de propriétés de masse entre les systèmes

Propriétés	Unigraphics	PDGS	Aries	ICAD
Aire (po ²)	149.0443	149.0620	149.062	149.0453
Centre de masse X (po)	4.1697	4.16701	4.1693	4.1635
Centre de masse Y (po)	0.0216	0.02211	0.0215	0.0231
Centre de masse Z (po)	0.03756	0.3751	0.3755	0.037484
Moment de masse X (lb-po ²)	113.8157	113.7209 ³	113.829	Non calculé
Moment de masse Y (lb-po ²)	485.5142	484.9821 ³	485.548	Non calculé
Moment de masse Z (lb-po ²)	428.9584	428.5118 ³	429.004	Non calculé
Produit de masse YZ (lb-po ²)	1.35545	1.35547 ³	1.35936	Non calculé
Produit de masse XZ (lb-po ²)	37.19014	37.12567 ³	37.1949	Non calculé
Produit de masse XY (lb-po ²)	1.88022	1.92943 ³	1.869	Non calculé
Rayon de giration X (po)	2.27747	Non calculé	2.27736	Non calculé
Rayon de giration Y (po)	4.70384	Non calculé	4.7035	Non calculé
Rayon de giration Z (po)	4.42139	Non calculé	4.42116	Non calculé

Tableau tiré de CLARK, A. L. et STALEY, S. M. (1995). STEP AP203 data exchange study. Proceedings of the 3rd Symposium on Solid Modeling and Applications, 213-224.

Tableau 2.5 Comparaisons aller-retour de propriétés de masse

Propriétés	Unigraphics	PDGS	UG (aller-retour)
Volume	77.5646	77.5372	77.4595
Masse (lb)	21.9430	21.9350 ³	21.9133
Aire (po ²)	149.0443	149.0620	148.9534
Centre de masse X (po)	4.1697	4.1670	4.1743
Centre de masse Y (po)	0.0216	0.0221	0.0216
Centre de masse Z (po)	0.3756	0.3750	0.3756

Tableau 2.5 (suite) Comparaisons aller-retour de propriétés de masse

Propriétés	Unigraphics	PDGS	UG (aller-retour)
Moment principal X (lb-po ²)	110.8381	110.7555 ³	110.7198
Moment principal Y (lb-po ²)	100.9414	101.0506 ³	100.5722
Moment principal Z (lb-po ²)	47.2961	47.4776 ³	47.0922
Axe principale XPX	0.9989	0.9989	0.9989
Axe principale XPY	0.0153	0.0149	0.0146
Axe principale XPZ	-0.00448	-0.0453	-0.0444
Axe principale YPX	-0.0162	-0.0159	-0.0155
Axe principale YPY	0.9996	0.9997	0.9997
Axe principale YPZ	-0.0211	0.0211	-0.0212
Axe principale ZPX	0.0445	0.0450	0.0440
Axe principale ZPY	0.0218	0.0218	0.0219
Axe principale ZPZ	0.9988	0.9987	0.9988

Tableau tiré de CLARK, A. L. et STALEY, S. M. (1995). STEP AP203 data exchange study. Proceedings of the 3rd Symposium on Solid Modeling and Applications, 213-224.

2.7.3 Améliorations à apporter

Lorsqu'une norme est créée, il lui faut un certain temps avant qu'elle puisse atteindre des standards de performance très élevés et ce, sur tous les aspects sans exception. La norme ISO 10303 ne fait pas exception à cette règle. Même si la norme peut être considérée, par certains, comme étant à un niveau de perfection élevé ci-haut, sa mise en œuvre ne l'est pas nécessairement. C'est pourquoi certaines améliorations sont à apporter afin d'atteindre les critères d'efficacité souhaités.

Selon Clark et Staley (1995), il y aurait six améliorations à apporter. La première amélioration concerne les cas administratifs solides “Solid Business Case”. Il faut absolument que les échanges de données qui sont effectués entre deux parties soient non seulement sans erreur, mais qu’ils soient robustes et automatiques. De plus, il faut être capable d’échanger des données d’une autre nature que celle de la géométrie solide.

La seconde amélioration concerne les logiciels de traduction de STEP. En effet, ceux-ci ont besoin de développement importants ainsi que d’un plus grand nombre de tests, puisqu’il arrive assez souvent de devoir éliminer les erreurs dans les logiciels.

La troisième amélioration consiste en la validation des outils de STEP. De tels outils sont nécessaires afin de pouvoir effectuer des vérifications de données sortantes des logiciels de traduction. Par ailleurs, si ces mêmes outils étaient conformes à la norme, ils pourraient aider à trouver les erreurs se glissant dans les fichiers physiques “physical files” de STEP.

La quatrième amélioration porte sur le besoin de protocoles de test plus rigoureux. Ces protocoles permettraient non seulement de s’assurer que les logiciels de traduction soient capables d’effectuer des traductions mais que ces dernières soient élaborées soigneusement, c’est-à-dire sans erreurs, de façon automatique et qu’une connaissance approfondie de STEP ne soit pas requise.

La cinquième amélioration va dans le même sens que les deux dernières. En effet, pour la troisième on parlait de besoin d’outils, pour la quatrième on parlait de protocoles alors que pour la cinquième on souligne le besoin d’analyses rigoureuses d’erreurs observées lors d’échanges de données. Pour bien comprendre ce que l’on entend par erreur, il faut de donner une définition brève de ce mot. Par erreur, on se réfère au genre d’écart qu’il y a entre des données de départ et celle à l’arrivée. On pourrait prendre comme exemple le

volume d'un objet qui au départ a une valeur v et qu'à l'arrivée a une valeur $v \pm x$, x représentant l'écart.

Enfin, la dernière et sixième amélioration renvoie au besoin de logiciels pour le support des contrôles de processus d'échange de données. En fait, on doit arriver à ce que l'ingénieur de conception n'ait seulement qu'à effectuer une opération simple pour que la conversion de son fichier s'effectue et soit envoyée à la destination désirée. En résumé, on désire combiner la traduction d'un fichier et son expédition vers une destination précise en une seule opération.

À la lumière de la revue de la littérature, il s'avère pertinent de développer une méthodologie d'évaluation de STEP pour sa mise en place.

CHAPITRE 3

MÉTHODOLOGIE D'ÉVALUATION DE STEP POUR SA MISE EN PLACE

Dans ce chapitre, il sera question de la pertinence de mettre au point une méthodologie d'évaluation de STEP pour sa mise en place. La description générale et détaillée de cette méthodologie sera présentée.

3.1 PERTINENCE DE LA MÉTHODOLOGIE

Dans un premier temps, une figure représentant les différents niveaux de communication entre deux entités (personnes ou systèmes) est introduite.

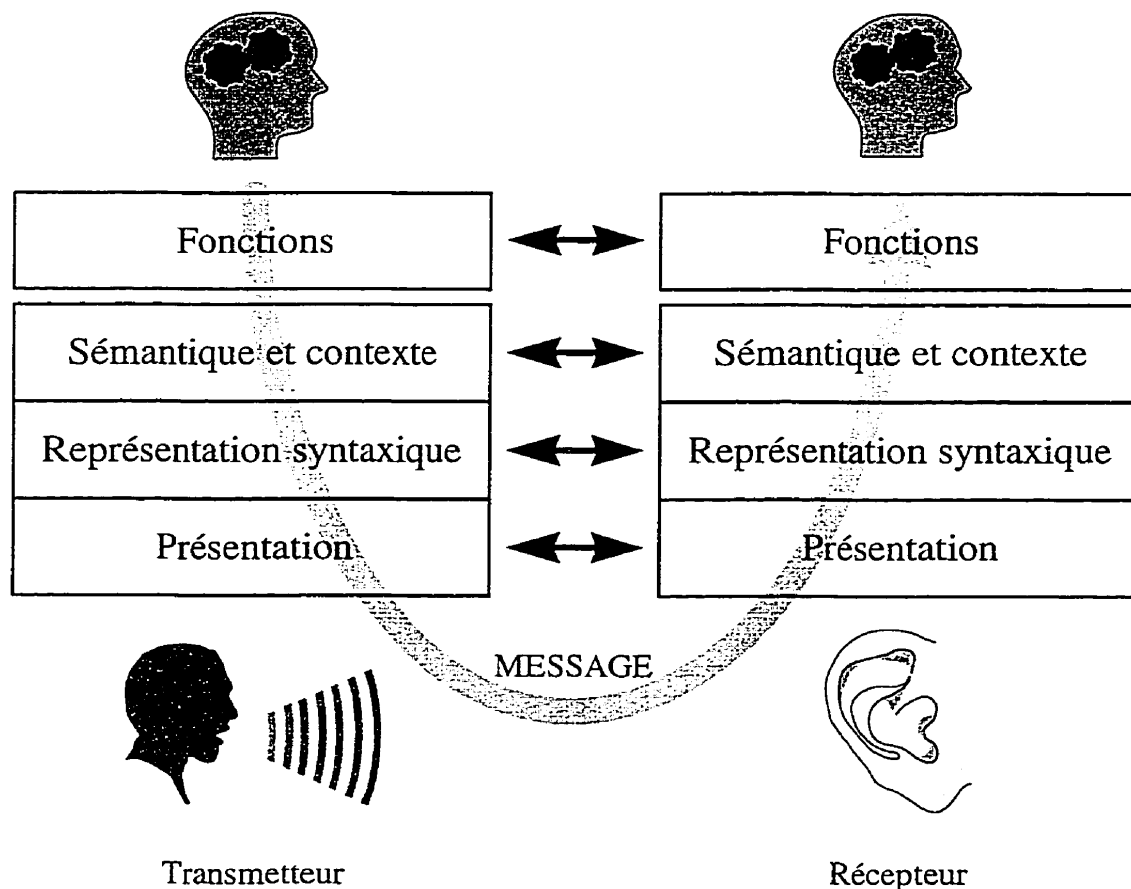


Figure 3.1 Structure de communication entre deux partis

Les quatre niveaux représentés à la figure 3.1, se définissent comme suit. Le niveau présentation comprend la représentation physique d'un message, par exemple l'élocution

pour un langage humain et l'apparence à l'écran pour un système CAO¹. La représentation syntaxique comprend essentiellement la grammaire du langage utilisé. Elle définit donc les règles d'utilisation des mots, c'est-à-dire des noms, des adjectifs, des verbes, etc. Le niveau de la sémantique, selon le dictionnaire Petit Robert (Robert et al, 1983), se définit comme l'étude du langage considéré du point de vue du sens. La sémantique étudie les relations du signifiant au signifié, les changements de sens, la synonymie, la polysémie, la structure du vocabulaire. En référence au Petit Larousse (1998), la sémantique est définie comme étant l'étude scientifique du sens des unités linguistiques et de leurs combinaisons ou encore, ce qui est relatif au sens, à la signification des unités linguistiques. La sémantique établit donc les liens entre la représentation du vocabulaire et son sens compris dans un contexte donné. Ce dernier éclairant la sémantique. Finalement, le niveau des fonctions comprend les concepts, les actions, la pensée même qui doivent être véhiculés entre le transmetteur et le récepteur.

Pour qu'un message soit bien compris, il faut une correspondance conforme entre chacun des niveaux du récepteur et du transmetteur. Chaque niveau assure le passage au niveau supérieur. Chaque niveau demande aussi une entente négociée, c'est-à-dire un protocole, entre chaque niveau correspondant du transmetteur et du récepteur. Par exemple, il faut que la même représentation syntaxique soit utilisée par les deux et qu'il y ait entente. Ainsi, la flèche allant de gauche à droite dans la figure 3.1, illustre le chemin parcouru par un message allant de la pensée du transmetteur à la pensée du récepteur. Pour valider la qualité de l'échange entre deux partis il faut donc établir une méthodologie qui vérifie chacun des niveaux du processus de communication. Afin de mieux comprendre la teneur de la figure 3.1, voici quelques précisions.

¹ "Systèmes CAO", "logiciels de CAO" et "logiciels de modélisation" sont trois expressions synonymes signifiant des logiciels utilisés pour la Conception Assistée par Ordinateur (CAO). Elles sont employées tout au long de ce chapitre ainsi que dans le chapitre 4.

Lorsqu'un commandant d'une division de soldats donne l'ordre de tourner vers la droite ce commandement doit être compris par tous les soldats à l'effet qu'il faut effectuer une rotation de 90° vers la droite. Cette rotation doit s'effectuer en deux mouvements comme suit: tout d'abord, les soldats doivent tourner leur corps vers la droite de 90° en prenant comme points de référence pour cette rotation le talon du pied droit et les orteils du pied gauche, finalement, les soldats doivent amener leur jambe gauche parallèlement au sol avec un angle de 90° entre la cuisse et le mollet et ensuite déposer le pied gauche sur le sol à côté du pied droit.

Pour que l'ordre soit bien interprété et exécuté par les soldats le commandant doit tout d'abord donner l'ordre "Division, à droite tourner". L'ordre signifie ici le niveau des fonctions. Le niveau de la sémantique est représenté par l'information implicite, contenu dans l'ordre "à droite, tourner", qu'il faut tourner d'un angle de 90° vers la droite. Le contexte est celui de l'exécution d'exercices militaires². Le niveau de la syntaxe est représenté par les mots division, à droite et tourner dans cette ordre. Finalement, le niveau de la présentation est associé à l'intonation de la voix du commandant ainsi que la prononciation des mots faisant partie de l'ordre donné.

Si les quatre niveaux commentés ci-dessus sont respectés, les soldats seront en mesure de comprendre l'ordre de tourner vers la droite de 90°. En effet, le message est bien compris puisque les conditions suivantes sont respectés:

- 1) l'ordre est clair, c'est-à-dire qu'il y a une bonne intonation et une bonne prononciation (niveau présentation)
- 2) l'ordre est bien structuré, c'est-à-dire qu'il y a une bonne position des mots (niveau de la syntaxe)

² Des exercices militaires signifient des mouvements tels de la marche et des virages militaires. Exercices militaires est la traduction française du terme anglais "drill".

- 3) l'information implicite de l'ordre est connue des soldats puisqu'ils ont eu des cours sur les exercices militaires et le contexte est clairement défini (niveau de la sémantique et du contexte)
- 4) l'ordre est donné (niveau des fonctions)

La figure 3.2 ci-dessous illustre les niveaux présentés précédemment avec leur caractéristiques d'implantation existant dans les logiciels de CAO et les normes d'échanges de données produit.

Fonctions	Implicite
Sémantique et contexte	Implicite
Représentation syntaxique	Activités actuelles des organismes de normalisation
Présentation	Réalisé par les applications du logiciel CAO

Figure 3.2 Niveaux de représentation d'un langage

Les normes d'échanges de données produit telles IGES et STEP, concentrent leurs efforts presque exclusivement sur la représentation syntaxique. Dans le domaine de la conception produits procédé, le niveau présentation est laissé aux applications CAO qui génèrent eux-mêmes l'affichage de l'information. Ce niveau n'est donc pas couvert par la méthodologie proposée dans ce mémoire. Le niveau des fonctions est présentement peu couvert par les normes d'échange de données produit (STEP), niveau pourtant bien important dans le processus de communication. En effet, STEP inclut ce niveau dans certains AP tel l'AP 213 mais ceux-ci sont toujours au stade de développement. Beaucoup de travail reste à faire relativement à la sémantique, au contexte ainsi qu'aux fonctions.

caractéristiques aux modèles. Aujourd'hui, tous ces genres de modélisation sont possibles et les maquettes virtuelles pouvant représenter un produit complet tel un avion ou une voiture, connaissent une utilisation grandissante.

L'emploi des systèmes CAO fait maintenant partie intégrante de la définition des produits dans la vie quotidienne des entreprises. En fait, leur utilisation augmente de manière exponentielle, ce qui entraîne un partage fréquent d'information. Ce partage se réalise par l'entremise de format neutre. Au début vers 1980, la norme IGES fût développée mais on s'est vite aperçu qu'elle était limitée puisqu'elle définissait un fichier physique neutre et non une structure d'échange indépendante du format du fichier. Ce lien de dépendance entre la structure de données et le fichier ne permet toujours pas la définition d'une base de données neutre ou d'interfaces plus générales.

Pour contrer au problème des normes comme IGES, la norme STEP est développée. Elle inclue la réalisation d'échanges de données très complexes. Toutefois, elle demeure très récente, d'où la nécessité de l'implantation d'une méthodologie pour la réalisation de tests d'échanges de données. Ces tests doivent être réalisés de manière à pouvoir analyser graduellement la réussite des échanges, c'est-à-dire analyser les échanges par niveaux de complexité. Aussi, une méthodologie rigoureuse est essentielle à l'aboutissement d'une production scientifique car sans son apport, on risque fort de glisser vers une production plus ou moins valable.

De plus, une méthodologie est considérée comme centrale parce qu'elle balise l'activité que le chercheur doit mener. Autrement dit, elle oriente la démarche en posant des jalons et des critères de scientificité. Une méthodologie pertinente et très rigoureuse s'impose lorsqu'il s'agit de l'exécution de tests.

Le choix d'axer la recherche sur la mise au point d'une méthodologie d'évaluation de STEP pour sa mise en place, plutôt que simplement sur l'exécution de séries de tests et d'analyses des résultats obtenus de ces tests, est basé sur cinq constats.

En premier lieu, il n'existe pas de méthodologie de ce type. Il n'y a que des tests d'échanges de données entre des systèmes de CAO spécifiques qui ont été effectués.

Le deuxième constat, en lien avec le premier, se rapporte aux logiciels de CAO: lorsqu'une série de tests est effectuée, elle est dépendante des versions des logiciels de CAO utilisés lors de ces tests tandis qu'une méthodologie ne l'est pas. Une explication est de mise afin d'exprimer ce lien de dépendance. Au moment de l'exécution de tests, chaque système CAO utilisé (deux ou plus) est à un niveau d'avancement bien précis, communément appelé, la version du logiciel (ex: version 2 pour la deuxième version). Les logiciels de modélisation comme tout autre logiciel évoluent dans le temps, ce qui affecte directement les résultats des tests: ceux-ci deviennent de plus en plus désuets au fur et à mesure que l'on progresse. En conséquence, la pertinence des tests effectués avec des logiciels de modélisation est malheureusement de courte durée. Contrairement à des tests spécifiques, une méthodologie d'évaluation doit être indépendante des logiciels qui seront utilisés pour effectuer cette évaluation.

Le troisième constat est relié à la structure de la communication puisque celle-ci est un processus globale et que la méthodologie s'appuie sur l'ensemble du processus.

Le quatrième constat fait référence à l'évolution des normes d'échange qui sont au point où les efforts ont été au niveau de la syntaxe et que la méthodologie va plus loin puisqu'elle s'applique aussi au niveau de la sémantique et du contexte ainsi qu'au niveau des fonctions.

Finalement, le dernier constat est que la méthodologie englobe la fonctionnalité des maquettes virtuelles puisque l'évolution des systèmes permettent maintenant une fonctionnalité d'une maquette virtuelle complète.

Eu égard à ces considérations, il a été jugé pertinent de travailler au développement d'une méthodologie d'évaluation de STEP pour sa mise en place. La présentation de la description générale de cette méthodologie suit. Cette dernière ainsi que les autres parties de ce chapitre sont structurée en quatre sections.

Tout d'abord, un bref résumé du contenu de la section est présenté. Ce dernier est suivi d'une figure représentant le contenu. Troisièmement, les principales étapes sont présentées dans un style direct. Finalement, une description exhaustive du contenu est formulée.

3.2 PROCÉDURE MÉTHODOLOGIQUE

La méthodologie développée lors de cette étude vise à fournir un guide de travail pour l'exécution d'échanges de données entre plusieurs entreprises. Dans cette visée, des objectifs d'échanges doivent être fixés afin de guider les tests et les analyses qui les suivent. Ensuite, des fichiers sont sélectionnés et traduits d'un système CAO à un autre via STEP. Lorsque ces fichiers sont dans le format du système CAO de réception, les tests de conformité, de sémantique et de fonctions sont exécutés. Lors de l'exécution de ces tests, si des problèmes surviennent, des actions correctives sont posées afin de les résoudre. Enfin, le processus d'évaluation prend fin lors de la rédaction d'un rapport contenant les informations importantes recueillies lors de l'exécution de la méthodologie.

Le diagramme de la figure 3.4, illustré à la page suivante, représente la vue d'ensemble de la méthodologie d'évaluation de STEP pour sa mise en place.

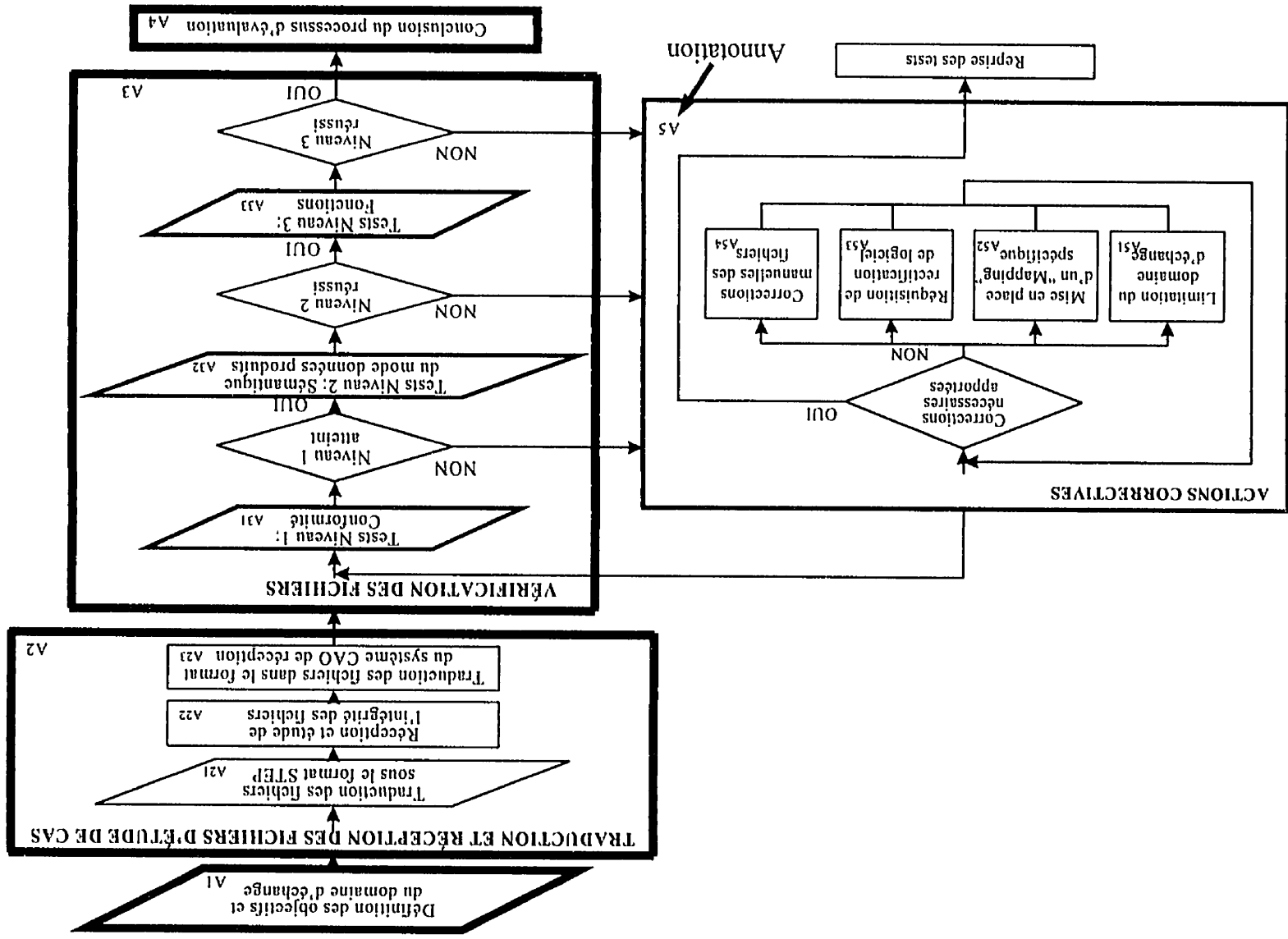


Figure 3.4 Méthodologie d'évaluation de STP pour sa mise en place

Les annotations (ex: A5) inscrites dans le coin inférieur droit de chacune des boîtes sont présentées afin d'aider le lecteur à faire le lien entre la partie texte de la méthodologie et la représentation graphique. Il sera plus facile d'associer la représentation graphique des étapes ou sous-étapes de la méthodologie à la description textuelle car les annotations sont aussi inscrites à l'extrémité droite des titres de chacune de ces étapes.

La méthodologie d'évaluation représentée à la figure 3.4 est séparée en quatre grandes parties lesquelles sont représentées par des blocs dont les bordures sont en gras:

- 1) définition des objectifs et du domaine d'échange
- 2) traduction et réception des fichiers d'étude de cas
- 3) vérification des fichiers
- 4) conclusion du processus d'évaluation

La première partie, celle de la définition des objectifs et du domaine d'échange, consiste principalement à définir les objectifs que les partis impliqués désirent atteindre lors des échanges de données et à sélectionner les fichiers qui seront utilisés pour effectuer ces échanges en référence aux objectifs ciblés. Le choix des fichiers d'étude de cas se fait donc selon les objectifs choisis.

La deuxième partie, celle de la traduction et de la réception des fichiers d'étude de cas, consiste tout d'abord à traduire les fichiers en format STEP, ensuite à recevoir ces fichiers et enfin, à les traduire sous la forme du logiciel de modélisation de réception.

La troisième partie, celle de vérification des fichiers qui constitue le cœur de la méthodologie, consiste principalement à effectuer une série d'analyses et de tests présentant trois niveaux et ce, afin de valider le degré de réussite des échanges par rapport aux objectifs fixés au début de la procédure méthodologique.

À la figure 3.4, un cinquième bloc identifié “actions correctives” est présent. Ce bloc n'est pas considéré comme étant un bloc principal car il est relié de très près au bloc de vérification des fichiers. Aussi, des actions correctives sont effectuées lorsqu'un problème est rencontré lors des tests réalisés à la partie de vérification des fichiers. Bref, les actions correctives existent pour fins de résolution de problèmes survenus lors des échanges de données, problèmes détectés lors de la vérification. Soulignons qu'un problème peut être détecté à l'un des trois niveaux de tests et que des actions correctives peuvent être effectuées à l'un de ces trois niveaux. De plus, un problème peut nécessiter des actions correctives à plus d'un niveau de tests.

Finalement, la quatrième partie, celle de la conclusion du processus d'évaluation, consiste en la rédaction d'un rapport selon les critères de l'entreprise qui opérationnalise la méthodologie. Ce rapport doit contenir l'ensemble des informations jugées pertinentes recueillies au cours de l'exécution de la méthodologie. Parmi toutes ces informations, on retrouve les objectifs des échanges et les problèmes rencontrés ainsi que les solutions retenues.

Les trois premières parties sont plus détaillées dans les sections 3.3 à 3.5 qui suivent. La définition des objectifs et du domaine d'échange est d'abord présentée.

3.3 DÉFINITION DES OBJECTIFS ET DU DOMAINE D'ÉCHANGE (A1)

La définition des objectifs d'échanges constitue la première étape de la méthodologie laquelle consiste à déterminer les paramètres et les résultats recherchés par l'entreprise. Ces objectifs servent de guide à la réalisation de ladite méthodologie, en balisant par la sélection des cas utilisés pour les échanges. Cette sélection consiste en un ensemble de fichiers comblant les exigences énoncées dans les objectifs.

Le diagramme de la figure 3.5 se veut représentatif de la définition des objectifs à atteindre lors des échanges de données ainsi que du domaine d'échange. Définition dont il sera question dans cette section.

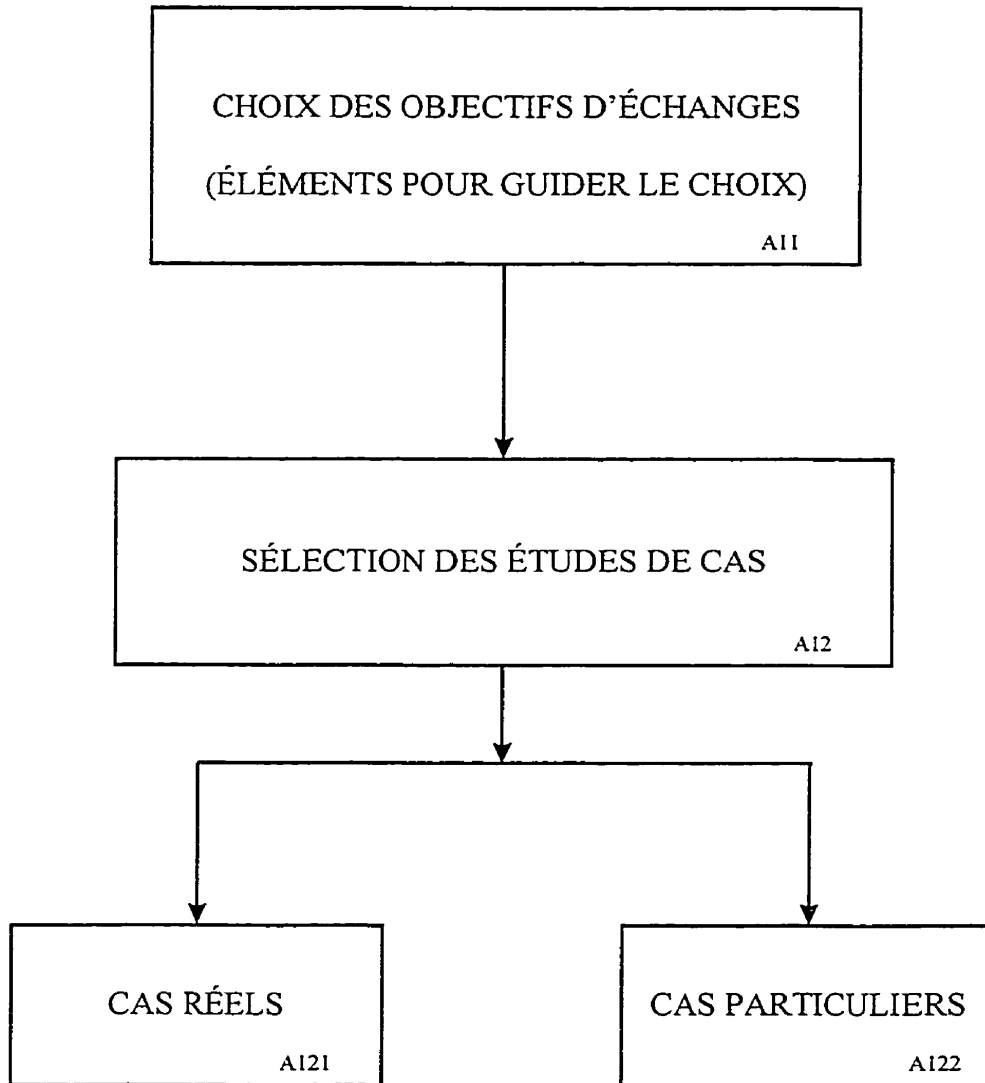


Figure 3.5 Définition des objectifs et du domaine d'échange

Comme on peut le constater à la lecture de la figure 3.5 ci-dessus, les phases importantes de cette étape de la méthodologie sont:

- 1) Choix des objectifs (paramètres et résultats recherchés)
- 2) Sélection des fichiers pour la réalisation des tests

Des détails concernant ces deux phases sont maintenant présentés.

3.3.1 Définition des objectifs d'échanges (A11)

Les échanges de données effectués entre plus d'un partenaire est une procédure connue par plusieurs et varie selon le type d'échanges ainsi que selon les partis impliqués. Comme on le sait, chaque entreprise possède des objectifs qui lui sont particuliers. Elles n'ont pas toutes les mêmes priorités et exigences pour chaque étape de la mise sur le marché d'un produit. Il arrive parfois que les objectifs, les priorités et les exigences varient d'une entreprise à une autre.

Peu importe si les partis qui désirent échanger des données ont la même ligne de pensée ou non, il est central de fixer des objectifs d'échanges afin d'éviter toute confusion. En effet, le processus pour échanger des données ne consiste pas qu'à faire un simple échange mais à le faire de façon cohérente et efficace. Ces objectifs sont des éléments qui guident le choix des cas qui sont utilisés pour les échanges et la spécificité de ces objectifs varie selon les entreprises, la complexité des échanges et les exigences à satisfaire.

Les objectifs d'échanges énumérés à cette partie de la méthodologie sont liés très étroitement avec le troisième niveau de la partie portant sur les tests de vérification des fichiers. En effet, la majorité des objectifs sont déterminés en fonction des applications pouvant être réalisées avec un fichier récupéré.

Ces objectifs d'échanges sont répartis en deux classes principales: les objectifs où seul un choix de réponse (oui ou non) est requis et les objectifs où une réponse manuscrite est requise. Différents types d'objectifs sont inscrits dans les tableaux 3.1 et 3.2 ci-dessous. Ils représentent une liste d'objectifs sommaire qui doivent être complétés par rapport aux exigences de l'entreprise.

Tableau 3.1 Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Utilise le(s) système(s) CAO: en interne..... avec des contracteurs.....		
Modèles sous forme: 3D: filaire..... CSG..... B-Rep..... surfacique..... 2D: avec annotations..... sans annotations.....		
Fichiers contenant: 1) une seule pièce..... 2) assemblage (2 pièces et plus)...		
Documentation technique et de maintenance		
Intégrer les fichiers pour: 1) conception..... 2) fabrication.....		
Taille des fichiers d'échanges: petite..... moyenne..... grande.....		

Tableau 3.1 (suite) Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Désire réaliser les applications spécifiques suivantes: STL..... FEA..... Si oui: utilise le module de FEA du système CAO..... utilise un logiciel spécialisé en FEA..... Étude cinématique..... Apport de modifications..... Application de FAO.....		
Nombre de systèmes CAO impliqués dans les échanges: un seul..... deux..... trois et plus.....		
Utilisation des fichiers: Échange unidirectionnel..... Échanges bidirectionnels: retournés une seule fois..... retournés plusieurs fois (processus itératif).....		

Tableau 3.2 Objectifs d'échanges à réponse manuscrite

OBJECTIFS D'ÉCHANGES	RÉPONSES MANUSCRITES
Budget disponible	
Quel est l'équipement informatique (hardware)	

Une fois les objectifs énoncés, les fichiers qui sont à l'étude peuvent être sélectionnés.

3.3.2 Sélection des études de cas (A12)

La sélection des études de cas consiste à choisir les fichiers types qui seront utilisés pour effectuer les tests de vérification d'échanges de données. Les cas doivent être choisis en fonction des objectifs d'échanges, sinon les tests perdent leur validité.

Il est préférable de choisir plus d'un cas, c'est-à-dire de choisir un ensemble de cas. Si l'on désire tester des assemblages, on peut choisir des fichiers composés de plusieurs pièces. Si toutefois, on désire tester l'efficacité de procédés de fabrication, on peut choisir des fichiers contenant chacun qu'une seule pièce. Différents types de fichiers peuvent être choisis pour étude mais le choix fait, se doit toujours d'être lié aux objectifs retenus.

L'ensemble des cas peut être exclusivement composés de cas réels ou particuliers mais peut aussi être la résultante du mélange des deux. Pour fins de compréhension, une courte description des cas réels et particuliers suit.

Cas réels (A121)

Les cas réels sont des fichiers qui existent déjà dans une entreprise et qui appartiennent à un modèle faisant partie d'un produit particulier de l'entreprise. Par exemple, pour une compagnie qui fabrique des voitures, un cas réel peut être un modèle de la porte du conducteur.

Cas particuliers (A122)

Lorsqu'on parle de cas particuliers, on fait allusion à des fichiers qui ne sont pas des cas réels provenant d'un modèle de l'entreprise mais des fichiers qui ont été modélisés

spécialement pour exécuter les tests d'échanges de données par rapport aux objectifs. Ces fichiers représentent des géométries réalistes qui sont fréquemment utilisées à l'intérieur de modèles réels dont la complexité varie. Ce type de cas est utilisé lorsqu'une entreprise veut préserver la confidentialité de ses produits ou s'il n'existe pas de fichiers contenant les aspects spécifiques sujets à analyse.

3.4 TRADUCTION ET RÉCEPTION DES FICHIERS D'ÉTUDE DE CAS (A2)

Cette partie présente les procédures suivies pour réaliser les trois principales phases de la traduction et réception des fichiers d'étude de cas. Ces phases sont: la traduction, directe ou par l'intermédiaire de IGES, des fichiers du format natif vers STEP; l'envoi des fichiers STEP ainsi que les analyses à effectuer lorsqu'ils sont reçus; et la traduction des fichiers STEP vers le format du système CAO de réception.

3.4.1 Traduction des fichiers (A21)

La traduction des fichiers en format STEP peut s'effectuer directement, sans étape intermédiaire, ou par l'entremise de la norme IGES. Cette dernière démarche existe afin de s'orienter graduellement vers l'utilisation de la norme STEP et de délaisser IGES qui est la norme la plus connue et la plus utilisée présentement.

La traduction directe est réalisée lorsqu'un processeur STEP est disponible. Elle consiste à ouvrir à l'écran les fichiers qui doivent être traduits pour fin de vérification et à effectuer la traduction en choisissant le bon protocole d'application ainsi que le bon type de format de traduction. Lorsque la traduction via IGES est utilisée, il s'agit de commencer par la traduction des fichiers en format IGES et par la suite, d'effectuer la migration de ces fichiers de IGES vers STEP à l'aide d'outils, tel le logiciel STEP/Works ou d'autres logiciels spécialisés dans la migration de IGES vers STEP. Il est à noter que l'une de ces

deux démarches s'emploie lorsqu'un processeur STEP ou un processeur IGES est disponible. Toutefois, si tel n'est pas le cas, les fichiers doivent être soumis à un organisme afin d'être traduit dans le format STEP.

Le diagramme de la figure 3.6 ci-dessous représente les différentes voies à exploiter pour effectuer une traduction d'un fichier d'un logiciel de modélisation vers STEP.

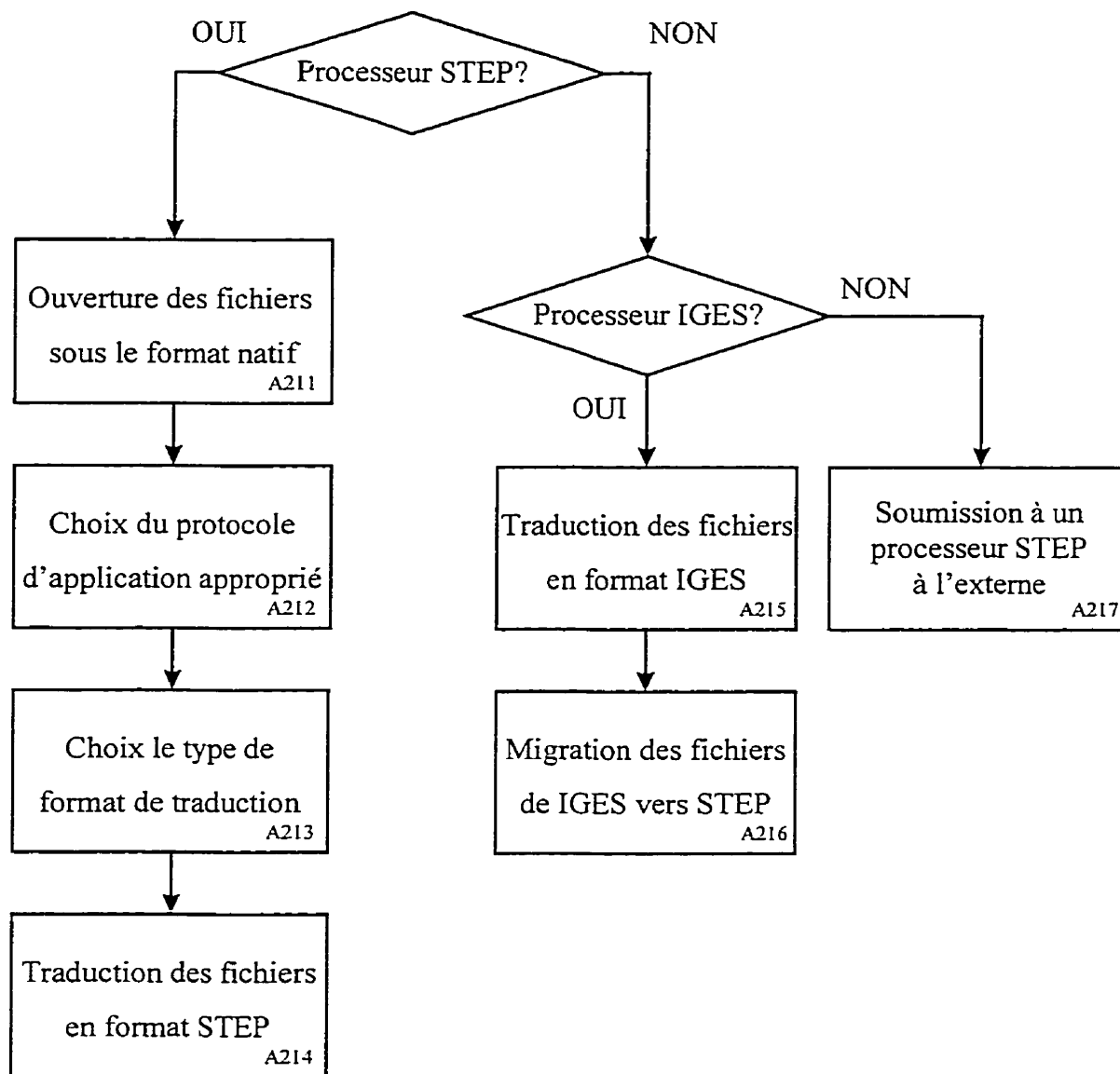


Figure 3.6 Traduction des fichiers sous le format STEP

Comme on peut le constater à la vue de la figure 3.6 ci-dessus, les principales phases de la traduction des fichiers sont réalisées en fonction des conditions à remplir. Ces phases sont:

- 1) Processeur STEP disponible:
 - Ouverture à l'écran des fichiers à traduire sur le logiciel d'origine

- Choix du protocole d'application et des autres paramètres (ex: repère)
 - Choix du type de format de traduction (ex: ASCII ou binaire)
 - Traduction des fichiers en format STEP
- 2) Processeur IGES disponible:
- Traduction des fichiers en format IGES
 - Conversion des fichiers IGES en format STEP à l'aide d'un logiciel capable d'effectuer cette opération (ex: le logiciel STEP/Works)
- 3) Aucun processeur disponible:
- Soumission des fichiers à un organisme pour traduction en format STEP

Voici maintenant des spécifications concernant les phases présentées ci-dessus.

Comme énoncé précédemment, il existe trois voies possibles pour traduire les fichiers en format STEP en lien avec les conditions suivantes. Dispose-t-on d'un pré-processeur STEP? Possède-t-on un pré-processeur IGES? La prise de décision est orientée par les conditions à remplir.

La première voie s'applique lorsque la première condition est remplie, c'est-à-dire que l'on dispose d'un pré-processeur STEP. Elle consiste à suivre les quatre phases suivantes: ouverture des fichiers désirés sous le format d'origine, choix du protocole d'application approprié (s'il y a lieu), choix du type de format de traduction (ex: ASCII) et traduction des fichiers en format STEP.

La deuxième voie s'applique lorsque la première condition ne peut être remplie mais que la deuxième l'est, c'est-à-dire que l'on détient un pré-processeur IGES et que l'on souhaite passer par IGES pour effectuer les échanges de données. La procédure se compose alors de deux phases: traduire les fichiers sous le format IGES et réaliser la

migration de IGES vers STEP. Cette option est disponible jusqu'à ce que STEP devienne la norme couramment utilisée.

La troisième voie envisagée s'applique dans le cas où on ne dispose pas d'un pré-processeur STEP ni d'un pré-processeur IGES ou encore, que l'on détient un pré-processeur IGES mais que l'on ne désire pas passer par IGES pour effectuer les échanges de données. À ce moment-là, il faut soumettre les fichiers à un centre STEP ou à un organisation qui possède un pré-processeur STEP, pour la traduction des fichiers (A217³). Le recours à cette voie est pertinent pour deux types de situation: lorsqu'une entreprise doit absolument réaliser des échanges de données avec une autre entreprise n'ayant pas le même système CAO et qu'aucun processeur STEP ou IGES n'est disponible; lorsqu'une entreprise désire effectuer des tests préliminaires de STEP afin de déterminer si elle plantera cette norme ou non.

Première voie

Cette première voie renvoie aux quatre étapes qui suivent.

Ouverture des fichiers (A211)

L'ouverture des fichiers est indispensable pour certains logiciels de modélisation tel Pro-E, mais ne l'est pas nécessairement pour d'autres comme UG et CATIA. Pour Pro-E, omettre d'ouvrir les fichiers empêche leur traduction puisque c'est l'ouverture qui rend active l'option de traduction en format STEP. Pour UG et CATIA, les fichiers peuvent être traduits sans avoir été préalablement affichés à l'écran. Même si cette ouverture n'est pas obligatoire, elle est recommandée puisqu'elle permet de vérifier quels sont les

³ Numérotation indiquée dans le bloc de la figure 3.3 associé à cette phase pour faciliter le lecteur à le repérer.

fichiers à traduire. On peut parler en termes de sélection de fichiers qui est une opération très simple.

Choix du protocole d'application (A212)

Lorsqu'un système CAO possède l'option de traduction vers le format STEP, il y a toujours au moins un protocole d'application présent avec cette option. C'est le protocole d'application qui spécifie l'information produit relative à une application, laquelle est nécessaire lors d'échanges de données. Un protocole d'application spécifie un sous-ensemble des ressources d'échange définies par la norme pour un domaine d'application particulier.

Le choix du protocole d'application n'est pas réalisé à chaque fois qu'un échange de données est effectué car certains logiciels de modélisation n'ont qu'un seul protocole d'application disponible. Par conséquent, le choix du protocole n'est pas requis dans ce cas puisque tout ce fait de façon automatisée.

Par ailleurs, certains systèmes CAO (ex: CATIA) offrent la possibilité de choisir parmi plus d'un protocole d'application. Les AP ne transfèrent pas le même type d'information, il est important de choisir le bon protocole afin que l'information échangée soit bien celle désirée en regard des objectifs et du domaine d'application (ex: secteur automobile). Il faut aussi s'assurer que le système récepteur utilise le même protocole d'application.

Jusqu'à tout récemment, le protocole d'application le plus utilisé, soit dans les proportions de 90%, est l'AP203. Toutefois, certains protocoles d'application tels l'AP202 et l'AP214 commencent à être utilisés sur le marché mais la plupart sont encore à la phase d'approbation. Quant au protocole d'application 214, il connaît une utilisation

grandissante puisqu'il réalise non seulement le même travail que l'AP203, mais le surpasse. Le contenu de l'AP214 est donc plus complet que celui de l'AP203.

Choix du type de format de traduction (A213)

Avant d'effectuer la traduction des fichiers dans le format STEP, il faut déterminer le type de format désiré pour la traduction. Pour ce faire, deux types d'informations doivent être spécifiées: le fichier STEP⁴ doit-il être en ASCII ou binaire et son contenu doit-il être en "short names" ou "long names". L'exemple 3.6 de la section 3.5.1 à la page 104 explique la différence entre les types "short names" et "long names".

Traduction des fichiers en format STEP (A214)

Les trois premières étapes une fois complétées, on passe à l'étape finale qui consiste à traduire les fichiers en format STEP. Cette dernière requiert normalement peu d'investissement de la part de l'opérateur puisque la traduction est une opération informatique, assurée par le logiciel qui effectue l'ensemble du travail.

Quand au nombre de paramètres (ex: repère, nom du fichier STEP, etc..) à identifier lors d'échanges, ils varient pour chaque logiciel de modélisation. À cet égard, il est nécessaire de bien maîtriser les éléments qui sont reliés à la procédure de traduction de fichiers pour le logiciel de modélisation utilisé. Les points 3 à 7 de l'exemple 3.1 ci-dessous présente les paramètres à spécifier pour le logiciel de modélisation CATIA.

⁴ Fichier qui est dans le format prescrit par la partie 21 de la norme ISO 10303.

Procédure de traduction de fichiers en format STEP pour trois systèmes CAO

Comme chaque système CAO traduit des fichiers en format STEP d'une manière qui lui est propre, voici trois exemples qui présente comment les systèmes CAO CATIA, Pro-Engineer et Unigraphics effectuent ce processus de traduction.

Exemple 3.1 Processus de traduction en format STEP pour CATIA

- 1) Choisir la fonction UTILITY
- 2) Choisir l'option CATSTP: Translate Catia Model (s) into STEP file(s)
- 3) Choisir le répertoire où est le fichier
- 4) Choisir le fichier
- 5) Choisir du nom du fichier STEP
- 6) Choisir l'emplacement où le fichier STEP doit être placé
- 7) Choisir le format du fichier STEP: "short names" ou "long names" (OPTIONEL)
- 8) Déterminer si un fichier STEP du même nom mais plus ancien doit être remplacé par le plus récent (OPTIONEL)
- 9) Traduire le fichier en format STEP en exécutant la fonction
- 10) Fermer l'option de traduction

Exemple 3.2 Processus de traduction en format STEP pour Pro-Engineer

- 1) Ouvrir le fichier
- 2) Choisir la fonction INTERFACE
- 3) Choisir l'option EXPORT
- 4) Choisir l'option STEP
- 5) Déterminer le nom du fichier STEP ou accepter celui que le système donne par défaut
- 6) Choisir le type du contenu à traduire:
 - Wireframe

- Surfaces
 - Solid
 - Shells
 - Datum curves, points or axes
- 7) Sélectionner le repère
- Create (créer un repère)
 - Select (sélectionner un déjà existant)
 - Default (prendre celui que le système donne par défaut)
- 8) Traduire le fichier en format STEP

Exemple 3.3 Processus de traduction en format STEP pour Unigraphics

- 1) Choisir la fonction STEP203UG dans le menu de UG (traduction avec l'AP 203)
- 2) Choisir la direction de traduction de UG vers STEP
- 3) Choisir les opérations de contrôle de la traduction
 - Mode de conversion:
 - ⇒ On-line (affichage à l'écran des opérations lors de la traduction)
 - ⇒ Batch (contraire de on-line)
 - Le nom du fichier où sauver les valeurs par défaut (ex: couleurs, tolérances, repère) lors de la traduction de UG à STEP
 - Le nom du fichier contenant les valeurs par défaut à récupérer (ex: couleurs, tolérances, repère) lors de la traduction de STEP à UG
 - Le nom du fichier où sont placés les informations obtenues lors de la traduction
 - Le genre de messages à inclure dans le fichier d'information (log file)
 - Le genre de messages à afficher à l'écran lors de la traduction
 - Le type du contenu à traduire
 - ⇒ Product Data
 - ⇒ Wireframe

- ⇒ Surfaces
 - ⇒ Solid
 - ⇒ All
 - ⇒ None
- 4) Choisir les option de traduction
- La valeur de la tolérance
 - Méthode de correction par défaut pour la validation d'un solid
 - ⇒ Tolerant Modeling (correction des erreurs comme des jeux, des trous ou des chevauchements, à l'intérieur de la tolérance spécifiée)
 - ⇒ None
- 5) Sélectionner le fichier
- Choisir le répertoire où est le fichier
 - Choisir le fichier
 - Choisir le répertoire où placer le fichier STEP
- 6) Traduire le fichier en format STEP
- 7) Fermer l'option de traduction

Deuxième voie

Cette deuxième voie comporte les deux étapes ci-dessous décrites.

Traduction des fichiers sous le format IGES (A215)

La procédure de traduction de fichiers en format IGES est sensiblement la même que celle utilisée pour traduire des fichiers en format STEP. La traduction s'effectue par le logiciel de modélisation d'origine des fichiers à traduire. L'exercice est simple, il faut choisir l'option de traduction du format du système CAO vers IGES et ensuite spécifier tous les paramètres (ex: nom du fichier, le répertoire où il est situé etc.) qui sont requis

pour effectuer la traduction désirée. Ces paramètres peuvent varier d'un système CAO à un autre en raison de la façon dont le pré-processeur IGES a été implanté dans le système CAO. L'exemple 3.4 ci-dessous explique la procédure de transfert d'un fichier CATIA en format IGES.

Exemple 3.4 Processus de traduction en format IGES pour CATIA

- 1) Choisir la fonction UTILITY
- 2) Choisir l'option CATIGE: Convert data from Catia format to IGES format
- 3) Spécifier les paramètres⁵
- 4) Traduire le fichier en format IGES en exécutant la fonction
- 5) Fermer l'option de traduction

Migration des fichiers de IGES vers STEP (A216)

Dès que les fichiers sont sous le format IGES, il faut les traduire en format STEP. Cette traduction peut s'effectuer à l'aide de la fonction de traduction de IGES vers STEP du logiciel STEP/Works. Ce dernier est un logiciel qui permet de visualiser ainsi qu'analyser des fichiers STEP et possède aussi un convertisseur IGES à STEP. Si l'entreprise ne détient pas les outils mentionnés ci-dessus, il faut alors soumettre les fichiers en format IGES à un organisme externe qui possède l'outillage nécessaire pour effectuer la migration de IGES vers STEP.

3.4.2 Réception et étude de l'intégrité des fichiers (A22)

Lorsque les fichiers sont sous le format STEP, la procédure méthodologique passe à l'étape suivante qui consiste, tout d'abord, à recevoir les fichiers et ensuite, à vérifier si ces derniers n'ont pas été corrompus lors du transfert.

⁵ La liste des paramètres à déterminer est présentée en détail à l'annexe A3.

Actuellement, il existe cinq différentes voies par lesquelles des fichiers peuvent être transmis: le protocole de transfert de fichier d'Internet soit le "file transfer protocol - ftp", le modem, la bande magnétique, la disquette et le réseau interne d'une compagnie. Chacune de ces voies peut être utilisée seule ou en combinaison avec d'autres. De plus, les échanges de données peuvent être effectués à l'intérieur d'une entreprise et/ou entre deux entreprises. Peu importe qui sont les partis impliqués dans les échanges, c'est-à-dire deux départements d'une même entreprise et/ou deux entreprises différentes, il faut noter la façon dont ces fichiers ont été transférés.

Lorsque la manière utilisée pour effectuer le transfert des fichiers est connue, une vérification visant à s'assurer que les fichiers n'ont pas été corrompus lors du transfert, c'est-à-dire vérifier leur intégrité, s'impose. Il s'agit donc ici de mettre en place un mécanisme qui s'assure de l'intégrité de la transmission après le transfert du fichier. Un exemple d'un mécanisme est une vérification qui consiste à examiner le statut des fichiers c'est-à-dire, analyser si la taille (l'espace mémoire, le nombre de "bytes", que le fichier utilise) des fichiers est réaliste et quand le transfert a été effectué. Elle consiste aussi à effectuer une analyse sommaire du fichier, en affichant son contenu en format texte à la console et ce, afin de vérifier si, d'emblée, il est conforme à la norme.

Il serait aussi possible d'effectuer un test simple "checksum" sur le fichier pour s'assurer qu'il a été transféré au complet. Ce test simple ne fait toutefois pas partie de la norme présentement.

Lorsque les fichiers STEP sont reçus, il faut les transformer dans le format du système CAO de réception afin de pouvoir procéder aux tests d'analyses.

3.4.3 Traduction des fichiers dans le format du système CAO de réception (A23)

Il faut d'abord effectuer une transformation qui consiste à traduire les fichiers du format STEP au format du logiciel de modélisation.

Pour procéder à cette traduction, il faut tout simplement choisir, sur le système CAO désiré, l'option de traduction de STEP vers le format du logiciel de modélisation. Pour des logiciels comme CATIA ou UG, il faut spécifier avant de procéder à la traduction, des paramètres tels le nom du fichier, le répertoire où le fichier traduit doit être inséré etc.

Puisque cette partie de la méthodologie se réalise en une seule étape, aucune figure ni de description sous forme télégraphique sont présentées. Toutefois, cette partie est elle aussi représentée sur la figure 3.1. Lorsque cette partie de la méthodologie est complétée, il reste une dernière étape à réalisée avant la rédaction du rapport qui est la vérification des fichiers.

3.5 VÉRIFICATION DES FICHIERS (A3)

La vérification des fichiers est l'étape qui suit celles d'analyses de bases. Elle constitue le coeur de la méthodologie puisque que c'est à ce niveau que l'on étudie le degré de réussite des échanges de données à travers les niveaux de test, ce qui entraîne des analyses avec des degrés de complexité différents.

Les niveaux de vérification de test sont les diverses étapes par lesquelles un fichier échangé entre deux systèmes CAO est analysé afin de déterminer s'il est utilisable au plan fonctionnel pour une entreprise. Il y a trois niveaux de test: le premier niveau est la vérification de la conformité; le deuxième niveau consiste en la vérification de la

sémantique; finalement, le troisième niveau renvoie à la vérification des fonctions pouvant être effectuées.

La procédure utilisée pour vérifier les trois niveaux de test n'est pas unique. C'est pourquoi une démarche générale est proposée. Par ailleurs, certaines conditions doivent être satisfaites avant de pouvoir passer d'un niveau à l'autre. Si les exigences d'un des niveaux ne sont pas satisfaites, il faut revenir en arrière et procéder à des actions correctives, autrement dit, corriger le(s) problème(s). Les corrections apportées, de nouveaux tests sont exécutés afin de déterminer si le(s) problème(s) est résolu. Cette démarche peut être poursuivie jusqu'à ce que les fichiers atteignent un niveau de réussite satisfaisant.

Il importe maintenant de présenter en quoi consistent les différentes étapes relatives à chacun des niveaux de tests afin de réaliser ces tests de façon chronologique et selon le principe de la complexité croissante.

Il est à noter que les résultats des tests doivent satisfaire aux exigences des partis impliqués dans la réalisation de la méthodologie avant que l'on puisse passer à la prochaine étape. Cette condition concerne les trois niveaux de tests.

3.5.1 Niveau 1 Conformité (A31)

On peut définir le niveau 1 comme étant la vérification de la conformité du format d'un fichier par rapport aux spécifications de la norme ISO 10303 (STEP). Il s'agit d'une opération de vérification visant à voir si le fichier échangé est conforme avec la partie 21⁶. Cette dernière est un format de structure d'échange qui transforme les

⁶ Spécifications inscrites à la section 5.4 de la norme 10303-21.

éléments du langage EXPRESS en une représentation donnée en utilisant un code de texte clair.

Afin de vérifier si les fichiers remplissent les exigences du niveau 1, il y a une procédure à suivre. Tout d'abord, il faut effectuer un test de base consistant en l'ouverture du fichier sur le système CAO de réception. Ce test a pour but de détecter toute erreur majeure (ex: pièce manquante) présente dans le fichier échangé. Si ce dernier ne semble pas présenter de problèmes, on peut passer aux tests du niveau 2. Par contre, si un ou plusieurs problèmes se manifestent, il est alors nécessaire de procéder à deux analyses afin de recueillir le plus d'informations possibles sur les fichiers en question: la pré-analyse des fichiers (A311) et la vérification des données dans le langage EXPRESS (A312). Pour une meilleure compréhension de l'exercice, une description de ces analyses s'impose.

Les diagrammes des figures 3.7 et 3.8 représentent les tests qui doivent être effectués afin de vérifier que les fichiers échangés sont corrects au plan syntaxique.

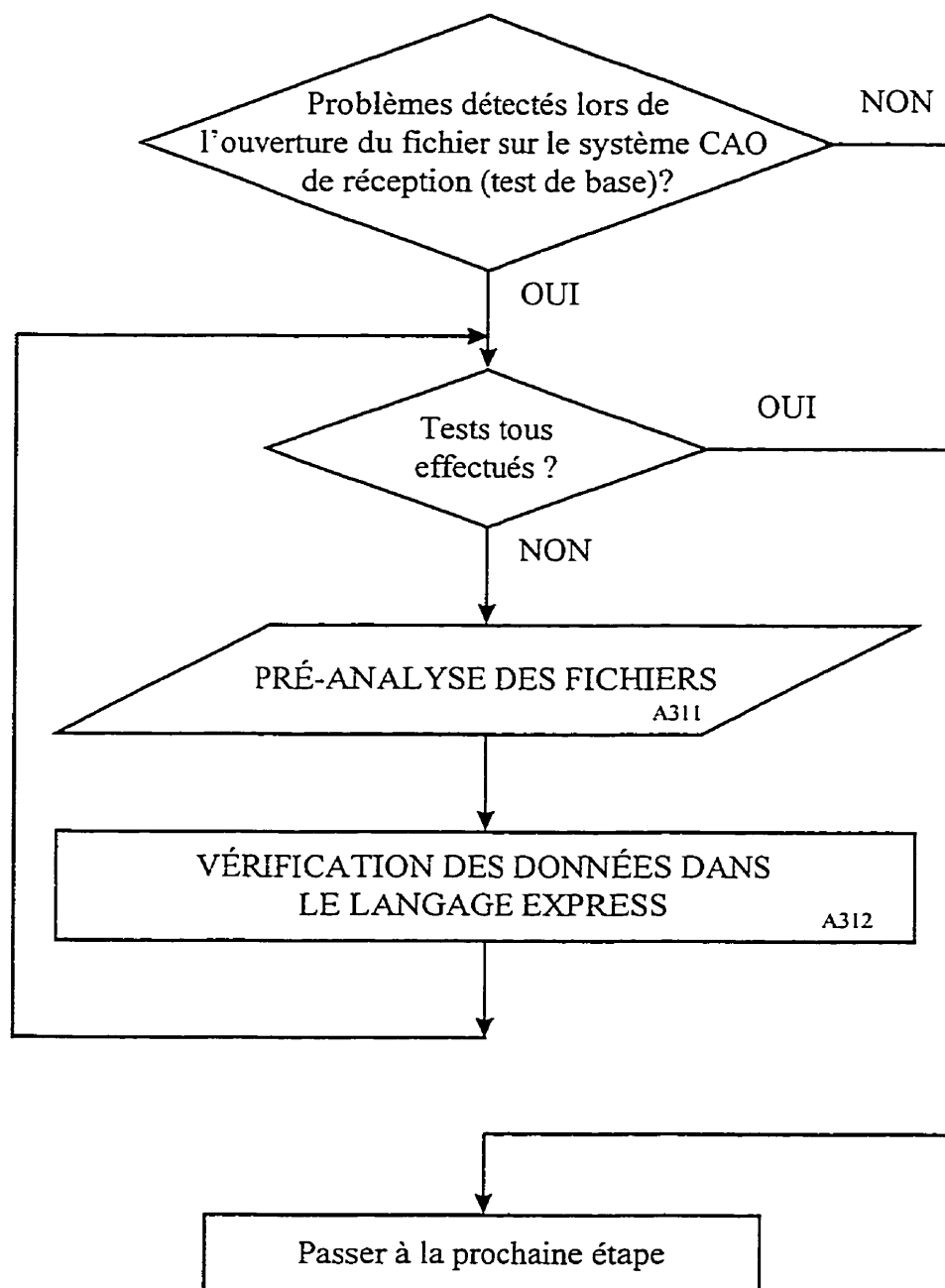


Figure 3.7 Tests Niveau 1: Conformité

À la lecture de la figure 3.7 ci-dessus, on constate que les étapes principales des tests de vérification de la conformité sont:

- 1) Détection d'erreurs majeures du fichier transféré (test de base)
 - Si non:
 - ⇒ Passe à la deuxième série de tests
 - Si oui:
 - ⇒ Analyses des informations contenues dans le fichier STEP (pré-analyse des fichiers)
 - ⇒ Vérification de la syntaxe des éléments texte, des caractères et des entités de définition géométrique dans le fichier STEP (vérification des données dans le langage EXPRESS)

Des explications plus détaillées des étapes à réaliser lorsque des erreurs majeures sont détectés lors du test de base sont maintenant formulées.

Pré-analyse des fichiers (A311)

La pré-analyse des fichiers se définit comme une série d'analyses des informations de l'ensemble des données reliées à la structure d'échange elle-même.

Cette partie des tests de vérification de la conformité est elle-même décomposée en plusieurs sous étapes. Ces dernières consistent en la vérification des données d'information de la structure d'échange à l'intérieur du fichier STEP ainsi qu'à la détermination du protocole d'application utilisé pour les échanges.

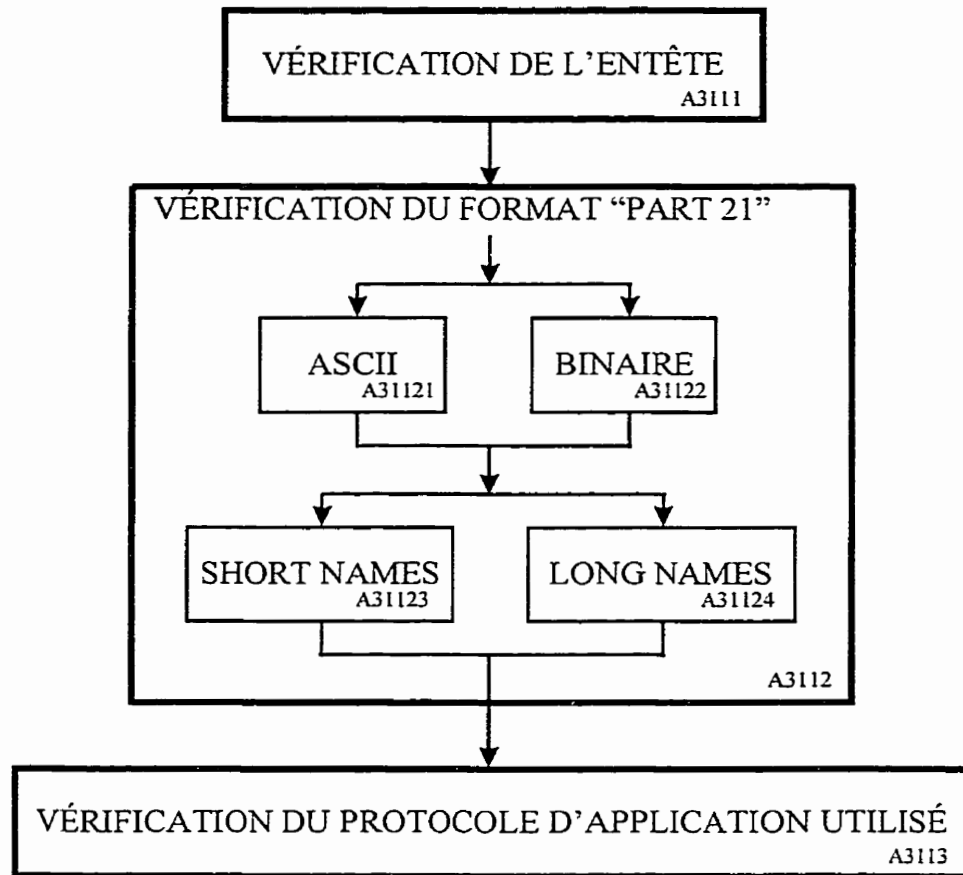


Figure 3.8 Pré-analyse des fichiers

Les étapes principales de la pré-analyse des fichiers se résume comme suit:

- 1) Vérification de l'entête du fichier STEP qui comprend, en autre, le nom du fichier
- 2) Vérification du format du fichier STEP
 - ASCII ou binaire
 - "short names" ou "long names"
- 3) Vérification du protocole d'application utilisé pour les échanges

Une description exhaustive de ces sous étapes est maintenant présentée.

Vérification de l'entête (A3111)

La vérification de l'entête a pour but de vérifier diverses informations pertinentes concernant un fichier échangé. Ces informations contenues dans l'entête du fichier STEP, facilitent le dépistage ainsi que la correction d'erreurs, se regroupent en trois classes: "file_description", "file_name" et "file schema".

Dans la première classe, on retrouve une description informelle du contenu du fichier ainsi que le niveau de mise en oeuvre⁷ que le système CAO récepteur employé doit soutenir pour les échanges.

La deuxième classe contient plusieurs informations relatives au contenu de la structure d'échange. Ces informations sont facilement repérables: le nom du fichier, la date et l'heure de création de la structure d'échange, le nom et l'adresse de la personne qui crée la structure d'échange, l'organisation de cette personne, le nom et la version du logiciel utilisé pour générer la structure d'échange, le nom du système d'origine du fichier utilisé pour créer la structure d'échange et enfin, le nom et l'adresse de la personne qui a autorisée l'envoi de la structure d'échange.

Pour sa part, la troisième classe contient l'information sur les "schémas EXPRESS, lesquels spécifient "the entity instances" dans la section des données"⁸ (ISO/DIS 10303-21 TC184/SC4, 1994).

Parmi l'ensemble de l'information que peut contenir l'entête d'un fichier, seulement une partie de cette information est réellement présente lors de la majorité des échanges de données. De plus, l'information fournie varie selon le système CAO utilisé car ils ne

⁷ Le niveau de capacité du système pour analyser les données contenues dans la structure d'échange.

⁸ Traduction libre.

fournissent pas tous le même contenu. Toutefois, les trois classes y sont toujours présentes.

Afin de mieux comprendre ce qui est contenu dans l'entête d'un fichier STEP, voici un exemple.

Exemple 3.5 Entête d'un fichier en format STEP

```
ISO-10303-21;
HEADER;
/ *-----
 * Exchange File generated by ST-DEVELOPER v1.4
 * Conforms to ISO 10303-21
 */
FILE_DESCRIPTION ((' ', '1');
FILE_NAME9 ('106177322', '1996-05-23T12:34:10-04:00', (' ', (' ',
 'ST-DEVELOPER v1.4', 'EDS - UNIGRAPHICS 11.0', (' '));
FILE_SCHEMA (('CONFIG_CONTROL_DESIGN'));
ENDSEC;
```

Explication des données d'information contenues dans les trois classes

```
FILE_DESCRIPTION ((' ', '1');
                ↑  ↑
                A  B
```

A: Description informelle du contenu

= Information absente

⁹ Les informations comprises dans cette classe sont inscrites sur deux lignes afin de minimiser l'espace de l'exemple mais elles doivent être inscrites sur des lignes distinctes.

B: Niveau de mise en oeuvre que le système CAO récepteur
doit soutenir

= 1 (Données de ges-
tion de configuration)

```

          C              D              E  F
          ↓              ↓              ↓  ↓
FILE_NAME ('106177322', '1996-05-23T12:34:10-04:00', ('), ('),
'ST-DEVELOPER v1.4', 'EDS - UNIGRAPHICS 11.0', '');

```

↑
G

↑
H

↑
I

C: Nom du fichier

= 106177322

D: Date et heure de création de la structure
d'échange

= 23/05/96 à 12h34 10 sec, 4
heures à l'ouest de Greenwich

E: Nom et adresse de l'auteur de la structure
d'échange

= Information absente

F: L'organisation de l'auteur de la structure
d'échange

= Information absente

G: Nom et version du logiciel de création de la
structure d'échange

= ST- Developer, version 1.4

H: Nom du système d'origine du fichier utilisé
pour créer la structure d'échange

= EDS-Unigraphics version 11.0

I: Nom et adresse de la personne qui a autorisée
la structure d'échange

= Information absente

```

FILE SCHEMA (('CONFIG_CONTROL_DESIGN'));

```

↑
J

J: Protocole d'application utilisé pour
 créer la structure d'échange = AP 203 ⇔ Configuration Control Design

Vérification du format "part 21" (A3112)

La vérification du format consiste à déterminer si la forme informatique du fichier STEP est en code ASCII (A31121) ou binaire (A31122). Pour déterminer s'il s'agit d'un fichier ASCII ou binaire, il faut faire apparaître le contenu du fichier à la console. Si ce dernier est lisible, c'est un fichier ASCII sinon, c'est un fichier binaire car ce type de fichier est une série de caractères codés.

Elle consiste aussi à vérifier si la partie 21 est composée de nom court "short names" (A31123) ou nom long "long names" (A31124). Un nom long est un mot qui est écrit dans son entier tandis que le nom court est une abréviation du nom long. Afin de mieux comprendre ce qu'est un fichier utilisant les "short names" ou "long names", voici un exemple.

Exemple 3.6 "Short names" et "long names"

<u>LONG NAMES</u>	<u>SHORT NAMES</u>
cartesian_point	cpt
vertex	vx

Vérification du protocole d'application utilisé (A3113)

Il est important de vérifier quel protocole d'application a été utilisé pour la traduction d'un fichier en format STEP car chaque protocole ne traduit pas nécessairement le même

contenu. En effet, l'AP 203 échange de la géométrie en trois dimensions (solide B-Rep) tandis que l'AP 201 échange de la géométrie en deux dimensions.

Il est possible d'effectuer cette vérification en regardant l'entête du fichier STEP à la ligne où est inscrit "FILE SCHEMA((""))". Le titre du protocole d'application utilisé y est inscrit. Si on se réfère à l'exemple 3.5, on peut constater qu'entre les guillemets de la ligne "FILE SCHEMA" il y a une inscription du type: "CONFIG_CONTROL_DESIGN", ce qui signifie "Configuration Control Design" et constitue le titre du protocole d'application 203.

Vérification des données dans le langage EXPRESS¹⁰ (A312)

Lorsque des problèmes surviennent, les causes sont parfois reliées à une erreur de définition d'une entité dans le fichier STEP qui comprend deux niveaux: les aspects de syntaxe et les aspects géométriques.

On parle d'aspects de syntaxe car on se réfère aux éléments texte et aux caractères, c'est-à-dire aux symboles (ex: \$). Pour mieux cerner ce que l'on entend par aspects de syntaxe, on peut prendre comme exemple la vérification d'une section du texte qui doit être en lettres majuscules et non en lettres minuscules ou encore la vérification qu'un signe \$ est bien représenté par un signe \$ et non un signe #.

Lorsqu'on parle d'aspects géométriques, on fait référence à la définition géométrique d'une entité, de la manière dont une entité, telle une ligne, doit être définie. Chaque système CAO définit ses entités géométriques d'une façon qui lui est propre. Toutefois, ceci ne doit en aucun cas influencer la manière dont les processeurs définissent les entités

¹⁰ Les vérifications effectuées ici se situent au niveau du langage EXPRESS puisque les spécifications des protocoles d'applications sont expliquées sous cette forme.

géométriques des fichiers STEP lors de la conversion CAO-STEP ou STEP-CAO. Il faut rappeler qu'il existe une seule manière dont les entités géométriques doivent être écrites dans les fichiers STEP. Une ligne peut être définie par un logiciel de CAO par 2 points mais la norme 10303 définit une ligne par un point cartésien et un vecteur. Cette sous-partie consiste alors à effectuer la vérification que la ligne est définie dans le fichier STEP par un point cartésien et un vecteur tel que dans l'exemple suivant:

```
# line (cartesian_point, vector)
```

Les analyses de cette partie de la méthodologie consistent à vérifier si les aspects de syntaxe et géométriques des données dans le langage EXPRESS sont inscrits correctement dans le fichier STEP. Il est impératif de mentionner que ces analyses sont réalisées seulement si des messages indiquant la présence d'un tel type d'erreur sont remarqués lors du test de base effectué au début du niveau 1. De plus, la vérification des données dans le langage EXPRESS est réalisée afin de palier au manque existant au niveau de la mise en oeuvre des tests de conformité. En effet, ces tests ne sont pas encore réalisés, par les laboratoires accrédités, sur les processeurs STEP des systèmes CAO afin de vérifier s'ils sont conformes avec la norme STEP.

Pour réaliser la vérification des données dans le langage EXPRESS, des informations sont recherchées dans la partie 21 de la norme et les parties de la série 200. Comme mentionné précédemment, la partie 21 contient l'ensemble de l'information relatives aux données contenues dans le fichier STEP sous forme physique. Ces données sont le résultat du "mapping" des données correspondantes dans le langage EXPRESS. Afin de bien cerner la différence existant entre le langage EXPRESS et la partie 21, des précisions suivent.

Une face avancée (`advanced_face`) est un "type de `face_surface` qui est contraint afin de s'assurer que la géométrie est définie directement et complètement. Une face avancée est

l'entité de niveau le plus élevé utilisée pour formuler la signification exacte d'une surface topologiquement entourée"¹¹ (ISO/IS 10303-203 TC184/SC4, 1994). Une `face_surface` est un type de face. La spécification EXPRESS de cette face s'écrit comme suit:

```

ENTITY advanced_face
  SUBTYPE OF (face_surface)
  WHERE
    WR1: SIZEOF ([ 'CONFIG_CONTROL_DESIGN.ELEMENTARY_SURACE',
                  'CONFIG_CONTROL_DESIGN.B_SPLINE_SURFACE',
                  CONFIG_CONTROL_DESIGN.SWEPT_SURFACE' ] *
                TYPEOF (face_geometry)) = 1;
    WR2: SIZEOF (QUERY (bnd <* bounds |
                      NOT (SIZEOF ([CONFIG_CONTROL_DESIGN.EDGE_LOOP',
                      CONFIG_CONTROL_DESIGN.VERTEX_LOOP'] * TYPEOF (bnd )) =
                      1))) = 0;
    WR3: SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* bounds |
                      'CONFIG_CONTROL_DESIGN.EDGE_LOOP' IN TYPEOF (bnds.bound)
                      ) | NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
                      NOT ('CONFIG_CONTROL_DESIGN.EDGE_CURVE' IN
                      TYPEOF (oe.edge_element)))))) = 0 ))) = 0;
    WR4: SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* bounds |
                      'CONFIG_CONTROL_DESIGN.EDGE_LOOP' IN TYPEOF (bnds.bound)
                      ) | NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
                      NOT (('CONFIG_CONTROL_DESIGN.VERTEX_POINT' IN TYPEOF
                      (oe.edge_start)) AND
                      ('CONFIG_CONTROL_DESIGN.VERTEX_POINT' IN TYPEOF
                      (oe.edge_end)))))) = 0))) = 0;

```

¹¹ Traduction libre.

```

WR5: SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* bounds |
    'CONFIG_CONTROL_DESIGN.EDGE_LOOP' IN TYPEOF (bnds.bound)
    ) | NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
    NOT (SIZEOF ([CONFIG_CONTROL_DESIGN.LINE',
    'CONFIG_CONTROL_DESIGN.CONIC',
    'CONFIG_CONTROL_DESIGN.POLYLINE',
    'CONFIG_CONTROL_DESIGN.PCURVE',
    'CONFIG_CONTROL_DESIGN.B_SPLINE_CURVE'] *
    TYPEOF (oe.edge_element\edge_curve.edge_geometry)) = 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* bounds |
    'CONFIG_CONTROL_DESIGN.EDGE_LOOP' IN TYPEOF (bnds.bound)
    ) | NOT ('CONFIG_CONTROL_DESIGN.ORIENTED_PATH' IN
    TYPEOF (elp_fbnds.bound)))) = 0;
WR7: (NOT ('CONFIG_CONTROL_DESIGN.SWEPT_SURFACE' IN
    TYPEOF(face_geometry)))
    OR
    (SIZEOF(['CONFIG_CONTROL_DESIGN.LINE',
    'CONFIG_CONTROL_DESIGN.CONIC',
    'CONFIG_CONTROL_DESIGN.POLYLINE',
    'CONFIG_CONTROL_DESIGN.B_SPLINE_CURVE'] *
    TYPEOF (face_geometry\swept_surface.swept_curve)) = 1);
WR8: SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* bounds |
    'CONFIG_CONTROL_DESIGN.VERTEX_LOOP' IN TYPEOF
    (bnds.bound)) |
    NOT (('CONFIG_CONTROL_DESIGN.VERTEX_POINT' IN
    TYPEOF (vlp_fbnds.bound\vertex_loop.loop_vertex)) AND
    ('CONFIG_CONTROL_DESIGN.CARTESIAN_POINT' IN
    TYPEOF

```

```

        (vlp_fbnds.bound #vertex_loop.loop_vertex\vertex_point.vertex_geometry)
        )))) = 0;
END_ENTITY;
(*

```

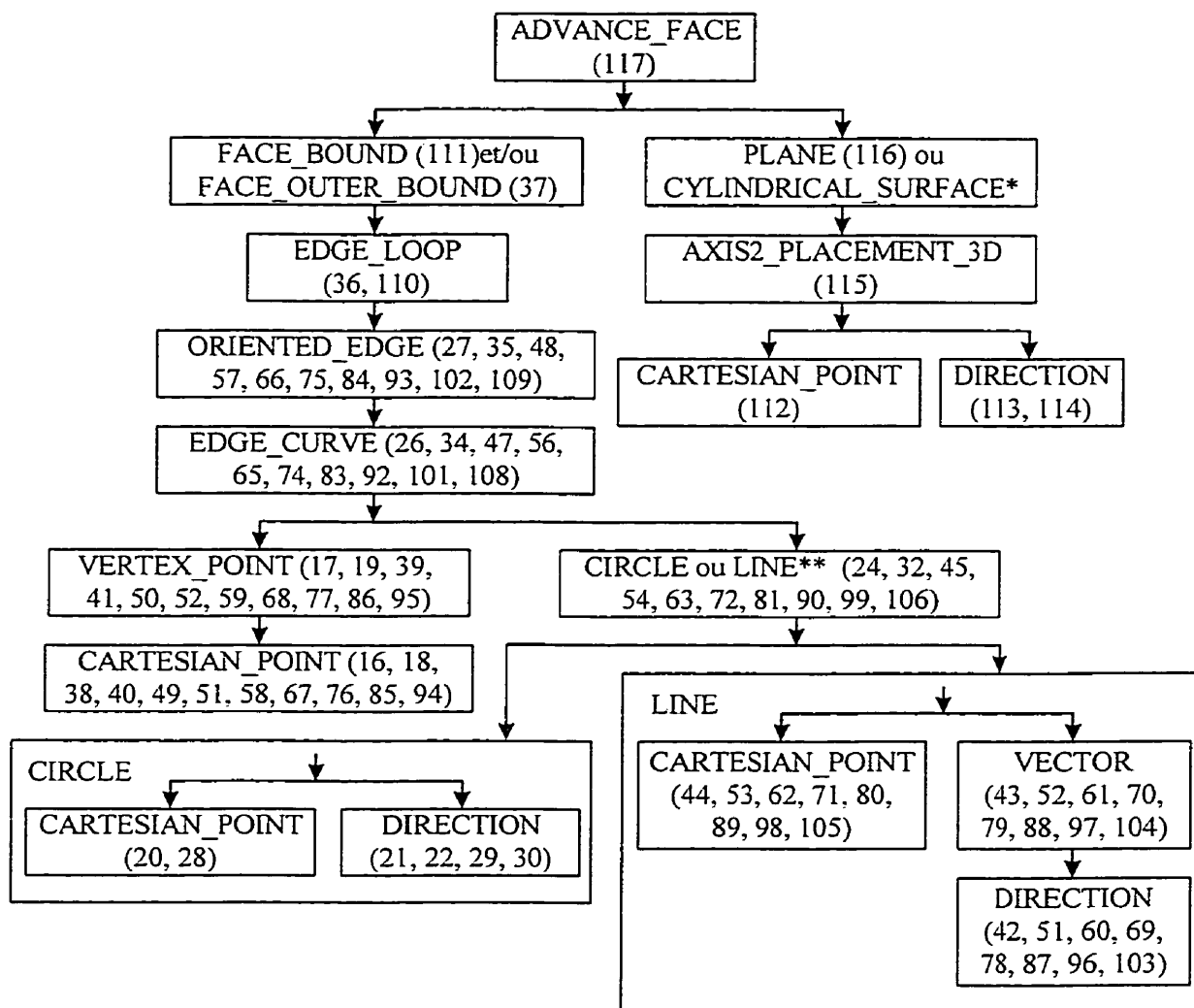
La signification des règles WR1 à WR8 sont les suivantes:

- WR1: La face doit être une “elementary_surface”, une “swept_surface” ou une “b_spline_surface”.
- WR2: Les frontières de la face avancée doivent être définies par des “vertex_loops” ou des “edge_loops”.
- WR3: La géométrie de toutes les arêtes qui entourent la face doivent être définies comme des “edge_curves”.
- WR4: Tous les sommets utilisés dans la définition de la face qui ont une frontière définie par une “edge_loop” doivent être de type “vertex_point”.
- WR5: Les types de courbes utilisées pour définir la géométrie des arêtes doivent être restreint à des lignes, des “conics”, des “polylines” ou des courbes b_spline.
- WR6: L’utilisation de “oriented_paths” dans la définition de “edge_loops” de la face avancée est défendu.
- WR7: Si la face est du type “swept_surface” alors la “swept_curve” utilisée dans la définition doit être une ligne, un “conic”, une “polyline” ou une courbe b_spline.
- WR8: Tous les sommets qui à l’origine ont une “vertex_loop” qui est définie comme une frontière de la face avancée doivent être du type “vertex_point” qui doivent avoir comme origine un point cartésien.

Dans la partie 21, la spécification d’une face avancée (advanced_face) est un peu différente. Si on se réfère à l’annexe A2, à la ligne 117, on y retrouve ceci:

```
#117 = ADVANCED_FACE (*FAC1', (#37, #111), #116, .F.)
```

Lorsque l'on décode le contenu de cette ligne, on y retrouve plusieurs informations qui suivent. Le nom de la face est *FAC1, elle est composée des frontières indiquées aux lignes 37 et 111, elle est située sur le plan de la ligne 116 et sa direction est orientée inversement ($F = \text{False}$) à la normale de la face. L'entité #37 est la frontière extérieure (FACE_OUTER_BOUND) de la face tandis que l'entité #111 est la frontière intérieure de la face (FACE_BOUND). Chaque frontière est une boucle fermée composée d'arêtes orientées et en forme de courbes ou de lignes. Le schéma représenté à la figure 3.9 illustre la composition d'une face orientée, dans ses moindres détails, avec les spécifications de la partie 21 associées.



* Une face est délimitée par un contour et s'appuie sur un plan ou une surface cylindrique

** Une courbe arête (edge_curve) est délimitée par deux "vertex_point" et elle s'appuie sur une ligne, un cercle, une B-spline etc...

Notes: 1) Les chiffres du schéma correspondent aux numéros de lignes du fichier STEP situé à l'annexe A2.

2) Chacun des éléments qui compose la face orientée a des spécifications EXPRESS qui lui sont propres.

Figure 3.9 Schéma détaillé d'une face avancée avec association de lignes du fichier STEP

Les parties de la série 200, quant à elles, renferment l'information complète à propos des protocoles d'application. Lorsque l'on regarde dans la documentation des protocoles d'application, il est très important de choisir la documentation se rapportant à l'AP utilisé lors des échanges de données.

À l'intérieur de la documentation des AP, il y a deux procédures d'incluses pour la recherche d'information désirée. La première consiste à regarder dans l'index à la fin du document, à repérer l'information particulière recherchée et à consulter la page où l'information (description et spécification EXPRESS) est inscrite sous forme détaillée. La deuxième consiste à repérer l'information désirée à travers les diagrammes AIM (sous forme EXPRESS-G) situés à la fin des documents des protocoles d'application. Ces diagrammes constituent la représentation graphique des spécifications du protocole d'application. Par ailleurs, ils sont moins précis que les définitions textuelles que l'on retrouve dans la documentation de l'AP. Pour fin de compréhension, reprennent la face avancée discutée précédemment et regardons sur le diagramme AIM des figures A1.4 et A1.5 à l'annexe A1. On retrouve la face avancée qui est un type de "face_surface" étant elle-même un type de face. Les éléments qui définissent une face s'applique aussi pour la définition d'une face avancée.

On remarque donc que la face pointe vers ses frontières qui à leur tour, pointent vers la boucle pour ensuite pointer vers les courbes et les points. À ce niveau, les éléments qui définissent une face, sont des items géométriques et non topologiques, ce qui explique la raison pour laquelle ces éléments sont représentés sur un diagramme différent. Il est important de mentionner que sur les diagrammes EXPRESS-G, les éléments sont regroupés par catégorie, c'est-à-dire que les éléments géométriques se regroupent sur les mêmes feuilles, lorsque nécessaire en raison d'un manque d'espace, comme sont les éléments topologiques, ceux de configuration, ceux de définition du produit, etc.

Les analyses mentionnées dans cette partie de la méthodologie sont reliées au premier niveau de test soit, celui de la conformité. Toutefois, il existe d'autres analyses et éléments qui se situent au même niveau et dont il est fait état à la section qui suit.

Éléments qui font partie du premier niveau

Voici quelques analyses et éléments, non mentionnés auparavant dans cette partie, qui font partie du premier niveau de test:

- la traduction d'un fichier qui a été bien converti du format CAO (ex: un fichier en format UG) dans le format STEP;
- le fichier STEP lui-même;
- les tests de conformité de la norme ISO 10303.

Les tests du niveau 1 sont suivis par ceux du niveau 2 soit, les tests de vérification de la sémantique dont des spécifications sont maintenant présentés.

3.5.2 Niveau 2 Sémantique (A32)

Les tests de vérification au niveau de la sémantique comprennent deux principales phases. Tout d'abord, les objectifs à atteindre à ce niveau sont déterminés. Cette opération permet d'identifier les tests qui seront exécutés par la suite. La deuxième phase constitue le coeur des tests de vérification de la sémantique puisque c'est à cette partie qu'ils sont exécutés. Le nombre de tests à réaliser à ce stade dépend des attentes de l'entreprise dont il a été question à la première phase ainsi que dans la partie de la définition des objectifs au début de la méthodologie.

Le diagramme de la figure 3.10 ci-dessous représente les tests qui doivent être effectués afin de vérifier si les fichiers échangés sont adéquats au plan sémantique.

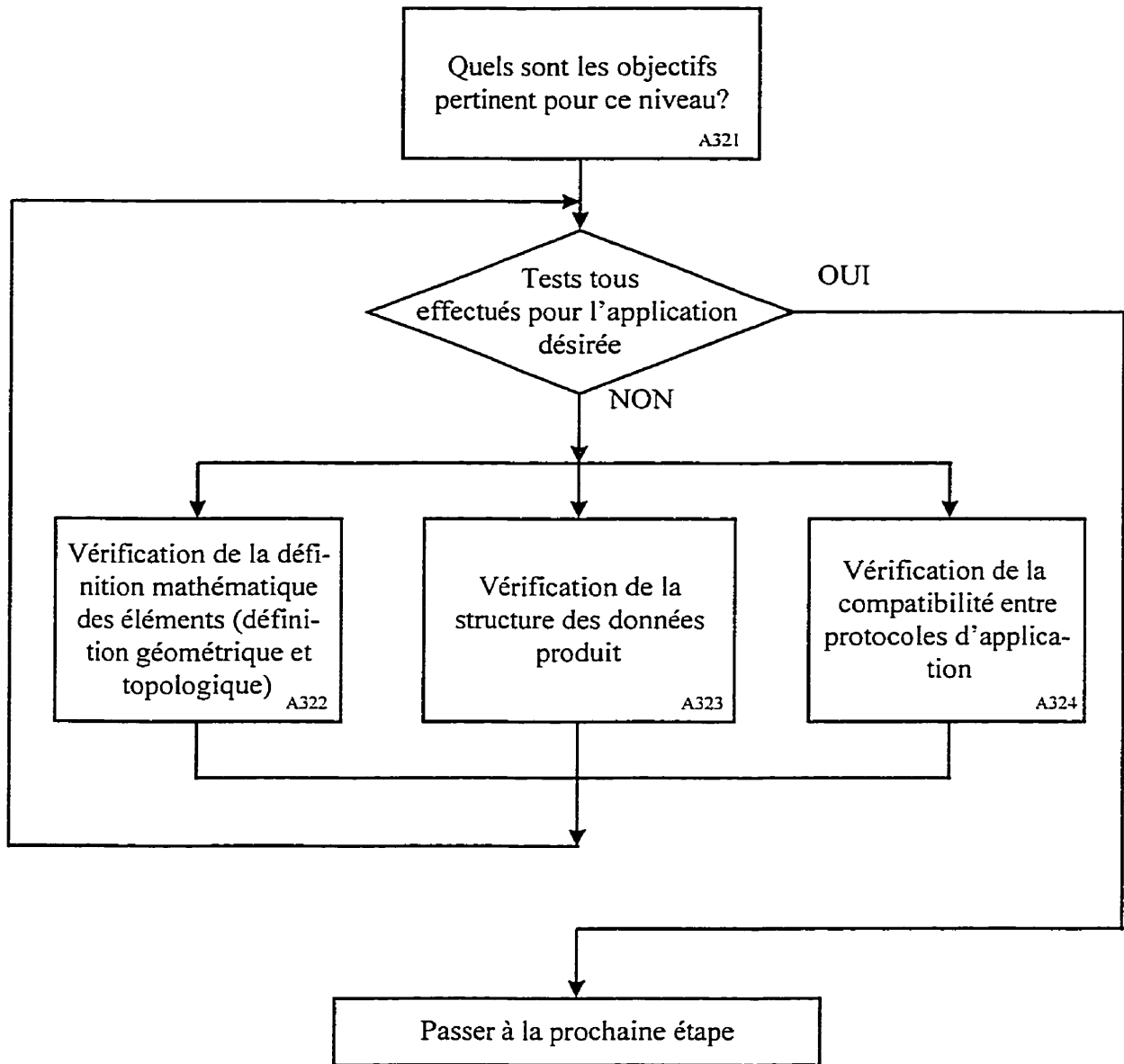


Figure 3.10 Tests Niveau 2: Sémantique

Comme on peut le constater sur la figure 3.10 ci-dessus, les principales phases du deuxième niveau de tests sont:

- 1) Détermination des objectifs pertinent pour ce niveau

- 2) Exécution des tests de vérifications du deuxième niveau, voici quelques exemples:
- Vérification de la définition mathématique des éléments (définition géométrique et topologique)
 - Vérification de la structure des données produit
 - Vérification de la compatibilité entre protocoles d'application

Lorsqu'on parle de vérification au niveau de la sémantique, on parle en termes de vérification de l'ensemble de l'information nécessaire, reliée à une entité, à savoir si elle est complète, claire et sans ambiguïté et qu'une seule interprétation de l'entité en question peut être perçue par tous les systèmes ou personnes qui interagissent avec ladite entité. L'organisation internationale des standards (International Standards Organization - ISO) définit la sémantique, du point de vue d'un langage de programmation, comme étant l'ensemble des relations entre les caractères ou groupes de caractères et leur signification, indépendamment de la façon de les employer ou de les interpréter. Afin d'avoir un meilleur aperçu de ce que l'on entend par vérification d'ordre sémantique, des clarifications s'imposent.

Vérification de la sémantique

Si on écrit Jean, on doit s'assurer que l'information associée à ce mot soit définie de manière claire afin que ce nom propre ne soit pas confondu avec un pantalon. Ici le contexte prend beaucoup d'importance. Voici des spécifications à ce sujet.

SIGNIFIANT	INFORMATIONS ADDITIONNELLES	SIGNIFIÉ
Jean	<ul style="list-style-type: none"> ● Prénom généralement donné au sexe masculin 	⇒ Nom propre

Jean

- Pantalons taillé dans un matériel de denim pouvant être de couleurs variées mais généralement de couleur bleu pâle ou marine

⇒ Pantalon de jeans

Le type de considérations présentes ci-dessus sont sont plutôt d'ordre linguistique. Or, dans cette étude, on s'intéresse davantage à l'aspect géométrique, au sens de la conception assistée par ordinateur (CAO). À ce sujet, les figures 3.11 et 3.12 font écho des précisions d'ordre sémantique au niveau de la CAO.

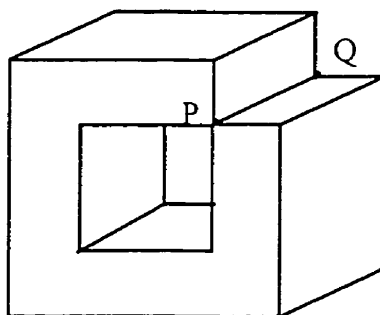


Figure 3.11 Solide

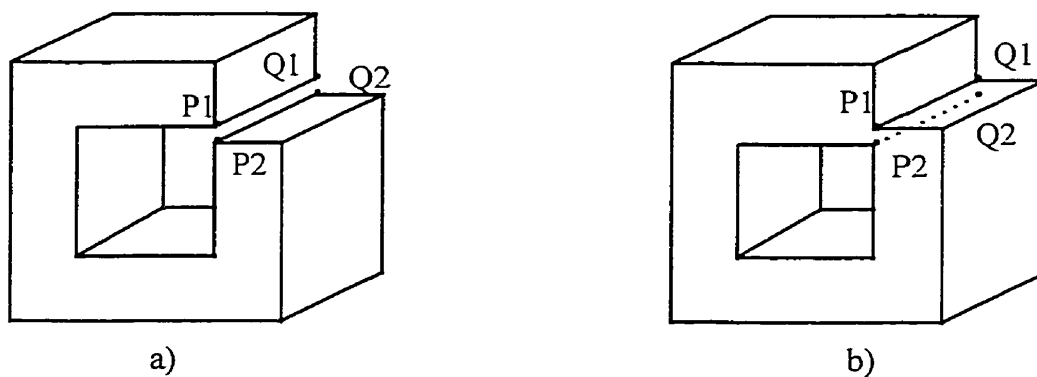


Figure 3.12 Possibilité de topologie du solide récupéré

La figure 3.11 illustre un solide dont l'arête PQ n'a pas d'épaisseur. En CAO, la modélisation solide doit permettre le calcul automatique de toute propriété géométrique d'un corps solide. Il faut donc une définition mathématique exacte des limites de la matière et établir si un point de l'espace est en dedans ou en dehors du solide et/ou de sa frontière. Elle doit aussi offrir une représentation des objets physiques qui se caractérise ainsi:

- 1) complète et non-ambigüe
- 2) appropriée pour tous les objets du monde de l'ingénierie
- 3) applicable sur les ordinateurs actuels

En outre, cette représentation doit permettre d'exclure les objets "absurdes", c'est-à-dire ceux que l'on ne peut pas fabriquer ou encore qui ne peuvent correspondre à une réalité physique.

L'arête PQ est une entité ambigüe et constitue un problème de sémantique en CAO puisqu'on ne sait pas de quel côté est la matière, c'est-à-dire qu'on ne sait pas si elle est en dedans ou en dehors du solide. De plus, le solide de la figure 3.11 représente un objet difficile à fabriquer et ne correspond donc pas à une réalité physique.

Si l'on récupère le solide de la figure 3.11, trois situations peuvent se produire:

- 1) Le solide est récupéré tel que celui illustré à la figure 3.11
- 2) Le solide est récupéré tel que celui illustré à la figure 3.12 a
- 3) Le solide est récupéré tel que celui illustré à la figure 3.12 b

Si le solide est récupéré comme c'est le cas sur l'une des deux illustrations de la figure 3.12, il y a un problème de sémantique puisque le solide récupéré n'est pas la représentation exacte de celui modélisé.

Le genre de géométrie tel que le solide de la figure 3.11 pose un problème non seulement lors d'échanges de données entre systèmes CAO, mais aussi lors de sa modélisation. En effet, certaines méthodes de modélisation de ce solide, telles l'union de deux solides en forme de L ou la soustraction de l'union de deux rectangles (rectangle à l'intérieur du cube et celui situé au coin supérieur droit de ce cube) à un cube, ne peuvent être réalisées. Toutefois, le système CAO CATIA permet de modéliser ce solide si on soustrait d'abord le rectangle du centre au cube et ensuite celui du coin supérieur droit au résultat de la première soustraction. Lorsque ce solide est échangé de CATIA-STEP et STEP-CATIA, le même solide est alors récupéré. Afin de vérifier si le solide récupéré est fermé, c'est-à-dire vérifier sa cohérence topologique, on peut mettre le fichier en mode de visualisation ombragée. S'il existe des incohérence dans le solide, par exemple des faces adjacentes qui ne possèdent pas d'arête commune, elles n'apparaîtront pas en mode ombragé.

La nature de la vérification sémantique étant cernée, des explications plus détaillées sur cette partie de la méthodologie suivent.

Afin de vérifier si les fichiers répondent aux exigences du niveau 2, il faut tout d'abord fixer les objectifs qui sont pertinent pour ce niveau (A321). Cette opération est nécessaire afin de connaître quels seront les tests à réaliser au niveau 2.

Puisque chaque entreprise se fixe des objectifs plus ou moins semblables, il est difficile d'énumérer toutes les analyses qui doivent être effectuées à cette étape. Toutefois, quelques exemples d'analyses peuvent être cités. Parmi toutes les analyses possibles, trois peuvent être mentionnés: la vérification de la définition mathématique des éléments

(définition géométrique et topologique) (A322), la vérification de la structure des données produit (A323) et la vérification de la compatibilité entre les protocoles d'application (A324).

La vérification de la définition mathématique des éléments consiste à vérifier l'exactitude de la représentation des surfaces c'est-à-dire la vérification du nombre de surfaces, une seule par rapport à plusieurs¹². Elle consiste aussi à analyser les définitions géométriques et topologiques d'entités, c'est-à-dire cibler les raisons pourquoi une entités est définie telle qu'elle est dans le fichier et à qui incombe la responsabilité (au logiciel ou à l'opérateur). Il est à noter que l'analyse des définitions géométriques et topologiques d'entités est réalisée seulement si une erreur a été repérée lors du même type d'analyse des tests du niveau 1.

La vérification de la structure des données produit consiste à vérifier comment ces données d'un fichier sont structurées, comme par exemple, la vérification que la représentation contenue dans un fichier, lorsqu'un fichier contient une pièce mais reproduite plusieurs fois (un "ditto" dans CATIA), est la pièce modélisée ou les copies de cette pièce. Des données produit sont, selon la norme ISO 10303, une représentation de faits, de concepts, ou d'instructions sur un ou plus d'un produit, décrite d'une manière formelle appropriée pour la communication, l'interprétation, ou le traitement par des êtres humains ou par de moyens automatiques. Un autre exemple est de vérifier comment est structuré un assemblage.

La vérification de la compatibilité entre les protocoles d'application consiste à vérifier si les résultats obtenus en échangeant un fichier avec deux protocoles d'application différents dans le même échange, c'est-à-dire échange d'un système CAO-STEP avec un AP et échange de STEP- système CAO avec un autre AP, sont compatibles.

¹² Cette vérification peut être réalisée en allant voir les résultats dans la liste lors du décodage.

Lorsque les tests de vérification de la sémantique sont terminés c'est-à-dire, qu'ils sont à un niveau satisfaisant pour l'entreprise, il est temps de passer aux tests de vérification des fonctions soit le troisième niveau.

3.5.3 Niveau 3 Fonctions (A33)

Le troisième niveau se définit comme étant les analyses des actions (ex: modifications, génération d'un fichier STL ...) qui peuvent être effectuées à l'intérieur d'une entreprise une fois le fichier récupéré. Ce niveau consiste à exécuter ces analyses qui, comme mentionné précédemment, sont directement liés aux objectifs d'échanges fixés au début de la méthodologie.

Le diagramme de la figure 3.13 ci-dessous représente les tests qui doivent être effectués afin de vérifier les fonctions, énoncées dans les objectifs d'échanges, qui peuvent être réalisées sur les fichiers échangés.

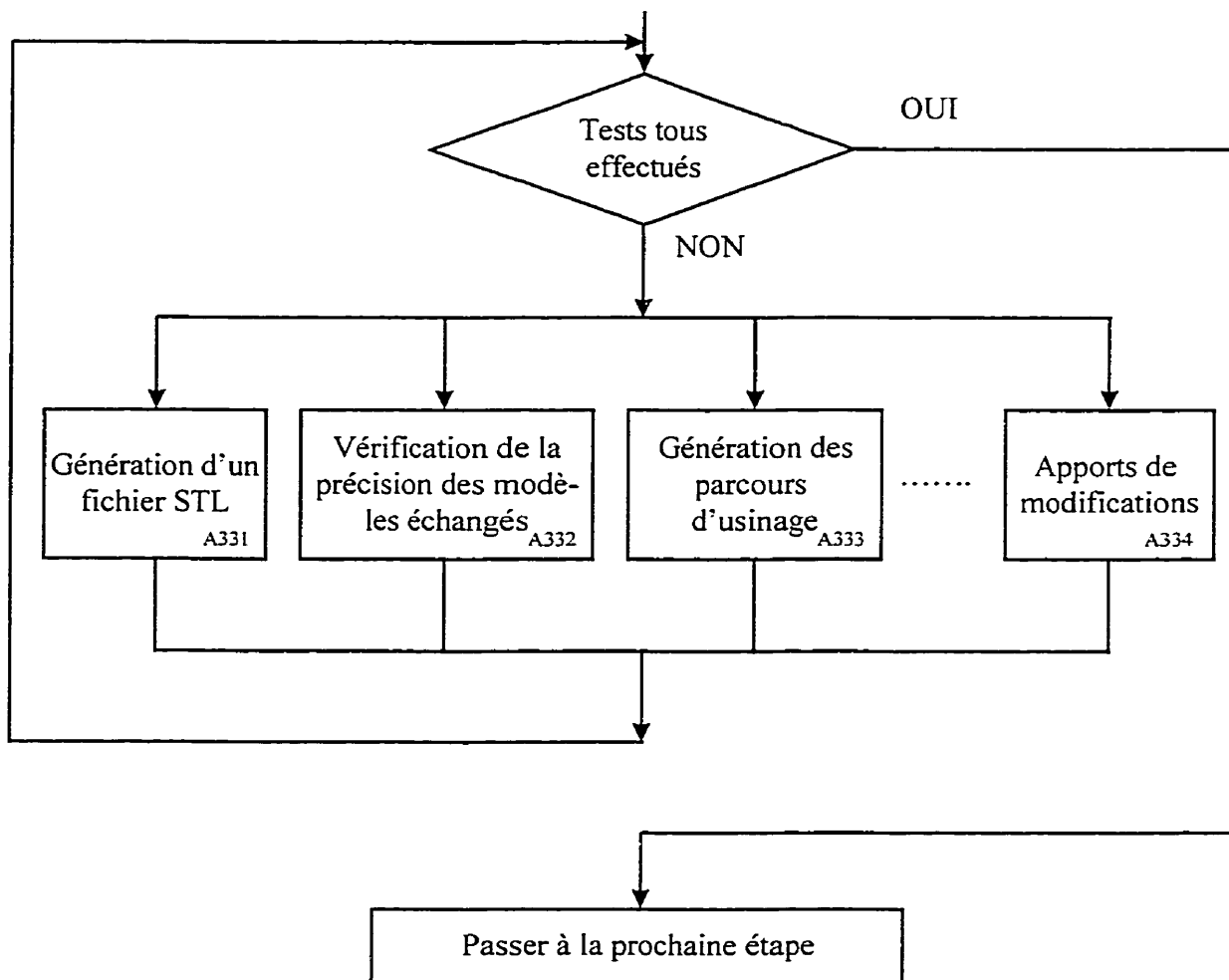


Figure 3.13 Tests Niveau 3: Fonctions

Comme on peut le constater à la figure 3.13, seulement quelques tests de vérification des actions pouvant être réalisées avec un fichier échangé sont présentés. Ces tests renvoient aux actions suivantes:

- Génération d'un fichier STL
- Vérification de la précision des modèles échangés
- Génération des parcours d'usinage
- Apports de modifications

Le passage de ce niveau de test est crucial pour une entreprise car c'est à ce moment que les objectifs de l'entreprise sont mis sur la sellette. C'est à ce niveau que l'on examine la gamme des opérations qu'il sera possible de réaliser avec les fichiers récupérés et ce, par rapport à ce que l'entreprise désire réaliser avec ces mêmes fichiers. Les tests sont aussi directement liés aux activités de l'entreprise. Par exemple, si une entreprise machine des pièces, des tests dirigés vers la génération de programmes d'usinage d'une pièce seront effectués pour savoir s'il est possible de réaliser une telle activité (vérifier qu'il n'existe pas de jeu entre les surfaces de la pièce). Si une entreprise est spécialisée dans les études de résistance des matériaux, des tests orientés vers le maillage de pièces seront réalisés.

Il existe une multitude de tests qui peuvent être effectués et comme mentionné au point 3.4.2, chaque entreprise se fixe des objectifs plus ou moins semblables aux autres entreprises, ce qui constitue un obstacle à l'énumération de l'ensemble des analyses pouvant être effectuées à cette étape. Toutefois, quelques exemples d'analyses peuvent être cités tels la génération d'un fichier STL (A331), la vérification de la précision des échanges (A332), la génération des parcours d'usinage (A333), la réalisation de modifications (A334) en autre.

Avant d'aller plus loin, il est important de mentionner que lorsque des échanges bidirectionnels sont requis, il existe une possibilité de transformation non-réversible de ces échanges. En effet, pour certains systèmes CAO, il arrive parfois, pour des entités, qu'il n'y ait pas d'opérateurs inverses. On est donc en présence d'une dégénérescence de l'information des fichiers originaux et ce, à partir de leur traduction dans un format neutre. Cette situation est un problème réel où dans le pire des cas il y a irréversibilité des échanges. Malheureusement, aucune norme ne solutionne ce type de problème.

Au niveau du développement d'un produit, il doit être constamment modifié au cours de son élaboration. Il faut donc que le modèle CAO produit soit conçu d'une façon qui

assure la facilité de modification. Au cours des dix dernières années, les systèmes CAO ont beaucoup évolués sous cet aspect et la venue des modeleurs paramétriques et variationnels (sans contraintes) tel Pro-Engineer a beaucoup influencé les développements. Tous les systèmes CAO offrent donc une fonctionnalité évoluée de modification des solides; cette réalité n'est pas étrangère à leur pénétration rapide dans le milieu industriel. Il faut donc que les échanges entre les systèmes CAO préservent la structure d'information requise pour assurer cette fonctionnalité de modification rapide des modèles.

Cette fonctionnalité est assurée en grande partie par quatres caractéristiques principales:

- 1) La construction d'une pièce par combinaison de primitives et d'opérations booléennes;
- 2) La paramétrisation de contours géométriques définis à partir de côtes et de contraintes sous jacentes: le solide est alors généré par extrusion;
- 3) La génération automatique d'arrondis entre les primitives; et finalement
- 4) La gestion d'assemblage d'un grand nombre de pièces de façon interactive.

Pour ce qui est de la facilité de modification au niveau 3, la norme d'échange doit donc s'assurer que la définition produit garde ses propriétés de modification à travers les échanges STEP. Ce type de fonctionnalité est donc vérifié au niveau 3.

La figure 3.14 ci-dessous représente l'arbre de construction d'une vis, c'est-à-dire les paramètres qui permettent de réaliser facilement une modification.

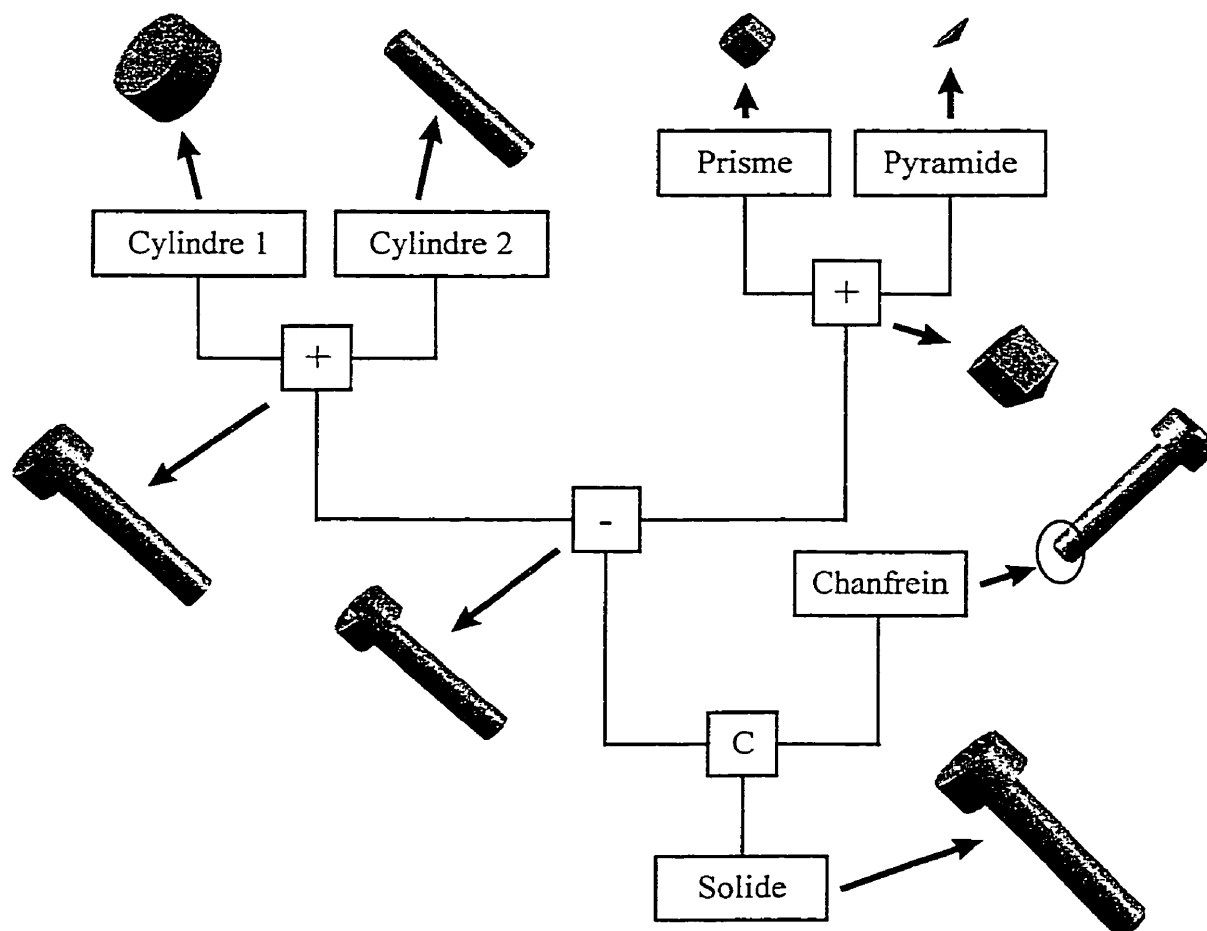


Figure 3.14 Arbre de construction d'une vis

Lorsque l'arbre de construction disparaît du fichier, il devient alors long et fastidieux d'y faire des modifications. Celles-ci doivent être effectuées sur chacune des faces touchant les modifications à réaliser ou il faut reconstruire les solides à partir des arêtes, lorsque celles-ci sont présentes dans le fichier et qu'elles peuvent être utilisées. Malheureusement, sur certains systèmes CAO tel Pro-Engineer, les entités de construction d'une pièce ne peuvent être employées.

La figure 3.15 ci-dessous illustre la vis de la figure 3.14 récupérée sur CATIA après avoir été échangée de CATIA-STEP et STEP-CATIA avec l'AP 203.

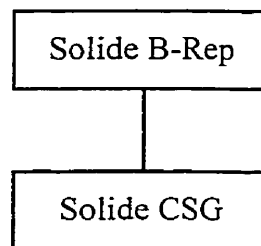
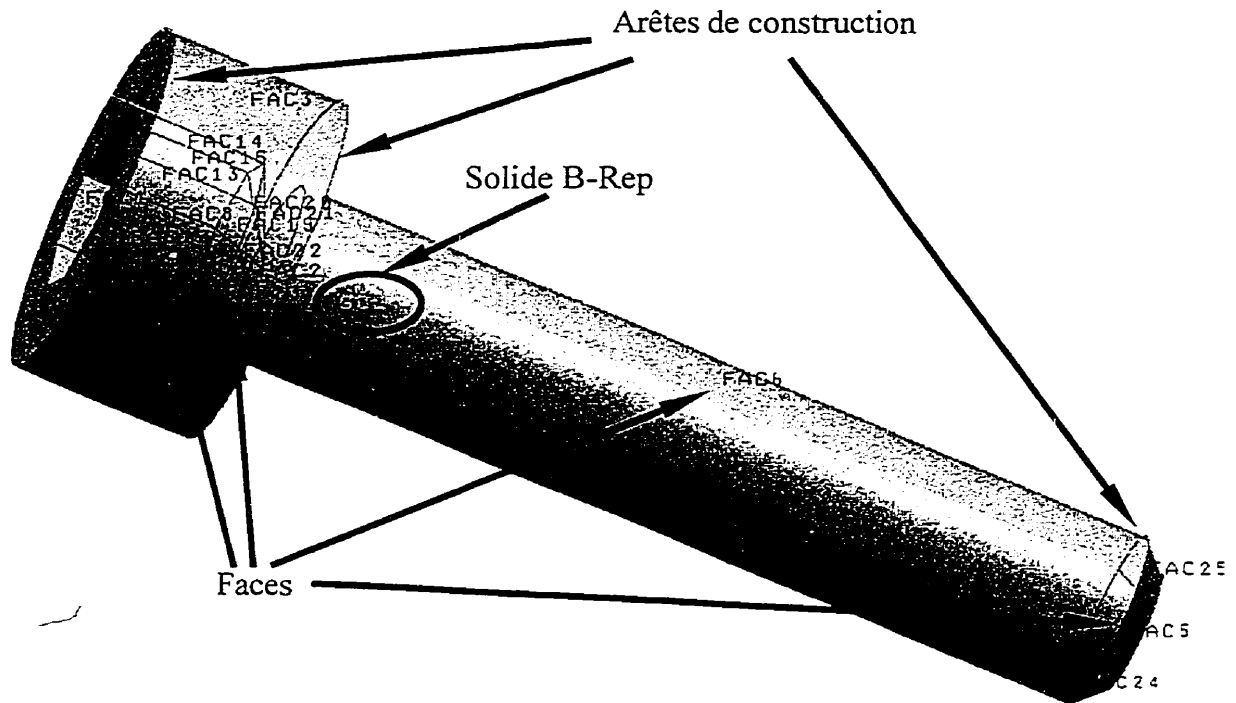


Figure 3.15 Fichier contenant une vis récupérée d'un échange via STEP

L'arbre de construction CSG de la vis sur la figure ci-dessus a été perdu puisque l'AP 203 ne transfère que de la géométrie solide B-Rep. Pour avoir un arbre de construction CSG, il faut transformer le solide B-Rep en solide CSG comme illustré sur la figure 3.15. On remarque que la vis récupérée est composée d'un solide B-Rep et de plusieurs faces. Pour effectuer des modifications sur ce solide, il faut soit tout reprendre à partir des arêtes de construction, ou faire les modifications une face à la fois.

Pour le moment, STEP ne permet pas d'effectuer facilement des modifications car l'AP 203 n'échange pas de solide CSG et que d'autres protocoles d'application avec cette possibilité, tel l'AP 214, sont peu présents sur les systèmes CAO.

L'associativité entre le modèle du produit et la programmation à contrôle numérique est aussi une problématique à évaluer au niveau 3. La fonctionnalité désirée à ce chapitre, est cependant plus complexe puisque le modèle produit est contrôlé par l'ingénieur et la fabrication ne peut modifier, en général, le modèle de définition du produit. Cependant, pour faciliter la génération des parcours d'usinage, certaines entreprises définissent une méthodologie de modélisation du produit qui est particulière pour un type de fabrication comme les pièces usinées, les pièces de métal en feuille et les pièces de fonderie.

Dans le cas d'un fournisseur qui fabrique des pièces pour un donneur d'ordre, il doit donc avoir accès aux informations du produit pour générer les programmes de contrôle numérique. Dans ce cas, les transferts sont en général unidirectionnel puisque le modèle est rarement retourné au donneur d'ordre. Les tests de niveau 3, doivent donc vérifier ce type de fonctionnalité afin de s'assurer que la définition produit reçue est exacte ainsi que de grande précision et que les parcours d'usinage puissent être facilement générés.

Un exemple de problématique simple en deux dimensions dans ce domaine est lorsque certaines entités sont partiellement (A) ou totalement (B) superposées comme montré à la figure ci-dessous.

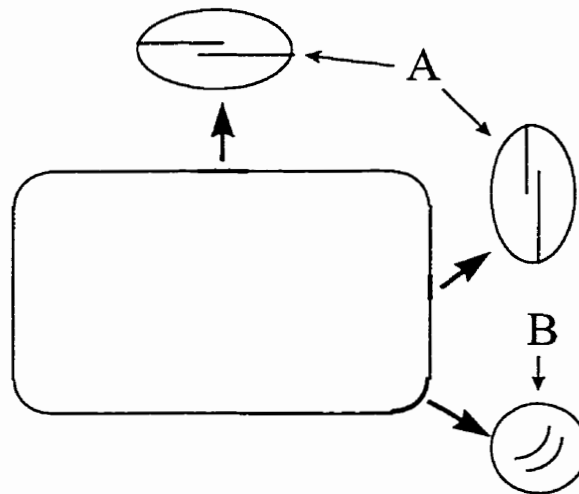


Figure 3.16 Superposition d'entités

Des problèmes plus sérieux peuvent survenir avec des modèles surfaciques et solides plus complexes. Ce type de vérification est donc essentielle pour les entreprises manufacturières.

De plus, l'associativité entre les parcours d'usinage et le modèle du produit permet de modifier de façon automatique ces parcours à partir d'une mise à jour effectuée sur le modèle. Cette fonctionnalité est intéressante pour une PME où l'ingénierie de conception et de fabrication sont tout à fait intégrées.

Cependant, dans une grande entreprise ces changements doivent être contrôlés de manière stricte afin de s'assurer que les bonnes configurations du produit soient maintenues. Une configuration du produit comprend les versions appropriées des pièces fabriquées telles qu'approuvées par l'ingénierie puisque le produit évolue au fil du temps.

L'AP 203 ne supporte pas cette associativité entre la définition et la programmation de contrôle numérique, cependant ce protocole d'application propose une certaine gestion des versions des pièces et des produits. Cette fonctionnalité est aussi du ressort du niveau 3.

Par ailleurs, l'AP 214 propose une structure de donnée d'échange possédant des liens entre le modèle produit et la programmation de contrôle numérique.

Les analyses dont il a été question dans cette partie de la méthodologie sont reliées au troisième niveau de test soit, celui des fonctions. Toutefois, il existe d'autres analyses et éléments qui se situent au même niveau et qui sont présentés au point suivant.

Éléments qui font partie du troisième niveau

Voici quelques éléments qui sont considérés comme faisant partie de troisième niveau de test:

- la construction d'un assemblage (les fichiers utilisés pour réaliser cette construction doivent être complets et pouvoir être manipulés sans restrictions);
- le niveau fonctionnel d'une entreprise comme, par exemple, les procédés de fabrication (transformation du fichier afin de pouvoir être utilisé dans la chaîne de fabrication) ou encore des analyses d'éléments finis;

Le concept de "form, fit and functions" est un autre élément qui est considéré au troisième niveau des tests de vérification. Lorsqu'on parle de ce concept on fait référence à trois aspects au niveau de la conception (définition) du produit sujet à contrôle: sa forme, ses jeux, ses tolérances ainsi que ses ajustements et ses fonctions. Ce concept prend une plus grande valeur lorsque l'on parle de STEP puisque cette norme a pour visée de soutenir un produit tout au long de son cycle de vie. Dans ce cas, on désire contrôler le développement de l'ingénierie ce qui comprend en premier lieu les procédés, c'est-à-dire les gammes de fabrication, les parcours d'usinage, la programmation des commandes numériques, en deuxième lieu on retrouve l'utilisation du produit, c'est-à-dire sa maintenance et finalement en troisième lieu on parle du recyclage d'un produit.

Il importe maintenant de décrire de façon plus exhaustive en quoi consistent les actions correctives.

3.5.4 Actions correctives (A5)

Les actions correctives sont des rectifications apportées à des fichiers présentant des problèmes dans le but de les éliminer. Elles peuvent prendre plusieurs formes. On s'entend pour reconnaître quatre principales classes d'actions correctives: limitation du domaine d'échange (A51); mise en place d'un "mapping" spécifique (A52); réquisition de rectification de logiciel (A53); corrections manuelles des fichiers (A54).

Les deux premières classes d'actions correctives existent pour deux raisons. La première, commune aux deux classes, est pour combler la norme puisque sa mise en œuvre n'est pas complètement instaurée. La deuxième raison diffère. La limitation du domaine d'échange existe aussi car ce n'est pas tous les systèmes CAO qui possèdent les mêmes capacités et par conséquent, certaines restrictions doivent être imposées afin de permettre de bons échanges entre les deux systèmes CAO. La mise en place d'un "mapping" spécifique, quant à elle, existe aussi pour faire le lien entre les protocoles d'applications.

Une réquisition de rectification de logiciel est nécessaire lorsqu'un problème est directement lié au logiciel. Ce type de problèmes peut parfois être corrigé manuellement, ce qui renvoie à la quatrième classe d'actions correctives qui concerne les corrections manuelles de fichiers. Afin de mieux comprendre cette dernière classe, voici quelques exemples.

Exemple 3.7 Exemple d'actions correctives manuelles

- Changement manuelle des ensembles de référence;
- Mise à jour de l'identification d'un produit;
- Changement de symboles dans le fichier STEP;
- Retrait d'entités en trop directement sur le logiciel de modélisation ou dans le fichier STEP.

3.6 VERS UNE PROCÉDURE D'ANALYSE

La conception d'une méthodologie d'évaluation de STEP pour sa mise en place se doit d'être analysée systématiquement afin de démontrer sa valeur heuristique. Exercice essentiel à tout développement d'une méthodologie de ce type.

Le chapitre suivant fera état des modalités d'analyse privilégiées, c'est-à-dire des tests réalisés pour fin de confirmer sa pertinence. Les résultats y seront présentés et discutés.

CHAPITRE 4

DESCRIPTION DES TESTS ET ANALYSE DE LA MÉTHODOLOGIE

L'objectif de ce chapitre consiste en la présentation des tests réalisés afin d'analyser la méthodologie du chapitre précédent. Tout d'abord, un bref rappel sur l'importance d'une méthodologie ainsi que sur la manière de l'analyser sont présentés. Ensuite, une explication de l'environnement dans lequel les tests ont été exécutés est exposée. Pour continuer, les tests sont présentés en deux parties: premièrement, deux études de cas sont décrites en détail et deuxièmement un résumé de l'ensemble des tests est exposé. Les deux études de cas sont présentées en trois phases: un résumé du fichier utilisé pour le test, une illustration de ce fichier et les détails des tests réalisés avec ledit fichier. Enfin, des prospectives sont mises en avant pour fins de raffinement de la présente étude.

4.1 PROCÉDURE D'ANALYSE DE LA MÉTHODOLOGIE

Comme mentionné au chapitre 3, contrairement à une simple exécution de tests pour vérifier la réussite d'échanges de données via une norme, une méthodologie d'évaluation est indépendante des logiciels utilisés pour effectuer ladite évaluation. Il importe de rappeler que la méthodologie présentée au chapitre précédent, expose les grandes lignes à suivre lors de l'exécution de tests visant à déterminer le niveau de réussite d'échanges de données via STEP et ce, peu importe les logiciels de CAO qui sont retenus.

Une méthodologie d'évaluation ne peut s'avérer pertinente sans vérification expérimentale. Autrement dit, il importe de recourir à des tests pour analyser cette méthodologie. Dans le même sens, une méthodologie d'évaluation se valide à l'aide de tests concrets en lien avec les entreprises, c'est-à-dire que ces dernières fixent les objectifs et fournissent les fichiers. Afin de pouvoir réaliser l'analyse de la méthodologie présentée au chapitre précédent, deux points méritent questionnement et clarification.

Le premier aspect est directement lié à la norme ISO 10303, puisqu'il est important de connaître le niveau d'implantation de celle-ci lorsque les tests d'analyse sont effectués.

Ces derniers se voient affectés lorsque la partie de la norme utilisée est implantée qu'en partie. Tel que spécifié au chapitre 3, STEP est une norme récente en constante évolution.

Le deuxième aspect est relié à la pertinence des tests, car il est essentiel de montrer qu'ils supportent de manière efficace la méthodologie. En termes plus précis, les tests doivent démontrer que chacune des parties de la méthodologie est pertinente.

Les deux aspects composant l'analyse de la méthodologie exposés, la présentation des tests suit.

4.2 PRÉSENTATION DES TESTS

Les tests sont exécutés dans le cadre d'une plate-forme d'échanges de données basée sur les principes de la technologie de STEP, mise au point à l'Institut des Technologies de Fabrication Intégrée (ITFI) du Conseil National de Recherches du Canada (CNRC) (Ottawa). Cette plate-forme a été présentée lors d'un colloque international du nom de "Joint ISO/TPO Meeting" qui a eu lieu à Toronto du 4-11 octobre 1996. La figure 4.1 ci-dessous illustre cette plate-forme.

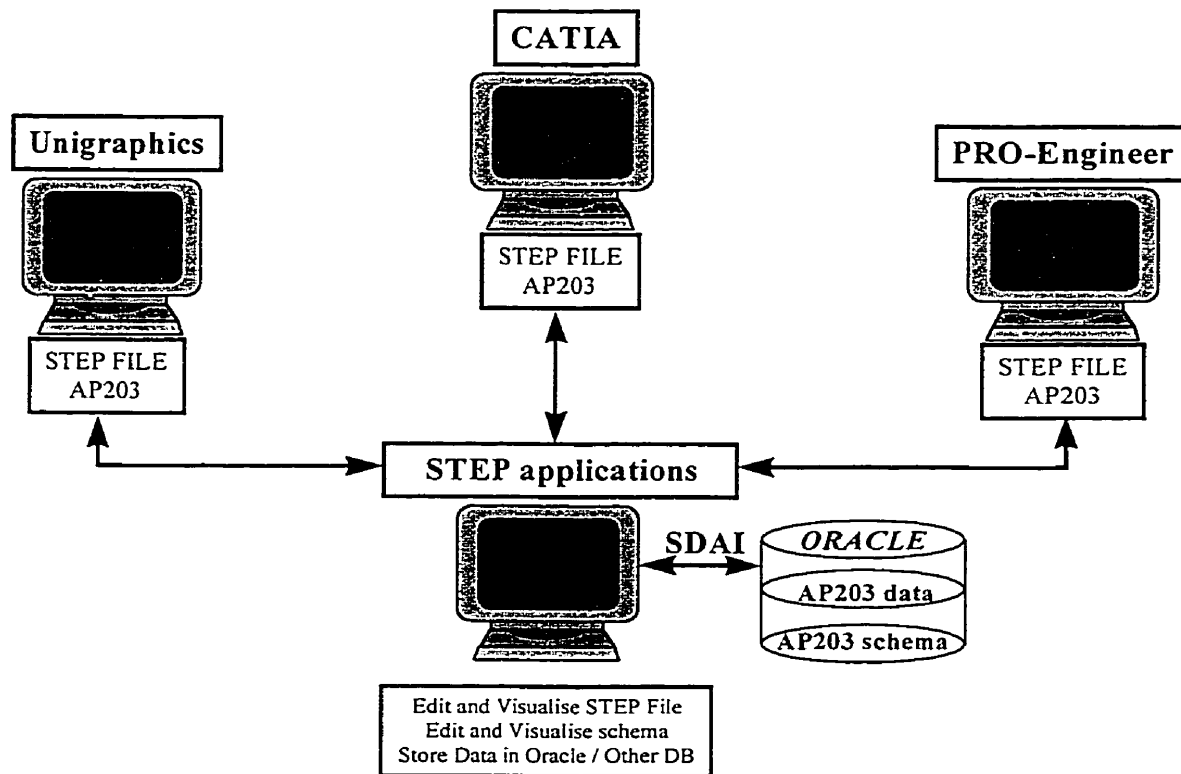


Figure 4.1 Plate-forme de tests du CNRC

La plate-forme ci-dessus a été développée pour l'exécution de tests reliés à STEP, dont ceux réalisés lors de cette étude. Ses objectifs sont de fournir, aux entreprises canadiennes, une base d'expérience reliée à STEP afin d'évaluer l'interopérabilité des produits STEP (échanges entre systèmes CAO, application de base de données), de montrer les possibilités de migration entre normes (IGES à STEP) ainsi que les capacités du "mapping". Elle a pu être mise sur pied grâce à la collaboration de plusieurs entreprises. Ces dernières ainsi que les logiciels utilisés pour la plate-forme qu'elles ont fournies, sont présentés ci-dessous.

ENTREPRISES

LOGICIELS

VERSIONS

EDS Canada

Unigraphics

11.0.0, 11.1.0

Rand technologies	Pro-Engineer	16.0
École Polytechnique	Catia	4.1.4, 4.1.6
ITI	STEP/Works	1.0, 2.0 beta et 2.0
STEP Tools Inc.	ST-Oracle, ST-WB ¹ , et ST-Developer	
CSTB ²	XIG tools	

Les entreprises General Motors Diesel et Delco ainsi que l'École Polytechnique de Montréal ont collaboré en fournissant des fichiers pour les échanges à réaliser. Deux autres sources de fichiers ont été utilisées soit le bureau de conception interne de l'ITFI et le répertoire de démonstration de fichier sur le système CAO Pro-Engineer. Les fichiers provenant de GM et ceux du bureau de conception sont issus d'Unigraphics, les fichiers de Delco de Pro-Engineer et ceux de l'École Polytechnique de CATIA.

Plusieurs personnes ont contribué au développement de cette plate-forme. Parmi celles-ci, on compte messieurs Helium Mak, Charles Mony, Syed Shafee Ahamed, Chang Shu, Steve Lamarche, John Dickinson, Jonathan Kofman ainsi que l'auteure même de la présente étude.

Afin de s'assurer que les tests d'échanges de données entre les systèmes CAO soient effectués selon les critères rigoureux, une démarche précise a été utilisée pour leur exécution. Cette démarche, qui constitue la méthodologie exposée au chapitre 3, assure une analyse systématique des fichiers et permet de détecter plus aisément la présence de problèmes. Cette méthodologie est le résultat d'une démarche qui débuta par la mise en place de la plate-forme, s'est poursuivie par l'exécution d'une première série de tests qui

¹ ST = Step Tools et WB = WorkBench.

² CSTB = Centre Scientifique des Technologies du Bâtiment.

ont amenés l'émergence d'une méthodologie. Au fur et à mesure que la méthodologie s'établit, d'autres tests ont été exécutés en parallèle afin de l'enrichir et de vérifier certains aspects de celle-ci. C'est un processus itératif qui a été suivi entre les tests et la méthodologie pour arriver au résultat présenté au chapitre 3.

Description de la présentation des études de cas

Tous les échanges de données ont été exécutés à l'aide du protocole d'application 203, car il est le plus utilisé à ce jour. Signalons également que, parmi tous les logiciels de modélisation utilisés aux fins de cette recherche, ce protocole est le seul commun à ces derniers. Plusieurs fichiers ont été utilisés pour effectuer les tests mais seulement les plus significatifs sont mentionnés dans ce mémoire.

Chaque étude de cas comprend trois parties: tout d'abord une brève présentation du fichier utilisé pour l'étude incluant sa provenance, son nom et son contenu (nombre de pièces et d'entités) est exposée; ensuite, une visualisation du fichier est présentée et, enfin, les tests réalisés avec ce fichier sont détaillés. Dans cette troisième partie, les tests pertinents, associés à chacune des étapes de la méthodologie du chapitre 3, sont décrits dans un style direct, en suivant rigoureusement la chronologie des étapes.

4.2.1 Première étude de cas

La première étude de cas décrit les tests effectués avec un fichier provenant de l'entreprise General Motors, division Diesel située à London, Ontario. Le fichier contient un assemblage qui porte le nom de 10617741 et comprend un total de 41 pièces et 1841 entités. La figure 4.2 ci-dessous permet de visualiser cet assemblage.

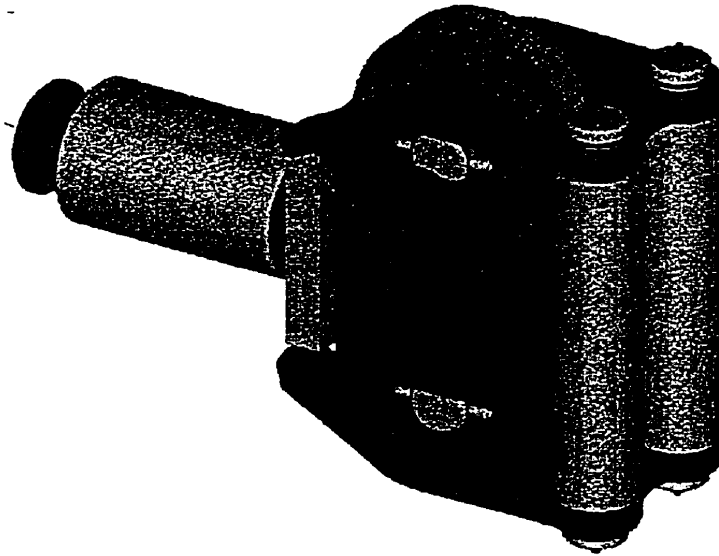


Figure 4.2 Assemblage 10617741 de General Motor Diesel de London (Ontario)

Le contenu du fichier et sa représentation graphique étant connus, la description des tests réalisés avec ce fichier suit en débutant par la définition des objectifs.

Définition des objectifs et du domaine d'échange

- Échanger un fichier contenant un assemblage de complexité moyenne entre 3 systèmes CAO;
- Échanger un fichier composé de géométrie solide;

Les fichiers choisis pour les échanges de données contiennent des cas réels. Par réels on se réfère à des fichiers déjà existants dans le système informatique de l'entreprise.

Les tableaux 4.1 et 4.2 ci-dessous fournissent des informations additionnelles sur les objectifs d'échanges.

Tableau 4.1 Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Utilise le(s) système(s) CAO: en interne..... avec des contracteurs.....	√	√
Modèles sous forme: 3D: filaire..... CSG..... B-Rep..... surfacique..... 2D: avec annotations..... sans annotations.....	√	√ √ √ √ √
Fichiers contenant: 1) une seule pièce..... 2) assemblage (2 pièces et plus)...	√	√
Documentation technique et de maintenance		√
Intégrer les fichiers pour: 1) conception..... 2) fabrication.....		√ √
Taille des fichiers d'échanges: petite..... moyenne..... grande.....	√ √	√
Désire réaliser les applications spécifiques suivantes: STL..... FEA..... Si oui: utilise le module de FEA du système CAO..... utilise un logiciel spécialisé en FEA..... Étude cinématique..... Apport de modifications..... Application de FAO.....	√ √ √	√ √

Tableau 4.1 (suite) Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Nombre de systèmes CAO impliqués dans les échanges:		
un seul.....		√
deux.....		√
trois et plus.....	√	
Utilisation des fichiers:		
Échange unidirectionnel.....	√	
Échange bidirectionnel:		
retournés une seule fois.....	√	
retournés plusieurs fois (processus itératif).....		√

Tableau 4.2 Objectifs d'échanges à réponse manuscrite

OBJECTIFS D'ÉCHANGES	RÉPONSES MANUSCRITES
Budget disponible	N/A ³
Quel est l'équipement informatique (hardware)	Systèmes CAO Unigraphics et Pro-Engineer sur le système d'exploitation SUN, système CAO CATIA sur le système d'exploitation HP

Traduction et réception des fichiers d'étude de cas

- Fichier reçu par bande magnétique;
- Traduction directe⁴ du format du système CAO d'origine (Unigraphics) vers STEP par le protocole d'application 203;
- Échanges de données entre trois systèmes CAO réalisés à l'intérieur de l'institut ITFI du CNRC et avec l'École Polytechnique;
- Impossibilité d'ouvrir le fichier sur le logiciel d'analyse STEP/Works;
- Vérification de la taille du fichier STEP sur une ligne de commande;

³ N/A: ne s'applique pas.

⁴ Traduction effectuée sans interface.

- Traduction directe du fichier en format STEP vers le format des systèmes CAO de réception (Catia et Pro-Engineer).

Plusieurs messages indiquant des erreurs sont apparus à l'écran lors de la tentative d'ouverture du fichier sur STEP/Works. Ces messages signalant des erreurs de syntaxe puisque le caractère \$ est inscrit sur plusieurs lignes, dans le fichier STEP, à un endroit où une chaîne de caractères soit, .AHEAD., doit être inscrite. Des précisions sont présentées à la section du niveau 1 (conformité) dans le présent exemple.

Vérification des fichiers

- Visualisation du fichier sur Pro-Engineer, ce qui amena à constater les erreurs suivantes: pièces manquantes et positionnement incorrect de pièces.

La vérification se fait à trois niveaux soit: niveau 1, la conformité, la vérification syntaxique; niveau 2, la sémantique, la vérification que l'information sémantique reste la même d'un système CAO à un autre; niveau 3, les fonctions, la vérification des actions qu'il est possible de réaliser avec le fichier récupéré.

Niveau 1: Conformité

- Vérification de la date de création du fichier STEP et de la version du logiciel de création dans l'entête du fichier STEP;
- Vérification de la syntaxe de quelques lignes de texte du fichier STEP telles celles dont le numéro est 31300, 31660, 32490, 33630, 34290, 35170. 36640 etc., afin de vérifier si elles sont inscrites de la bonne manière;

Des erreurs de syntaxe ont été détectées à ce niveau et des précisions concernant les correctifs suivent.

Actions correctives employées pour éliminer les erreurs de syntaxe

Les erreurs observées lors de la tentative de visualiser le fichier sur STEP/Works sont associées à des erreurs de syntaxe comme constaté lors de l'analyse de la présente section. Ces erreurs furent examinées plus en détail. En fait, elles révèlent qu'un caractère autorisé par la norme, "\$", est inséré à un endroit non approprié. Les messages indiquant des erreurs consistent en une répétition du même message se lisant comme suit:

Error (SAFP_TOKEN_MISMATCH)

Inprocessing entity #____, field → sense expected a int, Token read was : found

DOLLAR SIGN

Les erreurs sont bien des erreurs de syntaxe. En effet, un signe de dollar était inscrit au lieu de ".AHEAD.". Dans le but de corriger ces erreurs, deux essais ont été effectués. Lors du premier essai, les signes de dollar ont été remplacés par ".ahead." Puisque les messages d'erreurs apparaissaient toujours, une modification des ".ahead." par ".AHEAD." a été réalisée. À la suite à cette tentative, les messages d'erreurs étaient disparues.

Niveau 2: Sémantique

- Vérification de la similitude du fichier récupéré sur le logiciel Pro-Engineer par rapport au fichier d'origine sur UG;
- Vérification de la liste des entités contenues dans le fichier sur Pro-Engineer.

Des erreurs de définition d'assemblage ont été observées à ce niveau et des ajustements ont été apportés.

Actions correctives employées pour l'élimination des erreurs de définition d'assemblage

Il importe d'abord de préciser que les erreurs de définition d'assemblage se définissent comme étant des erreurs dans la description des éléments composant un assemblage. La figure 4.3 ci-dessous illustre l'assemblage 10617741 avec ce type d'erreurs, c'est-à-dire le résultat obtenu sur le logiciel de modélisation Pro-Engineer à la suite du transfert du fichier 10617741 de UG vers Pro-E via STEP.

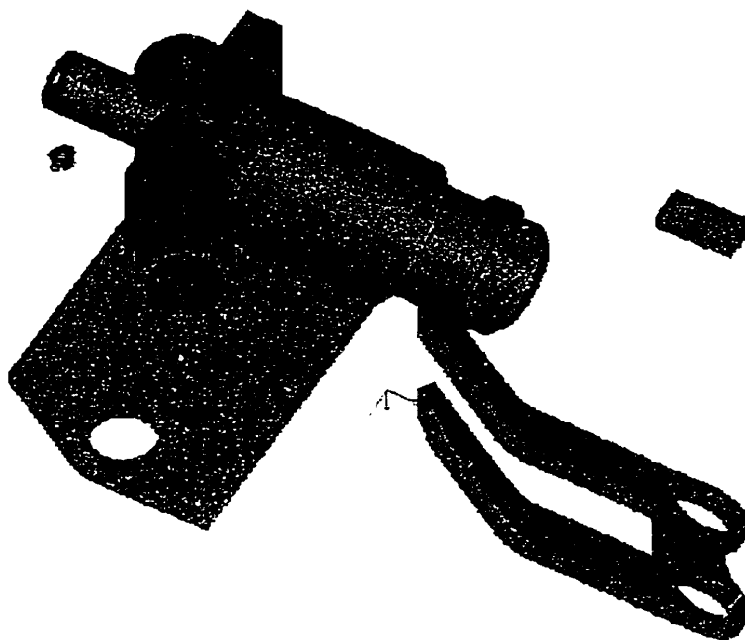


Figure 4.3 Assemblage 10617741 de GM Diesel avec erreurs de définition d'assemblage

Le même type de problème que celui rencontré à la figure 4.3, renvoie à la façon dont un assemblage est modélisé. Afin de pouvoir résoudre un tel type de problème, il s'agit de connaître comment cet assemblage fût construit, c'est-à-dire s'il est construit à partir d'un repère global⁵ ou local, si les pièces sont positionnées les unes par rapport aux autres ou par rapport au repère global, etc. Lorsque ces informations sont connues, le problème peut très souvent être résolu sans trop de difficultés.

Avant d'essayer de trouver la manière dont l'assemblage de GM Diesel a été modélisé, une étude a été exécutée afin de trouver la cause des problèmes rencontrés.

Le premier problème relié à l'assemblage fût observé sur STEP/Works. En fait, le fichier ne s'ouvrait pas, ce qui causait la fermeture instantanée du logiciel. Une précision s'impose car, ce premier problème fût rencontré seulement avec la version 1.0 de STEP/Works et non avec les versions 2.0 beta et 2.0. Après cette observation, une conclusion a pu être tirée à l'effet que le problème relevait de la grosseur du fichier. En effet, les versions 2.0 beta et 2.0 du logiciel STEP/Works sont capables de soutenir des fichiers dont la grosseur est plus importante qu'avec la version 1.0 du même logiciel.

Le deuxième problème concernant l'assemblage fût remarqué lors de la première visualisation sur Pro-Engineer. Ce problème était relié au mauvais emplacement des pièces ce qui indique une perte de référenciel de celles-ci, comme on peut le constater sur la figure 4.3. En effet, les pièces ne sont pas à l'endroit où elles doivent être afin de former l'assemblage.

Une étude plus approfondie de ce fichier devenait nécessaire afin de trouver où résidait le deuxième problème.

⁵ Un repère global est le repère du fichier et un repère local est le repère d'un élément géométrique particulier.

Jusqu'à maintenant, le deuxième problème dont il a été question plus haut était attribué au mauvais positionnement des pièces. Toutefois, une étude exhaustive du fichier a permis d'observer que lors de la traduction de l'assemblage de UG vers STEP, ce dernier ne gardait pas les éléments suivants:

- 1) les ensembles de référence par défaut;
- 2) les pièces copiées plus d'une fois;
- 3) l'identification du produit⁶;
- 4) la position relative des pièces ainsi que leur position dans le repère global du fichier.

Il importe de rappeler que la norme ISO 10303 spécifie que chaque pièce d'un assemblage doit avoir une identification unique.

Une fois les problèmes identifiés, une première tentative pour les corriger a été mise en oeuvre. Cette dernière consistait à ajouter une sphère à la schématisation de l'assemblage afin de permettre à l'ensemble de la géométrie d'être recalculée. Pour mieux comprendre le principe du recalcul, voici des précisions.

Un cylindre doit passer à travers un rectangle afin d'y percer un trou. La figure 4.4 illustre ces deux objets et le résultat désiré est illustré à la figure 4.5. Pour obtenir le résultat de cette dernière figure, il est nécessaire de faire recalculer le fichier de la figure 4.4, c'est-à-dire il faut procéder à une opération de soustraction entre le rectangle et le cylindre (soustraire le cylindre du rectangle).

⁶ Nom attribué d'une pièce (produit) d'un assemblage.

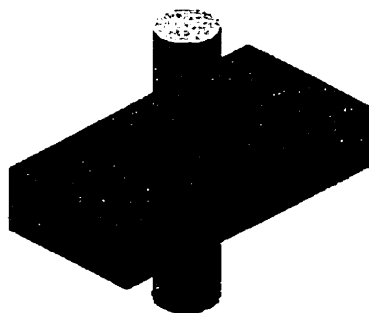


Figure 4.4 Fichier avant recalcul

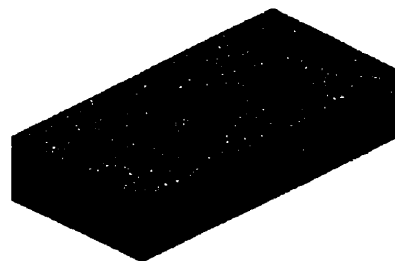


Figure 4.5 Fichier après recalcul

Ensuite, ce fichier fût traduit en format STEP et ouvert sur la version 2.0 beta de STEP/Works. Aucun changement concernant le fichier ne fût remarqué sauf en ce qui a trait à l'ajout de la sphère. Les pièces de l'assemblage étaient toujours mal positionnées et celles qui devaient être présentes plus d'une fois ne l'étaient pas.

Puisqu'aucun changement n'avait été observé, une analyse encore plus approfondie devenait nécessaire. Aussi un autre type de tests fût effectué avec ce fichier. En effet, des congés⁷ supplémentaires ont été ajoutés à l'assemblage afin de permettre à nouveau au fichier de recalculer la géométrie. Par la suite, le fichier fût traduit dans le format STEP et visualisé sur STEP/Works. Les congés additionnels étaient présents mais par ailleurs les pièces étaient toujours mal positionnées.

Après que toutes ces tentatives aient été entreprises afin de corriger les problèmes, il n'y avait toujours pas de changements positifs: les problèmes persistaient. À cette étape, le personnel de EDS Canada, groupe concepteur du logiciel Unigraphics, fût contacté afin de discuter de ces problèmes. Le résultat émanant de cette analyse vont en ce sens: les

⁷ Surface concave raccordant deux surfaces.

problèmes rencontrés étaient reliés non pas à la norme STEP mais au logiciel Unigraphics et au système de gestion de l'entreprise d'où provient le fichier.

Des alternatives ont été trouvées afin d'éviter les inconvénients rencontrés. Le changement manuel des ensembles de référence à nul (null) ainsi que la mise à jour de l'identification du produit dans le fichier UG, dans le fichier STEP ou encore, la création d'une routine remplissant cette information à partir du système de gestion font partie des alternatives plausibles et pertinentes. Pour mettre à jour l'identification du produit il s'agit d'inscrire un numéro ou un nom différent à chaque pièce contenue dans l'assemblage. Il importe de souligner que l'information sur l'identification du produit doit provenir du système de gestion de l'entreprise et non du système CAO: chacune des pièces d'une entreprise est unique et l'information reliée à celle-ci doit être uniforme pour tous les systèmes qui utilisent cette pièce.

Niveau 3: Fonctions

- Génération d'un fichier STL directement à partir du fichier STEP;
- Génération d'une vue explosée de l'assemblage sur Pro-E;
- Apports de modifications (congé et ajout d'une sphère).

La génération d'un fichier STL a permis la réalisation d'une pièce en stéréolithographie. Cette pièce a été réalisée sans aucun problème. Les apports de modifications dans le fichier sont décrites à la page précédente dans la description de l'étude plus approfondie du niveau 2.

4.2.2 Deuxième étude de cas

La deuxième étude de cas décrit les tests effectués avec un fichier provenant de l'École Polytechnique de Montréal et nommé support3-modifier. Ce dernier contient une pièce avec un total de 145 entités. La figure 4.6 ci-dessous illustre la configuration de cette pièce.

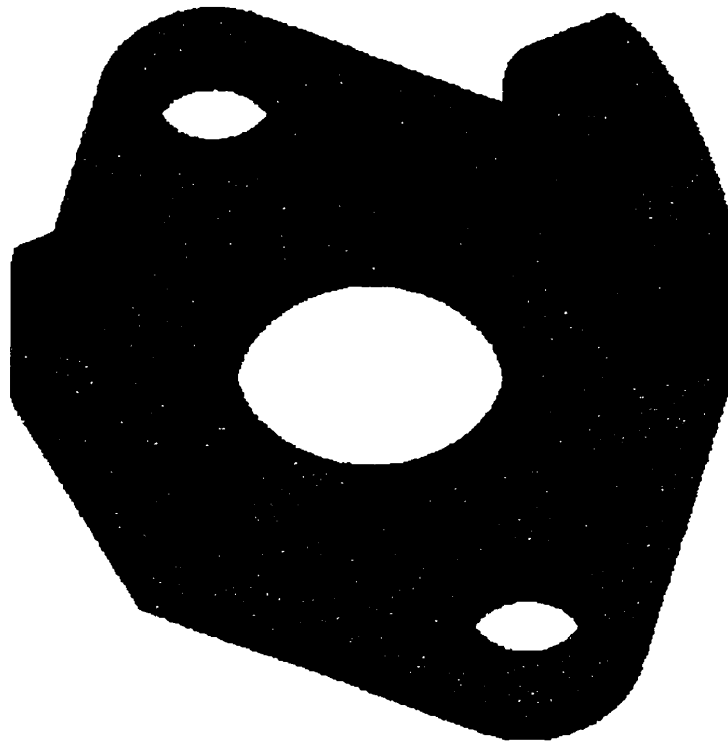


Figure 4.6 Pièce support3-modifier de l'École Polytechnique de Montréal

Le contenu du fichier et sa représentation graphique étant connus, la description des tests réalisés avec ce fichier suit en débutant par les objectifs choisis.

Définition des objectifs et du domaine d'échange

- Échanger un fichier contenant un solide de complexité moyenne entre 3 systèmes CAO;
- Échanger un fichier composé de géométrie solide;
- Effectuer des échanges de données avec des fichiers créés sur un système CAO complexe (CATIA).

Les fichiers choisis pour les échanges de données contiennent des cas particuliers. Comme mentionné à la section 3.3.2 du chapitre 3, un cas particulier est un fichier créé spécialement pour les échanges de données d'une étude de cas.

Les tableaux 4.3 et 4.4 ci-dessous fournissent des informations additionnelles sur les objectifs d'échanges.

Tableau 4.3 Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Utilise le(s) système(s) CAO: en interne..... avec des contracteurs.....	√	√
Modèles sous forme: 3D: filaire..... CSG..... B-Rep..... surfacique..... 2D: avec annotations..... sans annotations.....	√	√ √ √ √ √
Fichiers contenant: 1) une seule pièce..... 2) assemblage (2 pièces et plus)...	√	√
Documentation technique et de maintenance		√

Tableau 4.3 (suite) Objectifs d'échanges à choix de réponse

OBJECTIFS D'ÉCHANGES	RÉPONSES	
	OUI	NON
Intégrer les fichiers pour: 1) conception..... 2) fabrication.....		√ √
Taille des fichiers d'échanges: petite..... moyenne..... grande.....	√ √	√
Désire réaliser les applications spécifiques suivantes: STL..... FEA..... Si oui: utilise le module de FEA du système CAO..... utilise un logiciel spécialisé en FEA..... Étude cinématique..... Apport de modifications..... Application de FAO.....	√	√ √ √ √
Nombre de systèmes CAO impliqués dans les échanges: un seul..... deux..... trois et plus.....	√	√ √
Utilisation des fichiers: Échange unidirectionnel..... Échange bidirectionnel: retournés une seule fois..... retournés plusieurs fois (processus itératif).....		√

Tableau 4.4 Objectifs d'échanges à réponses manuscrites

OBJECTIFS D'ÉCHANGES	RÉPONSES MANUSCRITES
Budget disponible	N/A ⁸
Quel est l'équipement informatique (hardware)	Systemes CAO Unigraphics et Pro-Engineer sur le système d'exploitation SUN, système CAO CATIA sur le système d'exploitation HP

Traduction et réception des fichiers d'étude de cas

- Fichier reçu par ftp;
- Traduction directe du format du système CAO d'origine (Catia) vers STEP par le protocole d'application 203;
- Échanges effectués du système CAO d'origine vers deux autres systèmes CAO (Pro-Engineer et Unigraphics) situé à l'institut ITFI du CNRC;
- Ouverture du fichier sur le logiciel d'analyse STEP/Works;
- Vérification de la taille du fichier STEP et visualisation du format texte de la partie 21 à l'aide d'une fenêtre d'éditeur de texte;
- Traduction directe du fichier en format STEP vers le format des systèmes CAO de réception (Unigraphics et Pro-Engineer).

Vérification des fichiers

- Visualiser le fichier sur Unigraphics ce qui amena à constater qu'une des faces de la pièce n'apparaissait pas;
- Visualiser le fichier sur Pro-Engineer: aucun problème ne fût observé.

La vérification se fait à trois niveaux soit: niveau 1, la conformité, la vérification

⁸ N/A: ne s'applique pas.

syntaxique; niveau 2, la sémantique, la vérification que l'information sémantique reste la même d'un système CAO à un autre; niveau 3, les fonctions, la vérification des actions qu'il est possible de réaliser avec le fichier récupéré.

Niveau 1: Conformité

- Vérification de la date de création du fichier STEP et de la version du logiciel de création dans l'entête du fichier STEP;
- Vérification de la syntaxe de quelques lignes de texte du fichier STEP telles celle dont le numéro est 415, 415 et 453, afin de vérifier la bonne description qu'elles doivent prendre;
- Vérification de la définition géométrique de certaines entités telles 1945, 1950, 1992, 2005, 2088, 2101 etc., à l'aide de la documentation de l'AP 203, pour trouver les entités qui leur sont reliées.

Des entités en surplus⁹ étaient présentes dans le fichier et ont été détectées à ce niveau. La procédure de correction est explicitée dans les lignes qui suivent.

Actions correctives employées pour éliminer les entités en surplus

La figure 4.7 ci-dessous illustre la pièce support3-modifier avec les entités en surplus. Elle est construite en mode filaire avec lignes cachées. Les entités en trop, apparaissant en plus pâle sur cette figure, sont celles de construction de la pièce.

⁹ Entités excédentes et non nécessaires pour des opérations futures (ex: génération d'un fichier STL) du fichier.

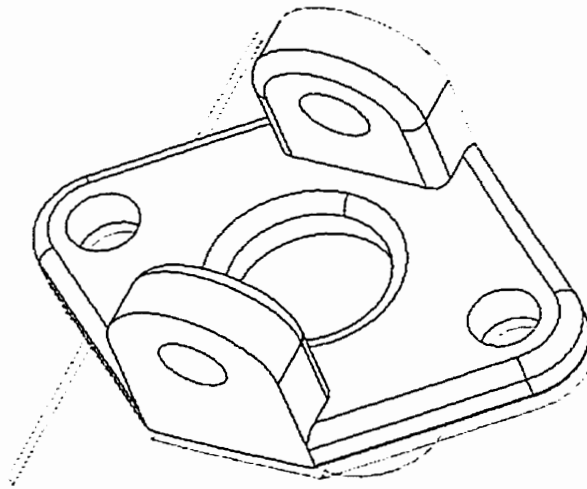


Figure 4.7 Pièce support3-modifier avec entités en trop

Pour pouvoir remédier au problème des entités en surplus, chaque numéro relié à ces entités a été détecté ainsi que toutes les entités leur étant reliées (parents ou enfants)¹⁰. Ces informations ont été obtenues par l'entremise des fonction "parents" et "children" du logiciel STEP/Works. La définition géométrique de ces entités, c'est-à-dire l'information sur les éléments qui composent une entité, a aussi été examinée, à l'aide de la documentation fournie à la partie 203 de la norme ISO 10303, afin de vérifier si toutes les entités reliées aux entités en surplus avaient été détectées. Il importe de noter que les informations obtenues par le logiciel STEP/Works peuvent être aussi obtenues à la lecture du fichier STEP.

Lorsque toutes les entités ont été identifiées elles ont été effacées directement dans le fichier STEP. Cette étape a été réalisée à maintes reprises car, à certains endroits, il y

¹⁰ Une entité enfant est une entité qui découle d'une entité parent (ex: un "point" est le parent d'un "point cartésien" et est aussi l'enfant d'un "item de représentation géométrique", dans l'AP 203 de la norme ISO 10303).

avait plusieurs entités empilées les unes sur les autres. Trois points en surplus, représentés par des X sur la figure 4.7, sont toujours présents lorsque l'on regarde le fichier après correction sur Pro-E. Ces trois points sont essentiels pour la définition du repère global du système qui lui est indispensable à la mise en forme de la géométrie dans un fichier.

Niveau 2: Sémantique

- Vérification de la liste des entités contenues dans le fichier sur Unigraphics (test de base);
- Vérification de la similitude du fichier récupéré à l'aide des logiciels Pro-Engineer et Unigraphics par rapport au fichier d'origine sur Catia.

Après ces vérifications, des erreurs de tolérance géométrique ont été observées au plan de l'organisation géométrique. Le terme "tolérance géométrique" renvoie à l'intervalle maximal admissible entre deux entités géométriques. L'explication suivante est formulée pour fins de compréhension: les extrémités de deux surfaces qui sont côtes à côtes doivent être assez rapprochées, c'est-à-dire à l'intérieur d'un intervalle maximal (tolérance), afin que ces deux extrémités puissent être considérées comme une arête commune aux deux surfaces. Le terme "tolérance géométrique" renvoie aussi à la tolérance géométrique du modelleur.

Des spécifications sur la recherche des causes des erreurs de tolérance géométrique ainsi que les actions correctives employées pour les corriger sont rapportées.

Actions correctives employées pour les erreurs de tolérance géométrique

La figure 4.8 ci-dessous représente la pièce support3-modifier ayant une erreur de "tolérance géométrique". Tel que déjà mentionné, cette "erreur de tolérance" se traduit par le manque d'une des faces de la pièce, soit celle de l'extrémité droite comme le montre la figure 4.8 ci-dessous présentée.

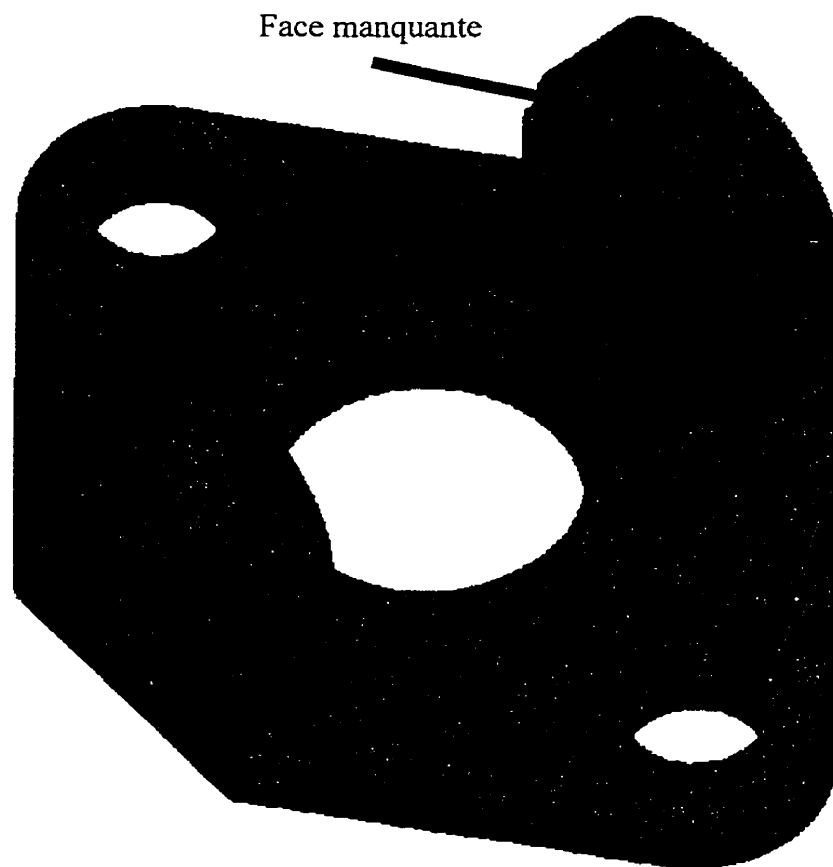


Figure 4.8 Pièce support3-modifier ayant une erreur de tolérance géométrique

Dans UG, on remarque l'absence de cette face (entité #453) lorsque le modèle est en mode ombragé. Toutefois, lorsque le modèle n'est pas dans ce mode, il est très difficile de la détecter. L'analyse du fichier step203ug.log, un fichier informatif donnant toutes les

entités trouvées ainsi que les messages d'avertissements et d'erreurs lors du transfert d'un fichier STEP en format Unigraphics, a permis de détecter un problème de tolérance géométrique du modèleur.

Avant d'aborder la démarche de résolution du problème de tolérance, une tentative de transfert de Pro-E à UG a été faite. La procédure suivie pour effectuer ce transfert est la suivante: le fichier fût acheminé sur Pro-E, un autre fichier STEP fût créé à partir de Pro-E et ensuite traduit de STEP vers UG. Quelques différences ont été remarquées dont, en autres, des faces sous forme d'arrondie n'étaient plus présentes alors qu'elles l'étaient lors du premier transfert direct de CATIA vers STEP vers UG. La face manquante initialement n'apparaît toujours pas.

Afin de tenter de résoudre le problème, le fichier a été traduit plusieurs fois tout en variant les paramètres. En effet, pour chaque essai, une valeur de tolérance différente a été retenue et des modifications de la méthode de correction de validation par défaut du solide "solid validation fault correction method" furent effectuées. Ces valeurs sont spécifiées lors de la traduction d'un fichier du format STEP vers Unigraphics sur ce dernier. La liste ci-dessous fait état de la valeur de tolérance retenue ainsi que des modifications de la méthode de correction de validation par défaut du solide effectuées.

TOLERANCE (mm)	SOLID VALIDATION FAULT CORRECTION METHOD
0.11	None
0.09	Tolerant modeling
1.0	Tolerant modeling
0.05	Tolerant modeling
0.05	None

0.001	Tolerant modeling
0.0001	Tolerant modeling
0.1	Tolerant modeling
0	Tolerant modeling

Après l'exécution de ces tests, aucun changement ne fût remarqué dans le fichier: il y avait toujours une face manquante. Pour cette raison, le personnel de EDS Canada fût contacté afin de discuter du problème. Après analyse, le problème rencontré a été associé à l'utilisation du logiciel Unigraphics et non au recours à la norme STEP. Les gens de EDS Canada ont pris en considération le problème et ont travaillé à sa résolution. Le problème de tolérance a été résolu par le recours à la version 11.1.0 de Unigraphics et autres (éventuellement).

Niveau 3: Fonctions

- Apports de modifications (congs).

Des congés additionnels ont été apportés au fichier support3-modifier. Il a été récupéré sur Pro-E avant d'effectuer le transfert de Pro-E vers UG. La procédure de ce transfert est expliquée à la section 3.4.3 du présent chapitre. Les congés additionnels étaient présents dans le fichier récupéré sur UG.

4.3 ANALYSE DE LA MÉTHODOLOGIE¹¹

Lorsque des tests sont réalisés dans le but d'analyser une méthodologie, il est nécessaire qu'ils soient évalués par rapport à la méthodologie d'intérêt afin de démontrer que la

¹¹ Les analyses réalisées dans cette partie se bases sur les études de cas décrites précédemment dans ce chapitre ainsi qu'à partir d'autres tests réalisés mais qui ne sont pas décrits dans le présent chapitre.

méthodologie a dépassée le stade hypothétique, qu'elle est bien fonctionnelle. Pour ce faire, il faut étudier chaque partie de la méthodologie séparément afin de savoir si chacune d'elle a été investiguée par les tests.

Rappelons que puisque que la méthodologie peut s'appliquer à plusieurs domaines et plusieurs types d'échanges de données, les tests peuvent prendre plusieurs directions. En d'autres termes, les tests à effectuer pour remplir les objectifs fixés et pour la réalisation de la méthodologie, varient: les tests dépendent des objectifs visés.

Les partie du texte qui suivent présentent l'analyse de la méthodologie par les tests effectués. Chaque point renvoie à une des parties de la méthodologie dont il a été question plus avant.

4.3.1 Définition des objectifs et du domaine d'échange

La première étape de la méthodologie est de définir des objectifs pour guider le déroulement de celle-ci et de choisir les fichiers qui seront utilisés pour les tests.

Définition des objectifs

Les objectifs d'échanges constituent un aspect central de la méthodologie car c'est en référence à eux que les tests détermineront si les échanges peuvent être considérés comme réussis ou non réussis. Les tests qui ont été réalisés dans le cadre de cette étude avaient pour but de démontrer qu'il est possible d'effectuer des échanges de données via STEP lors de l'exécution d'échanges de données de cas réels et particuliers. Ces tests avaient aussi pour but de mettre à l'épreuve la fiabilité de la norme ainsi que d'illustrer que STEP surpasse IGES sur plusieurs plan, dont la possibilité d'échanger des assemblages. Les autres objectifs fixés dans le cadre du développement de cette méthodologie portaient sur

des échanges de fichiers contenant des assemblages de complexité moyenne et de la géométrie solide.

Il importe de rappeler que cette partie de la méthodologie n'est pas seulement réservée au choix des objectifs mais aussi au domaine d'échange soit, la sélection des fichiers qui sont utilisés pour les échanges.

Définition du domaine d'échange

Les fichiers sélectionnés pour les tests réalisés dans cette étude contiennent des cas réels et des cas particuliers.

Les cas réels sont principalement composés de pièces et d'assemblages de complexité faible (voir figure 4.6) et moyenne (voir figure 4.2). Les cas particuliers sont, en autres, composés de courbes de Bézier et Bspline, de géométries de base (ex: sphère, cube) et de géométrie plus complexe (ex: pièces d'un micro-moteur, solides).

4.3.2 Traduction et réception des fichiers d'étude de cas

Lorsque les objectifs sont fixés et que les fichiers cibles sont sélectionnés, on doit procéder à la traduction en format STEP de ces fichiers pour ensuite les acheminer à l'endroit approprié.

Traduction des fichiers en format STEP

Il existe deux manières de traduire des fichiers en format STEP, soit directement, soit par l'intermédiaire du format IGES. Même si les deux moyens ont été utilisés, la grande majorité des fichiers ont été transférés directement en format STEP, avec un pourcentage

de réussite de 94 %. Soulignons que des pré-processeurs STEP étaient disponibles et qu'ils étaient nécessaires afin de démontrer la possibilité de réaliser des échanges par le recours à la norme STEP.

Lors de la traduction en format STEP, certains logiciels de CAO nécessitaient la mise en place de quelques paramètres (ex: géométrie solide ou surfacique?, les repères utilisés, etc.) avant de passer à la traduction tandis que pour d'autres, la traduction s'effectuait directement. Ces observations ont été faites en référence au type de situation.

La procédure de traduction du fichier en format STEP, passant par IGES, a été réalisée sur deux fichiers dans le but d'expérimenter cette procédure. Ces essais ont été exécutés avec la fonction de traduction de IGES à STEP du logiciel STEP/Works.

Le premier essai a été réalisé avec un fichier de GM du nom de 106177321. Ce dernier a été traduit en format IGES sur le logiciel Unigraphics (de UG vers IGES). Après le transfert de IGES vers STEP, le problème avait disparu. Il importe de noter que la version STEP de ce fichier, (traduction faite de UG vers STEP sur le logiciel UG), présentait le même type de problème d'ouverture sur le logiciel STEP/Works que le fichier 10617741.

Le deuxième essai a été réalisé avec une pièce du nom de IMS conçue sur le logiciel CATIA. Le but de cet essai était de procéder à une comparaison entre les résultats du fichier traduit de CATIA à UG directement (CATIA-IGES, IGES-UG) et celui traduit indirectement (CATIA-IGES, IGES-STEP sur STEP/Works, STEP-UG). La conclusion tirée de ce deuxième essai est la suivante: les deux types d'échange de ce fichier donnent le même résultat final sur UG, c'est-à-dire que la pièce récupérée sur ce logiciel montre l'absence d'entités.

La procédure passant par IGES ne fût utilisée que pour ces deux fichiers car le besoin n'était pas ressenti ailleurs. Dans le cas des autres fichiers, les pré-processeurs STEP étaient disponibles, le transfert direct vers STEP s'avérait plus simple et plus efficace que le passage de IGES vers STEP. Toutefois, cette procédure est manifestement intéressante pour les entreprises qui travaillent avec IGES et veulent intégrer graduellement l'utilisation de la norme STEP pour les échanges de données.

Une fois les fichiers traités en format STEP, ils sont prêt pour être acheminés.

Réception des fichiers

Le mode de réception des fichiers s'est effectuée par bande magnétique, par "ftp" ainsi que par réseau interne. L'opération s'est déroulée sans problème.

Une attention particulière a été portée à la provenance des fichiers, au nombre de fichiers présents sur les bandes magnétiques lors de leur réception et à l'identification du logiciel de modélisation d'origine des fichiers. Toutes ces actions apportent des informations additionnelles sur les fichiers et peuvent être utiles lors de recherche de solutions reliées à des problèmes de cet ordre. La vérification de l'intégrité de certains fichiers a aussi été faite par l'examen de la taille des fichiers ainsi que de leur contenu en format texte.

Lorsque toutes les analyses pouvant être réalisées dans cette partie ont été terminées, les fichiers ont été traduits dans le format des systèmes CAO lesquels sont utilisés pour le reste de l'exécution de la méthodologie.

Traduction des fichiers en format CAO

L'étape de la traduction des fichiers en format STEP vers les différents systèmes CAO a été réalisée avec un pourcentage de réussite de 92 %. Elle s'avérait indispensable afin de démontrer que les échanges par STEP sont du domaine du plausible.

Comparaison entre IGES et STEP

Une comparaison entre IGES et STEP a été réalisée avec un fichier réel et assez complexe, provenant d'une entreprise dont la confidentialité doit être gardée. La figure 4.9 ci-dessous sert d'illustration.

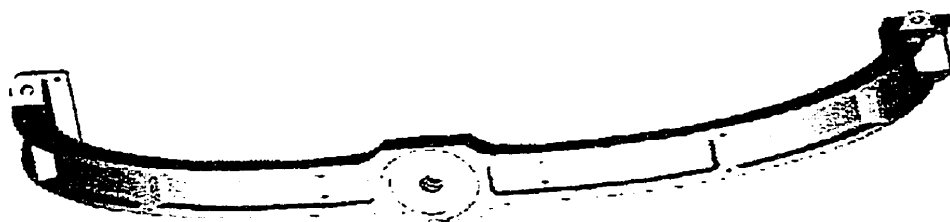


Figure 4.9 Fichier traduit via IGES et STEP

La comparaison c'est faite à l'intérieur de l'institut ITFI du CNRC. Les gens du bureau de design de l'ITFI ont travaillé plusieurs semaines pour transférer ce fichier par IGES. Le même fichier a ensuite été soumis au site de la plate-forme de tests pour traduction en format STEP afin de savoir si ce type de traduction était possible avec ce fichier particulier. Le transfert s'est effectué à l'intérieur d'une trentaine de minutes et s'avéra un succès puisqu'aucun problème ne fut rencontré et que la pièce récupérée est comme celle de départ.

Naturellement, le fichier n'a pas transité de IGES vers STEP. Par ailleurs, à la suite de la comparaison réalisée, on sait qu'il existe deux voies pouvant être exploitées pour la traduction de ce fichier spécifique.

4.3.3 Vérification des fichiers

Cette partie s'amorce lorsque les fichiers sont dans le format des systèmes CAO de réception. Elle comprend trois niveaux de tests différents présentant chacun leur importance.

Niveau 1: Conformité

Ce niveau débute par une vérification de base, c'est-à-dire pour l'ouverture des fichiers sur le(s) système(s) CAO de réception. Cette vérification s'est effectuée pour tous les fichiers tel que prévu. Lorsqu'aucun problème n'est observé lors de cette vérification, il est possible de passer au deuxième niveau mais lorsque des anomalies sont détectées, il faut procéder à une série d'analyses.

Comme c'est à ce niveau que certaines anomalies avaient été initialement détectées, il fallait trouver la cause de celles-ci. Pour ce faire, les deux parties qui composent le fichier STEP, l'entête et le développement, furent analysées. Il importe de souligner que dans la partie développement, seules les entités avec anomalies furent étudiées.

Les anomalies trouvées sont de deux types: erreurs de syntaxe et entités en surplus. Pour apporter les corrections aux premier types d'anomalies, il a fallu vérifier la documentation de l'AP 203 et de la partie 21 de la norme puisque ces anomalies étaient reliées à des erreurs de syntaxe. Pour corriger le deuxième type d'anomalies, il a fallu

regarder leur définition géométrique dans la documentation de l'AP 203 afin de pouvoir les éliminer et considérer toutes les autres entités en lien avec elles.

Les analyses du niveau 1 se terminaient par la vérification du protocole d'application qui ne fût pas réalisé pour les tests de cette étude car tous les échanges ont été faits avec l'AP 203 tel que déjà mentionné à la partie de la description de la présentation des études de cas.

Lorsque toutes les analyses du premier niveau ont été terminées, le deuxième niveau de tests s'est amorcé.

Niveau 2: Sémantique

Le début du deuxième niveau ressemble beaucoup à celui du premier car l'opération débute aussi avec une vérification de base qui consiste à déterminer les objectifs pertinents à ce niveau qui ne fût pas réalisée. Toutefois, les pièces contenues dans les fichiers furent analysées brièvement afin de déterminer si celles-ci étaient composées de faces, surfaces, volumes ou solides. Ces analyses furent réalisées de deux façons: visualisation à l'écran des fichiers lors de la vérification de base du niveau 1 et examen de la liste des entités contenues dans les fichiers.

Les analyses mentionnées ci-dessus ont été suivies par une série d'analyses qui ont un intérêt lorsqu'un problème détecté au premier niveau est d'ordre sémantique et non syntaxique ou encore, lorsqu'un problème est détecté à ce deuxième niveau.

Parmi toutes les analyses qui peuvent être accomplies au deuxième niveau, des analyses de définitions géométriques ont été réalisées. Elles ont permis de trouver que l'un des problèmes relié à certains fichiers étaient dû au logiciel de modélisation et non à la

personne qui a modélisé ces fichiers. En effet, le logiciel de modélisation sur lequel fût créé les fichiers définissait certains paramètres comme les tolérances géométriques, d'une façon différente que d'autres logiciels du même type (systèmes CAO) ne le font. Ces différences de paramètres entraînaient des erreurs au plan de la sémantique. Ce type d'erreurs a été corrigé avec les versions améliorées des systèmes CAO soit, les nouvelles versions.

Des analyses concernant les manières de modéliser un assemblage réalisées à ce niveau, ont permis d'associer des anomalies relatives au mauvais emplacement de pièces d'un assemblage et au manque d'autres pièces (observées au premier niveau), à des problèmes de sémantique. Des étapes de correction de définition d'assemblage visant à retrouver la sémantique initiale, soit celle définie dans le système CAO d'origine, ont été apportées au fichier arborant ce type de problème.

Un troisième type d'analyse, soit celle de la vérification de la similitude des fichiers, a été réalisé à ce niveau. Pour ce faire, le contenu des fichiers sur leur système CAO d'origine et celui présent dans les fichiers récupérés sur les systèmes CAO différents ont été comparés. Soulignons que cette analyse a pu être exécutée car les logiciels de modélisation d'origine des fichiers reçus étaient disponibles.

Il existe d'autre type d'analyses qui peuvent être faites mais certaines d'entre elles appartiennent à un niveau plus élevé. Dans cette étude, ce niveau est présenté comme le troisième niveau, c'est-à-dire celui des fonctions. Des spécifications sur le troisième niveau sont présentées à la section 3.5.3 du chapitre 3.

Niveau 3: Fonctions

Comme il a déjà été mentionné précédemment, les tests à exécuter étaient étroitement liés aux objectifs fixés au début de la méthodologie. Cette précision concerne tous les niveaux de tests mais, tout particulièrement, le troisième niveau. Tous les tests de ce niveau sont déterminés directement à partir des objectifs établis.

Les objectifs des tests étant assez limités dans cette partie, seulement quelques analyses furent réalisées à ce niveau. Les analyses réalisées constituaient des apports de modifications sur les pièces des fichiers, des vérifications de précision des échanges ainsi que la génération d'une vue explosée de l'assemblage et d'un fichier STL à partir du fichier représenté à la figure 4.2.

Concernant le fichier de la figure 4.2, une application intéressante du troisième niveau fût réalisée. En effet, à partir de ce fichier, une pièce fût construite par stéréolithographie à partir d'un fichier STL. La génération de ce fichier s'est faite directement à partir du fichier STEP et aucune modification n'a dû être effectuée. Il importe de souligner que lorsque cette procédure est employée avec un fichier IGES, des modifications mineures doivent d'abord être apportées afin que l'on soit en mesure de générer un fichier STL. En effet, puisque dans un fichier IGES il y a souvent des trous dans un solide et que l'on ne sait plus où est la matière du solide, il faut boucher les trous ainsi qu'indiquer où est la matière du solide, c'est-à-dire trouver la normale des surfaces, avant de générer le fichier STL.

4.3.4 Actions correctives

Les actions correctives sont effectuées tout au long de la vérification des fichiers puisqu'elles peuvent être appliquées à n'importe quel niveau lorsque le besoin est ciblé.

Certaines actions correctives ont pu être effectuées au cours des tests, ce sont des corrections manuelles pour des erreurs trouvées dans les fichiers, ainsi qu'une réquisition de rectification de logiciel. Par contre, d'autres types d'actions correctives comme la limitation du domaine d'échange et la mise en place d'un "mapping" spécifique, n'ont pu être réalisées car aucun problème ne nécessitait ce genre de corrections. Toutefois, à l'intérieur d'une entreprise, des corrections manuelles d'enlèvement d'entités en surplus peuvent être effectuées par le recours à un programme de "mapping" afin d'accélérer le processus de corrections et de rendre celles-ci plus efficaces. Ce programme a pour but de corriger certains types d'erreurs bien définies dans un fichier à chaque fois qu'elles apparaissent.

4.3.5 Discussion

À la lecture de l'information présentée dans cette section, on peut affirmer que les tests réalisés permettent de bien analyser la méthodologie proposée au chapitre précédent. Cette affirmation peut être avancée puisque cette analyse a démontrée que chacune des parties a été examinée par le recours aux tests et chacune d'elles s'avère utile. Ces diverses parties sont donc pertinentes et nécessaires pour la réalisation des échanges de données efficaces.

La définition des objectifs est l'étape la plus importante car elle guide l'application de la méthodologie. La sélection des cas qui sont à l'étude a aussi son importance car les tests seront réalisés avec les fichiers choisis.

La traduction des fichiers d'un système CAO à un autre doit être bien exécutée afin de permettre le bon déroulement du reste de la méthodologie. Une mauvaise exécution à cet endroit pourrait fausser les résultats subséquents.

La troisième étape de la méthodologie, soit celle de la vérification des fichiers, constitue le moment crucial où il est possible de savoir si les échanges sont réussis ou non réussis. C'est à cette étape que les fichiers sont examinés après le transfert d'un système CAO à un autre, que les anomalies sont détectées et résolues, lorsque c'est possible. C'est aussi à cette étape que l'on effectue des tests pour examiner les possibilités de pouvoir effectuer des opérations avec les fichiers récupérés. Chaque niveau de tests joue un rôle bien particulier.

Le premier niveau permet de vérifier que la syntaxe d'un fichier est bien définie par rapport à la norme ISO 10303. C'est aussi à ce niveau que les premiers résultats des échanges est observés. Quant au deuxième niveau, il permet de vérifier si la sémantique reliée à un fichier est complète afin que la géométrie contenue dans ce fichier soit interprétée de la même manière sur différents systèmes CAO. C'est à ce niveau que l'on peut remarquer les différences qui existent entre les deux systèmes CAO. Enfin, le troisième niveau évalue les opérations pouvant être exécutées avec un fichier récupéré lors d'un échange de données. Ce niveau est critique pour une entreprise puisque toutes les applications (ex: modifications, analyses de FAO etc.) de l'entreprise se situent à ce niveau.

La dernière étape de la méthodologie, soit celle de la conclusion du processus d'évaluation, est très importante car elle permet de conserver toutes les informations recueillies lors de l'exécution de la méthodologie.

4.4 ÉTUDES FUTURES

Il importe de rappeler brièvement que la norme ISO 10303 (STEP) est très récente et en constante évolution. À ce jour, il existe que peu de tests qui ont été réalisés par rapport à

cette norme et ceux effectués lors de cette étude en font partie. Plusieurs études devront être réalisées dans le futur afin de vérifier l'avancement et la fiabilité de la norme. À ce sujet, quelques suggestions peuvent être mises en avant.

La première des suggestions concerne les tests déjà effectués avec STEP. Seulement quelques études ont été repérées mais il en existe peut-être plusieurs autres telle l'organisation PDES Inc. Cette dernière est un consortium d'entreprises qui effectue des tests reliés à STEP. Plusieurs documents reliés aux projets de cette organisation sont maintenant disponibles sur l'Internet (<http://www.scra.org/pdesinc/news.html>) mais ne l'étaient pas au moment de la recherche, du présent mémoire, concernant d'autres projets reliés à STEP. D'autres tests par rapport à STEP sont en cours de réalisation en France par l'organisation GOSET¹². Des recherches devraient être menées afin de vérifier s'il existe d'autres études reliées à la fonctionnalité de STEP que celles mentionnées et sur les types de tests qui sont en cours de réalisation en France.

La deuxième suggestion est reliée aux protocoles d'application. Puisque que l'AP 203 est le protocole le plus connu et le plus utilisé, les tests d'échanges de données ne sont effectués qu'avec ce dernier. Il serait très avantageux et intéressant de réaliser des tests d'échanges de données avec d'autres protocoles d'application que l'AP 203 tel que l'AP 202, qui est relié à l'aspect "drafting" et l'AP 214, lequel est relié au secteur automobile et qui est plus complet que l'AP 203. De plus, des tests entre protocoles devraient être réalisés comme par exemple, un échange de données entre deux systèmes CAO, l'un avec l'AP 203 et l'autre avec l'AP 214 (système CAO-STEP AP 203 à STEP AP 214-système CAO).

¹² Organisation initialement créée pour le standard SET et formée par des entreprises privées dans le domaine de l'aérospatiale. Aujourd'hui, cette organisation se concentre sur tout ce qui touche les échanges de données.

La troisième suggestion concerne la norme IGES. La plupart des entreprises qui effectuent des échanges de données entre systèmes CAO distincts le font via IGES car c'est la norme la plus connue et elle existe depuis une vingtaine d'années. Il est plausible de penser que plus STEP sera utilisée et publicisée, plus les entreprises se verront forcées de migrer de IGES vers STEP. Peu de tests ont été réalisés de IGES vers STEP. Il serait donc très pertinent de réaliser des essais de migration de IGES vers STEP. Pour ce faire, on peut penser au logiciel STEP/Works qui possède une option visant à effectuer une telle migration.

La dernière suggestion est directement reliée aux entreprises du marché. Comme la méthodologie couvre plusieurs domaines et qu'à chaque fois qu'elle sera appliquée, elle le sera par rapport à une série d'objectifs spécifiques, reliés aux tests à réaliser, des essais se rapprochant davantage des objectifs d'une entreprise devraient être définis. Plusieurs séries d'essais peuvent être effectuées en variant les objectifs qu'une entreprise fixerait si elle devait faire de tels échanges. Les objectifs devraient donc être reliés de très près à des situations réelles.

CHAPITRE 5

CONCLUSION

La norme STEP est plus complète et plus performante que les autres normes, mais elle demeure récente et en constante évolution, ce qui explique la raison pour laquelle peu de plates-formes de tests ont été conçues pour vérifier le niveau d'implantation de la norme. Quelques plates-formes ont été réalisées mais le désavantage de ces dernières est qu'elles sont dépendantes non seulement des logiciels utilisés pour réaliser la plate-forme de tests mais aussi des objectifs de ces tests. Les conclusions tirées de ces plates-formes ne peuvent être appliquées à d'autres tests. Ce facteur a amené l'idée de mettre au point une méthodologie d'évaluation de STEP pour sa mise en place. Cette méthodologie présente plusieurs avantages intéressants, elle est indépendante des logiciels utilisés pour réaliser les tests et elle couvre un nombre important de domaines. De plus, elle permet de réaliser une série de tests avec des objectifs précis et ce, de façon structurée et méthodique. Elle permet aussi aux entreprises, qui désirent utiliser STEP de réaliser une série de tests, avant d'effectuer des échanges de données entre partenaires. Mais encore de détecter les problèmes potentiels, qui pourraient survenir lors des échanges, et de les solutionner, dans la mesure du possible.

La méthodologie d'analyse de STEP a été développée concurremment à des tests d'échanges de données entre logiciels de modélisation, c'est-à-dire que des tests ont été réalisés avant la mise sur pied de la méthodologie, pendant et après celle-ci. Ces tests ont permis d'analyser la méthodologie afin de démontrer la pertinence de chacune de ses parties. Naturellement, cette méthodologie demeure perfectible puisqu'elle couvre un champ d'application très large et que pour cette raison, peu de tests ont pu être exécutés. Les objectifs d'échanges sont établis par rapport aux types de fichiers à échanger, aux projets auxquels sont reliés les échanges ainsi qu'aux entreprises impliquées dans ces échanges. La méthodologie demeure donc perfectible aussi parce que l'exécution de la méthodologie est directement liée avec les objectifs d'échanges. En effet, des modifications peuvent être apportées à la méthodologie après qu'une entreprise l'ait appliquée.

Cette méthodologie est novatrice. Elle est indépendante des systèmes informatiques et elle permet la migration de la norme IGES vers STEP. C'est important puisque IGES est la norme d'échanges de données par interface normalisée la plus connue et la plus utilisée à ce jour. Elle donne aussi un support méthodologique aux entreprises désirant effectuer des échanges de données par l'entremise de la norme ISO 10303 et permet à celles-ci une économie de temps au niveau de la recherche des causes d'erreurs qui surviennent lors des échanges ainsi que de leurs solutions. Finalement, elle s'applique sur l'ensemble du processus global de communication, puisqu'elle inclue les niveaux de la sémantique, du contexte et des fonctions. Elle permet même d'aller jusqu'au niveau des maquettes virtuelles.

Les tests effectués ont fait face à certaines limites et contraintes, tels un ensemble de tests limité ainsi qu'un seul protocole d'application disponible. Les tests exécutés n'ont couverts qu'une infime partie des échanges de données, qu'il est possible de réaliser à travers plusieurs domaines, car les objectifs de ces tests étaient très généraux. À cette première limite s'ajoute la contrainte de la disponibilité d'un seul protocole d'application. En effet, seulement l'AP 203 était disponible sur tous les logiciels de modélisation et souvent le seul disponible. Donc, aucun test n'a pu être réalisé avec un autre protocole.

Étant donné que peu de plate-formes de tests ont été réalisés, la recherche d'autres plate-formes serait enrichissante. Une seconde étude tourne autour du fait que seulement un protocole d'application et trois systèmes CAO ont été utilisés pour l'exécution des tests, il serait intéressant et avantageux de faire des tests avec d'autres protocoles d'application tel l'AP 214 et d'autres systèmes CAO tels AUTOCAD ou EUCLID. Les objectifs des tests étaient très généraux et énoncés dans l'optique de démontrer que STEP est fonctionnel. Ces objectifs n'étaient pas assez proches des objectifs d'une entreprise, en conséquence il serait donc avantageux de tester la méthodologie proposée dans un milieu industriel, à

partir d'objectifs particuliers. Une dernière étude intéressante à réaliser serait d'augmenter le nombre de tests reliés à la migration de IGES vers STEP car, pour les années à venir, ce processus sera utilisé par plusieurs entreprises.

Enfin, on sait que les échanges de données produit font partie d'une réalité quotidienne des entreprises et qu'ils deviennent de plus en plus complexes. Une méthodologie d'évaluation est donc un atout indispensable pour la réalisation d'échanges de données produit fiable et efficace à tous les niveaux de communication.

RÉFÉRENCES

ARBOUY, S., BEZOS, A., CUTTING-DECELLE, A.-F., DIAKONOFF, P., GERMAIN-LACOUR, P., LETOUZEY, J.-P. et VIEL, C. (1994). STEP Concepts fondamentaux, AFNOR, Paris, 188 pages.

BRODSKY, J. (1995). Using manufacturing knowledge in the design of PCA products and processes: STEP is the standard for data exchange. Proceedings of the Technical Program - NEPCON WEST '95, 2, 1163-1172.

CLARK, A. L. et STALEY, S. M. (1995). STEP AP203 data exchange study. Proceedings of the 3rd Symposium on Solid Modeling and Applications, 213-224.

CRUME, D. T. (1993). STEPs to better product data exchange. Aerospace America, 31, 30-32.

HERMANN et LAVOISIER (1991). Dictionnaire d'informatique, éditeurs des sciences et des arts et Technique et Documentation, Paris, 673 pages.

HUANG, J., WANG, T., WAN, L., ZHANG, X. et ZHOU, J. (1994). STEP-based CAD/CAPP/CAM integrated systems: CADMIS. Engineering Data Management: Integrating the Engineering Enterprise ASME Database Symposium 1994, 155-161.

ISO/DIS 10303-1 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 1: Overview and fundamental principles, ISO, Genève, 14 pages.

ISO/DIS 10303-11 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 11: Description methods: EXPRESS language reference manual, ISO, Genève, 208 pages.

ISO/DIS 10303-21 TC184/SC4 (1994). Industrial automation systems and integration - Product data representation exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure, ISO, Genève, 55 pages.

ISO/DIS 10303-31 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: General concepts, ISO, Genève, 31 pages.

ISO/CD 10303-33 TC184/SC4 (1995). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: Structure and use of abstract test suites, ISO, Genève, 7 pages.

ISO/IS 10303-203 TC184/SC4 (1994). Industrial automation systems and integration - Product data representation exchange - Part 203: Application protocols: Configuration controlled design, ISO, Genève, 513 pages.

ISO/TR 10303-304 TC184/SC4 (1996). Industrial automation systems and integration - Product data representation exchange - Part 304: Abstract test suites: Mechanical design using boundary representation, ISO, Genève, 297 pages.

KRISHNASWAMI, R. (1995). AutoSTEP gets in high gear. Actionline, 36-38.

LAURANCE, N. (1994). A high level view of STEP. Manufacturing Review, vol. 7 no. 1, 39-46.

LEGGE, D.I. (1994). Integration of design, inspection and quality management systems. 27th ISATA - Proceedings for the Dedicated Conferences on Mechatronics and Supercomputing Applications in the Transportation Industries, 421-427.

MCMAHON, C. et BROWNE, J. (1994). CADCAM From principles to practice. Addison-Wesley, 508 pages.

NELL, J. G. (1995). STEP on a page. Sur la page web de STEP on a Page: US Product Data Association. [Http://www.scra.org/uspro/stds/stepage.html](http://www.scra.org/uspro/stds/stepage.html).

OWEN, J. (1993). STEP An introduction, Information Geometers Ltd, Winchester, 143 pages.

RICHTER, D. (1991). Concepts and problems concerning the implementation of STEP. Second International Conference on Computer-Aided Design and Computer Graphics, 334-341.

ROBERT, P., REY, A. et REY-DEBOVE, J. (1983). Le petit robert: dictionnaire alphabétique et analogique de la langue française, nouvelle édition revue, corrigée et mise à jour pour 1983, Le Robert, Paris, 2171 pages.

THU-HUA, L. et FISCHER, G.W. (1995). An assembly code classification and coding scheme based on a STEP mechanical product model. Manufacturing Review, vol. 8 no. 1, 33-46.

TRAPP, G. (1993). The emerging Step standard for product-model data exchange. Computer, 26, 85-87.

VERGEEST, J.S.M. (1991). CAD surface data exchange using STEP. Computer Aided Design, vol. 23 no.4, 269-281.

VERHOEF, M., LIEBICH, T. et AMOR, R. (1995). A multi-paradigm mapping method survey. Modeling of Buildings through their Life-Cycle, Fisher, Law and Luiten, CIB/W78-TG10 publication 180, Stanford University, 233-247.

WANG, C.-Y. et RADACK G.M. (1995). Function modeling and process description of forging for STEP AP development. Computers in engineering 1995: and proceedings of the 1995 Database Symposium, presented at the 15th Annual International Computers in Engineering Conference, the 9th Annual ASME Engineering Database Symposium, 1085-1095.

XIAOHU, Y., JINXIANG, D. et ZHIJUNM, H. (1993). The role and application of STEP in CAD/CAPP/CAM integration. Proceedings of TENCOM '93 - IEEE Region 10 International Conference on Computers, Communications and Automation, 2, 746-749.

ZHAO, R., ZHAO, Z., SUN, J. et XIAO, K. (1996). Research on the product data representation and exchange using STEP. Proceedings of the SPIE - The International Society for Optical, 2644, 402-407.

(1998). Petit Larousse illustré, Larousse, Paris, 1720 pages.

BIBLIOGRAPHIE

ARBOUY, S., BEZOS, A., CUTTING-DECELLE, A.-F., DIAKONOFF, P., GERMAIN-LACOUR, P., LETOUZEY, J.-P. et VIEL, C. (1994). STEP Concepts fondamentaux, AFNOR, Paris, 188 pages.

BERNHARD, P.J. (1994). A reduced test suite for protocol conformance testing. ACM Transactions on Software Engineering and Methodology - TOSEM, vol. 3 no. 3, 201-220.

BLOOR, M.S. et OWEN, J. (1995). Learning lessons from conformance testing. Computer Standards and Interfaces, vol. 17 no. 3, 231-252.

BLOOR, M. S. et OWEN, J. (1991). CAD/CAM product-data exchange: the next step. Comuter Aided Desing, vol. 23 no. 4, 237-243.

BRODSKY, J. (1995). Using manufacturing knowledge in the design of PCA products and processes: STEP is the standard for data exchange. Proceedings of the Technical Program - NEPCON WEST '95, 2, 1163-1172.

BROOKS, S.L. et GREENWAY, R.B, Jr. (1995). Using STEP to integrate design features with manufacturing features. Computers in engineering 1995: and proceedings of the 1995 Database Symposium, presented at the 15th Annual International Computers in Engineering Conference, the 9th Annual ASME Engineering Database Symposium, 579-586.

CANADA, BUREAU DE LA TRADUCTION (1995). Termium sur CD-ROM, version 2.16. Ottawa, Ontario: Travaux publics et services gouvernementaux canada, bureau de la traduction. Disques au laser, 4 ¾ po.

CHASSÉ, D et WHITNEY, G. (1997). Guide de rédaction des références bibliographiques, Éditions de l'École Polytechnique de Montréal, Montréal, 178 pages.

CLARK, A. L. et STALEY, S. M. (1995). STEP AP203 data exchange study. Proceedings of the 3rd Symposium on Solid Modeling and Applications, 213-224.

COENEN, M. J. (1994). EMC workbench: testing methodology, module level testing and standardization. Philips Journal of Research, vol. 48 no. 1, 83-116.

CORBI, C. et SISTO, G. (1992). Common abstract test suite for the connection oriented presentation protocol embedded under different application contexts. Interfaces in Industrial Systems for Production and Engineering - Transactions C (Communication Systems), C-3, 143-149.

CRUME, D. T. (1993). STEPs to better product data exchange. Aerospace America, 31, 30-32.

CURRAN, L. (1994). STEP bridges the way to better product modeling. Machine Design International, vol. 66 no. 5, 137-140.

DANNER, W.F. et YANG, Y. (1992). STEP development methods: specification of semantics for information sharing. National Institute of Standards and Technology, 20 pages.

DAVIES, M. (1991). The status and future of product data exchange. Proceedings of the International Conference on Computer Integrated Manufacturing - Manufacturing Enterprises of the 21st Century, 259-262.

DESROCHERS, A. et CLÉMENT, A. (1994). A dimensioning and tolerancing assistance model for CAD/CAM systems. International Journal of Advanced Manufacturing Technology, vol. 9 no. 6, 352-361.

DIBUZ, S. (1993). A frame-based approach to conformance testing. Microprocessing and Microprogramming, vol. 39 no. 2, 191-194.

EVANS, F. (1994). Why are we so out of STEP. Computing and Control Engineering Journal, vol. 5 no. 3, 155-158.

FORTIER, S.C. et WILMOT, A. (1992). Test and diagnostics product data representation. Conference Record AUTOTESTCON '92 - The IEEE Systems Readiness Technology, 41-47.

GOH, A., HUI, S.C., SONG, B. et WANG, F.Y. (1994). A study of SDAI implementation on object-oriented databases. Computer Standards & Interfaces, vol. 16 no. 1, 33-43.

GRABOWSKI, R., ANDERL, R., LI, X, et SCHMITT, M. (1993). Exchange of freeform surfaces using standard interfaces. Proceedings on Interfaces in Industrial Systems for Production and Engineering, B-10, 201-219.

HAENISCH, J. (1990). CAD-exchange-towards a first step implementation. IECON '90 - 16th Annual Conference of IEEE Industrial Electronics Society, 1, 734-739.

HERMANN et LAVOISIER (1991). Dictionnaire d'informatique, éditeurs des sciences et des arts et Technique et Documentation, Paris, 673 pages.

HUANG, J., WANG, T., WAN, L., ZHANG, X. et ZHOU, J. (1994). STEP-based CAD/CAPP/CAM integrated systems: CADMIS. Engineering Data Management: Integrating the Engineering Enterprise ASME Database Symposium 1994, 155-161.

ISO/DIS 10303-1 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 1: Overview and fundamental principles, ISO, Genève, 14 pages.

ISO/DIS 10303-11 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 11: Description methods: EXPRESS language reference manual, ISO, Genève, 208 pages.

ISO/DIS 10303-21 TC184/SC4 (1994). Industrial automation systems and integration - Product data representation exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure, ISO, Genève, 55 pages.

ISO/DIS 10303-31 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: General concepts, ISO, Genève, 31 pages.

ISO/DIS 10303-32 TC184/SC4 (1996). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: Requirements on testing laboratories and clients, ISO, Genève, 27 pages.

ISO/CD 10303-33 TC184/SC4 (1995). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: Structure and use of abstract test suites, ISO, Genève, 7 pages.

ISO/CD 10303-34 TC184/SC4 (1995). Industrial automation systems and integration - Product data representation exchange - Part 31: Conformance Testing methodology and framework: Abstract test methods, ISO, Genève, 15 pages.

ISO/DIS 10303-41 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 41: Integrated generic resources: Fundamentals of product description and support, ISO, Genève, 143 pages.

ISO/DIS 10303-42 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 42: Integrated generic resources: Geometric and topological representation, ISO, Genève, 207 pages.

ISO/DIS 10303-43 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 43: Integrated generic resources: Representation structures, ISO, Genève, 31 pages.

ISO/DIS 10303-44 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 44: Integrated generic resources: Product structure configuration, ISO, Genève, 53 pages.

ISO/CD 10303-46 TC184/SC4 (1992). Industrial automation systems and integration - Product data representation exchange - Part 46: Integrated generic resources: Visual presentation, ISO, Genève, 179 pages.

ISO/DIS 10303-101 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 101: Integrated applicaiton resources: Draughting, ISO, Genève, 40 pages.

ISO/DIS 10303-201 TC184/SC4 (1993). Industrial automation systems and integration - Product data representation exchange - Part 201: Application protocols: Explicit draughting, ISO, Genève, 476 pages.

ISO/IS 10303-203 TC184/SC4 (1994). Industrial automation systems and integration - Product data representation exchange - Part 203: Application protocols: Configuration controlled design, ISO, Genève, 513 pages.

ISO/TR 10303-304 TC184/SC4 (1996). Industrial automation systems and integration - Product data representation exchange - Part 304: Abstract test suites: Mechanical design using boundary representation, ISO, Genève, 297 pages.

HOFFMANN, C. M. (1989). Geometric and solid MODELING: An Introduction, Morgan Kaufmann Publishers Inc., 338 pages.

KEUFFEL, W. (1996). Battle of the modeling techniques. DBMS, vol. 9 no.9, 83-97.

KIRK, J.W., THU-HUA, L. et FISCHER, G.W. (1992). PDES/STEP-based information model for CAE and CAM integration. International Journal of Systems Automation: Research & Applications, vol. 2 no. 4, 375-394.

KLUEH, D. W. et CASHMAN, J. E. (1991). Opening mechanical desing automation via industry standards. AUTOFACT '91 - Conference Proceedings, 22-19 - 22-30.

KRISHNASWAMI, R. (1995). AutoSTEP gets in high gear. Actionline, 36-38.

KROSZYNSKI, U.I., SORENSEN, T. et TROSTMANN, E. (1993). Issues in modelling techniques for the STEP based exchange of robotics data. Interfaces in Industrial Systems for Production and Engineering - Transactions B (Applications in Technology), B-10, 101-113.

LAZAR, R. A. (1994). Conformance testing for the spatial data transfer standard. Cartography and Geographic Information Systems, vol. 21 no. 3, 159-161.

LAURANCE, N. (1994). A high level view of STEP. Manufacturing Review, vol. 7 no. 1, 39-46.

LEGG, D.I. (1994). Integration of design, inspection and quality management systems. 27th ISATA - Proceedings for the Dedicated Conferences on Mechatronics and Supercomputing Applications in the Transportation Industries, 421-427.

LIZHUANG, M. et ZHIJUN, H (1996). Methods and techniques for advanced curve and surface modeling. Proceedings of the SPIE - The International Society for Optical, 2644, 366-374.

LIZHUANG, M., ZHIJUN, H. et JINXIANG, D. (1993). Advanced curve and surface design in CAD system using STEP. Proceedings of TENCON '93- IEEE Region 10 International Conference on Computers, Communications and Automation, 1, 581-584.

LLOYD, M. (1991). Conformance testing for FTAM, CTS-WAN FTAM and CTS-2 FTAM. Conformance Testing and Certification in Information Technology and Telecommunications, 159-163.

MACILWAIN, C. (1992). Grand pulls firms into step with data exchange standard. The Engineer, 275, 32.

MCKAY, A. et BLOOR, M. S. (1991). The role of product models in effective CAD/CAM. Proceedings of the Institution of Mechanical Engineers - Effective CAD/CAM '91, 113-119.

MCMAHON, C. et BROWNE, J. (1994). CAD/CAM From principles to practice, Addison-Wesley, 508 pages.

MOLLENAUER, J. F. (1991). Which communications standards for integrated manufacturing. Information strategy: The Executive's Journal, vol. 7 no. 2, 24-32.

NAKAMURA, I., KOJIMA, T., KUGAI, Y. et KIMURA, F. (1993). A CAD database interface based on STEP. Interfaces in Industrial Systems for Production and Engineering - Transactions B (Applications in Technology), B-10, 115-129.

NELL, J. G. (1995). STEP on a page. Sur la page web de STEP on a Page: US Product Data Association. [Http://www.scra.org/uspro/stds/stepage.html](http://www.scra.org/uspro/stds/stepage.html).

NOTAISE, J., BARDA, J. et DUSANTER, O. (1995). Dictionnaire du multimédia, AFNOR, Paris, 886 pages.

OWEN, J. (1993). STEP An introduction, Information Geometers Ltd, Winchester, 143 pages.

PONCE, A.M. (1994). Documenting the process of conformance testing. Microprocessing and Microprogramming, vol. 40 no. 10, 715-718.

RICHTER, D. (1991). Concepts and problems concerning the implementation of STEP. Second International Conference on Computer-Aided Design and Computer Graphics, 334-341.

ROBERT, P., REY, A. et REY-DEBOVE, J. (1983). Le petit robert: dictionnaire alphabétique et analogique de la langue française, nouvelle édition revue, corrigée et mise à jour pour 1983, Le Robert, Paris, 2171 pages.

SHANPING, L. et JUSTER, N.P. (1993). An exchange strategy for product information. Proceedings on Interfaces in Industrial Systems for Production and Engineering, B-10, 175-186.

SHEVELKOV, V. (1994). The method of conformance testing for presentation data transfer syntax. Automatic Control and Computer Sciences, vol. 28 no. 1, 28-36.

THU-HUA, L. et FISCHER, G.W. (1995). An assembly code classification and coding scheme based on a STEP mechanical product model. Manufacturing Review, vol. 8 no. 1, 33-46.

TORO, M. (1993). Active tester based conformance testing strategy. Computer Standards and Interfaces, vol. 15 no. 4, 337-342.

TRAPP, G. (1993). The emerging Step standard for product-model data exchange. Computer, 26, 85-87.

TRAPP, G. (1991). Concurrent engineering PDES/STEP. DICE/CERC, 2-11.

VERGEEST, J.S.M. (1991). CAD surface data exchange using STEP. Computer Aided Design, vol. 23 no.4, 269-281.

VERHOEF, M., LIEBICH, T. et AMOR, R. (1995). A multi-paradigm mapping method survey. Modeling of Buildings through their Life-Cycle, Fisher, Law and Luiten, CIB/W78-TG10 publication 180, Stanford University, 233-247.

WANG, C.-Y. et RADACK G.M. (1995). Function modeling and process description of forging for STEP AP development. Computers in engineering 1995: and proceedings of the 1995 Database Symposium, presented at the 15th Annual International Computers in Engineering Conference, the 9th Annual ASME Engineering Database Symposium, 1085-1095.

XIAOHU, Y., JINXIANG, D. et ZHIJUNM, H. (1993). The role and application of STEP in CAD/CAPP/CAM integration. Proceedings of TENCOM '93 - IEEE Region 10 International Conference on Computers, Communications and Automation, 2, 746-749.

XINLI, Z., MING, G. et XINXIONG, Z, (1994). A form-feature technique based on CSG boolean operations. Proceedings of 1994 International Conference on Data and Knowledge Systems for Manufacturing and Engineering, 1, 25-30.

ZHANG, S., HOU, X. et WANG, Z. (1995). STEP application protocol interoperability for CAD/CAPP integration. Computers in engineering 1995: and proceedings of the 1995 Database Symposium, presented at the 15th Annual International Computers in Engineering Conference, the 9th Annual ASME Engineering Database Symposium, 687-690.

ZHANG, Y. et DAI, Y. (1992). Product data exchange specification and its implementation. Interfaces in Industrial Systems for Production and Engineering - Transactions B (Applications in Technology), B-1, 309-314.

ZHAO, R., ZHAO, Z., SUN, J. et XIAO, K. (1996). Research on the product data representation and exchange using STEP. Proceedings of the SPIE - The International Society for Optical, 2644, 402-407.

ZYWIETZ, C. et WILLEMS, J.L. (1993). European conformance testing services for computerized electrocardiography: New procedures and standards. Journal of Electrocardiology, 26, 137-146.

(1998). Petit Larousse illustré, Larousse, Paris, 1720 pages.

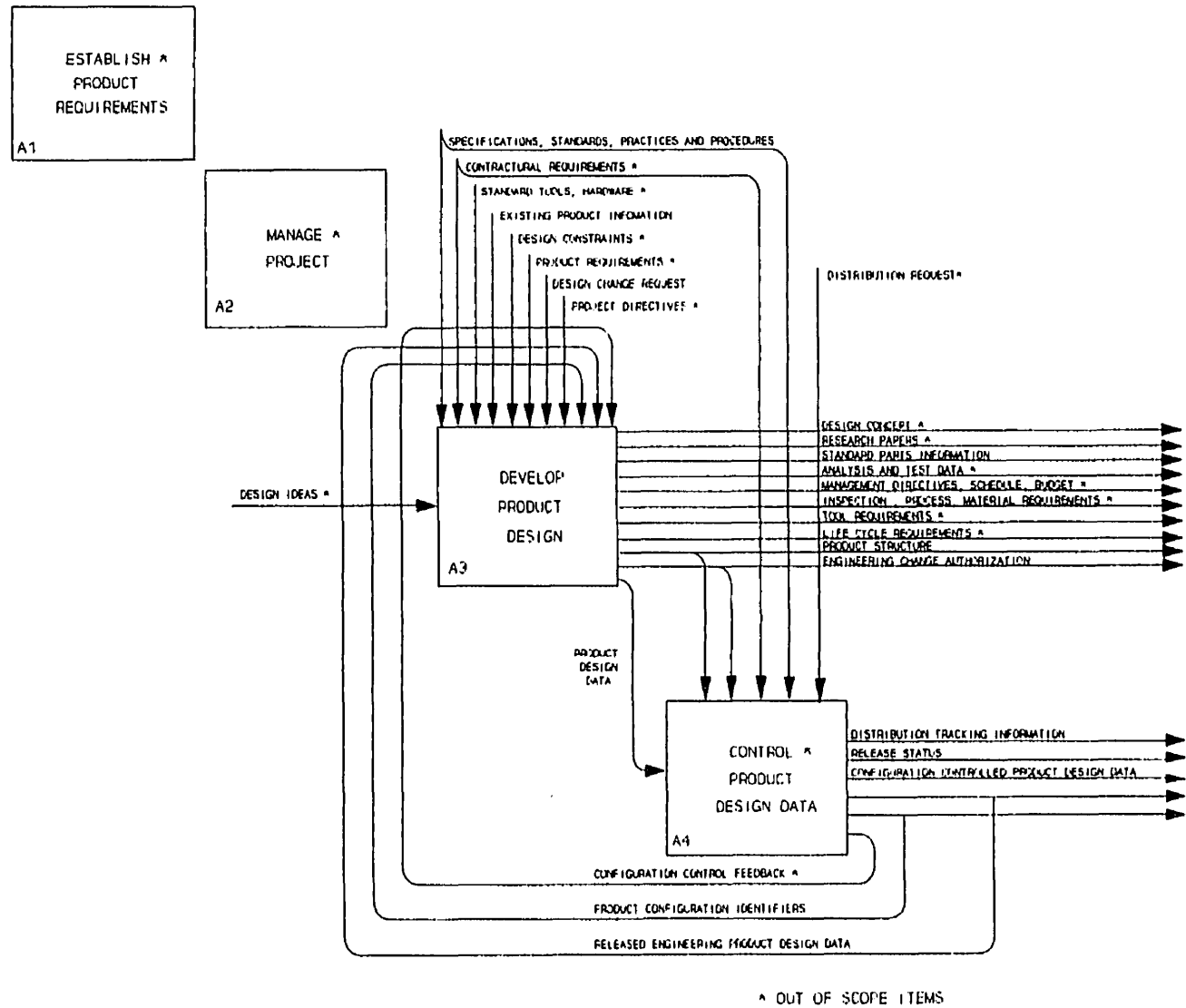
(1994). A STEP forward in product data exchange. Tooling & Production, 60, 19-20.

(1993). A giant STEP for data exchange. Manufacturing Engineering, vol. 111 no. 2, 16-

ANNEXE A1

DIAGRAMMES AAM, ARM ET AIM

Figure A1.1¹ Exemple d'un AAM de l'AP 203



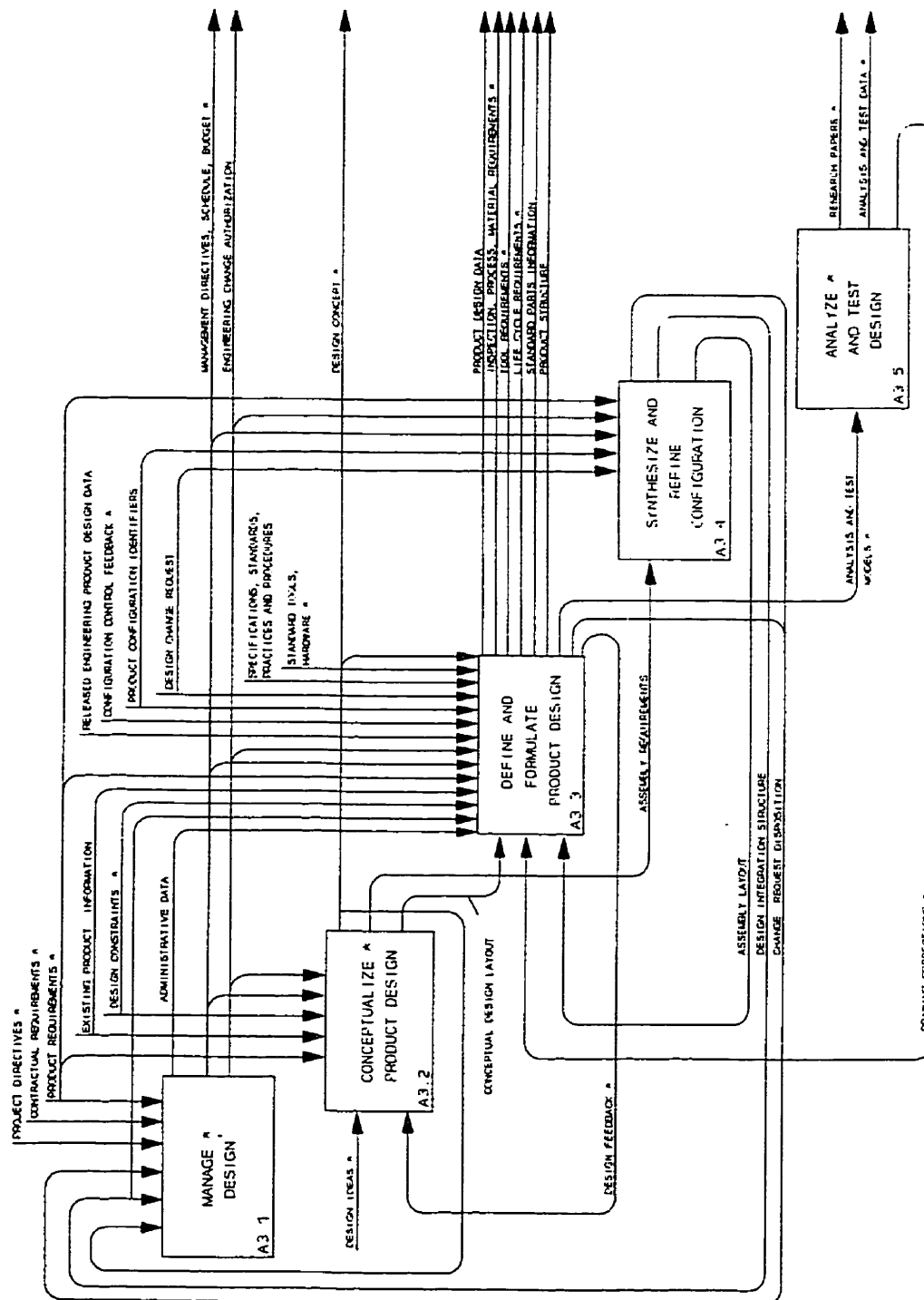
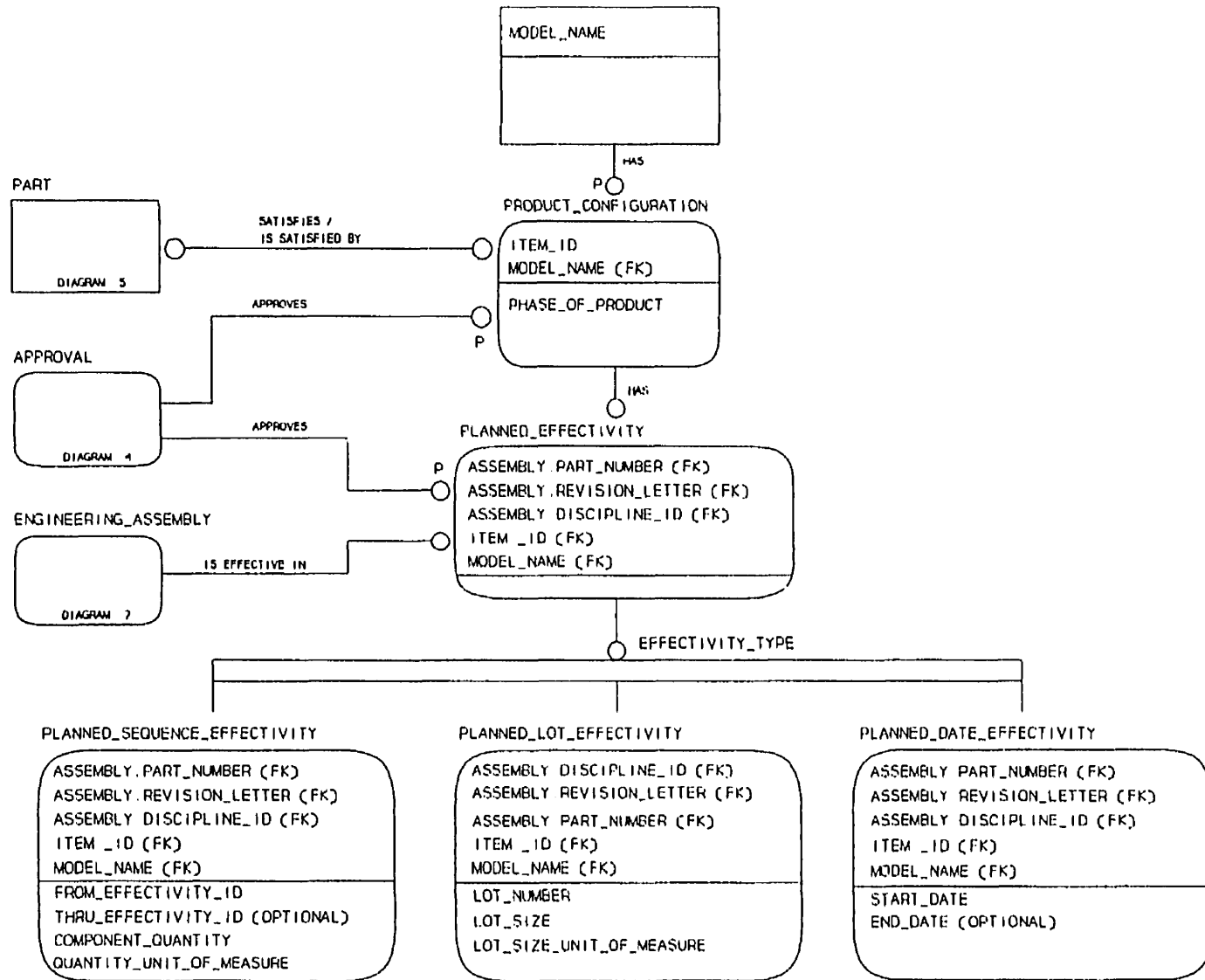


Figure A1.2¹ Exemple d'un AAM détaillé de l'AP 203

Figure A1.31 Exemple d'un ARM de l'AP 203



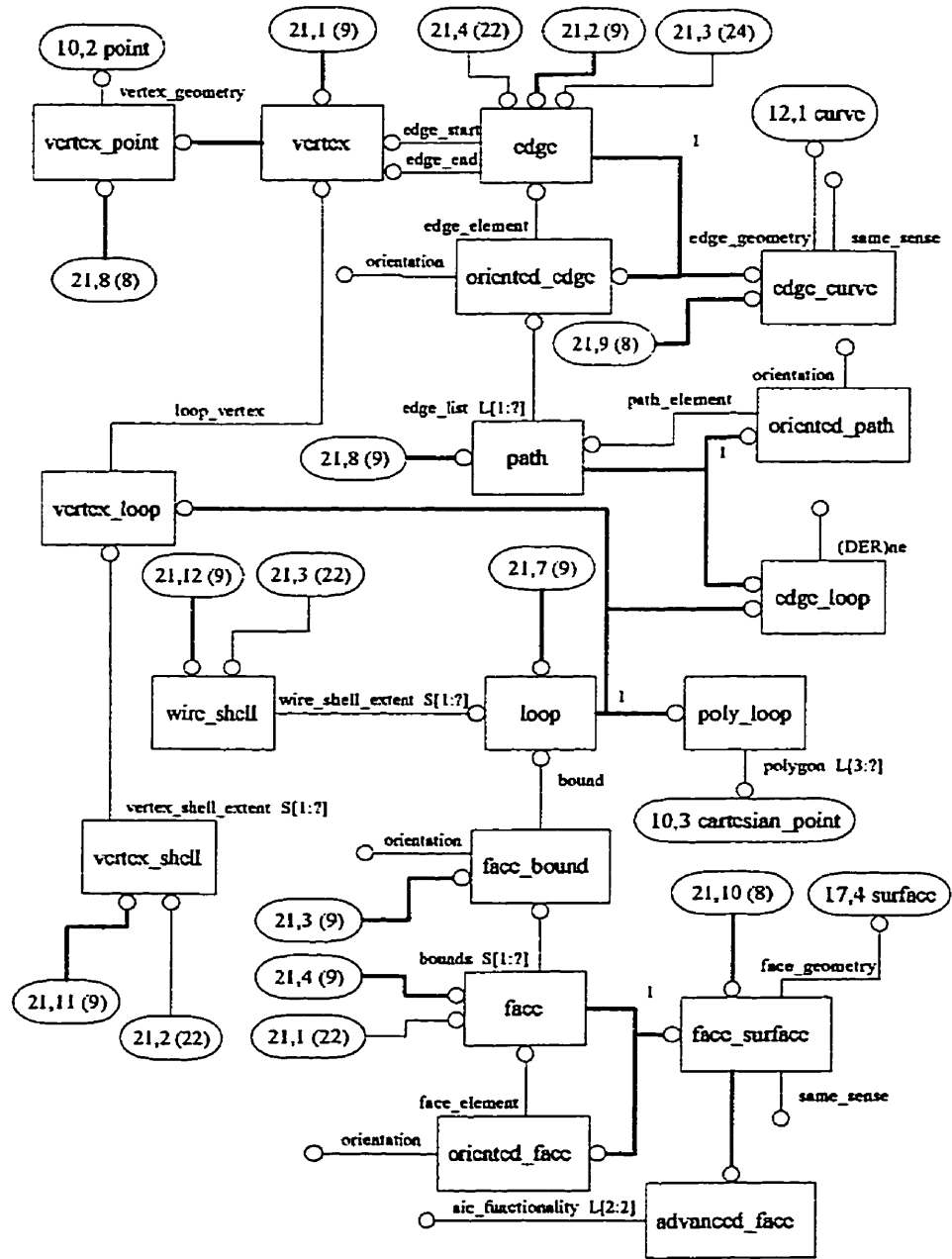


Figure A1.4¹ Premier exemple d'un AIM de l'AP 203

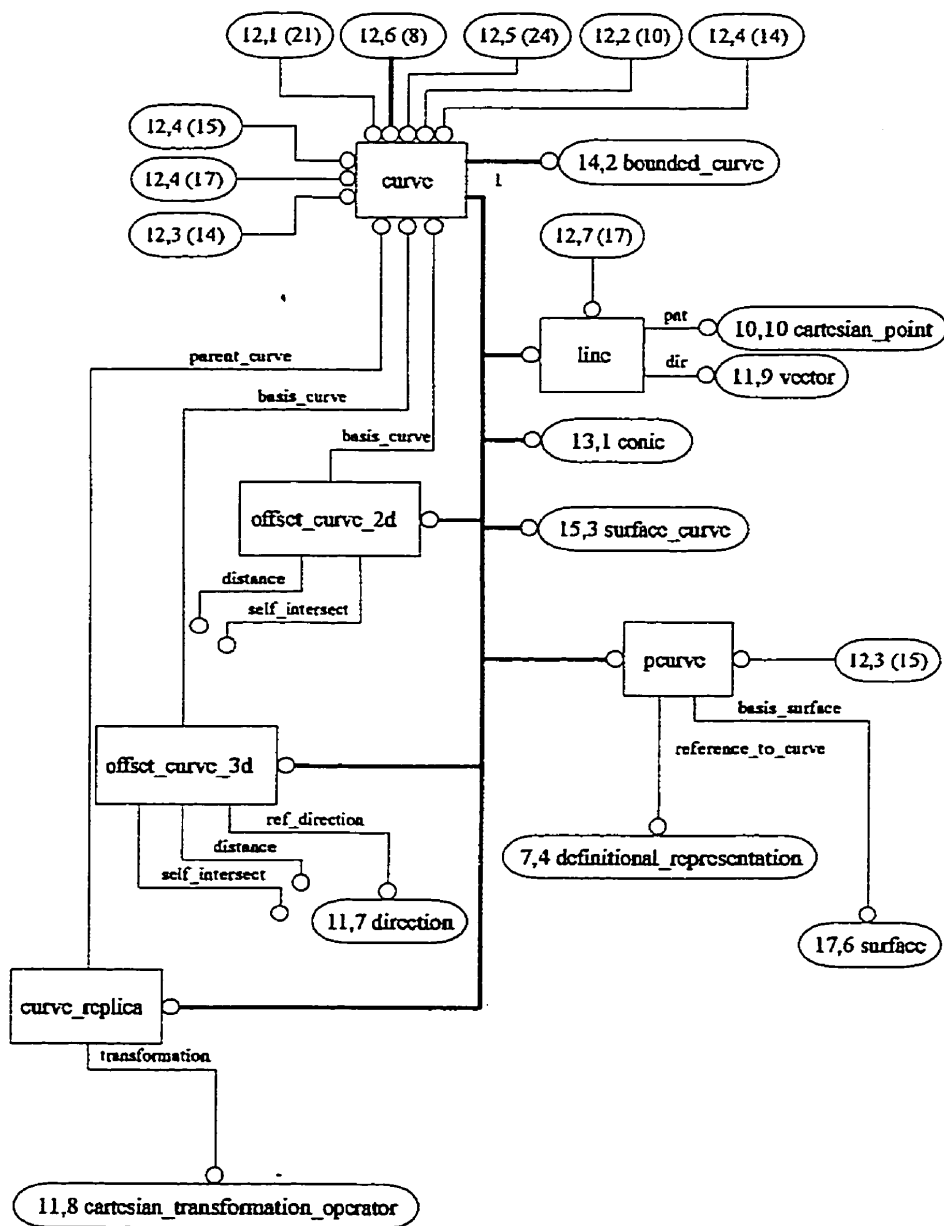


Figure A1.5¹ Deuxième exemple d'un AIM de l'AP 203

¹ Toutes les figures de l'annexe A1 sont tirées de ISO/IS 10303-203 TC184/SC4 (1994). Industrial automation systems and integration - Product data representation exchange - Part 203: Application protocols: Configuration controlled design, ISO, Genève, 513 pages.

ANNEXE A2

FICHER STEP

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('', '1');
FILE_NAME('VIS M2 FIXATION BOUCHON',
'1997-07-17T16:00:12-04:00',
('),
('),
'CATIA.STEP INTERFACE',
'CATIA SOLUTIONS V4  RELEASE 1.7 FR 4.1.7',
");
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;

DATA;
#1=DIMENSIONAL_EXPONENTS(0.00000000000,0.00000000000,0.00000000000,0.0
0000000000,0.00000000000,0.00000000000,0.00000000000);
(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#4=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0174
532925199),#3);
(CONVERSION_BASED_UNIT('DEGREE',#4)NAMED_UNIT(#1)PLANE_ANGLE_
UNIT());
(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASS
IGNED_CONTEXT((#8))GLOBAL_UNIT_ASSIGNED_CONTEXT((#2,#5,#6))REPR
ESENTATION_CONTEXT('*MASTER','WORKSPACE'));
#8=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000
00),#2,'TOL_CRV','CONFUSED CURVE UNCERTAINTY');

```

```
#9=CARTESIAN_POINT('NONE',(20.6600000000,0.0000000000,20.6300000000));
#10=DIRECTION('NONE',(1.0000000000,0.0000000000,0.0000000000));
#11=DIRECTION('NONE',(0.0000000000,0.0000000000,1.0000000000));
#12=AXIS2_PLACEMENT_3D('NONE',#9,#11,#10);
(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#14))GLOBAL_UNIT_ASSIGNED_CONTEXT((#2,#5,#6))REPRESENTATION_CONTEXT('VIS M2','WORKSPACE'));
#14=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000),#2,'TOL_CRV','CONFUSED CURVE UNCERTAINTY');
#15=MANIFOLD_SOLID_BREP('*SOL7',#736);
#16=CARTESIAN_POINT('NONE',(-3.18928766869E-12,-2.00000000000,-2.00000000000));
#17=VERTEX_POINT('NONE',#16);
#18=CARTESIAN_POINT('NONE',(-3.19106402553E-12,-2.00000000000,2.00000000000));
#19=VERTEX_POINT('NONE',#18);
#20=CARTESIAN_POINT('NONE',(0.0000000000,-2.0000000000,0.0000000000));
#21=DIRECTION('NONE',(0.0000000000,0.0000000000,-1.0000000000));
#22=DIRECTION('NONE',(0.0000000000,1.0000000000,0.0000000000));
#23=AXIS2_PLACEMENT_3D('NONE',#20,#22,#21);
#24=CIRCLE('*CRV11',#23,2.0000000000);
#26=EDGE_CURVE('NONE',#17,#19,#24,.T.);
#27=ORIENTED_EDGE('NONE',*,*,#26,.T.);
#28=CARTESIAN_POINT('NONE',(0.0000000000,-2.0000000000,0.0000000000));
#29=DIRECTION('NONE',(0.0000000000,0.0000000000,-1.0000000000));
#30=DIRECTION('NONE',(0.0000000000,1.0000000000,0.0000000000));
#31=AXIS2_PLACEMENT_3D('NONE',#28,#30,#29);
#32=CIRCLE('*CRV12',#31,2.0000000000);
```

```
#34=EDGE_CURVE('NONE',#19,#17,#32,.T.);
#35=ORIENTED_EDGE('NONE',*,*,#34,.T.);
#36=EDGE_LOOP('NONE',(#27,#35));
#37=FACE_OUTER_BOUND('NONE',#36,.F.);
#38=CARTESIAN_POINT('NONE',(-0.414213562379,-2.000000000000,
1.000000000001));
#39=VERTEX_POINT('NONE',#38);
#40=CARTESIAN_POINT('NONE',(-1.000000000000,-2.000000000000,
0.414213562382));
#41=VERTEX_POINT('NONE',#40);
#42=DIRECTION('NONE',(-0.707106781187,-0.000000000000,-0.707106781187));
#43=VECTOR('NONE',#42,1.000000000000);
#44=CARTESIAN_POINT('NONE',(-0.707106781187,-
2.000000000000,0.707106781187));
#45=LINE('*LN124',#44,#43);
#47=EDGE_CURVE('NONE',#39,#41,#45,.T.);
#48=ORIENTED_EDGE('NONE',*,*,#47,.T.);
#49=CARTESIAN_POINT('NONE',(-1.000000000000,-2.000000000000,-
0.414213562365));
#50=VERTEX_POINT('NONE',#49);
#51=DIRECTION('NONE',(0.000000000000,0.000000000000,1.000000000000));
#52=VECTOR('NONE',#51,1.000000000000);
#53=CARTESIAN_POINT('NONE',(-1.000000000000,-2.000000000000,
0.000000000000));
#54=LINE('*LN125',#53,#52);
#56=EDGE_CURVE('NONE',#50,#41,#54,.T.);
#57=ORIENTED_EDGE('NONE',*,*,#56,.F.);
```



```
#58=CARTESIAN_POINT('NONE',(-0.414213562380,-2.00000000000,-  
1.00000000000));  
#59=VERTEX_POINT('NONE',#58);  
#60=DIRECTION('NONE',(-0.707106781187,0.00000000000,0.707106781187));  
#61=VECTOR('NONE',#60,1.00000000000);  
#62=CARTESIAN_POINT('NONE',(-0.707106781187,-2.00000000000,-  
0.707106781187));  
#63=LINE('*LN126',#62,#61);  
#65=EDGE_CURVE('NONE',#59,#50,#63,.T.);  
#66=ORIENTED_EDGE('NONE',*,*,#65,.F.);  
#67=CARTESIAN_POINT('NONE',(0.414213562364,-2.00000000000,-  
1.00000000000));  
#68=VERTEX_POINT('NONE',#67);  
#69=DIRECTION('NONE',(1.00000000000,0.00000000000,0.00000000000));  
#70=VECTOR('NONE',#69,1.00000000000);  
#71=CARTESIAN_POINT('NONE',(0.00000000000,-2.00000000000,-1.00000000000));  
#72=LINE('*LN127',#71,#70);  
#74=EDGE_CURVE('NONE',#59,#68,#72,.T.);  
#75=ORIENTED_EDGE('NONE',*,*,#74,.T.);  
#76=CARTESIAN_POINT('NONE',(1.00000000001,-2.00000000000,-  
0.414213562374));  
#77=VERTEX_POINT('NONE',#76);  
#78=DIRECTION('NONE',(-0.707106781187,-0.00000000000,-0.707106781187));  
#79=VECTOR('NONE',#78,1.00000000000);  
#80=CARTESIAN_POINT('NONE',(0.707106781187,-2.00000000000,-  
0.707106781187));  
#81=LINE('*LN128',#80,#79);  
#83=EDGE_CURVE('NONE',#77,#68,#81,.T.);
```

```
#84=ORIENTED_EDGE('NONE',*,*,#83,.F.);
#85=CARTESIAN_POINT('NONE',(1.00000000000,-2.00000000000,
0.414213562373));
#86=VERTEX_POINT('NONE',#85);
#87=DIRECTION('NONE',(6.12323399574E-17,0.00000000000,1.00000000000));
#88=VECTOR('NONE',#87,1.00000000000);
#89=CARTESIAN_POINT('NONE',(1.00000000000,-2.00000000000,-6.12323399574E-
17));
#90=LINE('*LN129',#89,#88);
#92=EDGE_CURVE('NONE',#77,#86,#90,.T.);
#93=ORIENTED_EDGE('NONE',*,*,#92,.T.);
#94=CARTESIAN_POINT('NONE',(0.414213562378,-2.00000000000,
1.000000000001));
#95=VERTEX_POINT('NONE',#94);
#96=DIRECTION('NONE',(-0.707106781187,0.00000000000,0.707106781187));
#97=VECTOR('NONE',#96,1.00000000000);
#98=CARTESIAN_POINT('NONE',(0.707106781187,-2.00000000000,
0.707106781187));
#99=LINE('*LN130',#98,#97);
#101=EDGE_CURVE('NONE',#86,#95,#99,.T.);
#102=ORIENTED_EDGE('NONE',*,*,#101,.T.);
#103=DIRECTION('NONE',(1.00000000000,0.00000000000,0.00000000000));
#104=VECTOR('NONE',#103,1.00000000000);
#105=CARTESIAN_POINT('NONE',(0.00000000000,-2.00000000000,
1.00000000000));
#106=LINE('*LN131',#105,#104);
#108=EDGE_CURVE('NONE',#39,#95,#106,.T.);
#109=ORIENTED_EDGE('NONE',*,*,#108,.F.);
```

```
#110=EDGE_LOOP('NONE',(#48,#57,#66,#75,#84,#93,#102,#109));
#111=FACE_BOUND('NONE',#110,.F.);
#112=CARTESIAN_POINT('NONE',(0.000000000000,-2.000000000000,
0.000000000000));
#113=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#114=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#115=AXIS2_PLACEMENT_3D('NONE',#112,#114,#113);
#116=PLANE('NONE',#115);
#117=ADVANCED_FACE('*FAC1',(#37,#111),#116,.F.);
#120=CARTESIAN_POINT('NONE',(-3.19106402553E-12,0.000000000000,
2.000000000000));
#121=VERTEX_POINT('NONE',#120);
#122=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#123=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#124=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#125=AXIS2_PLACEMENT_3D('NONE',#122,#124,#123);
#126=CIRCLE('*CRV13',#125,2.000000000000);
#128=EDGE_CURVE('NONE',#170,#121,#126,.T.);
#129=ORIENTED_EDGE('NONE',*,*,#128,.T.);
#130=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#131=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#132=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#133=AXIS2_PLACEMENT_3D('NONE',#130,#132,#131);
#134=CIRCLE('*CRV14',#133,2.000000000000);
#136=EDGE_CURVE('NONE',#121,#170,#134,.T.);
#137=ORIENTED_EDGE('NONE',*,*,#136,.T.);
#138=EDGE_LOOP('NONE',(#129,#137));
#139=FACE_OUTER_BOUND('NONE',#138,.T.);
```

```
#140=CARTESIAN_POINT('NONE',(-1.59464383434E-12,0.00000000000,-  
0.999999999999));  
#141=VERTEX_POINT('NONE',#140);  
#142=CARTESIAN_POINT('NONE',(-1.59553201276E-12,0.00000000000,  
0.999999999999));  
#143=VERTEX_POINT('NONE',#142);  
#144=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));  
#145=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));  
#146=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));  
#147=AXIS2_PLACEMENT_3D('NONE',#144,#146,#145);  
#148=CIRCLE('*CRV15',#147,1.00000000000);  
#150=EDGE_CURVE('NONE',#141,#143,#148,.T.);  
#151=ORIENTED_EDGE('NONE',*,*,#150,.T.);  
#152=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));  
#153=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));  
#154=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));  
#155=AXIS2_PLACEMENT_3D('NONE',#152,#154,#153);  
#156=CIRCLE('*CRV16',#155,1.00000000000);  
#158=EDGE_CURVE('NONE',#143,#141,#156,.T.);  
#159=ORIENTED_EDGE('NONE',*,*,#158,.T.);  
#160=EDGE_LOOP('NONE',(#151,#159));  
#161=FACE_BOUND('NONE',#160,.F.);  
#162=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));  
#163=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));  
#164=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));  
#165=AXIS2_PLACEMENT_3D('NONE',#162,#164,#163);  
#166=PLANE('NONE',#165);  
#167=ADVANCED_FACE('*FAC2',(#139,#161),#166,.T.);
```

```
#168=ORIENTED_EDGE('NONE',*,*,#26,.F.);
#169=CARTESIAN_POINT('NONE',(-5.34378146473E-24,0.00000000000,-
2.00000000000));
#170=VERTEX_POINT('NONE',#169);
#171=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#172=VECTOR('NONE',#171,1.00000000000);
#173=CARTESIAN_POINT('NONE',(-5.34378146473E-24,-0.00000000000,-
2.00000000000));
#174=LINE(*LN132',#173,#172);
#176=EDGE_CURVE('NONE',#17,#170,#174,.T.);
#177=ORIENTED_EDGE('NONE',*,*,#176,.T.);
#178=ORIENTED_EDGE('NONE',*,*,#128,.T.);
#179=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#180=VECTOR('NONE',#179,1.00000000000);
#181=CARTESIAN_POINT('NONE',(-5.34378146473E-24,-0.00000000000,
2.00000000000));
#182=LINE(*LN133',#181,#180);
#184=EDGE_CURVE('NONE',#19,#121,#182,.T.);
#185=ORIENTED_EDGE('NONE',*,*,#184,.F.);
#186=EDGE_LOOP('NONE',(#168,#177,#178,#185));
#187=FACE_OUTER_BOUND('NONE',#186,.F.);
#188=CARTESIAN_POINT('NONE',(0.00000000000,-2.00000000000,
0.00000000000));
#189=DIRECTION('NONE',(-2.67189073236E-24,0.00000000000,-1.00000000000));
#190=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#191=AXIS2_PLACEMENT_3D('NONE',#188,#190,#189);
#192=CYLINDRICAL_SURFACE(*SUR1',#191,2.00000000000);
#194=ADVANCED_FACE(*FAC3',(#187),#192,.T.);
```

```
#195=ORIENTED_EDGE('NONE',*,*,#34,.F.);
#196=ORIENTED_EDGE('NONE',*,*,#184,.T.);
#197=ORIENTED_EDGE('NONE',*,*,#136,.T.);
#198=ORIENTED_EDGE('NONE',*,*,#176,.F.);
#199=EDGE_LOOP('NONE',(#195,#196,#197,#198));
#200=FACE_OUTER_BOUND('NONE',#199,.F.);
#201=ADVANCED_FACE('*FAC4',(#200),#192,.T.);
#204=CARTESIAN_POINT('NONE',(-8.92861133206E-06,10.0000000000,
0.7000000000131));
#205=VERTEX_POINT('NONE',#204);
#206=B_SPLINE_CURVE_WITH_KNOTS('*CRV17',5,(#207,#208,#209,#210,#211,#2
12,#213,#214,#215,#216,#217),.UNSPECIFIED.,U.,U.,(6,5,6),(0.0000000000,1.00000
000000,2.00000000000),.UNSPECIFIED.);
#207=CARTESIAN_POINT('NONE',(-8.92861133212E-06,10.0000000000,-
0.7000000000131));
#208=CARTESIAN_POINT('NONE',(-0.219922519834,10.0000000000,-
0.699919547887));
#209=CARTESIAN_POINT('NONE',(-0.440018614181,10.0000000000,-
0.613948622003));
#210=CARTESIAN_POINT('NONE',(-0.613982337129,10.0000000000,-
0.440033011880));
#211=CARTESIAN_POINT('NONE',(-0.699922166145,10.0000000000,-
0.219871568593));
#212=CARTESIAN_POINT('NONE',(-0.700000095365,10.0000000000,-
3.33066907388E-16));
#213=CARTESIAN_POINT('NONE',(-
0.699939623134,10.0000000000,0.219876888431));
```

```
#214=CARTESIAN_POINT('NONE',(-
0.613917660654,10.0000000000,0.440049991875));
#215=CARTESIAN_POINT('NONE',(-0.440093229931,10.0000000000,
0.613965536418));
#216=CARTESIAN_POINT('NONE',(-0.219812335383,10.0000000000,
0.699931816186));
#217=CARTESIAN_POINT('NONE',(-8.92861133187E-06,10.0000000000,
0.700000000131));
#218=EDGE_CURVE('NONE',#664,#205,#206,.T.);
#219=ORIENTED_EDGE('NONE',*,*,#218,.T.);
#220=B_SPLINE_CURVE_WITH_KNOTS('*CRV18',5,(#221,#222,#223,#224,#225,#2
26,#227,#228,#229,#230,#231),UNSPECIFIED.,U.,U.,(6,5,6),(0.0000000000,1.00000
000000,2.0000000000),UNSPECIFIED.);
#221=CARTESIAN_POINT('NONE',(8.92861133200E-06,10.0000000000,
0.700000000131));
#222=CARTESIAN_POINT('NONE',(0.219922519834,10.0000000000,0.699919547887
));
#223=CARTESIAN_POINT('NONE',(0.440018614181,10.0000000000,0.613948622003
));
#224=CARTESIAN_POINT('NONE',(0.613982337129,10.0000000000,0.440033011880
));
#225=CARTESIAN_POINT('NONE',(0.699922166145,10.0000000000,0.219871568593
));
#226=CARTESIAN_POINT('NONE',(0.700000095365,10.0000000000,1.27675647832
E-15));
#227=CARTESIAN_POINT('NONE',(0.699939623134,10.0000000000,-
0.219876888431));
```

```
#228=CARTESIAN_POINT('NONE',(0.613917660654,10.0000000000,-  
0.440049991875));  
#229=CARTESIAN_POINT('NONE',(0.440093229931,10.0000000000,-  
0.613965536418));  
#230=CARTESIAN_POINT('NONE',(0.219812335383,10.0000000000,-  
0.699931816186));  
#231=CARTESIAN_POINT('NONE',(8.92861133225E-06,10.0000000000,-  
0.700000000131));  
#232=EDGE_CURVE('NONE',#205,#664,#220,.T.);  
#233=ORIENTED_EDGE('NONE',*,*,#232,.T.);  
#234=EDGE_LOOP('NONE',(#219,#233));  
#235=FACE_OUTER_BOUND('NONE',#234,.T.);  
#236=CARTESIAN_POINT('NONE',(0.0000000000,10.0000000000,0.0000000000));  
#237=DIRECTION('NONE',(0.0000000000,0.0000000000,-1.0000000000));  
#238=DIRECTION('NONE',(0.0000000000,1.0000000000,0.0000000000));  
#239=AXIS2_PLACEMENT_3D('NONE',#236,#238,#237);  
#240=PLANE('NONE',#239);  
#241=ADVANCED_FACE('*FAC5',(#235),#240,.T.);  
#242=ORIENTED_EDGE('NONE',*,*,#150,.T.);  
#245=DIRECTION('NONE',(0.0000000000,1.0000000000,0.0000000000));  
#246=VECTOR('NONE',#245,1.0000000000);  
#247=CARTESIAN_POINT('NONE',(-2.67189073236E-24,-0.0000000000,  
1.0000000000));  
#248=LINE('*LN134',#247,#246);  
#250=EDGE_CURVE('NONE',#143,#285,#248,.T.);  
#251=ORIENTED_EDGE('NONE',*,*,#250,.T.);  
#252=CARTESIAN_POINT('NONE',(-2.67189073236E-24,9.7000000000,-  
1.0000000000));
```



```
#253=VERTEX_POINT('NONE',#252);
#254=B_SPLINE_CURVE_WITH_KNOTS('*CRV19',5,(#255,#256,#257,#258,#259,#2
60,#261,#262,#263),.UNSPECIFIED.,.U.,.U.,(6,3,6),(0.00000000000,1.00000000000,2.0
0000000000),.UNSPECIFIED.);
#255=CARTESIAN_POINT('NONE',(3.33066907388E-16,9.70000000000,-
1.00000000000));
#256=CARTESIAN_POINT('NONE',(-0.314157150628,9.70000000000,-
1.00000934996));
#257=CARTESIAN_POINT('NONE',(-0.628579947296,9.70000000000,-
0.877062675197));
#258=CARTESIAN_POINT('NONE',(-0.877046039200,9.70000000000,-
0.628465496529));
#259=CARTESIAN_POINT('NONE',(-1.12290417028,9.70000000000,
1.22211072341E-13));
#260=CARTESIAN_POINT('NONE',(-0.877046039200,9.70000000000,
0.628465496529));
#261=CARTESIAN_POINT('NONE',(-0.628579947296,9.70000000000,
0.877062675197));
#262=CARTESIAN_POINT('NONE',(-0.314157150628,9.70000000000,
1.00000934996));
#263=CARTESIAN_POINT('NONE',(4.77395900589E-15,9.70000000000,
1.00000000000));
#264=EDGE_CURVE('NONE',#253,#285,#254,.T.);
#265=ORIENTED_EDGE('NONE',*,*,#264,.F.);
#266=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#267=VECTOR('NONE',#266,1.00000000000);
#268=CARTESIAN_POINT('NONE',(-2.67189073236E-24,-0.00000000000,-
1.00000000000));
```

```

#269=LINE(*LN135',#268,#267);
#271=EDGE_CURVE('NONE',#141,#253,#269,.T.);
#272=ORIENTED_EDGE('NONE',*,*,#271,.F.);
#273=EDGE_LOOP('NONE',(#242,#251,#265,#272));
#274=FACE_OUTER_BOUND('NONE',#273,.T.);
#275=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#276=DIRECTION('NONE',(-2.67189073236E-24,0.00000000000,-1.00000000000));
#277=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#278=AXIS2_PLACEMENT_3D('NONE',#275,#277,#276);
#279=CYLINDRICAL_SURFACE(*SUR2',#278,1.00000000000);
#281=ADVANCED_FACE(*FAC6',(#274),#279,.T.);
#282=ORIENTED_EDGE('NONE',*,*,#158,.T.);
#283=ORIENTED_EDGE('NONE',*,*,#271,.T.);
#284=CARTESIAN_POINT('NONE',(-1.22464677243E-16,9.70000000000,
1.00000000000));
#285=VERTEX_POINT('NONE',#284);
#286=B_SPLINE_CURVE_WITH_KNOTS(*CRV20',5,(#287,#288,#289,#290,#291,#2
92,#293,#294,#295),.UNSPECIFIED..U.,U.,(6,3,6),(0.00000000000,1.00000000000,2.0
0000000000),.UNSPECIFIED.);
#287=CARTESIAN_POINT('NONE',(-2.22044604925E-16,9.70000000000,
1.00000000000));
#288=CARTESIAN_POINT('NONE',(0.314157150628,9.70000000000,1.00000934996)
);
#289=CARTESIAN_POINT('NONE',(0.628579947296,9.70000000000,0.877062675197
));
#290=CARTESIAN_POINT('NONE',(0.877046039200,9.70000000000,0.628465496529
));

```

```
#291=CARTESIAN_POINT('NONE',(1.12290417028,9.70000000000,-  
3.82404896145E-14));  
#292=CARTESIAN_POINT('NONE',(0.877046039200,9.70000000000,-  
0.628465496529));  
#293=CARTESIAN_POINT('NONE',(0.628579947296,9.70000000000,-  
0.877062675197));  
#294=CARTESIAN_POINT('NONE',(0.314157150628,9.70000000000,-  
1.00000934996));  
#295=CARTESIAN_POINT('NONE',(-4.48804488227E-15,9.70000000000,-  
1.00000000000));  
#296=EDGE_CURVE('NONE',#285,#253,#286,.T.);  
#297=ORIENTED_EDGE('NONE',*,*,#296,.F.);  
#298=ORIENTED_EDGE('NONE',*,*,#250,.F.);  
#299=EDGE_LOOP('NONE',(#282,#283,#297,#298));  
#300=FACE_OUTER_BOUND('NONE',#299,.T.);  
#301=ADVANCED_FACE('*FAC7',(#300),#279,.T.);  
#302=ORIENTED_EDGE('NONE',*,*,#47,.F.);  
#303=CARTESIAN_POINT('NONE',(-0.414213562373,-0.500000000000,  
1.00000000000));  
#304=VERTEX_POINT('NONE',#303);  
#305=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));  
#306=VECTOR('NONE',#305,1.00000000000);  
#307=CARTESIAN_POINT('NONE',(-0.414213562373,-0.00000000000,  
1.00000000000));  
#308=LINE('*LN136',#307,#306);  
#310=EDGE_CURVE('NONE',#39,#304,#308,.T.);  
#311=ORIENTED_EDGE('NONE',*,*,#310,.T.);  
#314=DIRECTION('NONE',(-0.707106781187,0.00000000000,-0.707106781187));
```

```
#315=VECTOR('NONE',#314,1.00000000000);
#316=CARTESIAN_POINT('NONE',(-0.707106781187,-0.500000000000,
0.707106781187));
#317=LINE('*LN137',#316,#315);
#319=EDGE_CURVE('NONE',#304,#501,#317,.T.);
#320=ORIENTED_EDGE('NONE',*,*,#319,.T.);
#321=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#322=VECTOR('NONE',#321,1.00000000000);
#323=CARTESIAN_POINT('NONE',(-0.999999999997,-0.000000000000,
0.414213562376));
#324=LINE('*LN138',#323,#322);
#326=EDGE_CURVE('NONE',#41,#501,#324,.T.);
#327=ORIENTED_EDGE('NONE',*,*,#326,.F.);
#328=EDGE_LOOP('NONE',(#302,#311,#320,#327));
#329=FACE_OUTER_BOUND('NONE',#328,.F.);
#330=CARTESIAN_POINT('NONE',(-0.999999999997,-2.000000000000,
0.414213562376));
#331=DIRECTION('NONE',(0.00000000000,-1.00000000000,0.00000000000));
#332=DIRECTION('NONE',(-0.707106781187,0.00000000000,0.707106781187));
#333=AXIS2_PLACEMENT_3D('NONE',#330,#332,#331);
#334=PLANE('NONE',#333);
#335=ADVANCED_FACE('*FAC8',(#329),#334,.F.);
#336=ORIENTED_EDGE('NONE',*,*,#108,.F.);
#337=ORIENTED_EDGE('NONE',*,*,#310,.T.);
#340=DIRECTION('NONE',(1.00000000000,0.00000000000,0.00000000000));
#341=VECTOR('NONE',#340,1.00000000000);
#342=CARTESIAN_POINT('NONE',(0.00000000000,-0.500000000000,
1.00000000000));
```

```
#343=LINE(*LN139',#342,#341);
#345=EDGE_CURVE('NONE',#304,#373,#343,.T.);
#346=ORIENTED_EDGE('NONE',*,*,#345,.T.);
#347=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#348=VECTOR('NONE',#347,1.00000000000);
#349=CARTESIAN_POINT('NONE',(0.414213562371,-0.00000000000,
1.00000000000));
#350=LINE(*LN140',#349,#348);
#352=EDGE_CURVE('NONE',#95,#373,#350,.T.);
#353=ORIENTED_EDGE('NONE',*,*,#352,.F.);
#354=EDGE_LOOP('NONE',(#336,#337,#346,#353));
#355=FACE_OUTER_BOUND('NONE',#354,.T.);
#356=CARTESIAN_POINT('NONE',(0.414213562371,-2.00000000000,
1.00000000000));
#357=DIRECTION('NONE',(0.00000000000,-1.00000000000,0.00000000000));
#358=DIRECTION('NONE',(-0.00000000000,-0.00000000000,-1.00000000000));
#359=AXIS2_PLACEMENT_3D('NONE',#356,#358,#357);
#360=PLANE('NONE',#359);
#361=ADVANCED_FACE(*FAC9',(#355),#360,.T.);
#362=ORIENTED_EDGE('NONE',*,*,#101,.F.);
#365=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#366=VECTOR('NONE',#365,1.00000000000);
#367=CARTESIAN_POINT('NONE',(1.00000000000,-0.00000000000,
0.414213562373));
#368=LINE(*LN141',#367,#366);
#370=EDGE_CURVE('NONE',#86,#401,#368,.T.);
#371=ORIENTED_EDGE('NONE',*,*,#370,.T.);
```

```
#372=CARTESIAN_POINT('NONE',(0.414213562372,-0.500000000000,
1.000000000000));
#373=VERTEX_POINT('NONE',#372);
#374=DIRECTION('NONE',(-0.707106781187,0.000000000000,0.707106781187));
#375=VECTOR('NONE',#374,1.000000000000);
#376=CARTESIAN_POINT('NONE',(0.707106781187,-0.500000000000,
0.707106781187));
#377=LINE('*LN142',#376,#375);
#379=EDGE_CURVE('NONE',#401,#373,#377,.T.);
#380=ORIENTED_EDGE('NONE',*,*,#379,.T.);
#381=ORIENTED_EDGE('NONE',*,*,#352,.F.);
#382=EDGE_LOOP('NONE',(#362,#371,#380,#381));
#383=FACE_OUTER_BOUND('NONE',#382,.F.);
#384=CARTESIAN_POINT('NONE',(0.414213562372,-2.000000000000,
1.000000000000));
#385=DIRECTION('NONE',(0.000000000000,-1.000000000000,0.000000000000));
#386=DIRECTION('NONE',(0.707106781187,0.000000000000,0.707106781187));
#387=AXIS2_PLACEMENT_3D('NONE',#384,#386,#385);
#388=PLANE('NONE',#387);
#389=ADVANCED_FACE('*FAC10',(#383),#388,.F.);
#390=ORIENTED_EDGE('NONE',*,*,#92,.F.);
#391=CARTESIAN_POINT('NONE',(1.000000000000,-0.500000000000,-
0.414213562363));
#392=VERTEX_POINT('NONE',#391);
#393=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#394=VECTOR('NONE',#393,1.000000000000);
#395=CARTESIAN_POINT('NONE',(1.000000000000,-0.000000000000,-
0.414213562363));
```

```
#396=LINE(*LN143',#395,#394);
#398=EDGE_CURVE('NONE',#77,#392,#396,.T.);
#399=ORIENTED_EDGE('NONE',*,*,#398,.T.);
#400=CARTESIAN_POINT('NONE',(1.000000000000,-0.500000000000,
0.414213562373));
#401=VERTEX_POINT('NONE',#400);
#402=DIRECTION('NONE',(6.12323399574E-17,0.000000000000, 1.000000000000));
#403=VECTOR('NONE',#402,1.000000000000);
#404=CARTESIAN_POINT('NONE',(1.000000000000,-0.500000000000,-
6.12323399574E-17));
#405=LINE(*LN144',#404,#403);
#407=EDGE_CURVE('NONE',#392,#401,#405,.T.);
#408=ORIENTED_EDGE('NONE',*,*,#407,.T.);
#409=ORIENTED_EDGE('NONE',*,*,#370,.F.);
#410=EDGE_LOOP('NONE',(#390,#399,#408,#409));
#411=FACE_OUTER_BOUND('NONE',#410,.F.);
#412=CARTESIAN_POINT('NONE',(1.000000000000,-2.000000000000,
0.414213562373));
#413=DIRECTION('NONE',(0.000000000000,-1.000000000000,0.000000000000));
#414=DIRECTION('NONE',(1.000000000000,0.000000000000,0.000000000000));
#415=AXIS2_PLACEMENT_3D('NONE',#412,#414,#413);
#416=PLANE('NONE',#415);
#417=ADVANCED_FACE(*FAC11',(#411),#416,.F.);
#418=ORIENTED_EDGE('NONE',*,*,#83,.F.);
#419=ORIENTED_EDGE('NONE',*,*,#398,.T.);
#422=DIRECTION('NONE',(-0.707106781187,0.000000000000,-0.707106781187));
#423=VECTOR('NONE',#422,1.000000000000);
```

```
#424=CARTESIAN_POINT('NONE',(0.707106781187,-0.500000000000,-  
0.707106781187));  
#425=LINE('*LN145',#424,#423);  
#427=EDGE_CURVE('NONE',#392,#455,#425,.T.);  
#428=ORIENTED_EDGE('NONE',*,*,#427,.T.);  
#429=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));  
#430=VECTOR('NONE',#429,1.000000000000);  
#431=CARTESIAN_POINT('NONE',(0.414213562369,-0.000000000000,-  
1.000000000000));  
#432=LINE('*LN146',#431,#430);  
#434=EDGE_CURVE('NONE',#68,#455,#432,.T.);  
#435=ORIENTED_EDGE('NONE',*,*,#434,.F.);  
#436=EDGE_LOOP('NONE',(#418,#419,#428,#435));  
#437=FACE_OUTER_BOUND('NONE',#436,.T.);  
#438=CARTESIAN_POINT('NONE',(0.414213562369,-2.000000000000,-  
1.000000000000));  
#439=DIRECTION('NONE',(0.000000000000,-1.000000000000,0.000000000000));  
#440=DIRECTION('NONE',(-0.707106781187,0.000000000000,0.707106781187));  
#441=AXIS2_PLACEMENT_3D('NONE',#438,#440,#439);  
#442=PLANE('NONE',#441);  
#443=ADVANCED_FACE('*FAC12',(#437),#442,.T.);  
#444=ORIENTED_EDGE('NONE',*,*,#74,.F.);  
#445=CARTESIAN_POINT('NONE',(-0.414213562376,-0.500000000000,-  
1.000000000000));  
#446=VERTEX_POINT('NONE',#445);  
#447=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));  
#448=VECTOR('NONE',#447,1.000000000000);
```



```
#449=CARTESIAN_POINT('NONE',(-0.414213562376,-0.000000000000,-  
1.000000000000));  
#450=LINE('*LN147',#449,#448);  
#452=EDGE_CURVE('NONE',#59,#446,#450,.T.);  
#453=ORIENTED_EDGE('NONE',*,*,#452,.T.);  
#454=CARTESIAN_POINT('NONE',(0.414213562364,-0.500000000000,-  
1.000000000000));  
#455=VERTEX_POINT('NONE',#454);  
#456=DIRECTION('NONE',(1.000000000000,0.000000000000,0.000000000000));  
#457=VECTOR('NONE',#456,1.000000000000);  
#458=CARTESIAN_POINT('NONE',(0.000000000000,-0.500000000000,-  
1.000000000000));  
#459=LINE('*LN148',#458,#457);  
#461=EDGE_CURVE('NONE',#446,#455,#459,.T.);  
#462=ORIENTED_EDGE('NONE',*,*,#461,.T.);  
#463=ORIENTED_EDGE('NONE',*,*,#434,.F.);  
#464=EDGE_LOOP('NONE',(#444,#453,#462,#463));  
#465=FACE_OUTER_BOUND('NONE',#464,.F.);  
#466=CARTESIAN_POINT('NONE',(0.414213562364,-2.000000000000,-  
1.000000000000));  
#467=DIRECTION('NONE',(0.000000000000,-1.000000000000,0.000000000000));  
#468=DIRECTION('NONE',(-0.000000000000,-0.000000000000,-1.000000000000));  
#469=AXIS2_PLACEMENT_3D('NONE',#466,#468,#467);  
#470=PLANE('NONE',#469);  
#471=ADVANCED_FACE('*FAC13',(#465),#470,.F.);  
#472=ORIENTED_EDGE('NONE',*,*,#65,.F.);  
#473=ORIENTED_EDGE('NONE',*,*,#452,.T.);
```

```
#474=CARTESIAN_POINT('NONE',(-1.00000000000,-0.50000000000,-  
0.414213562369));  
#475=VERTEX_POINT('NONE',#474);  
#476=DIRECTION('NONE',(-0.707106781187,0.00000000000,0.707106781187));  
#477=VECTOR('NONE',#476,1.00000000000);  
#478=CARTESIAN_POINT('NONE',(-0.707106781187,-0.50000000000,-  
0.707106781187));  
#479=LINE('*LN149',#478,#477);  
#481=EDGE_CURVE('NONE',#446,#475,#479,.T.);  
#482=ORIENTED_EDGE('NONE',*,*,#481,.T.);  
#483=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));  
#484=VECTOR('NONE',#483,1.00000000000);  
#485=CARTESIAN_POINT('NONE',(-1.00000000000,-0.00000000000,-  
0.414213562369));  
#486=LINE('*LN150',#485,#484);  
#488=EDGE_CURVE('NONE',#50,#475,#486,.T.);  
#489=ORIENTED_EDGE('NONE',*,*,#488,.F.);  
#490=EDGE_LOOP('NONE',(#472,#473,#482,#489));  
#491=FACE_OUTER_BOUND('NONE',#490,.T.);  
#492=CARTESIAN_POINT('NONE',(-1.00000000000,-2.00000000000,-  
0.414213562369));  
#493=DIRECTION('NONE',(0.00000000000,-1.00000000000,0.00000000000));  
#494=DIRECTION('NONE',(0.707106781187,0.00000000000,0.707106781187));  
#495=AXIS2_PLACEMENT_3D('NONE',#492,#494,#493);  
#496=PLANE('NONE',#495);  
#497=ADVANCED_FACE('*FAC14',(#491),#496,.T.);  
#498=ORIENTED_EDGE('NONE',*,*,#56,.F.);  
#499=ORIENTED_EDGE('NONE',*,*,#488,.T.);
```

```
#500=CARTESIAN_POINT('NONE',(-1.00000000000,-0.500000000000,
0.414213562380));
#501=VERTEX_POINT('NONE',#500);
#502=DIRECTION('NONE',(0.00000000000,0.00000000000,1.00000000000));
#503=VECTOR('NONE',#502,1.00000000000);
#504=CARTESIAN_POINT('NONE',(-1.00000000000,-0.500000000000,
0.00000000000));
#505=LINE('*LN151',#504,#503);
#507=EDGE_CURVE('NONE',#475,#501,#505,.T.);
#508=ORIENTED_EDGE('NONE',*,*,#507,.T.);
#509=ORIENTED_EDGE('NONE',*,*,#326,.F.);
#510=EDGE_LOOP('NONE',(#498,#499,#508,#509));
#511=FACE_OUTER_BOUND('NONE',#510,.T.);
#512=CARTESIAN_POINT('NONE',(-1.00000000000,-
2.00000000000,0.414213562380));
#513=DIRECTION('NONE',(0.00000000000,-1.00000000000,0.00000000000));
#514=DIRECTION('NONE',(1.00000000000,0.00000000000,0.00000000000));
#515=AXIS2_PLACEMENT_3D('NONE',#512,#514,#513);
#516=PLANE('NONE',#515);
#517=ADVANCED_FACE('*FAC15',(#511),#516,.T.);
#518=ORIENTED_EDGE('NONE',*,*,#379,.F.);
#521=DIRECTION('NONE',(-0.838716980803,0.419358490402,-0.347407948441));
#522=VECTOR('NONE',#521,1.00000000000);
#523=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#524=LINE('*LN152',#523,#522);
#526=EDGE_CURVE('NONE',#401,#634,#524,.T.);
#527=ORIENTED_EDGE('NONE',*,*,#526,.T.);
#528=DIRECTION('NONE',(-0.347407948440,0.419358490401,-0.838716980804));
```

```
#529=VECTOR('NONE',#528,1.000000000000);
#530=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#531=LINE('*LN153',#530,#529);
#533=EDGE_CURVE('NONE',#373,#634,#531,.T.);
#534=ORIENTED_EDGE('NONE',*,*,#533,.F.);
#535=EDGE_LOOP('NONE',(#518,#527,#534));
#536=FACE_OUTER_BOUND('NONE',#535,.F.);
#537=CARTESIAN_POINT('NONE',(0.414213562372,-0.500000000000,
1.000000000000));
#538=DIRECTION('NONE',(0.838716980803,-0.419358490402, 0.347407948441));
#539=DIRECTION('NONE',(0.316227766017,0.894427191000,0.316227766017));
#540=AXIS2_PLACEMENT_3D('NONE',#537,#539,#538);
#541=PLANE('NONE',#540);
#542=ADVANCED_FACE('*FAC16',(#536),#541,.F.);
#543=ORIENTED_EDGE('NONE',*,*,#407,.F.);
#546=DIRECTION('NONE',(-0.838716980806,0.419358490403, 0.347407948434));
#547=VECTOR('NONE',#546,1.000000000000);
#548=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#549=LINE('*LN154',#548,#547);
#551=EDGE_CURVE('NONE',#392,#634,#549,.T.);
#552=ORIENTED_EDGE('NONE',*,*,#551,.T.);
#553=ORIENTED_EDGE('NONE',*,*,#526,.F.);
#554=EDGE_LOOP('NONE',(#543,#552,#553));
#555=FACE_OUTER_BOUND('NONE',#554,.F.);
#556=CARTESIAN_POINT('NONE',(1.000000000000,-0.500000000000,
0.414213562373));
#557=DIRECTION('NONE',(0.838716980806,-0.419358490403,-0.347407948434));
#558=DIRECTION('NONE',(0.447213595500,0.894427191000,-1.18795477829E-17));
```

```
#559=AXIS2_PLACEMENT_3D('NONE',#556,#558,#557);
#560=PLANE('NONE',#559);
#561=ADVANCED_FACE('*FAC17',(#555),#560,.F.);
#562=ORIENTED_EDGE('NONE',*,*,#427,.F.);
#563=ORIENTED_EDGE('NONE',*,*,#551,.T.);
#564=DIRECTION('NONE',(-0.347407948437,0.419358490401,0.838716980806));
#565=VECTOR('NONE',#564,1.000000000000);
#566=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#567=LINE('*LN155',#566,#565);
#569=EDGE_CURVE('NONE',#455,#634,#567,.T.);
#570=ORIENTED_EDGE('NONE',*,*,#569,.F.);
#571=EDGE_LOOP('NONE',(#562,#563,#570));
#572=FACE_OUTER_BOUND('NONE',#571,.T.);
#573=CARTESIAN_POINT('NONE',(0.414213562369,-0.500000000000,-
1.000000000000));
#574=DIRECTION('NONE',(0.838716980806,-0.419358490401,-0.347407948437));
#575=DIRECTION('NONE',(-0.316227766017,-0.894427191000,0.316227766017));
#576=AXIS2_PLACEMENT_3D('NONE',#573,#575,#574);
#577=PLANE('NONE',#576);
#578=ADVANCED_FACE('*FAC18',(#572),#577,.T.);
#579=ORIENTED_EDGE('NONE',*,*,#461,.F.);
#582=DIRECTION('NONE',(0.347407948444,0.419358490401,0.838716980803));
#583=VECTOR('NONE',#582,1.000000000000);
#584=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#585=LINE('*LN156',#584,#583);
#587=EDGE_CURVE('NONE',#446,#634,#585,.T.);
#588=ORIENTED_EDGE('NONE',*,*,#587,.T.);
#589=ORIENTED_EDGE('NONE',*,*,#569,.F.);
```

```
#590=EDGE_LOOP('NONE',(#579,#588,#589));
#591=FACE_OUTER_BOUND('NONE',#590,.F.);
#592=CARTESIAN_POINT('NONE',(0.414213562364,-0.500000000000,-
1.000000000000));
#593=DIRECTION('NONE',(-0.347407948444,-0.419358490401,-0.838716980803));
#594=DIRECTION('NONE',(-4.96406566423E-18,0.894427191000,-0.447213595500));
#595=AXIS2_PLACEMENT_3D('NONE',#592,#594,#593);
#596=PLANE('NONE',#595);
#597=ADVANCED_FACE('*FAC19',(#591),#596,.F.);
#598=ORIENTED_EDGE('NONE',*,*,#481,.F.);
#599=ORIENTED_EDGE('NONE',*,*,#587,.T.);
#600=DIRECTION('NONE',(0.838716980805,0.419358490401,0.347407948437));
#601=VECTOR('NONE',#600,1.000000000000);
#602=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#603=LINE('*LN157',#602,#601);
#605=EDGE_CURVE('NONE',#475,#634,#603,.T.);
#606=ORIENTED_EDGE('NONE',*,*,#605,.F.);
#607=EDGE_LOOP('NONE',(#598,#599,#606));
#608=FACE_OUTER_BOUND('NONE',#607,.T.);
#609=CARTESIAN_POINT('NONE',(-1.000000000000,-0.500000000000,-
0.414213562369));
#610=DIRECTION('NONE',(-0.347407948443,-0.419358490402,-0.838716980803));
#611=DIRECTION('NONE',(0.316227766017,-0.894427191000,0.316227766017));
#612=AXIS2_PLACEMENT_3D('NONE',#609,#611,#610);
#613=PLANE('NONE',#612);
#614=ADVANCED_FACE('*FAC20',(#608),#613,.T.);
#615=ORIENTED_EDGE('NONE',*,*,#507,.F.);
#616=ORIENTED_EDGE('NONE',*,*,#605,.T.);
```

```
#617=DIRECTION('NONE',(0.838716980802,0.419358490401,-0.347407948446));
#618=VECTOR('NONE',#617,1.000000000000);
#619=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#620=LINE('*LN158',#619,#618);
#622=EDGE_CURVE('NONE',#501,#634,#620,.T.);
#623=ORIENTED_EDGE('NONE',*,*,#622,.F.);
#624=EDGE_LOOP('NONE',(#615,#616,#623));
#625=FACE_OUTER_BOUND('NONE',#624,.T.);
#626=CARTESIAN_POINT('NONE',(-1.000000000000,-0.500000000000,
0.414213562380));
#627=DIRECTION('NONE',(-0.838716980805,-0.419358490403,-0.347407948435));
#628=DIRECTION('NONE',(0.447213595500,-0.894427191000,1.04579619199E-17));
#629=AXIS2_PLACEMENT_3D('NONE',#626,#628,#627);
#630=PLANE('NONE',#629);
#631=ADVANCED_FACE('*FAC21',(#625),#630,.T.);
#632=ORIENTED_EDGE('NONE',*,*,#319,.F.);
#633=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#634=VERTEX_POINT('NONE',#633);
#635=DIRECTION('NONE',(0.347407948441,0.419358490402,-0.838716980803));
#636=VECTOR('NONE',#635,1.000000000000);
#637=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#638=LINE('*LN159',#637,#636);
#640=EDGE_CURVE('NONE',#304,#634,#638,.T.);
#641=ORIENTED_EDGE('NONE',*,*,#640,.T.);
#642=ORIENTED_EDGE('NONE',*,*,#622,.F.);
#643=EDGE_LOOP('NONE',(#632,#641,#642));
#644=FACE_OUTER_BOUND('NONE',#643,.F.);
```

```
#645=CARTESIAN_POINT('NONE',(-0.999999999997,-0.500000000000,
0.414213562376));
#646=DIRECTION('NONE',(-0.347407948441,-0.419358490402, 0.838716980803));
#647=DIRECTION('NONE',(-0.316227766017,0.894427191000, 0.316227766017));
#648=AXIS2_PLACEMENT_3D('NONE',#645,#647,#646);
#649=PLANE('NONE',#648);
#650=ADVANCED_FACE('*FAC22',(#644),#649,.F.);
#651=ORIENTED_EDGE('NONE',*,*,#345,.F.);
#652=ORIENTED_EDGE('NONE',*,*,#640,.T.);
#653=ORIENTED_EDGE('NONE',*,*,#533,.F.);
#654=EDGE_LOOP('NONE',(#651,#652,#653));
#655=FACE_OUTER_BOUND('NONE',#654,.T.);
#656=CARTESIAN_POINT('NONE',(0.414213562371,-0.500000000000,
1.000000000000));
#657=DIRECTION('NONE',(-0.347407948441,-0.419358490402, 0.838716980803));
#658=DIRECTION('NONE',(6.32535428746E-18,-0.894427191000,-0.447213595500));
#659=AXIS2_PLACEMENT_3D('NONE',#656,#658,#657);
#660=PLANE('NONE',#659);
#661=ADVANCED_FACE('*FAC23',(#655),#660,.T.);
#662=ORIENTED_EDGE('NONE',*,*,#296,.T.);
#663=CARTESIAN_POINT('NONE',(8.92861133206E-06,10.0000000000,-
0.700000000131));
#664=VERTEX_POINT('NONE',#663);
#665=DIRECTION('NONE',(-2.10449387314E-05,0.707106781184,0.707106780876));
#666=VECTOR('NONE',#665,1.00000000000);
#667=CARTESIAN_POINT('NONE',(0.000129464864320,5.34999999981,-
5.349999999829));
#668=LINE('*LN160',#667,#666);
```



```

#670=EDGE_CURVE('NONE',#253,#664,#668,.T.);
#671=ORIENTED_EDGE('NONE',*,*,#670,.T.);
#672=ORIENTED_EDGE('NONE',*,*,#232,.F.);
#673=DIRECTION('NONE',(2.10449387317E-05,0.707106781184,-0.707106780876));
#674=VECTOR('NONE',#673,1.00000000000);
#675=CARTESIAN_POINT('NONE',(-0.000129464864322,5.34999999981,
5.34999999829));
#676=LINE('*LN161',#675,#674);
#678=EDGE_CURVE('NONE',#285,#205,#676,.T.);
#679=ORIENTED_EDGE('NONE',*,*,#678,.F.);
#680=EDGE_LOOP('NONE',(#662,#671,#672,#679));
#681=FACE_OUTER_BOUND('NONE',#680,.T.);
#682=B_SPLINE_SURFACE_WITH_KNOTS('*SUR4',5,1,((#683,#684),(#685,#686),(#
687,#688),(#689,#690),(#691,#692),(#693,#694),(#695,#696),(#697,#698),(#699,#700),(#
701,#702),(#703,#704)),.UNSPECIFIED.,U.,U.,U.,(6,5,6),(2,2),(0.00000000000,1.00
000000000,2.00000000000),(0.00000000000,1.00000000000),.UNSPECIFIED.);
#683=CARTESIAN_POINT('NONE',(-1.22464677243E-16,9.70000000000,
1.00000000000));
#684=CARTESIAN_POINT('NONE',(8.92861133201E-06,10.0000000000,
0.700000000131));
#685=CARTESIAN_POINT('NONE',(0.314223983798,9.70000000000,0.999885102757
));
#686=CARTESIAN_POINT('NONE',(0.219930850515,10.0000000000,0.699918165012
));
#687=CARTESIAN_POINT('NONE',(0.628541451493,9.69999999999,0.877070037866
));
#688=CARTESIAN_POINT('NONE',(0.440014220706,10.0000000000,0.613948849355
));

```

```
#689=CARTESIAN_POINT('NONE',(0.877157645552,9.70000000000,0.628612342943
));
#690=CARTESIAN_POINT('NONE',(0.613983702083,10.00000000000,0.440028545072
));
#691=CARTESIAN_POINT('NONE',(0.999873925941,9.70000000000,0.314112736172
));
#692=CARTESIAN_POINT('NONE',(0.699920356317,10.00000000000,0.219879494377
));
#693=CARTESIAN_POINT('NONE',(0.999999904631,9.70000000000,1.77635683940
E-15));
#694=CARTESIAN_POINT('NONE',(0.700000095365,10.00000000000,1.27675649155
E-15));
#695=CARTESIAN_POINT('NONE',(0.999873925941,9.70000000000,-
0.314112736172));
#696=CARTESIAN_POINT('NONE',(0.699920356317,10.00000000000,-
0.219879494377));
#697=CARTESIAN_POINT('NONE',(0.877157645552,9.700000000001,-
0.628612342943));
#698=CARTESIAN_POINT('NONE',(0.613983702083,10.00000000000,-
0.440028545072));
#699=CARTESIAN_POINT('NONE',(0.628541451493,9.69999999999,-
0.877070037866));
#700=CARTESIAN_POINT('NONE',(0.440014220706,10.00000000000,-
0.613948849355));
#701=CARTESIAN_POINT('NONE',(0.314223983798,9.70000000000,-
0.999885102757));
#702=CARTESIAN_POINT('NONE',(0.219930850515,10.00000000000,-
0.699918165012));
```

```
#703=CARTESIAN_POINT('NONE',(9.71445146547E-17,9.70000000000,-  
1.00000000000));  
#704=CARTESIAN_POINT('NONE',(8.92861133203E-06,10.0000000000,-  
0.700000000131));  
#705=ADVANCED_FACE('*FAC24',(#681),#682,.T.);  
#706=ORIENTED_EDGE('NONE',*,*,#264,.T.);  
#707=ORIENTED_EDGE('NONE',*,*,#678,.T.);  
#708=ORIENTED_EDGE('NONE',*,*,#218,.F.);  
#709=ORIENTED_EDGE('NONE',*,*,#670,.F.);  
#710=EDGE_LOOP('NONE',(#706,#707,#708,#709));  
#711=FACE_OUTER_BOUND('NONE',#710,.T.);  
#712=B_SPLINE_SURFACE_WITH_KNOTS('*SUR3',5,1,((#713,#714),(#715,#716),(#  
717,#718),(#719,#720),(#721,#722),(#723,#724),(#725,#726),(#727,#728),(#729,#730),(  
#731,#732),(#733,#734)),UNSPECIFIED.,U.,U.,U.,(6,5,6),(2,2),(0.00000000000,1.00  
000000000,2.00000000000),(0.00000000000,1.00000000000),UNSPECIFIED.);  
#713=CARTESIAN_POINT('NONE',(-2.67189073236E-24,9.70000000000,-  
1.00000000000));  
#714=CARTESIAN_POINT('NONE',(-8.92861133201E-06,10.0000000000,-  
0.700000000131));  
#715=CARTESIAN_POINT('NONE',(-0.314223983798,9.70000000000,-  
0.999885102757));  
#716=CARTESIAN_POINT('NONE',(-0.219930850515,10.0000000000,-  
0.699918165012));  
#717=CARTESIAN_POINT('NONE',(-0.628541451493,9.70000000000,-  
0.877070037866));  
#718=CARTESIAN_POINT('NONE',(-0.440014220706,10.0000000000,-  
0.613948849355));
```

```
#719=CARTESIAN_POINT('NONE',(-0.877157645552,9.70000000000,-  
0.628612342943));  
#720=CARTESIAN_POINT('NONE',(-0.613983702083,10.0000000000,-  
0.440028545072));  
#721=CARTESIAN_POINT('NONE',(-0.999873925941,9.70000000000,-  
0.314112736172));  
#722=CARTESIAN_POINT('NONE',(-0.699920356317,10.0000000000,-  
0.219879494377));  
#723=CARTESIAN_POINT('NONE',(-0.999999904631,9.70000000000,-  
2.22044604925E-16));  
#724=CARTESIAN_POINT('NONE',(-0.700000095365,10.0000000000,-  
2.33146837818E-16));  
#725=CARTESIAN_POINT('NONE',(-0.999873925941,9.70000000000,  
0.314112736172));  
#726=CARTESIAN_POINT('NONE',(-0.699920356317,10.0000000000,  
0.219879494377));  
#727=CARTESIAN_POINT('NONE',(-0.877157645552,9.70000000000,  
0.628612342943));  
#728=CARTESIAN_POINT('NONE',(-0.613983702083,10.0000000000,  
0.440028545072));  
#729=CARTESIAN_POINT('NONE',(-0.628541451493,9.69999999999,  
0.877070037866));  
#730=CARTESIAN_POINT('NONE',(-0.440014220706,10.0000000000,  
0.613948849355));  
#731=CARTESIAN_POINT('NONE',(-0.314223983798,9.70000000000,  
0.999885102757));  
#732=CARTESIAN_POINT('NONE',(-0.219930850515,10.0000000000,  
0.699918165012));
```

```
#733=CARTESIAN_POINT('NONE',(1.22124532709E-15,9.700000000000,
1.000000000000));
#734=CARTESIAN_POINT('NONE',(-8.92861133324E-06,10.00000000000,
0.700000000131));
#735=ADVANCED_FACE('*FAC25',(#711),#712,.,T.);
#736=CLOSED_SHELL('NONE',(#117,#167,#194,#201,#241,#281,#301,#335,#361,#38
9,#417,#443,#471,#497,#517,#542,#561,#578,#597,#614,#631,#650,#661,#705,#735));
#737=ADVANCED_BREP_SHAPE_REPRESENTATION('VIS M2',(#15,#742),#13);
#738=ADVANCED_BREP_SHAPE_REPRESENTATION('*MASTER',(#744,#750,#75
6,#762),#7);
#739=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#740=DIRECTION('NONE',(1.000000000000,0.000000000000,0.000000000000));
#741=DIRECTION('NONE',(0.000000000000,0.000000000000,1.000000000000));
#742=AXIS2_PLACEMENT_3D('NONE',#739,#741,#740);
#743=REPRESENTATION_MAP(#742,#737);
#744=MAPPED_ITEM('*DIT1',#743,#12);
#745=CARTESIAN_POINT('NONE',(20.66000000000,0.00000000000,-
20.63000000000));
#746=DIRECTION('NONE',(1.000000000000,0.000000000000,0.000000000000));
#747=DIRECTION('NONE',(0.000000000000,0.000000000000,1.000000000000));
#748=AXIS2_PLACEMENT_3D('NONE',#745,#747,#746);
#749=REPRESENTATION_MAP(#742,#737);
#750=MAPPED_ITEM('*DIT2',#749,#748);
#751=CARTESIAN_POINT('NONE',(-20.66000000000,0.00000000000,
20.63000000000));
#752=DIRECTION('NONE',(1.000000000000,0.000000000000,0.000000000000));
#753=DIRECTION('NONE',(0.000000000000,0.000000000000,1.000000000000));
#754=AXIS2_PLACEMENT_3D('NONE',#751,#753,#752);
```

```
#755=REPRESENTATION_MAP(#742,#737);
#756=MAPPED_ITEM('*DIT3',#755,#754);
#757=CARTESIAN_POINT('NONE',(-20.6600000000,0.0000000000,-
20.6300000000));
#758=DIRECTION('NONE',(1.0000000000,0.0000000000,0.0000000000));
#759=DIRECTION('NONE',(0.0000000000,0.0000000000,1.0000000000));
#760=AXIS2_PLACEMENT_3D('NONE',#757,#759,#758);
#761=REPRESENTATION_MAP(#742,#737);
#762=MAPPED_ITEM('*DIT4',#761,#760);
(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASS
IGNED_CONTEXT((#764))GLOBAL_UNIT_ASSIGNED_CONTEXT((#2,#5,#6))REP
RESENTATION_CONTEXT('FIX BOUCHON','WORKSPACE'));
#764=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.10000000
0000),#2,'TOL_CRV','CONFUSED CURVE UNCERTAINTY');
#765=MANIFOLD_SOLID_BREP('*SOL1',#856);
#766=CARTESIAN_POINT('NONE',(-3.18928766869E-12,-2.0000000000,-
2.0000000000));
#767=VERTEX_POINT('NONE',#766);
#768=CARTESIAN_POINT('NONE',(-3.19106402553E-12,-2.0000000000,
2.0000000000));
#769=VERTEX_POINT('NONE',#768);
#770=CARTESIAN_POINT('NONE',(0.0000000000,-2.0000000000,
0.0000000000));
#771=DIRECTION('NONE',(0.0000000000,0.0000000000,-1.0000000000));
#772=DIRECTION('NONE',(0.0000000000,1.0000000000,0.0000000000));
#773=AXIS2_PLACEMENT_3D('NONE',#770,#772,#771);
#774=CIRCLE('*CRV21',#773,2.0000000000);
#776=EDGE_CURVE('NONE',#767,#769,#774,.T.);
```

```
#777=ORIENTED_EDGE('NONE',*,*,#776,.T.);
#778=CARTESIAN_POINT('NONE',(0.000000000000,-2.000000000000,
0.000000000000));
#779=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#780=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#781=AXIS2_PLACEMENT_3D('NONE',#778,#780,#779);
#782=CIRCLE('*CRV22',#781,2.000000000000);
#784=EDGE_CURVE('NONE',#769,#767,#782,.T.);
#785=ORIENTED_EDGE('NONE',*,*,#784,.T.);
#786=EDGE_LOOP('NONE',(#777,#785));
#787=FACE_OUTER_BOUND('NONE',#786,.F.);
#788=CARTESIAN_POINT('NONE',(0.000000000000,-2.000000000000,
0.000000000000));
#789=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#790=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#791=AXIS2_PLACEMENT_3D('NONE',#788,#790,#789);
#792=PLANE('NONE',#791);
#793=ADVANCED_FACE('*FAC26',(#787),#792,.F.);
#796=CARTESIAN_POINT('NONE',(-3.19106402553E-12,0.000000000000,
2.000000000000));
#797=VERTEX_POINT('NONE',#796);
#798=CARTESIAN_POINT('NONE',(0.000000000000,0.000000000000,0.000000000000));
#799=DIRECTION('NONE',(0.000000000000,0.000000000000,-1.000000000000));
#800=DIRECTION('NONE',(0.000000000000,1.000000000000,0.000000000000));
#801=AXIS2_PLACEMENT_3D('NONE',#798,#800,#799);
#802=CIRCLE('*CRV23',#801,2.000000000000);
#804=EDGE_CURVE('NONE',#824,#797,#802,.T.);
#805=ORIENTED_EDGE('NONE',*,*,#804,.T.);
```

```
#806=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#807=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#808=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#809=AXIS2_PLACEMENT_3D('NONE',#806,#808,#807);
#810=CIRCLE('*CRV24',#809,2.00000000000);
#812=EDGE_CURVE('NONE',#797,#824,#810,.T.);
#813=ORIENTED_EDGE('NONE',*,*,#812,.T.);
#814=EDGE_LOOP('NONE',(#805,#813));
#815=FACE_OUTER_BOUND('NONE',#814,.T.);
#816=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#817=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#818=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#819=AXIS2_PLACEMENT_3D('NONE',#816,#818,#817);
#820=PLANE('NONE',#819);
#821=ADVANCED_FACE('*FAC27',(#815),#820,.T.);
#822=ORIENTED_EDGE('NONE',*,*,#776,.F.);
#823=CARTESIAN_POINT('NONE',(-5.34378146473E-24,0.00000000000,-
2.00000000000));
#824=VERTEX_POINT('NONE',#823);
#825=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#826=VECTOR('NONE',#825,1.00000000000);
#827=CARTESIAN_POINT('NONE',(-5.34378146473E-24,-0.00000000000,-
2.00000000000));
#828=LINE('*LN162',#827,#826);
#830=EDGE_CURVE('NONE',#767,#824,#828,.T.);
#831=ORIENTED_EDGE('NONE',*,*,#830,.T.);
#832=ORIENTED_EDGE('NONE',*,*,#804,.T.);
#833=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
```



```
#834=VECTOR('NONE',#833,1.00000000000);
#835=CARTESIAN_POINT('NONE',(-5.34378146473E-24,-0.00000000000,
2.00000000000));
#836=LINE('*LN163',#835,#834);
#838=EDGE_CURVE('NONE',#769,#797,#836,.T.);
#839=ORIENTED_EDGE('NONE',*,*,#838,.F.);
#840=EDGE_LOOP('NONE',(#822,#831,#832,#839));
#841=FACE_OUTER_BOUND('NONE',#840,.F.);
#842=CARTESIAN_POINT('NONE',(0.00000000000,-2.00000000000,
0.00000000000));
#843=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#844=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#845=AXIS2_PLACEMENT_3D('NONE',#842,#844,#843);
#846=CYLINDRICAL_SURFACE('*SUR5',#845,2.00000000000);
#848=ADVANCED_FACE('*FAC28',(#841),#846,.T.);
#849=ORIENTED_EDGE('NONE',*,*,#784,.F.);
#850=ORIENTED_EDGE('NONE',*,*,#838,.T.);
#851=ORIENTED_EDGE('NONE',*,*,#812,.T.);
#852=ORIENTED_EDGE('NONE',*,*,#830,.F.);
#853=EDGE_LOOP('NONE',(#849,#850,#851,#852));
#854=FACE_OUTER_BOUND('NONE',#853,.F.);
#855=ADVANCED_FACE('*FAC29',(#854),#846,.T.);
#856=CLOSED_SHELL('NONE',(#793,#821,#848,#855));
#857=ADVANCED_BREP_SHAPE_REPRESENTATION('FIX
BOUCHON',(#765,#858),#763);
#858=MANIFOLD_SOLID_BREP('*SOL3',#944);
#859=CARTESIAN_POINT('NONE',(-1.59464383434E-12,0.00000000000,-
0.99999999999));
```

```
#860=VERTEX_POINT('NONE',#859);
#861=CARTESIAN_POINT('NONE',(-1.59553201276E-12,0.00000000000,
0.999999999999));
#862=VERTEX_POINT('NONE',#861);
#863=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#864=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#865=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#866=AXIS2_PLACEMENT_3D('NONE',#863,#865,#864);
#867=CIRCLE('*CRV25',#866,1.00000000000);
#869=EDGE_CURVE('NONE',#860,#862,#867,.T.);
#870=ORIENTED_EDGE('NONE',*,*,#869,.T.);
#871=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#872=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#873=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#874=AXIS2_PLACEMENT_3D('NONE',#871,#873,#872);
#875=CIRCLE('*CRV26',#874,1.00000000000);
#877=EDGE_CURVE('NONE',#862,#860,#875,.T.);
#878=ORIENTED_EDGE('NONE',*,*,#877,.T.);
#879=EDGE_LOOP('NONE',(#870,#878));
#880=FACE_OUTER_BOUND('NONE',#879,.F.);
#881=ADVANCED_FACE('*FAC30',(#880),#820,.F.);
#884=CARTESIAN_POINT('NONE',(-1.59553201276E-12,10.0000000000,
0.999999999999));
#885=VERTEX_POINT('NONE',#884);
#886=CARTESIAN_POINT('NONE',(0.00000000000,10.0000000000,0.00000000000));
#887=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#888=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#889=AXIS2_PLACEMENT_3D('NONE',#886,#888,#887);
```

```
#890=CIRCLE(*CRV27',#889,1.00000000000);
#892=EDGE_CURVE('NONE',#912,#885,#890,.T.);
#893=ORIENTED_EDGE('NONE',*,*,#892,.T.);
#894=CARTESIAN_POINT('NONE',(0.00000000000,10.0000000000,0.00000000000));
#895=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#896=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#897=AXIS2_PLACEMENT_3D('NONE',#894,#896,#895);
#898=CIRCLE(*CRV28',#897,1.00000000000);
#900=EDGE_CURVE('NONE',#885,#912,#898,.T.);
#901=ORIENTED_EDGE('NONE',*,*,#900,.T.);
#902=EDGE_LOOP('NONE',(#893,#901));
#903=FACE_OUTER_BOUND('NONE',#902,.T.);
#904=CARTESIAN_POINT('NONE',(0.00000000000,10.0000000000,0.00000000000));
#905=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#906=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#907=AXIS2_PLACEMENT_3D('NONE',#904,#906,#905);
#908=PLANE('NONE',#907);
#909=ADVANCED_FACE(*FAC31',(#903),#908,.T.);
#910=ORIENTED_EDGE('NONE',*,*,#869,.F.);
#911=CARTESIAN_POINT('NONE',(-2.67189073236E-24,10.0000000000,-
1.00000000000));
#912=VERTEX_POINT('NONE',#911);
#913=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#914=VECTOR('NONE',#913,1.00000000000);
#915=CARTESIAN_POINT('NONE',(-2.67189073236E-24,-0.00000000000,-
1.00000000000));
#916=LINE(*LN164',#915,#914);
#918=EDGE_CURVE('NONE',#860,#912,#916,.T.);
```

```
#919=ORIENTED_EDGE('NONE',*,*,#918,.T.);
#920=ORIENTED_EDGE('NONE',*,*,#892,.T.);
#921=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#922=VECTOR('NONE',#921,1.00000000000);
#923=CARTESIAN_POINT('NONE',(-2.67189073236E-24,-0.00000000000,
1.00000000000));
#924=LINE('*LN165',#923,#922);
#926=EDGE_CURVE('NONE',#862,#885,#924,.T.);
#927=ORIENTED_EDGE('NONE',*,*,#926,.F.);
#928=EDGE_LOOP('NONE',(#910,#919,#920,#927));
#929=FACE_OUTER_BOUND('NONE',#928,.F.);
#930=CARTESIAN_POINT('NONE',(0.00000000000,0.00000000000,0.00000000000));
#931=DIRECTION('NONE',(0.00000000000,0.00000000000,-1.00000000000));
#932=DIRECTION('NONE',(0.00000000000,1.00000000000,0.00000000000));
#933=AXIS2_PLACEMENT_3D('NONE',#930,#932,#931);
#934=CYLINDRICAL_SURFACE('*SUR6',#933,1.00000000000);
#936=ADVANCED_FACE('*FAC32',(#929),#934,.T.);
#937=ORIENTED_EDGE('NONE',*,*,#877,.F.);
#938=ORIENTED_EDGE('NONE',*,*,#926,.T.);
#939=ORIENTED_EDGE('NONE',*,*,#900,.T.);
#940=ORIENTED_EDGE('NONE',*,*,#918,.F.);
#941=EDGE_LOOP('NONE',(#937,#938,#939,#940));
#942=FACE_OUTER_BOUND('NONE',#941,.F.);
#943=ADVANCED_FACE('*FAC33',(#942),#934,.T.);
#944=CLOSED_SHELL('NONE',(#881,#909,#936,#943));
ENDSEC;
END-ISO-10303-21;
```

ANNEXE A3

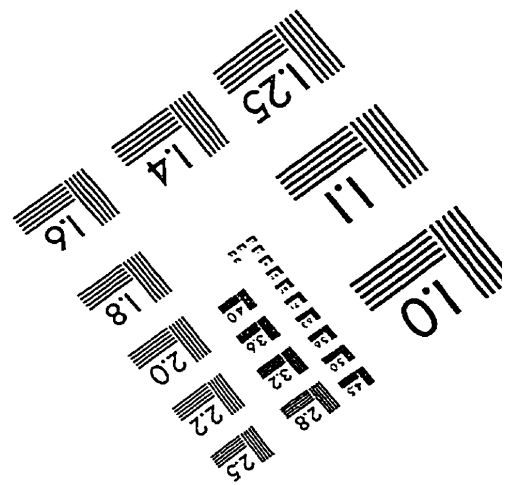
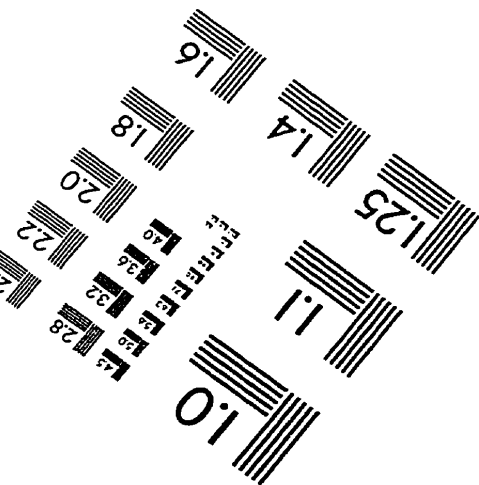
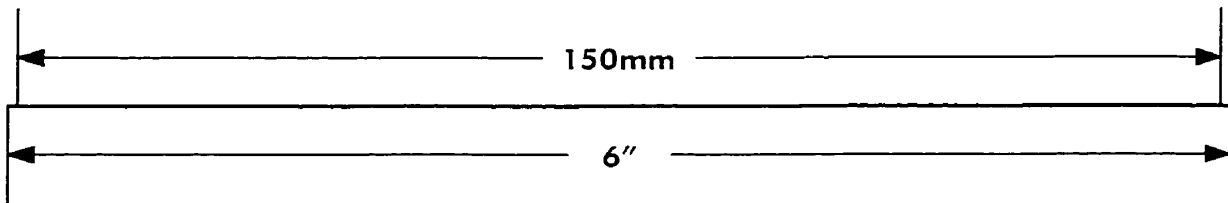
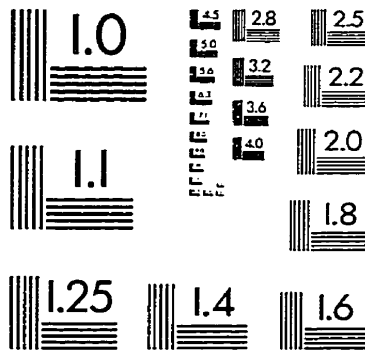
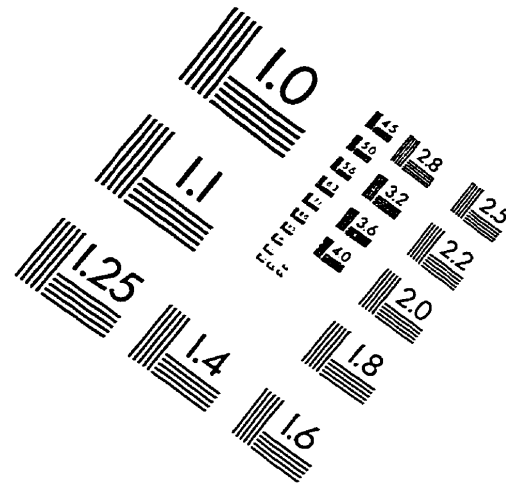
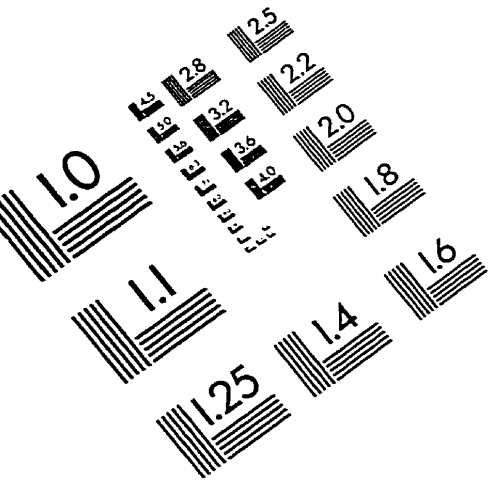
**PARAMÈTRES D'UN FICHER IGES À SPÉCIFIER SUR
CATIA**

Voici les paramètres à spécifier pour le processus de traduction en format IGES pour CATIA dans l'exemple 3.4:

- a) le répertoire où le fichier est situé
- b) le nom du fichier
- c) le répertoire où placer le fichier IGES et spécifier le nom désiré pour ce dernier
- d) le format du fichier IGES
- e) le créateur (optionel)
- f) la compagnie (optionel)
- g) le nom du fichier (optionel)
- h) le receveur du fichier (optionel)
- i) l'expéditeur (optionel)
- j) la version "prepro" (optionel)
- k) un message (optionel)
- l) des commentaires (optionel)
- m) transfert en VDA-IS (optionel)
- n) la version de IGES:
 - i) v3
 - ii) v4
 - iii) v5
 - iv) v5.1
 - v) v5.2
- o) le mode de conversion:
 - i) standard
 - ii) allongée
 - iii) réduite
- p) le transfert canonique:
 - i) standard

- ii) B-spline
- q) le transfert conique:
 - i) standard
 - ii) B-spline
- r) le transfert spline:
 - i) B-spline
 - ii) paramètre
- s) la norme de cotation:
 - i) aucune
 - ii) ANSI
 - iii) ISO
 - iv) DIN
 - v) JIS
- t) l'identification du "draft"
- u) l'application "string"
- v) la table des polices de caractères (pour la cotation)
- w) la grosseur de l'index (optionel)
- x) la grosseur des données (optionel)
- y) le niveau des messages:
 - i) indication des erreurs
 - ii) indication des avertissements
 - iii) information
 - iv) indication des traces

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved