



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Université Henri Poincaré, Nancy I  
UFR Sciences Techniques Mathématiques Informatique Automatique  
Ecole Doctorale IAE+M  
DFD Automatique et Production Automatisée



C.N.R.S. URA 821

## THESE

présentée en vue de l'obtention du



### DOCTORAT DE L'UNIVERSITE HENRI POINCARÉ NANCY I Spécialité Production Automatisée

par

**Frédéric CHAXEL**

#### **Contribution à la gestion et à la conduite des systèmes manufacturiers par les objets nomades de production**

soutenue publiquement le 21/12/1995 devant la Commission d'Examen:

- Président: M. VERON, Professeur à l'Université H. Poincaré, Nancy I
- Rapporteurs: S. TICHKIEWITCH, Professeur à l'Institut National Polytechnique de Grenoble  
F. VERNADAT, Professeur à l'Université de Metz
- Examineurs: E. BAJIC, Maître de Conférences HDR à l'Université H. Poincaré, Nancy I  
G. SEGARRA, Ingénieur, Coordinateur de la Recherche Informatique à la Direction de l'Organisation et de l'Ingénierie Informatique, Renault RNUR

Directeur de thèse: J. RICHARD, Professeur à l'Université H. Poincaré, Nancy I

## AVANT-PROPOS

Le travail présenté dans ce mémoire a été réalisé au Centre de Recherche en Automatique de Nancy (C.R.A.N.-U.R.A. 821 du C.N.R.S.) dans l'équipe Ingénierie de la Qualité (I.Q.) du Groupe Génie de la Production (G.G.P.).

A son terme, je tiens tout particulièrement à remercier:

- Monsieur Michel VERON, Professeur à l'Université H. Poincaré, Nancy I, directeur du CRAN, de m'avoir accueilli ces dernières années au sein de son laboratoire et de me faire l'honneur de présider ce jury.

- Monsieur Serge TICHKIEWITCH, Professeur à l'Institut National Polytechnique de Grenoble et Monsieur François VERNADAT, Professeur à l'Université de Metz pour avoir accepté de juger ce travail en qualité de rapporteurs.

- Monsieur Gerard SEGARRA, Coordinateur de la Recherche Informatique à la Direction de l'Organisation et de l'Ingénierie Informatique à la régie Renault, pour avoir accepté de participer à ce jury en tant qu'examinateur, ainsi que Madame Brigitte LONC, chef de projets de Recherche à la régie Renault pour le soutien qu'elle nous a apporté.

- Monsieur Eddy BAJIC, Maître de Conférences HDR à l'Université H. Poincaré - Nancy I, pour les nombreux conseils qu'il m'a prodigués. Son soutien sur le plan scientifique a largement contribué à l'aboutissement de ces travaux.

- Monsieur Jacques RICHARD, Professeur à l'Université H. Poincaré - Nancy I, pour m'avoir accueilli dans son équipe. La confiance qu'il m'a toujours accordée, son expérience et ses conseils ont été autant de facteurs qui ont permis d'orienter mes travaux dans des directions originales.

Je tiens également à exprimer ma profonde sympathie à tous les membres du laboratoire et de l'AIP Lorrain qui m'ont soutenu dans les moments les plus difficiles.

Enfin, je voudrais adresser une pensée toute particulière à ma famille et à ma compagne Nathalie pour la patience dont elles ont su faire preuve à mon égard et pour leur sens aigu de la grammaire.

# Sommaire

---

---

<b>INTRODUCTION .....</b>	<b>1</b>
<b>CHAPITRE I : SYSTEME DE PRODUCTION ET OBJETS NOMADES DE PRODUCTION .....</b>	<b>4</b>
<b>CHAPITRE II : INTEGRATION PAR LE MODELE DU PRODUIT .....</b>	<b>26</b>
<b>CHAPITRE III : DES PRODUITS AUX OBJETS NOMADES DE PRODUCTION .....</b>	<b>48</b>
<b>CHAPITRE IV : MISE EN OEUVRE ET APPLICATION DU CONCEPT .....</b>	<b>77</b>
<b>CONCLUSION &amp; PERSPECTIVES .....</b>	<b>100</b>
<b>BIBLIOGRAPHIE .....</b>	<b>105</b>
<b>ACRONYMES .....</b>	<b>114</b>
<b>ANNEXE : LE LANGAGE EXPRESS .....</b>	<b>116</b>

---

<b>INTRODUCTION</b>	<b>1</b>
---------------------	----------

---

<b>CHAPITRE I : SYSTEME DE PRODUCTION ET OBJETS NOMADES DE PRODUCTION</b>	<b>4</b>
---	----------

---

<b>1. INTRODUCTION</b>	<b>4</b>
<b>2. CONCEPT D'OBJETS NOMADES DE PRODUCTION</b>	<b>5</b>
2.1. APPROCHE CONSIDEREE	5
2.2. ETAT DE L'ART DES TRAVAUX A L'ORIGINE DU CONCEPT	7
2.2.1. <i>Le concept iso-hiérarchique issu de l'Université de Madison-Wisconsin</i>	8
2.2.2. <i>La métaphore biologique du CAM-I Japon</i>	9
2.2.3. <i>Le concept de "générateur de jetons"</i>	10
2.2.4. <i>Le projet Esprit "Holonc Manufacturing Systems"</i>	12
<b>3. APPORT DE L'IDENTIFICATION AUTOMATIQUE</b>	<b>13</b>
3.1. DEFINITION DE L'IDENTIFICATION AUTOMATIQUE	13
3.1.1. <i>La technologie du Code à Barres</i>	14
3.1.2. <i>Les technologies des étiquettes électroniques</i>	15
3.1.2.1. Les étiquettes à code fixe	16
3.1.2.2. Les étiquettes programmables	17
3.2. LES APPORTS DU PROJET ESPRIT MONOLIN	18
3.2.1. <i>Les résultats</i>	18
3.2.2. <i>L'interface applicative ou EMS-API</i>	19
3.2.3. <i>Aspects réseaux de communication</i>	20
<b>4. INFLUENCE SUR LES SYSTEMES DE PRODUCTION</b>	<b>20</b>
4.1. POSITIONNEMENT DANS UN CONTEXTE CIM	20
4.1.1. <i>De l'architecture classique</i>	20
4.1.2. <i>à une architecture à objets nomades</i>	21
4.2. APPORTS EN GESTION DE LA QUALITE	22
4.2.1. <i>Gestion de la qualité des produits</i>	22
4.2.2. <i>Traçabilité des produits</i>	23
4.3. IMPACT SUR LE SYSTEME D'INFORMATION	24
<b>5. CONCLUSION</b>	<b>25</b>

<b>CHAPITRE II : INTEGRATION PAR LE MODELE DU PRODUIT</b>	<b>26</b>
<b>1. INTRODUCTION</b> .....	<b>26</b>
<b>2. LE PRODUIT : MOTEUR DE L'ENTREPRISE</b> .....	<b>27</b>
<b>3. DEMARCHE D'INTEGRATION PAR LES DONNEES</b> .....	<b>29</b>
3.1. HISTORIQUE .....	30
3.2. ROLE DU MODELE .....	32
3.3. LA NORMALISATION INTERNATIONALE PIERRE ANGULAIRE DE L'EDIFICE .....	33
<b>4. LA SERIE DE NORMES STEP (ISO 10303)</b> .....	<b>34</b>
4.1. OBJECTIFS.....	34
4.2. STRUCTURATION .....	35
4.3. LES OUTILS DE BASE (PARTIES 11 A 29).....	36
4.3.1. <i>Le langage de modélisation EXPRESS 10303-11</i> .....	36
4.3.1.1. Les éléments du langage .....	38
4.3.1.2. Cadre d'utilisation et outils.....	38
4.3.1.3. Les travaux en cours .....	39
4.3.2. <i>L'échange par fichiers neutres 10303-21</i> .....	40
4.3.3. <i>L'interface d'accès aux données SDAI 10303-22</i> .....	41
4.4. VERS UN STANDARD UNIVERSEL ?.....	42
<b>5. ARCHITECTURES INTEGRANTES STEP</b> .....	<b>43</b>
5.1. LES OUTILS LOGICIELS .....	43
5.2. LES PLATES-FORMES PROTOTYPES .....	44
<b>6. CONCLUSION</b> .....	<b>46</b>
<b>CHAPITRE III : DES PRODUITS AUX OBJETS NOMADES DE PRODUCTION</b>	<b>48</b>
<b>1. INTRODUCTION</b> .....	<b>48</b>
<b>2. UN OBJET NOMADE PARTICULIER : L'OBJET INFORME</b> .....	<b>49</b>
2.1. LES DONNEES DU PRODUIT.....	51
2.1.1. <i>Formalisme de modélisation des données</i> .....	51
2.1.2. <i>Distribution des données</i> .....	53
2.1.2.1. Nécessité de la distribution.....	53
2.1.2.2. Critères de distribution.....	55

2.2. PROPOSITION D'UNE INTERFACE D'ACCES A L'OBJET NOMADE DE PRODUCTION .....	57
2.2.1. Structure d'accueil .....	57
2.2.2. Spécifications du module supportant l'interface .....	59
2.2.2.1. Modèle en couches .....	59
2.2.2.2. Analyse fonctionnelle du module .....	60
2.2.2.3. Accès aux données distribuées.....	62
2.2.2.4. La gestion des données.....	64
2.2.2.5. La couche "métier" .....	65
2.2.3. Génération de l'interface physique.....	66
<b>3. PROPOSITION D'UNE DEMARCHE DE CONCEPTION.....</b>	<b>68</b>
3.1. CONCEPTION DE L'APPLICATION .....	68
3.1.1. Modèle fonctionnel .....	70
3.1.2. Modèle dynamique.....	71
3.1.3. Modèle informationnel.....	72
3.1.3.1. Le Modèle organisationnel de données .....	72
3.1.3.2. Le Modèle organisationnel de traitements.....	73
3.2. VERIFICATION DES MODELES .....	73
<b>4. CONCLUSION.....</b>	<b>75</b>
<b>CHAPITRE IV : MISE EN OEUVRE ET APPLICATION DU CONCEPT .....</b>	<b>77</b>
<b>1. INTRODUCTION .....</b>	<b>77</b>
<b>2. TAG-STEP - OUTIL D'IMPLANTATION.....</b>	<b>78</b>
2.1. ANALYSE LEXICO-GRAMMATICALE (STEP TOOLKIT) .....	79
2.1.1. Outil Xpress .....	79
2.1.2. Outil STEPparse .....	80
2.2. TRAITEMENT DU MODELE DE DONNEES .....	80
2.2.1. Génération de la base de données relationnelle .....	80
2.2.2. Interface SDAI .....	82
2.2.3. Les convertisseurs de fichiers neutres .....	86
2.3. TRAITEMENT DES DONNEES DU MODELE.....	86
2.3.1. Répartition des données.....	86
2.3.2. Base de données relationnelle.....	87
2.3.3. Modèle physique de données sur les étiquettes électroniques .....	87
<b>3. UNE APPLICATION PARTICULIERE : LA CELLULE DE L'AIP LORRAIN .....</b>	<b>88</b>

---

3.1. DESCRIPTION DE LA CELLULE FLEXIBLE D'ASSEMBLAGE .....	89
3.1.1. <i>Partie Opérative de la cellule</i> .....	89
3.1.2. <i>Partie Commande de la cellule</i> .....	90
3.1.3. <i>Pièces et Produits assemblés</i> .....	91
3.2. MODELISATION FONCTIONNELLE DE LA CELLULE.....	91
3.3. MODELISATION DES DONNEES ASSOCIEES AUX PRODUITS .....	93
3.4. FORMALISATION DES REQUETES UTILISATEURS .....	95
3.5. PROTOTYPAGE DE L'APPLICATION PAR SIMULATION SUR SPEX.....	97
<b>4. CONCLUSION.....</b>	<b>98</b>
<b>CONCLUSIONS &amp; PERSPECTIVES</b> .....	<b>100</b>
<b>BIBLIOGRAPHIE</b> .....	<b>105</b>
<b>ACRONYMES</b> .....	<b>114</b>
<b>ANNEXE : LE LANGAGE EXPRESS</b> .....	<b>116</b>

---



# Introduction

---

## 1. Contexte de l'étude

---

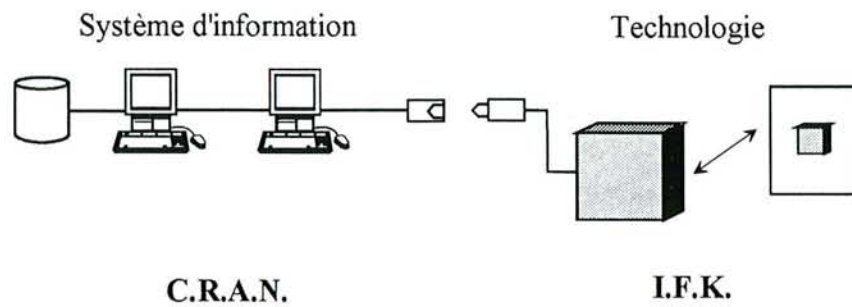
Ce mémoire présente nos travaux menés au sein du Centre de Recherche en Automatique de Nancy, dans le Groupe Génie de la Production, équipe Ingénierie de la Qualité. Il porte sur l'étude et la définition d'un système d'information distribué, en vue de la conduite de procédés et de la gestion de la qualité des produits, par l'intégration des technologies de l'Identification Automatique (IdA) dans les processus manufacturiers. Le terme emblématique de ce projet de recherche est "objets nomades de production".

Suivant cette approche, les objets quels qu'ils soient, propriétaires de leurs informations doivent s'intégrer dans la gestion dynamique du flux d'informations de la cellule, de l'atelier, ou du processus. Ils sont les clients d'une structure mise à leur disposition. Le processus peut ainsi être considéré comme le serveur ou le prestataire de services envers ces objets.

Cette approche innovante, développée par la communauté scientifique internationale pose le problème essentiel de la gestion individualisée des informations associées à chaque objet mobile circulant dans les structures de production.

Nos travaux défendent les idées que les technologies de l'identification automatique exploitant les **étiquettes électroniques** sont des outils adaptés à ce type de gestion d'information et peuvent s'intégrer dans les systèmes d'information existants, sans modification des méthodes de conception.

A cette fin, nous avons établi, début 1993 pour une durée de deux ans, un contrat de partenariat de coopération scientifique avec l'Institut Für Fordertechnik de l'Université de Karlsruhe (IFK) dans le cadre d'un programme Franco-Allemand PROCOPE N°94064. Le CRAN apporte sa maîtrise en productique et l'IFK son savoir faire technologique dans le domaine de l'identification automatique (cf. figure IT.1).



*Figure IT.1 : Intégration des technologies de l'Identification Automatique.*

## 2. Plan du mémoire

Notre mémoire, développé en quatre chapitres suit le plan suivant :

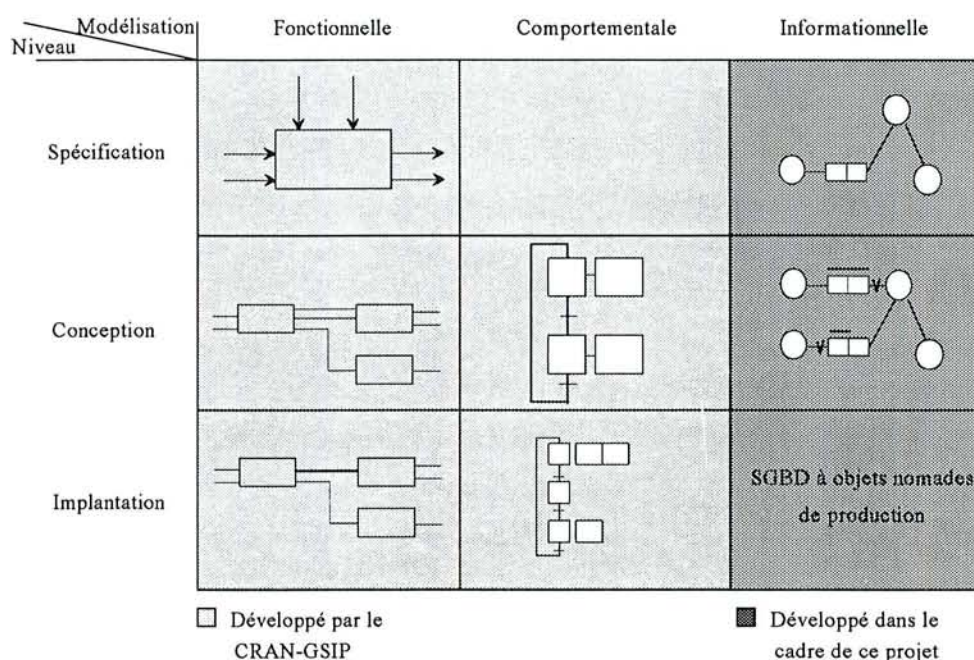
Le chapitre I présente le **concept d'objet nomade** de production et établit un état de l'art des différents travaux sur les approches multi-agents en productique qui considèrent les produits comme des entités autonomes de production. Hors du cadre de la simulation de systèmes bâtis sur ce paradigme, les différentes propositions tendent à considérer **les systèmes d'identification automatique** comme des supports privilégiés d'implantation. Nous montrerons que le passage des spécifications informationnelles de l'objet nomade de production vers sa mise en oeuvre sur de tels supports, bien que nécessaire, se révèle relativement complexe du fait d'un manque de méthodes et d'outils adaptés. Enfin, nous présenterons les **retombées** que peut avoir la manipulation de ce concept **sur les systèmes de production**.

Dans un cadre d'intégration, les spécifications informationnelles de l'objet nomade ne peuvent être dissociées des spécifications établies en conception. Le chapitre II présente le concept de **modèle de produit** ainsi que les différentes méthodes développées, ou en cours de développement, nécessaires à l'intégration des données afin d'éviter toute saisie supplémentaire d'information. Nous détaillons plus particulièrement les modèles, méthodes et outils relatifs au projet **STEP ISO 10303** visant l'intégration, par les données du produit, des fonctions de conception-fabrication.

Dans le chapitre III nous développons notre contribution à la constitution d'un **SGBD<sup>1</sup>** apte à gérer les informations du produit **sur des supports mobiles**, puis nous proposons une méthode de conception de systèmes automatisés de production reposant sur l'utilisation du

<sup>1</sup> Système de Gestion de Bases de Données

concept d'objet nomade de production (cf. figure IT.2). Nous montrons que cette méthode propose des modèles exécutables au sens CIM-OSA, c'est-à-dire qu'elle permet l'utilisation des modèles, successivement affinés, à des fins de prototypage puis d'implantation sur site. La figure IT.2 détaille les aspects couverts par la méthode proposée selon les deux vues fonction, modélisation fonctionnelle et comportementale, et information de CIM-OSA.



*Figure IT.2 : Aspects couverts par la méthode proposée.*

Dans le chapitre IV nous montrons comment nous sommes passé du concept d'objet nomade de production à une mise en oeuvre particulière utilisant les moyens techniques disponibles au CRAN et à l'AIP Lorrain. Nous présentons **l'outil Tag-STEP**, que nous avons développé dans le cadre de ce travail, permettant le traitement du modèle logique de données et la génération automatique de **l'interface d'accès** aux données portées par les produits. Puis nous appliquons la méthode afin de prototyper par simulation le fonctionnement de la **cellule flexible d'assemblage de l'AIP Lorrain**.

Nous terminons en présentant les **perspectives** ouvertes par ces travaux pour une évolution vers une gestion des informations du produit durant la totalité de son cycle de vie, objet d'une proposition de projet **Esprit IV**, ainsi que vers une intégration autour du concept de bus logiciel défini dans le projet Esprit AIT.

# Chapitre I

## Système de production et objets nomades de production

---

*Résumé : Dans ce chapitre nous présentons le concept d'objet nomade de production ainsi que les travaux réalisés par la communauté scientifique à l'origine du concept. Nous montrons dans quelle mesure les technologies de l'identification automatique, et plus particulièrement les technologies par étiquettes électroniques programmables, peuvent être utiles à la mise en oeuvre effective du concept. Nous présentons les travaux du projet Esprit MONOLIN dans ce domaine particulier des technologies de l'information, démontrant que longtemps considérées comme de simples supports d'octets, les étiquettes électroniques tendent à devenir de véritables bases de données en accord avec le concept. Nous montrons également que ce paradigme n'est pas uniquement applicable aux structures pour lesquelles il a été conçu mais est adaptable à un grand nombre de modes d'organisation de production.*

### 1. Introduction

---

La conception et l'implantation d'un système d'information dans un système intégré de production conditionnent de manière significative les capacités de ce dernier à gérer les informations complexes et hétérogènes du produit, du plan de fabrication et de l'état des ressources [BAJ 95].

Notre approche consiste à prendre en compte une partie de ce système d'information, richesse première de l'entreprise, et à proposer une méthode d'implantation de celui-ci (cf. chapitre III). Chaque produit en circulation dans les structures de production emporte avec lui et met à disposition l'ensemble organisé des informations qui le concerne. Cet ensemble d'information est défini de façon suffisant pour que les éléments du système de production puissent effectuer les travaux requis [CHA 91b].

Ce concept, que nous qualifions sous le vocable "objet nomade de production", ouvre des perspectives nouvelles en termes de flexibilité et réactivité des structures de production vis à vis de leur environnement. Les conditions *sine qua non* de la prise en compte industrielle du concept consistent en la définition d'une méthode, d'outils et de moyens technologiques de mise en oeuvre "productique", c'est à dire permettant :

- l'interconnexion des systèmes d'information de conception et de production, qualifiée d'intégration verticale,
- l'accès standardisé au système d'information de production par les composants du processus de fabrication, agissant sur le produit afin de lui conférer de la valeur ajoutée, qualifiée d'intégration horizontale.

## 2. Concept d'Objets Nomades de Production

---

### 2.1. Approche considérée

Le marché industriel actuel est caractérisé par des changements rapides et une évolution chaotique de la production [CIM 94] due au raccourcissement de la durée de vie des produits de consommation. Le respect des objectifs de flexibilité et des contraintes du type zéro stock, zéro délai, a entraîné la définition de structures de fabrication de plus en plus complexes de par les fonctions supportées ainsi que par le niveau de compétence requis pour leur utilisation et leur intégration.

Le concept de CIM (Computer Integrated Manufacturing) définit un cadre méthodologique de structuration des SAP (Système Automatisé de Production) permettant d'atteindre ces objectifs. Toutefois les expériences de mise en application ont fait observer une forte augmentation des coûts de développement et de maintenance de tels systèmes [DEW 90] [HEN 87].

Parmi les axes majeurs de recherche menés dans le domaine du CIM et liés à la structuration des systèmes de production, nous pouvons dégager plus particulièrement les deux thèmes suivants :

- la conception des systèmes d'information permettant :
  - ↳ d'une part d'assurer l'intégrité et la cohérence des données entre les différents services d'une même entreprise,



Dans ce cadre, le rôle dévolu aux machines est de savoir faire et celui dévolu aux produits est de savoir faire faire; ceci caractérise l'approche "objet nomade de production" comparée à l'approche classique illustrée par la figure I.2 [CHA 93]. Le produit possède le QUOI, caractérisé entre autre par sa gamme de fabrication [PAT 95], et la machine le COMMENT, soit la méthode à employer pour réaliser les opérations spécifiées dans la gamme du produit.

Nous n'utilisons toutefois le terme "objet nomade de production" qu'associé à la technologie des étiquettes électroniques programmables que nous considérons comme étant la plus adéquate à la mise en oeuvre de ce concept (chapitre I - § 3.1.2.2).

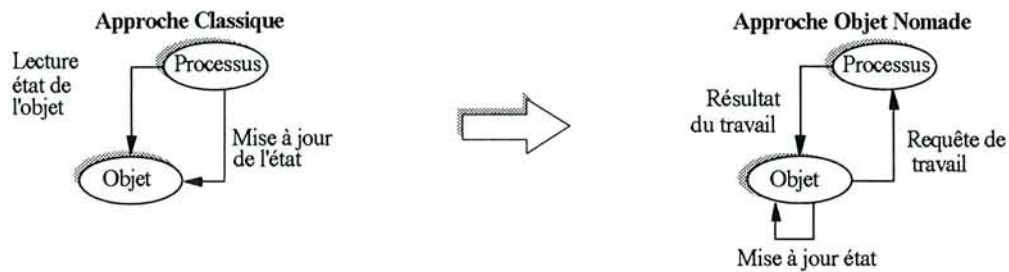


Figure I.2. : De l'approche classique à l'approche objet nomade.

Cette approche autorise à déporter la fonction de gestion des données du produit de la partie commande des machines vers l'entité objet nomade de production. Cela permet donc de simplifier la tâche de programmation des logiciels de commandes sur lesdites machines qui n'ont plus alors qu'à se focaliser sur leur métier.

Ce concept, initialement développé pour les architectures à caractère iso-hiérarchique par Duffie [DUF 86] peut être appliqué à tout type de paradigme de contrôle-commande et sur toute structure matérielle. En effet, si l'on se réfère à un cadre tel que CIM-OSA, seule la vue organisationnelle est modifiée. L'ensemble des fonctions et des informations manipulées par le système va rester globalement identique mais se trouvera conceptuellement et physiquement distribué différemment [PER 93].

## 2.2. Etat de l'art des travaux à l'origine du concept

Avec l'apparition de technologies de communication de plus en plus fiables et rapides [DOI 92], de nombreux chercheurs ont proposé des solutions visant à franchir un pas dans le niveau de répartition des processus de décision. D'abord centrés sur les machines, devenant

entièrement autonomes, ils ont naturellement poursuivi leurs réflexions en considérant les produits comme des entités autonomes, à égalité avec les machines. Ce paradigme, communément appelé multi-agents, prend ses sources tant dans les travaux des producticiens que des informaticiens [BON 92]. Pour illustrer ce concept, nous avons choisi différents exemples de réalisation qui nous sont apparus intéressants de par leurs aspects démonstratifs. Cependant la bibliographie, extrêmement vaste dans ce domaine, s'enrichit rapidement.

### 2.2.1. Le concept iso-hiérarchique issu de l'Université de Madison-Wisconsin

Les travaux sur les architectures iso-hiérarchiques<sup>1</sup> en production, entrepris par Duffie [DUF 86], sont basés sur le modèle portant le nom de réseau contractuel, ou réseau de contrats, dans la terminologie multi-agents. Chaque agent-produit émet un contrat vers l'ensemble des agents-machines afin d'obtenir d'eux la réalisation d'une des quatre actions élémentaires identifiées par une approche systémique : transporter, transformer, stocker et contrôler. La communication entre les machines autonomes et les produits est réalisée au travers d'un tableau noir unique ("*black board*" en anglais), ou boîte aux lettres commune, partagé par tous les agents. Les résultats obtenus par Duffie sur une maquette expérimentale ont démontré qu'une telle solution est réalisable. De plus, la productivité de ce système sans technique d'ordonnancement globale n'est que très faiblement inférieure aux solutions classiques prônées à l'époque, notamment les architectures hiérarchisées [DUF 89]. Par contre, il note des gains notables en durée de développement des logiciels de contrôle-commande des machines ainsi qu'un accroissement de la flexibilité structurelle et de la robustesse du système. L'ajout ou la suppression de machines se réalise sans difficulté majeure. Il en est de même pour la fabrication de produits ayant des gammes non prévues à la conception du système, à la condition expresse que le QUOI du produit ait une correspondance dans le COMMENT des machines.

La mise en oeuvre d'un tel concept nécessite la gestion de l'ensemble des informations associées à chaque produit en considérant le point suivant : plus les informations caractérisant un produit seront finement détaillées selon un axe des spécifications du produit permettant de dériver l'ensemble des traitements, plus le système autorisera un agencement dynamique des opérations de fabrication. Il possédera plus de degrés de liberté lui permettant de réagir aux aléas de fonctionnement.

L'approche proposée par Duffie [DUF 90] a tout d'abord consisté à utiliser un grand nombre de micro-ordinateurs du type PC connectés sur un même réseau local. Ces machines,

---

<sup>1</sup> nommées "heterarchical architectures" par Duffie



équipées d'un système d'exploitation multitâches, activent une tâche permettant d'assurer le lien bi-univoque nécessaire entre le produit physique et l'agent-produit. Les inconvénients majeurs formulés à l'encontre de cette technique de mise en oeuvre proviennent de la charge créée sur le réseau de communication et du nombre d'ordinateurs nécessaires afin de supporter une application dans laquelle un nombre variable de produits est en circulation. Ces inconvénients ont par la suite été partiellement levés par [VEE 93] grâce à l'utilisation de chariots filoguidés informatisés et connectés en réseau.

Avec la même philosophie, Maley [MAL 88] et Upton [UPT 92] proposent de fixer un ordinateur, connecté à un réseau radiofréquence, directement sur chaque palette emportant un produit et qualifient le tout par l'appellation "*The Computerized Product in a Society of Machines*". Nous la traduisons, sans doute en en perdant partiellement le sens, par "le produit informatisé dans une société de machines". Maley conclut toutefois que, pour des raisons de coûts de mise en oeuvre, il est plus réaliste d'utiliser des étiquettes électroniques (§ 3.1.2., chapitre I), approche également envisagée au LAMIH<sup>2</sup> à l'Université de Valenciennes et du Hainaut Cambresis [NOU 94].

### 2.2.2. La métaphore biologique du CAM-I Japon

Devant la carence constatée en flexibilité réelle des cellules de fabrication [OKI 94], le CAM-I Japon a ouvert, en 1985, une voie de recherche originale sous l'influence de deux scientifiques : K. Ueda de l'université de Kobe et N. Okino de l'université de Kyoto. Le terme générique qualifiant cette approche est "*Bionic Manufacturing Systems - Systèmes Manufacturiers Bioniques*".

Le système de production dans sa globalité est conçu par comparaison avec une société (cf. figure I.3), en utilisant le modèle d'organisation hiérarchique de la vie: ADN<sup>3</sup>, enzymes, cellules, organes, individus et finalement écosystèmes [UED 94].

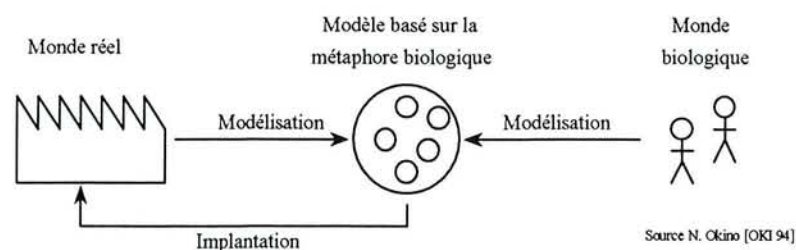


Figure I.3: Construction d'un système basé sur la métaphore biologique.

<sup>2</sup> Laboratoire d'Automatique Industrielle et Humaine

<sup>3</sup> Acide Désoxyribo Nucléique

L'approche adoptée va plus loin que la majorité des propositions effectuées dans les domaines multi-agents qui arrêtent leur comparaison aux niveaux individus et écosystèmes.

Selon N. Okino [OKI 94], le produit se développe, grâce à ses informations, par émission de messages vers les machines de production, les chariots filoguidés, les opérateurs de production, ... comme le ferait un individu à partir de son patrimoine génétique codé dans des chaînes d'ADN.

Qui plus est, le produit en tant qu'entité autonome a une existence non restreinte au système de production mais existe dès la phase de conception. C'est donc au produit de faire appel aux ressources de conception puis aux machines qui vont, pas à pas, modifier sa structure comme des cellules se spécialisent pour donner différents organes.

La vision de K. Ueda est sensiblement la même, si ce n'est qu'elle avance un concept plus abstrait afin d'identifier les produits autonomes : les bons de travaux, ou gammes des produits, évoluant dans l'écosystème de production.

Cependant, il nous faut considérer le fait que les biologistes ne connaissent pas les mécanismes permettant de donner la vie à partir des composants de bases A.G.C.T. (Adénine, Guanine, Cytosine, Thymine). Dans ces conditions, comment pouvons nous créer des systèmes vivants à partir d'un ensemble fini d'éléments quels qu'ils soient : ADN, cellules, organes, individus? Le problème est posé : si nous estimons disposer des briques élémentaires de la vie, combien de temps sera nécessaire pour que d'un ou de plusieurs des arrangements possibles de ces briques émerge la vie fut-elle artificielle?

Les recherches, particulièrement dans le domaine des algorithmes génétiques et des réseaux de neurones, tendent néanmoins à démontrer la possibilité d'émergence de comportements viables non initialement prévus par le concepteur, sans toutefois pouvoir les comparer à l'expression d'une vie artificielle.

### **2.2.3. Le concept de "générateur de jetons"**

Le modèle de commande décentralisée proposé par D. Brun-Picard, à l'origine destiné à la commande de robots, repose sur une entité de contrôle-commande immatérielle appelée I-Machine. Chaque I-Machine est en relation bi-univoque avec une machine ou station réelle, et possède trois caractéristiques : l'action, la décision et la communication. Une I-Machine est un acteur autonome de contrôle-commande capable de réaliser des tâches en coopération avec l'ensemble des autres acteurs.

La commande de ce système distribué est guidée par le produit qui apporte avec lui l'ensemble organisé des éléments des connaissances nécessaires à son élaboration, le tout contenu dans une entité informationnelle appelée "jeton".

Dès lors que l'on considère ce concept de jetons, il apparaît clairement [FER 93], comme l'avait d'ailleurs remarqué Duffie, que le fonctionnement global du système dépend en grande partie de la connaissance contenue dans ceux-ci. Un générateur de jetons, de même niveau fonctionnel et structurel que les I-Machines, permet de réaliser les liens entre activités de conception, de gestion de production et de fabrication (cf. figure I.4).

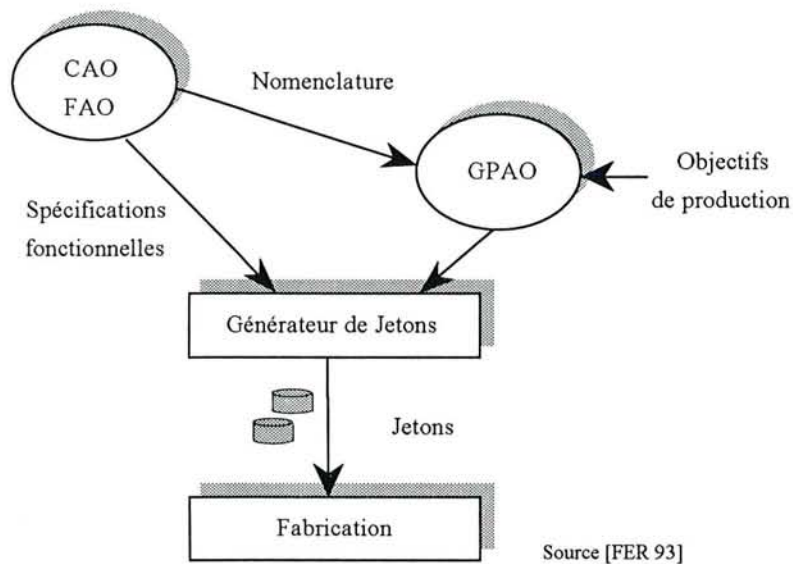


Figure I.4: Le générateur de jetons.

La simulation de systèmes de production bâtis sur les concepts de I-Machines et de Jetons a permis de valider l'approche proposée et de conclure que: *"Le fait d'avoir associé de la connaissance au produit pour guider la fabrication nous apparaît maintenant comme une étape. Il nous semble en effet très prometteur d'aller au delà en associant également des règles de décision, et donc une autonomie, aux produits"* [FER 95].

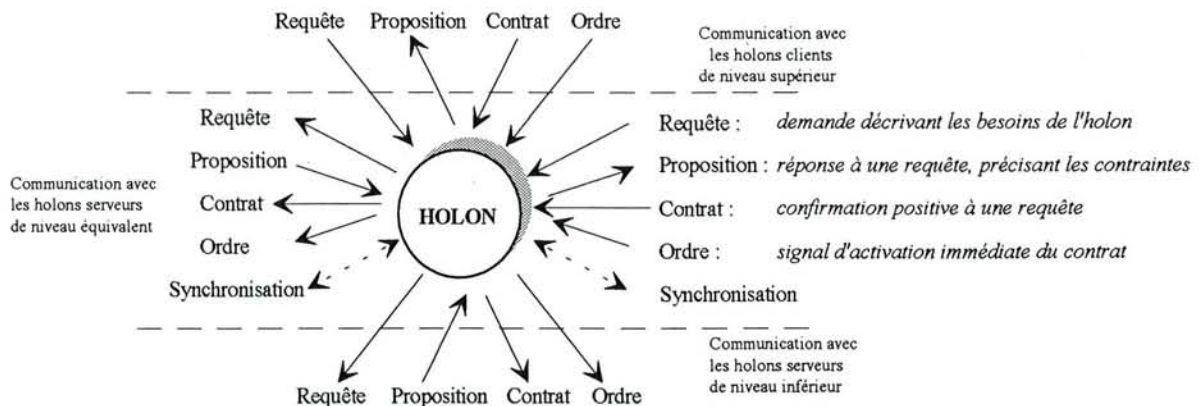
6 Concernant l'aspect de mise en oeuvre effective des jetons, il est proposé d'utiliser des étiquettes électroniques [FER 93]. Ces étiquettes électroniques, présentées comme une réponse technologique possible au problème posé, permettent le stockage et la gestion des informations de production directement sur les produits. De tels systèmes peuvent s'interfacer entre les I-Machines et le produit. La circulation des produits entraîne la distribution des données dans toute la structure de production, créant ainsi un flux d'informations indissociable du flux physique de produits [CHA 91b].

### 2.2.4. Le projet Esprit "Holonc Manufacturing Systems"

Le projet Esprit n° 7754 "Holonc Manufacturing Systems" fait partie d'une des six applications du programme international IMS<sup>4</sup>. Le modèle multi-agents retenu est celui du réseau contractuel [HEI 95]. Plus spécifiquement, l'émission de messages entre agents, appelés holons<sup>5</sup>, se fait par niveau de hiérarchie (cf. figure I.5). Un agent donné n'a la possibilité de communiquer qu'avec un nombre limité d'agents grâce au protocole de contrats défini par Smith [SMI 80].

Comme cela est proposé par K.Ueda (§ 2.2.2), le produit est caractérisé par un holon "order holon" assimilable à un "bon de travaux", considéré comme la meilleure représentation des besoins des clients. Il permet [BON 95]:

- de définir le produit,
- d'emporter ses données de fabrication,
- de gérer la séquence d'opérations nécessaires à l'obtention du but fixé initialement,
- de contenir les objectifs tels que les contraintes temporelles.



Source [HEI 95]

Figure I.5: Les modes de communication entre holons.

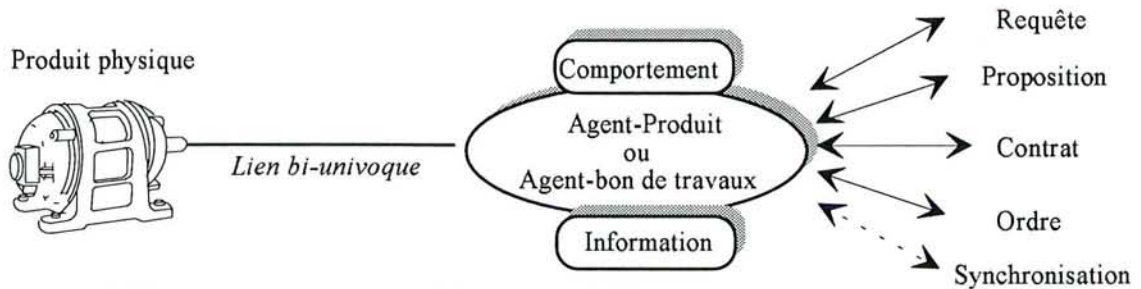
Le cadre de cette expérimentation, Esprit et IMS, démontre l'intérêt scientifique et industriel pour les concepts de distribution totale des fonctions de décision, ainsi que la validité des concepts consistant à considérer le produit, ou de façon plus abstraite, le bon de travaux, comme une entité autonome.

<sup>4</sup> Intelligent Manufacturing Systems

<sup>5</sup> Combinaison du mot Grecque *holos* - le tout - et du suffixe de *proton* suggérant la particule élémentaire

### 3. Apport de l'identification automatique

En dehors d'un cadre de simulation, bon nombre de propositions de mise en oeuvre des concepts de produits-agents ou produits-acteurs requièrent l'utilisation de technologies permettant de garantir une relation stable, dans le temps et dans l'espace, entre le produit physique et son agent-produit correspondant (cf. figure I.6). Sans cette relation, qu'il est envisageable de maintenir de façon purement logicielle, il devient impossible d'associer une gamme, un état, un délai de réalisation et des fonctions de décision à un produit réel.



**Figure I.6 :** Relation entre le produit physique et l'agent-produit.

Ce lien de correspondance est assuré par un marquage unique du produit pouvant être lu ou écrit automatiquement par les éléments composant le système de production. Ces technologies de marquage, dites de l'identification automatique, appartiennent à un vaste domaine d'étude que nous allons partiellement détailler.

#### 3.1. Définition de l'identification automatique

Le CETIM<sup>6</sup> désigne l'Identification Automatique (*IdA*) comme "*l'ensemble des moyens mis en oeuvre pour saisir une information en réduisant, voire supprimant, la part de l'intervention humaine dans cette opération de saisie*".

Les technologies actuellement disponibles sont très variées. Elles se justifient par la diversité des domaines d'utilisation de l'IdA. Parmi celles-ci, nous pouvons citer notamment l'holographie; les cartes perforées; le code à barres; les étiquettes électroniques; les cartes à mémoire; les pistes magnétiques; la reconnaissance des caractères; la reconnaissance vocale; et la reconnaissance de l'existant (forme, couleur, poids,...).

<sup>6</sup> Centre Technique des Industries Mécaniques

Un des grands principes commun entre ces différentes technologies est, comme nous l'avons dit, la relation bi-univoque entre l'identifiant et l'identifié. Tous les utilisateurs, où qu'ils se trouvent et quel que soit le secteur d'activité auquel ils appartiennent, doivent pouvoir être certains qu'il n'existe pour une entité donnée, qu'un seul code et que ce code ne corresponde qu'à cette seule entité [GRO 95]. Cette garantie peut concerner différents secteurs d'activités, néanmoins il est clair qu'au delà des limites de l'entreprise, une telle certitude ne peut être obtenue en dehors du respect de règles normalisées.

Les différentes technologies en question permettent de balayer les besoins allant des deux extrêmes [MON 94] pour lesquelles :

- le système d'identification ne transporte que l'identifiant du produit,
- le système d'identification permet de transporter toutes les données relatives au produit durant une phase considérée de son cycle de vie.

### 3.1.1. La technologie du Code à Barres

La technologie du code à barres a fait l'objet d'un nombre incalculable d'applications et concerne des domaines extrêmement variés tels que le suivi de fabrication [LUX 94], la gestion de stocks et plus encore le domaine de la grande distribution.

Les principaux avantages du code à barres sont [GRO 95] :

- une mise en oeuvre aisée dans des environnements non agressifs puisqu'il s'agit d'imprimer des symboles sur une étiquette, directement sur le produit à identifier, ou encore sur son emballage,
- la lecture à distance est possible sur des objets en mouvement rapide avec des profondeurs de champs variables allant jusqu'à plusieurs mètres,
- la disponibilité de standards internationaux tels le code 39, le code EAN 128, le code 2 parmi 5, permet une utilisation à la fois dans des contextes fermés mais surtout dans des contextes ouverts comme la grande distribution,
- l'existence de codification à deux dimensions offrant une capacité inhabituelle de stockage et une possibilité d'ajout d'informations au fur et à mesure des besoins. Nous pouvons donner en exemple le code PDF417<sup>7</sup> permettant le stockage de 2 Ko de données.

---

<sup>7</sup> Portable Data File

Par contre pour la mise en oeuvre du concept d'objet nomade de production dans le contexte des systèmes manufacturiers et plus particulièrement dans le domaine de la mécanique (assemblage et/ou usinage), le code à barres représente une solution minimaliste. Il correspond toutefois à certains besoins affirmés d'industriels, comme l'identification de caisses Kanban. Dans l'état actuel de la technologie, les problèmes posés par son utilisation [DUN 95] sont :

- les informations qu'il emporte ne sont pas ou peu évolutives. De ce fait, dans la majorité des applications le code à barres est utilisé comme identifiant clef du produit pour accéder aux informations stockées sur une base de données [HEN 87],
- il est rapidement illisible lorsqu'il est souillé et totalement inutilisable lorsqu'il est placé derrière un obstacle. Le taux d'échec en lecture réel ainsi constaté peut nécessiter la présence permanente d'un opérateur,
- les lecteurs code à barres performants sont encore d'un coût prohibitif par rapport aux lecteurs d'étiquettes électroniques car ils nécessitent la gestion simultanée de plusieurs faisceaux lasers.

### 3.1.2. Les technologies des étiquettes électroniques

Les étiquettes électroniques, également appelées "transpondeurs", constituent aujourd'hui une alternative au code à barres dans les domaines manufacturiers, voire dans les domaines à très forte valeur ajoutée.

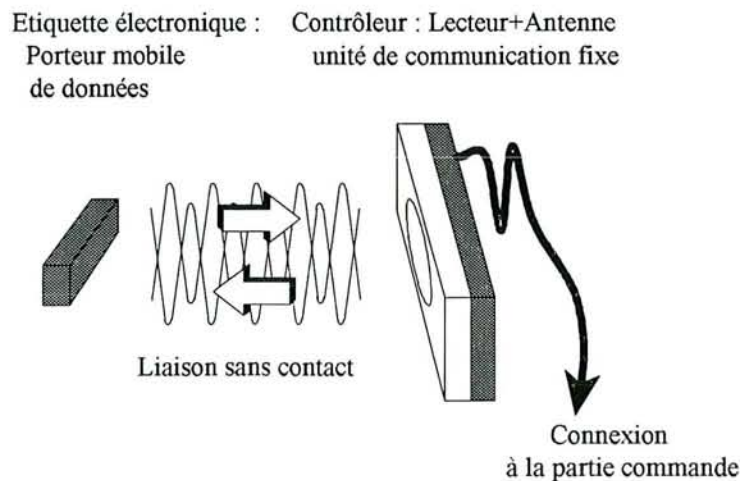


Figure I.7: Vue fonctionnelle d'un système par étiquettes électroniques.

Le principe de fonctionnement repose sur la modulation d'un signal électromagnétique ou optique [DUN 95], en fréquence ou en amplitude, qui est transmis de l'unité de contrôle fixe vers l'étiquette mobile ou inversement (cf. figure I.7).

Pour Dominique SAULNIER, fondateur de la société Statec, figurant parmi les premiers fournisseurs français de systèmes d'IdA, cette technologie, et plus particulièrement les étiquettes programmables, constitue la contribution des électroniciens aux différents concepts de produit-agents [CHA 91b].

En dehors d'applications fermées, le développement de cette technique se heurte aux problèmes du coût unitaire des étiquettes et surtout à l'absence totale de normalisation dans ce domaine obligeant l'utilisateur à employer des étiquettes et des unités de contrôle provenant du même fournisseur. Les différentes discussions portant sur ce sujet tant au sein du consortium Esprit MONOLIN que des instances de l'ISO n'ont pas à ce jour donné de résultat significatif, et ce malgré la motivation des utilisateurs.

Le développement d'applications multi-sectorielles est donc actuellement délicat, car il contraint l'ensemble des intervenants à s'équiper chez le même fournisseur.

### **3.1.2.1. Les étiquettes à code fixe**

Ce type d'étiquette se pose essentiellement en concurrent direct du code à barres car il offre sensiblement les mêmes caractéristiques fonctionnelles: la conservation de données statiques accessibles en lecture seulement. Elles permettent de garantir la relation bi-univoque entre identifiant et identifié tout au long du processus d'utilisation. Le cas d'utilisation le plus anecdotique est sans doute l'identification de pigeons voyageurs lors d'une compétition en Allemagne en Juin 1994 [BUR 94].

L'avantage de ce type de dispositif, comparé au code à barres, provient des possibilités de lecture dans des environnements agressifs dans une gamme de positions relativement importante de l'antenne de communication par rapport à l'étiquette [GRO 95]. Qui plus est, la fiabilité de ces systèmes est très élevée.

Ce type d'étiquette peut soit porter un code fixe inscrit à sa fabrication ou modifié par contact avec des dispositifs particuliers (en général des programmeurs d'EEPROM).

Le prix des étiquettes oblige l'utilisateur à les recycler. Cependant une expérience menée dans un supermarché Anglais [MIL 95] tente de valider leur usage pour la grande distribution. Nous pensons qu'une étiquette vendue à très faible coût aurait toutes les chances de signifier la fin du code à barres classique, d'autant plus que de telles étiquettes peuvent avoir une épaisseur équivalente à une feuille de papier.

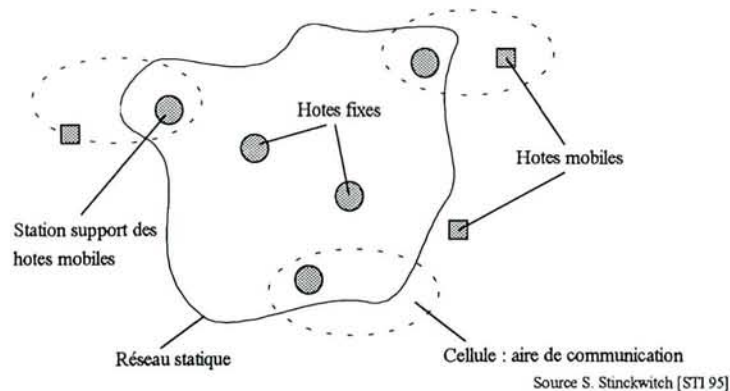


### 3.1.2.2. Les étiquettes programmables

Cette technologie est à notre avis la plus prometteuse de toutes pour les secteurs industriels où les étiquettes peuvent être recyclées (en raison de leur coût) ou vendues en tant que composant du produit, apportant ainsi une plus value à la gestion voire l'utilisation du produit.

Ces étiquettes ont une capacité mémoire allant de quelques bits à plusieurs centaines de Ko. Actuellement la gamme de capacité la plus répandue va de 2 Ko à 32 Ko, mais nous pouvons noter que l'évolution dans ce domaine est très rapide. Lorsque nous avons débuté ce travail en 1991 il était commun d'utiliser des étiquettes ne portant que quelques octets.

Les systèmes à base d'étiquettes électroniques actuellement sur le marché n'autorisent que des lectures/écritures d'octets ou de blocs d'octets, voire récemment de fichiers selon les règles d'un protocole de niveau bits ou octets. Ces fonctionnalités permettent une intégration matérielle simple à la structure de communication mais apportent une limitation quant à leur intégration au système d'information de l'entreprise, où la nature des informations est au delà du bit ou du mot.



**Figure I.8:** Zone de visibilité du produit et de ses informations [STI 95].

Notre approche a pour objectif d'intégrer ces outils d'identification automatique pour la gestion des flux de données du produit et de les transformer en gestionnaire intelligent d'informations structurées conformément aux concepts exposés. L'une des particularités de cette approche provient de la non accessibilité des informations associées au produit lorsque celui-ci n'est pas dans une zone de communication (cf. figure I.8) : il y a perte de visibilité du produit par la partie commande [PER 93]. De ce fait, l'emploi d'étiquettes électroniques comme support de données du produit contraint leurs utilisations à un nombre limité de systèmes de production. Nous donnerons l'exemple des systèmes multi-agents pour lesquels le

temps de négociation entre un produit et les machines, nécessitant la présence des informations du produit, peut avoir une durée supérieure au temps de présence de l'étiquette devant une unité de contrôle. En ce sens, les propositions faites par Maley [MAL 88] et Upton [UPT 92] (paragraphe 2.2.1.) d'associer à chaque produit un ordinateur connecté sur un réseau radiofréquence semble offrir de plus larges perspectives mais posent un problème de coût.

### 3.2. Les apports du projet Esprit MONOLIN

Initié en Juillet 1992 pour une durée de 3 ans, le projet Esprit N° 6903 MONOLIN (**MO**bile **NO**de **L**ogistics and **I**ndustrial **N**etworks) autour d'un consortium composé de Renault, Volvo, Baumer Ident, l'EPFL<sup>8</sup> et le RIT<sup>9</sup> propose des solutions, visant l'intégration des systèmes d'IdA en production et en logistique du transport. Ces travaux, figurant parmi les plus avancés dans ce domaine, ont mis en exergue la nécessité d'interchangeabilité et d'interconnectivité des systèmes d'identification automatique par étiquettes électroniques au travers de fonctions d'accès utilisateur banalisées.

#### 3.2.1. Les résultats

Les partenaires MONOLIN se sont principalement attaché à définir l'interface d'accès aux mémoires électroniques par les applications implantées sur des automates, des commandes numériques, des ordinateurs. Cette interface est appelée EMS-API (**E**scort **M**emory **S**ystem - **A**pplication **P**rogramme **I**nterface).

Les spécifications de l'EMS-API [MON 94] définissent un ensemble de primitives d'accès à une unité de contrôle virtuelle (VCU : **V**irtual **C**ontrol **U**nit) et à une mémoire embarquée virtuelle (VEM : **V**irtual **E**scort **M**emory). L'ensemble de ces deux composants forme le système virtuel d'identification par étiquettes électroniques (VEMS : **V**irtual **E**scort **M**emory **S**ystem). Le terme virtuel a ici la même signification que le terme "machine virtuelle" au sens de la norme MMS<sup>10</sup>, c'est à dire que le détail de l'implantation réelle est entièrement caché aux développeurs d'applications.

---

<sup>8</sup> Ecole Polytechnique Fédérale de Lausanne

<sup>9</sup> Royal Institute of Technology de Suède

<sup>10</sup> Manufacturing Messages Specification ISO 9501

### 3.2.2. L'interface applicative ou EMS-API

L'EMS-API se compose de deux éléments distincts (cf. figure I.9) : le serveur sur l'unité de contrôle d'étiquettes et le client sur le poste applicatif (automates, ordinateurs, contrôleurs de CN, ...). Le développement de l'API sur les deux cibles est à la charge de leurs fabricants respectifs. Il est par exemple réalisé, sous la forme d'un FB EMS-API (Blocs applicatifs fonctionnels) sur les automates SIEMENS. L'objectif est que n'importe quel EMS-API client puisse dialoguer avec n'importe quel EMS-API serveur au travers d'un profil de communication commun.

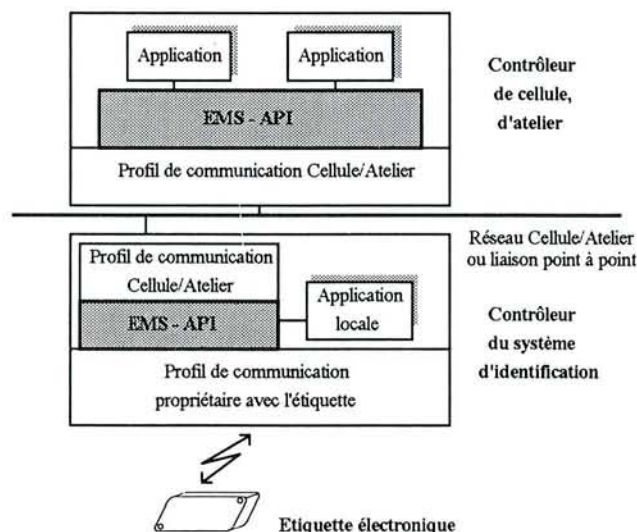


Figure I.9: Distribution de l'EMS-API.

Les primitives de l'EMS-API sont classées en cinq catégories différentes :

- les services de gestion d'accès aux étiquettes permettant d'établir un canal de communication avec une des étiquettes dans le champs de l'unité de contrôle,
- les services de gestion des étiquettes offrant les fonctions de test, d'initialisation ou encore de lecture des caractéristiques de celles-ci (capacité, nombre de blocs mémoires ou de fichiers alloués),
- les services de gestion mémoire des étiquettes tels que la création, la destruction de blocs mémoires ou de fichiers, ainsi que les opérations de lecture et d'écriture associées,
- les services de gestion des unités de contrôle comme l'initialisation, l'exécution de programmes préalablement enregistrés (équivalent à la primitive de démarrage d'un objet ProgramInvocation en MMS),

- et enfin des services auxiliaires portant sur les unités de contrôle tels que la gestion des entrées/sorties tout ou rien et la mise à jour de l'horloge interne.

### 3.2.3. Aspects réseaux de communication

Dans l'esprit de MONOLIN, tout comme dans le nôtre, les unités de contrôle ainsi que les étiquettes sont des éléments d'un réseau. Par exemple, une étude a été réalisée afin de mettre en oeuvre l'API au dessus de la couche application réseau MMS et plus particulièrement au dessus des profils réseaux MMS proposés sur FIP, PROFIBUS et INTERBUS-S [MON 94].

De même une MIB unité de contrôle et une MIB étiquette (**M**anagement **I**nformation **B**ase) ont été étudiés afin de fournir au gestionnaire du réseau les informations dont il pourrait avoir besoin. Cette MIB contient les informations désormais classiques en gestion des systèmes et des réseaux [LEC 94] telles que les caractéristiques du fabricant, le numéro de série, la date d'installation mais également des informations particulières au domaine comme le nombre d'antennes, la capacité mémoire, l'état de la pile, le nombre de transactions effectuées, etc.

## 4. Influence sur les systèmes de production

---

### 4.1. Positionnement dans un contexte CIM

Les systèmes d'*IdA* à base d'étiquettes électroniques programmables sont des équipements dont le rôle est la gestion et le transport de données, fussent-elles élémentaires, sur des supports liés à un objet. D'un point de vue fonctionnel, ils trouvent leur place comme ressources du système d'information de la cellule ou de l'atelier [BEN 94a].

#### 4.1.1. De l'architecture classique ...

L'architecture classique d'un système automatisé de production à objets discrets est rappelée par la figure I.11 :

- les capteurs assurent la détection de la présence de la pièce ou du produit,

- les informations issues des capteurs sont transmises aux parties commandes qui alimentent les parties opératives ainsi que le système de supervision et de suivi des produits,
- le flux d'information associé au flux produit est géré par le système de supervision nécessitant :
  - ↳ la connexion de toutes les stations de travail à ce système,
  - ↳ un logiciel complexe pour gérer le synchronisme entre le flux physique et le flux d'information. La complexité du logiciel est liée non pas aux algorithmes mis en oeuvre mais au nombre et à la diversité des sources d'information permettant le suivi des produits (capteurs de détection de la présence des produits, postes de travail, postes de qualification, ...).

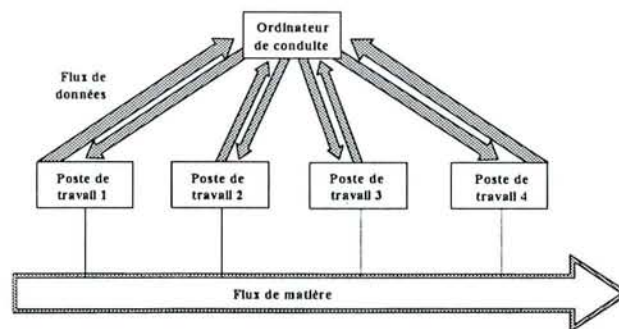


Figure I.11: Architecture sans étiquettes électroniques

La synchronisation entre les **deux flux distincts**, physique et d'information, est une opération complexe lorsque la structure physique du système de production permet un routage dynamique des produits [UDO 92]. Cette opération reste également lourde à mettre en oeuvre sur des lignes de production réalisant un produit unique à multiples variantes.

La centralisation des données tend à forcer l'ingénieur à simplifier son modèle de données du produit afin de ne pas surcharger le réseau de communication et la base de données même distribuée sur différents supports.

#### 4.1.2. ... à une architecture à objets nomades

Les modifications induites sur l'architecture classique par l'emploi du concept d'objet nomade de production, emportant avec lui **l'ensemble organisé des connaissances** nécessaires à sa réalisation, sont mises en évidence par la figure I.12:

- les informations nécessaires au traitement du flux sont véhiculées par les produits eux-même: il y a **synchronisation naturelle** du flux physique et du flux d'information associé,
- lorsque la totalité des informations associées aux produits sont portées par eux-même, la connexion des postes à une base de données et à un logiciel de suivi des produits peuvent devenir inutiles, sauf pour le premier et le dernier poste de la structure de production considérée.

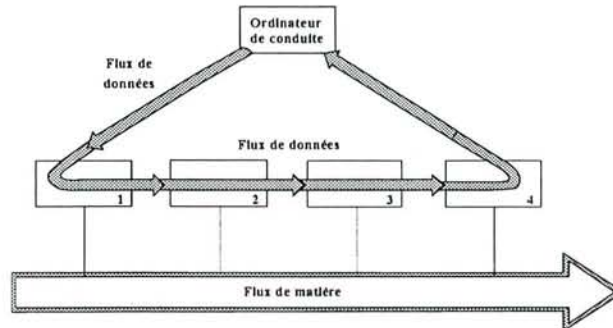


Figure I.12: Architecture idéale avec étiquettes électroniques programmables.

A la condition que les informations transportées par les produits soient jugées suffisantes pour éviter toute connexion des postes à un réseau de communication, les résultats obtenus par la mise en oeuvre de cette architecture sont :

- une autonomie du poste de travail,
- une simplification des architectures physique et logique du système : le coût est donc réduit dans des proportions bien supérieures aux coûts supplémentaires dus à l'emploi des étiquettes électroniques [HEN 87].
- une diminution des coûts de fonctionnement du fait d'une sensibilité réduite aux pannes, mais aussi du fait de la diminution du nombre de composants, de logiciels, et d'échanges d'informations entre les niveaux.

## 4.2. Apports en gestion de la qualité

### 4.2.1. Gestion de la qualité des produits

Très tôt dans l'analyse ayant amené à la réalisation du système de pilotage de la cellule auto-contrôlée d'usinage du CRAN [MRT 91] [DOI 92], l'impact sur la gestion de la qualité de ce

que nous appelons désormais "objet nomade de production" avait été identifié: *"Le concept de pièces intelligentes est intéressant à plusieurs points de vue car il permet, par exemple, de réaliser un auto-contrôle sur la qualité de réalisation des pièces suite à une opération de qualification"*.

Nous pensons, en effet, que la gestion centralisée de la quantité d'information nécessaire à cet auto-contrôle [COU 91] n'est réaliste que dans un système où le nombre de pièces ou produits en circulation est relativement faible. Ce type de gestion sur une base de données multi-utilisateurs accessible via un réseau, réalisé dans le cadre de la mise en oeuvre de la cellule auto-contrôlée du CRAN, a rapidement démontré ses limites en termes de charge imposée à la base, d'autant plus que le modèle de données du produit obtenu était complexe.

#### **4.2.2. Traçabilité des produits**

Selon les normes ISO 8402 et AFNOR NFX 50.120 la traçabilité *"est l'aptitude à retrouver l'historique, l'utilisation ou la localisation d'un article ou d'une activité, au moyen d'une identification enregistrée"* (cf. figure I.13). Cette définition générale, dont a hérité la définition donnée dans la série de normes ISO 9000, regroupe à la fois les problèmes de suivi du produit en phase de conception, ISO 9001, et en phase de fabrication, ISO 9002. En fin de ligne de production nous devons, si nous désirons suivre une démarche de traçabilité totale, être capable de fournir, pour chaque produit, les informations couvrant l'ensemble de son cycle de production, telles que:

- les références des pièces utilisées dans le processus d'assemblage tels que leur numéro de lot, leur fournisseur, ...,
- le numéro de lot, s'il existe, du produit considéré,
- les date et heure de passage sur l'ensemble des machines d'usinage ou d'assemblage, assurant l'identification des causes de dérives détectées par exemple par un outil SPC<sup>11</sup> sur un poste de qualification,
- le résultat de l'ensemble des opérations de qualification.

La remarque précédente, donnée dans le paragraphe 4.2.1, sur la gestion centralisée de ces informations s'applique également à ce cas précis.

---

<sup>11</sup> Statistical Process Control traduit en Maîtrise Statistique des Procédés

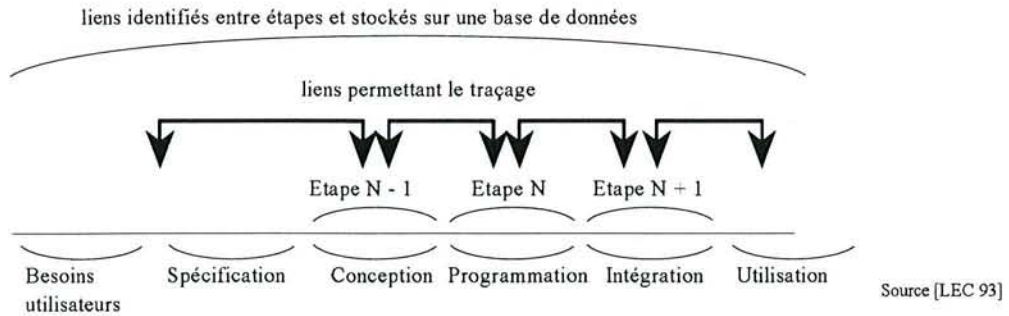


Figure I.13: Traçabilité en conception.

L'emploi du concept d'objet nomade de production, et de la technologie de mise en oeuvre adéquate, offre des perspectives importantes à l'ingénieur qualité, disposant ainsi d'un outil de choix pour **la collecte sur site d'informations destinées à la traçabilité** et le suivi individualisé de l'ensemble des produits. Nous noterons que ces informations collectées et stockées sur les étiquettes électroniques mobiles doivent être, en fin de production, introduite sur une base de données permettant la traçabilité.

Toutefois, cette gestion impose la définition d'outils permettant le traitement d'une quantité d'information non habituelle à ce jour.

### 4.3. Impact sur le système d'information

Grâce aux technologies de l'identification automatique (du code à barres aux étiquettes électroniques), le produit porte de manière naturelle un ensemble d'informations. Cet ensemble correspond à tout ou partie des informations nécessaires à son traitement par la partie commande de l'appareil industriel.

F. Pereyrol [PER 93] qualifie les informations portées physiquement par le produit d'informations fournies par la partie opérative à la partie commande. Cela implique que plus la partie opérative sera capable de fournir de l'information sur le produit, moins il sera nécessaire de reconstruire logiciellement l'état du produit au niveau de la partie commande.

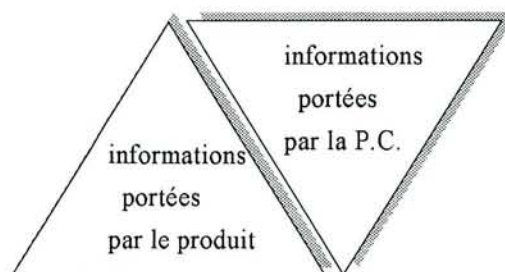


Figure I.14: Situation de l'information du produit [PER 93].



En conséquence, plus le produit va emporter d'information avec lui, moins il sera nécessaire d'en faire une modélisation ou une reconstruction au niveau de la partie commande (cf. figure I.14). Cette approche nécessite toutefois de considérer une entité capable de gérer et mettre à disposition les informations liées aux produits. Une démarche de mise en oeuvre d'une telle entité permet de séparer les informations liées aux produits et les mécanismes d'exécution des opérations de transport, d'assemblage ou d'usinage. Les logiciels de commande n'ont plus alors qu'à s'attacher à leur métier.

## 5. Conclusion

---

Le concept de produit-agent, ou tout du moins de produit gestionnaire de ses propres informations, semble désormais être considéré comme une solution viable et à fort potentiel par la communauté scientifique internationale comme le démontre, entre autre, l'existence du projet IMS et Esprit "Holonc Manufacturing Systems".

La mise en oeuvre de ce concept peut passer, selon de nombreux auteurs, par l'utilisation des étiquettes électroniques. A notre connaissance, hormis les propositions faites dans le cadre du projet Esprit MONOLIN, le seul moyen d'utilisation de ces systèmes reste la lecture de bits ou d'octets tandis que les besoins tendent vers des services équivalents à ceux offert par un SGBD.

Dans le cadre strict du CIM, il n'est pas envisageable de considérer le problème de la gestion des données du produit dans les ateliers de fabrication sans prendre en compte les phases amonts de son cycle de vie. Ces phases amonts de conception sont susceptibles de fournir les informations élémentaires de définition du produit lui même [PAT 95] (caractéristiques, gamme, entités d'usinage, ...), et constituent le réservoir dans lequel va s'approvisionner le "générateur de jetons" décrit dans le paragraphe 2.2.3. On peut alors parler d'intégration verticale.

Nous allons donc détailler les méthodes et outils d'échange de données éligibles pour réaliser la migration des données de conception vers les activités de fabrication.

# Chapitre II

## Intégration par le modèle du produit

---

***Résumé :** Nous abordons le problème de l'échange d'informations entre les différentes activités participant au cycle de vie d'un produit. Le point de départ du discours consiste à prendre en considération l'hétérogénéité des différents outils de X.A.O. comme un état de fait. L'aspect qui nous concerne le plus, mais qui n'est pas restrictif, est l'échange des informations attenantes aux produits entre les niveaux conception et production. De façon plus détaillée, nous examinons les différents éléments constitutifs aux travaux de normalisation de STEP (Standard for Transfert and Exchange of Product data model). Nous examinons également les différentes solutions de mise en oeuvre de STEP sur des architectures intégrant de systèmes d'information, et plus particulièrement les propositions émanant des résultats apportés par différents projets européens Esprit en établissant un parallèle avec les outils de l'intégration forte tels PCTE et SHOOD.*

### 1. Introduction

---

L'interfaçage entre les niveaux conception et production peut être supporté par le système d'information, assurant la migration des données de conception et de représentation des produits, des outils, des machines, ... en vue de leurs utilisations par les fonctions de conduite, de maintenance et de gestion de la qualité [MRT 91].

Le système d'information de production ne nécessite pas la connaissance de la totalité des spécifications contenues dans les différents modèles de conception : *"La cellule, ou structure de production, n'a besoin que du contenu des modèles indépendamment de la totalité de la sémantique représentée par les relations, les sous-typages et les contraintes du modèle CAO"* [BAJ 95]. L'objectif n'est pas de recréer un modèle de données exhaustif des objets considérés afin de le fournir à l'activité de production, mais bien de déterminer une structure simplifiée permettant de récupérer les éléments, définis en conception, pertinents vis à vis de la structure de production considérée (cellule, atelier, ...).

Ce constat général s'applique aux produits manufacturiers circulant dans les structures de production, dont le modèle doit toutefois être adapté pour récupérer les données de définition.

Le modèle de données du produit utilisé par l'activité de production, qu'il soit générique ou spécifique, est l'élément permettant le stockage et la distribution de la connaissance du produit vers l'appareil de production.

L'intégration verticale entre activités de conception et activités de fabrication permet d'assurer la cohérence et l'homogénéité entre l'objet conçu, voire imaginé, et l'objet réalisé. Dans notre cas, l'interprétation des modèles de définition permet la migration des informations du produit nécessaires à la fabrication sur les supports étiquettes électroniques afin d'être utilisées par l'appareil de production. De même, l'interprétation du modèle de données de l'objet nomade de production permet de récupérer les informations contenues sur ces étiquettes en fin de production vers les activités de conception, créant ainsi une boucle de rétroaction.

Cette tâche sera d'autant plus aisée si les différents modèles sont exprimés dans un langage commun et perçus comme des vues ou spécialisations du même objet. Cette intégration par les données fait actuellement l'objet de nombreux travaux de recherche dans la communauté scientifique internationale. Deux points de vues s'affrontent, l'intégration forte, la plus séduisante mais également la plus complexe à mettre en oeuvre, et l'intégration faible par échange de données, vision plus pragmatique, représentée par les travaux sur les normes internationales STEP.

## 2. Le produit: moteur de l'entreprise

D'un point de vue systémique, toute entreprise industrielle peut être vue comme un objet dont le flux entrant est constitué par de l'information (état du marché, commandes clients) et le flux sortant par des bénéfices d'exploitation [GAR 92]. Les autres flux, tels les matières premières ou les impôts peuvent alors être considérés comme des flux auxquels l'entreprise ne peut que se soumettre.

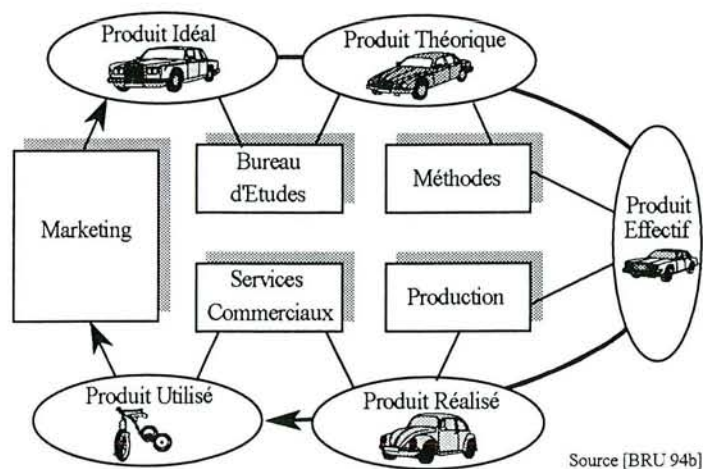


Figure II.1. : Cycle de vie des produits manufacturiers.

Ces bénéfices d'exploitation, et valeur ajoutée au produit dans ses différentes étapes, sont les résultats émergeant du travail fourni par l'entreprise. Le produit, dans son acception la plus large, est donc l'élément clef de toute organisation industrielle, qu'il soit matériel (un véhicule, un four) ou immatériel (un logiciel, un service), fabriqué ou en cours de conception.

L'environnement concurrentiel, la mondialisation des échanges et la chute de la consommation obligent les entreprises à fournir des produits de qualité croissante pour des coûts les plus faibles possibles [CIM 94] et en réponse rapide aux attentes des consommateurs tout en conservant une valeur ajoutée la plus élevée possible. Finalement, l'entreprise est dans l'obligation de trouver un juste équilibre entre des objectifs souvent contradictoires.

Afin de maintenir ces contraintes à un niveau acceptable, les entreprises doivent maîtriser pleinement leurs processus de conception, de fabrication et de vente des produits.

Cette maîtrise passe par une organisation efficace des activités intervenant dans le cycle de vie du produit, autour d'un système d'information orienté vers les produits [TIC 95], permettant d'éviter les pertes d'information d'une phase du cycle de vie à l'autre (cf. figure II.1).

L'élément fondamental est donc la maîtrise des technologies avancées de l'information permettant d'intégrer et de coordonner les différentes activités durant le cycle de vie de développement d'un produit [CIM 94] comme l'illustre la figure II.2 [KLA 93].

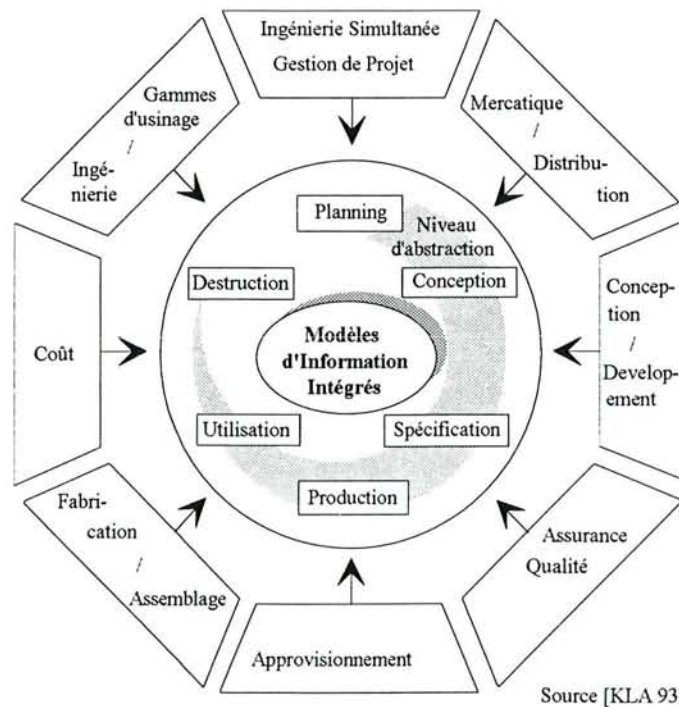


Figure II.2 : Activités concernées par le cycle de vie du produit.

"L'objectif majeur de ces besoins en technologies de l'information est la modélisation du produit, qui crée un réservoir d'informations des données complètes du produit afin de supporter les différentes activités à différentes phases de développement du produit" (Traduit de [KLA 93]).

### 3. Démarche d'intégration par les données

Le processus de conception-réalisation-maintenance d'un produit manufacturier nécessite l'utilisation de différentes méthodes et outils travaillant chacun selon une vue distincte du produit considéré. L'emploi ordonné, séquentiel ou concourant, de ces outils permet de répondre au besoin global. Le partage ou l'échange d'informations entre les intervenants (hommes et outils) est donc le point essentiel de la démarche d'intégration.

Comme proposé dans [WAK 93], cette démarche nécessite de disposer de (cf. figure II.3):

- une interface de présentation de données unique telle que X11-Motif, réalisant le lien entre outils et utilisateurs,
- une architecture système adaptée et modulaire reposant sur des structures de communication à hautes performances tels que Ethernet, FIP, ...,
- un accès uniforme aux données au travers de standards reconnus comme MMS, SQL ou les standards d'échanges.

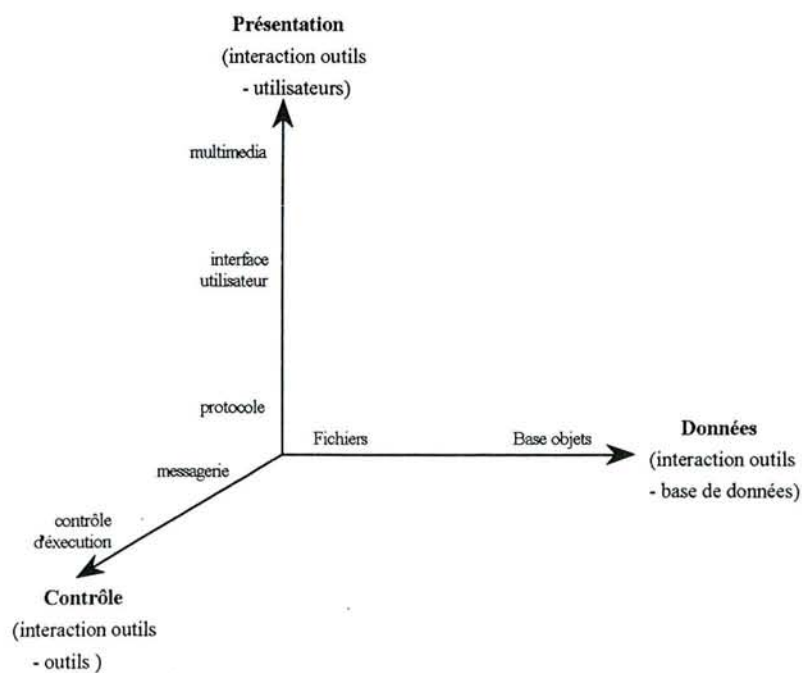


Figure II.3: Dimensions de l'intégration selon [WAK 93].

### 3.1.Historique

La première étape pour une intégration par les données consiste à offrir des outils faiblement connectés, ayant chacun leur propre système d'information et communiquant entre eux par l'intermédiaire d'une interface commune. Ce type d'échanges, s'il est préférable à des outils totalement déconnectés, imposant un transfert manuel, duplique les données et laisse aux outils la charge de vérifier la cohérence des informations qu'ils génèrent. Ce type d'échange est coûteux du fait de la multiplicité d'interfaces nécessaires en fonction du nombre d'outils connectés. Ces interfaces spécifiques deux à deux, n'offrent que très peu de possibilités de réutilisation et croissent avec le nombre d'outils suivant le facteur  $n*(n-1)$  où  $n$  est le nombre d'outils. Cette démarche est par exemple celle proposée au CRAN [PAN 91] en CAO d'automatique pour intégrer les activités de conception préliminaire et de spécification détaillée entre les outils ORCHIS<sup>1</sup> et SPEX<sup>2</sup>.

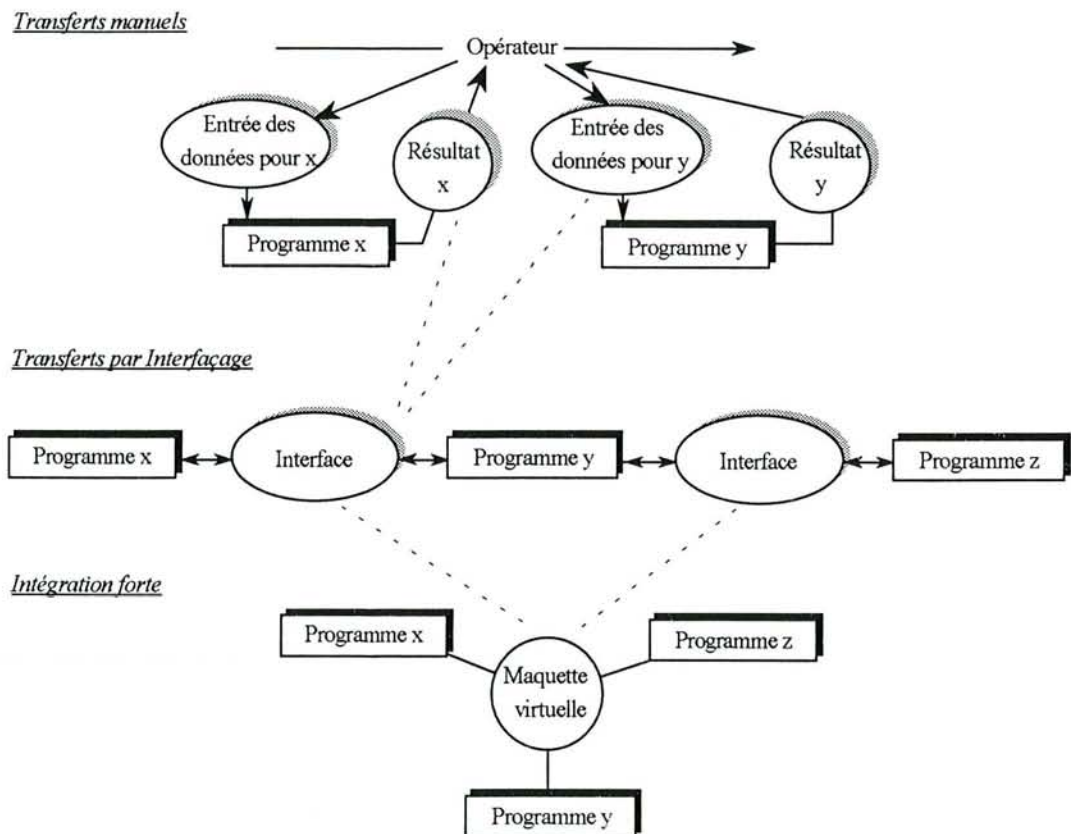


Figure II.4: Du transfert manuel à l'intégration.

<sup>1</sup> Outil d'analyse fonctionnelle au formalisme SADT : © TNI

<sup>2</sup> Outil de spécifications comportementales de composants d'automatismes : © CRAN-TNI-SPIE TRINDEL

Lorsque l'échange est réalisé au travers d'un modèle de données d'un domaine d'étude particulier, nous parlons d'intégration par les données. Cette intégration est dite faible lorsque le transfert de données est réalisé par échange de fichiers. Elle est dite forte lorsque chaque outil concerné par le domaine de modélisation utilise les données générées par les autres applications sur une base de données commune et partageable. Les outils travaillent alors sur la même maquette virtuelle avec des points de vues différents. Cette solution ne supprime pas pour autant la base de données locale à l'outil afin de gérer les données permettant d'optimiser certaines fonctions; par exemple la fonction d'affichage graphique sur des logiciels de CAO.

En soit, les démarches d'intégration faible et forte possèdent un point commun: le modèle de données du domaine d'application. Les méthodes d'échanges de données restent foncièrement différentes et l'intégration forte nécessite de casser les structures logicielles classiques pour les connecter à une base commune [LOM 94].

Selon notre point de vue, l'intégration faible par échange de fichiers de données peut être vue comme une phase de prototypage pragmatique ou de faisabilité de l'intégration forte.

Base-PTA [PTA 92] et CRISTAL<sup>3</sup> [LHO 94] sont typiquement des projets ayant cette démarche dans le cadre des applications de CAO d'automatismes. L'approche pragmatique ascendante de CRISTAL consiste à modéliser, dans un premier temps, les systèmes d'information respectifs de deux outils particuliers: SPEX et CADEPA<sup>4</sup>. Le domaine couvert va de la spécification comportementale d'un automatisme par SPEX (Grafset, Ladder, équations booléennes) à la programmation d'automates par CADEPA. Le transfert de données entre les deux outils s'effectue au travers d'un fichier neutre pour être par la suite, réalisé par l'intermédiaire d'une base de données partagées.

*In fine* l'intégration forte peut être réalisée de façon propriétaire: le concepteur du logiciel offre la possibilité d'accéder à son système d'information, qu'il considère alors comme une référence pour le domaine considéré. Cette offre est largement répandue sur les outils de CAO mécaniques tels que Catia, Cadds, Euclid. Elle est d'autant plus simple à proposer que ces outils sont bâtis autour d'une base de données techniques. Nous pouvons citer, par exemple, l'intégration de l'outil Cadds avec les outils de simulation robotique et métrologie CimStation de Silma au travers de l'interface propriétaire de ComputerVision CV-Doors.

D'un point de vue pragmatique, il nous faut considérer **l'hétérogénéité des différents outils de X.A.O. comme état de fait** et qu'il est hors de question de la remettre en cause par une

---

<sup>3</sup> © CRAN-EURILOR

<sup>4</sup> © EURILOR

quelconque réécriture de logiciels [LOM 94], la solution propriétaire ne peut ainsi être considérée que comme un pis-aller.

### 3.2. Rôle du modèle

De manière analogue à la définition des modèles de données dans les SGBD [GAR 85], nous pouvons appréhender le modèle descriptif du produit de trois façons [GAR 91] :

- la ou les représentations externes, orientées vers l'utilisateur avec le double objectif: d'une part de présentation compréhensible par l'utilisateur et d'autre part de gestion de contraintes de confidentialité, de droit d'accès, ...,
- la représentation logique de l'objet : cette représentation correspond au modèle conceptuel dans la terminologie Bases de Données. Cette représentation logique **est le modèle**. Elle comprend des données et des algorithmes permettant de reconnaître la structure de l'objet, ses composants, ses paramètres,
- la représentation programmée correspondant à l'agrégation des modèles logique et physique au sens Base de Données. Il s'agit ici de la réalisation informatique liée à la technologie.

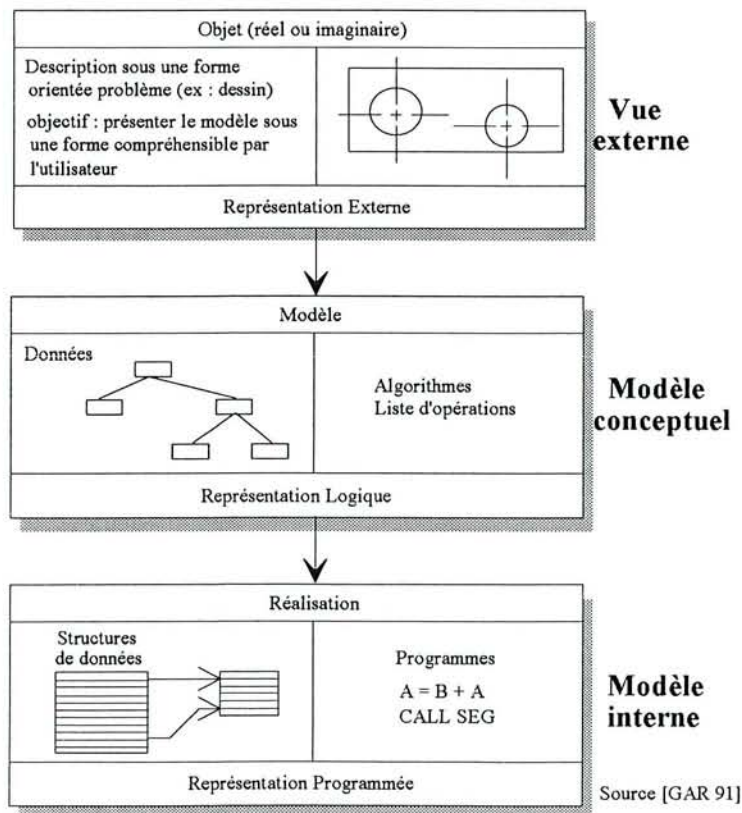


Figure II.5: Représentation d'un objet.



### 3.3. La normalisation internationale pierre angulaire de l'édifice

L'intégration faible ou forte par les données nécessite, comme nous l'avons vu, la spécification d'un modèle de données apte à stocker les données représentatives de l'objet modélisé. Or le nombre d'outils concernés par l'objet en question est relativement élevé (CAO mécanique, calcul de structures, génération de gamme d'usinage, génération de programmes d'usinage, centre d'usinage, ...). Il est donc relativement improbable, voire **non souhaitable**, qu'un seul fournisseur couvre l'ensemble de ces activités avec ses outils.

Le problème se pose donc en termes d'hétérogénéité de matériels, de logiciels et de fournisseurs. De ce fait, le ou les modèles de données permettant de couvrir tout ou partie du cycle de vie d'un produit doivent être publics et stables dans le temps, afin de pouvoir être utilisés par l'ensemble des fournisseurs. Les deux adjectifs public et stable peuvent être considérés comme des qualificatifs du terme "normalisé". Dans ce domaine, les travaux sont fort nombreux tant au niveau national qu'au niveau international.

La figure II.6 présente les efforts de standardisation dans le domaine de l'échange des données du produit dans l'ordre chronologique de leur apparition. Le standard STEP<sup>5</sup>, que nous allons détailler, est à la fois le plus récent, le plus ambitieux, et le plus consensuel de tous.

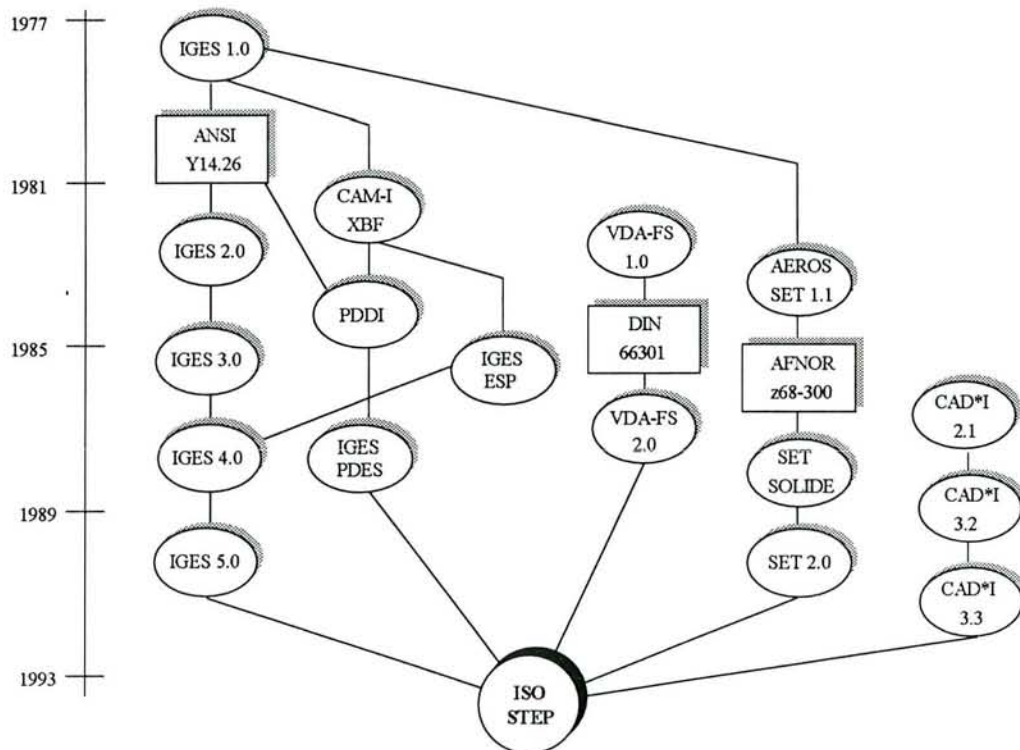


Figure II.6: Historique des standards d'échange de données produits [HAE 90].

<sup>5</sup> Standard for Transfert and Exchange of Product data model

#### 4. La série de normes STEP (ISO 10303)

Le projet STEP a débuté en 1983 au sein du sous-comité ISO/TC184/SC4 "Système d'automatisation industrielle et intégration" et rassemble aujourd'hui près d'une trentaine de pays et environ 300 personnes [AFN 94]. STEP est non seulement considéré comme un projet de normes, mais également comme un projet de recherche et développement par un bon nombre des participants à cet effort de normalisation.

En effet, ce projet ne cherche pas seulement à normaliser des modèles de données mais bien à définir des modèles de références, par extraction des invariants conceptuels, contenus dans les modèles particuliers, connus par les experts du domaine d'application considéré. En ce sens, les modèles normalisés, appelés protocoles d'applications, peuvent être considérés comme des éléments partiels voire génériques de la vue information selon l'approche CIM-OSA.

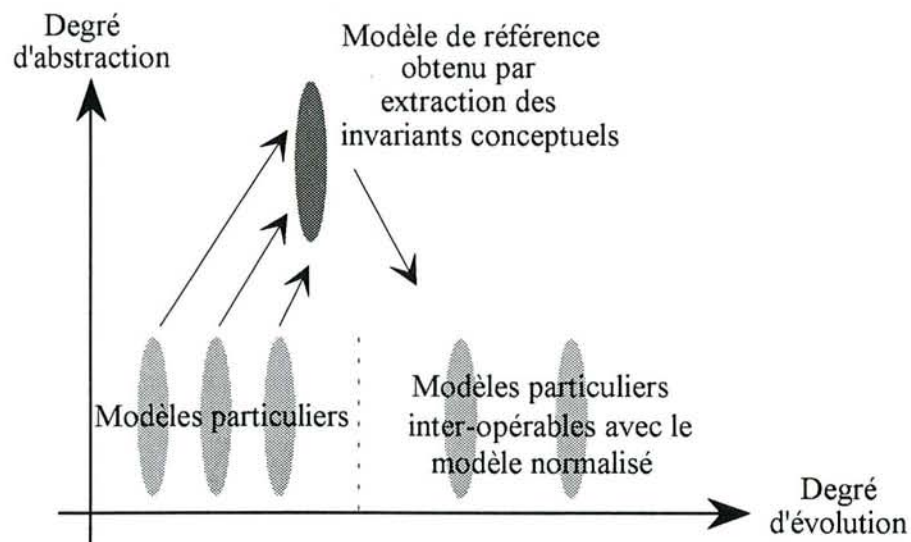


Figure II.7: Construction d'un modèle de référence.

##### 4.1.Objectifs

Face à la problématique posée par le CIME<sup>6</sup>, STEP constitue une réponse technologique normalisée et internationale de nouvelle génération. Il s'agit d'avantage de définir un modèle de données de définition de produit pour toute l'entreprise, plutôt qu'un simple format d'échange [VIE 94]. Ce modèle devra permettre, à terme, la représentation de toutes les informations associées à la définition des produits de l'entreprise tout au long de leur cycle de vie. La nature de cette description permet non seulement l'échange comme avec un format neutre classique (IGES, VDA) mais en plus, elle peut être utilisée comme référence pour réaliser une base de

<sup>6</sup> Computer Integrated Manufacturing and Engineering

données partageable. Ceci pour permettre l'intégration des applications mises en place dans les entreprises et la pratique de l'ingénierie simultanée [CZE 94].

Ces objectifs sont par ailleurs clairement définis dans la partie 1 "Principes fondamentaux" de la série de normes 10303 :

*"ISO 10303 est une série de Normes Internationales pour la représentation et l'échange informatique de données des produits. L'objectif est de fournir les mécanismes capables de décrire les données des produits tout au long de leur cycle de vie, indépendamment de tout système particulier. La nature de cette description la rend utilisable non seulement pour l'échange neutre de données, mais aussi comme la base pour l'implantation de systèmes d'information produits partagés ainsi que pour l'archivage"* (Traduit de la Norme 10303-1).

## 4.2.Structuration

L'organisation nécessaire à l'aboutissement des travaux menés dans le projet STEP est structurée autour des sept aires fonctionnelles suivantes (cf. figure II.8) [WIL 92]:

- les principes fondamentaux et guides d'utilisation de la série de normes,
- les méthodes de description contenant les spécifications des langages de modélisation tels que EXPRESS et ses extensions futures,
- les méthodes de mise en oeuvre telles que le format du fichier d'échange neutre ou les spécifications de l'interface SDAI d'accès aux bases de données STEP,
- les spécifications de tests de la conformité des outils,
- les ressources intégrées contenant les briques de base des modèles d'informations telles que les entités géométriques, de tolérancement, ou plus simplement le format d'une date: nous dirons ici les éléments génériques de la vue information CIM-OSA,
- les modèles ou protocoles d'information applicatifs définissant les modèles de données pour un domaine particulier (dessin technique, géométrie surfacique, électricité, construction navale, ...),
- les tests abstraits associés aux protocoles d'applications.

Un nombre important de projets européens Esprit, Eureka ou Brite sont impliqués dans la définition de protocoles applications. Parmi ceux-ci nous pouvons citer :

- le projet Esprit N° 2032 COMBINE dans le domaine de l'architecture,
- le projet Esprit N° 6040 PRODEX dans le domaine de la CAO,
- le projet Esprit N° 6041 MARITIME dans le domaine de la construction navale.

Par ailleurs, l'appel d'offre CCE<sup>7</sup> Esprit IV encourage explicitement les projets pouvant contribuer à STEP ou utilisant STEP dans les domaines où la modélisation produit s'avère nécessaire.

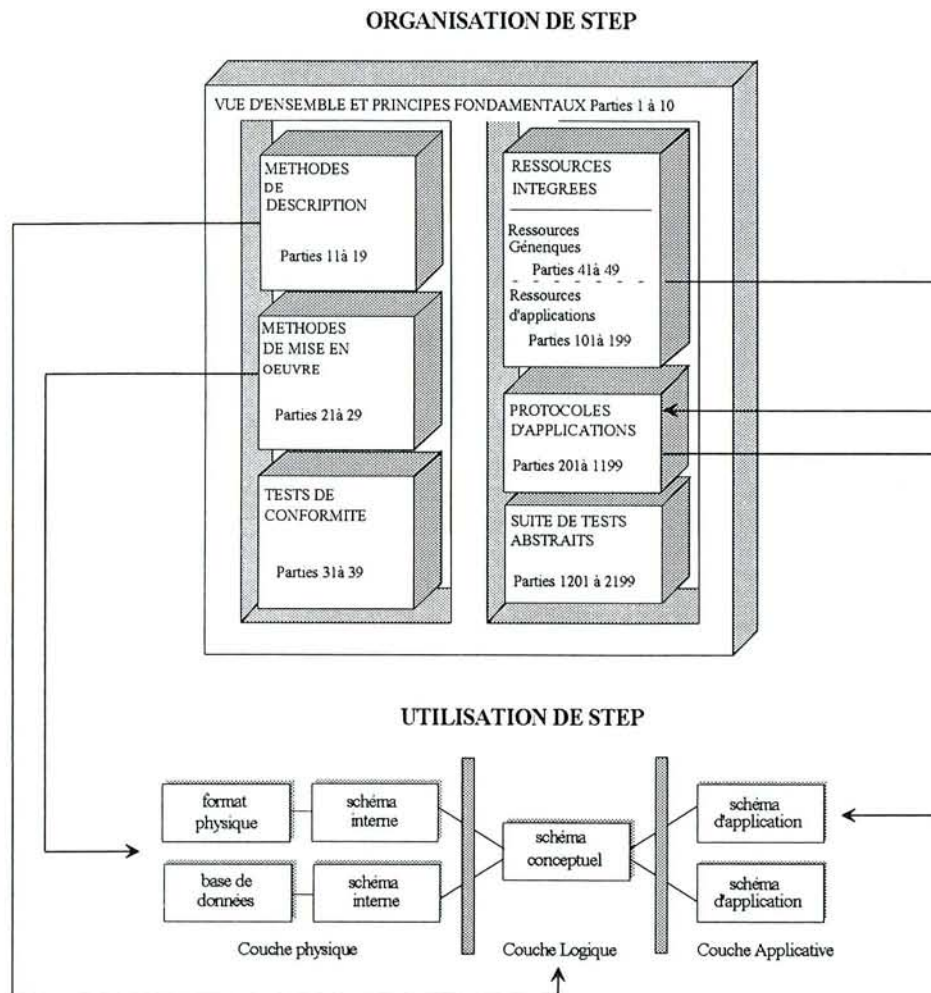


Figure II.8: Structure de la norme ISO 10303.

### 4.3. Les outils de base (Parties 11 à 29)

#### 4.3.1. Le langage de modélisation EXPRESS 10303-11

EXPRESS [ISO 1] est le résultat de la partie "méthodes de description" du projet STEP. C'est un langage de définition des modèles d'information décrivant l'ensemble des étapes du cycle de vie d'un produit. EXPRESS est un langage de modélisation qui s'est inspiré de certains

<sup>7</sup> Commission des Communautés Européennes

langages de programmation structurés comme Pascal ou C pour la syntaxe des instructions et la définition des données, et de langages orientés objets pour introduire des concepts tels que l'héritage, le polymorphisme ou l'orthogonalité.

EXPRESS possède deux formats distincts (cf. figure II.9) :

- le langage formel, avec une notation lexicale et une syntaxe définie par une grammaire formelle dite LALR(1). Ceci rend le modèle, que nous pouvons qualifier de modèle logique, défini dans ce langage, interprétable par une machine. En ce sens, EXPRESS est à rapprocher du langage formel MSS (Manufacturing System Specification) défini par F. Vernadat [VER 94].
- le langage EXPRESS-G, permettant la construction graphique du modèle de données. Il permet une description de la structure d'un modèle EXPRESS à l'aide de symboles graphiques pour une représentation des relations, des cardinalités et des types de données. Si ce format aide à avoir une vue globale d'un modèle, il reste insuffisant pour décrire les règles et les contraintes imposées aux données de façon aussi complète qu'avec EXPRESS. On peut également noter que les formalismes IDEF1X ou NIAM sont bien plus souvent utilisés que le formalisme EXPRESS-G lors du développement de protocoles d'applications STEP. Ceci étant dû en partie, selon notre avis, à un manque d'outils adaptés.

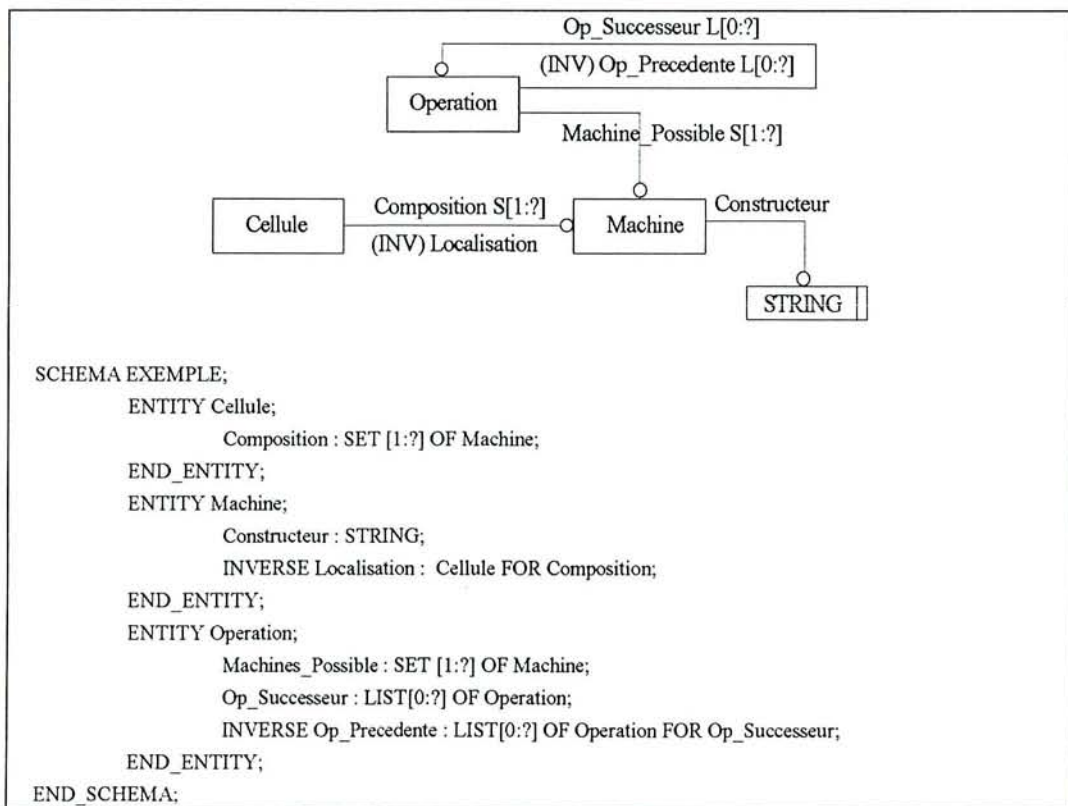


Figure II.9: Exemple de modèles EXPRESS-G et EXPRESS équivalents.

#### 4.3.1.1. Les éléments du langage

Le langage EXPRESS est doté d'éléments permettant de décomposer, classer et caractériser des objets d'un domaine du discours. Un modèle EXPRESS est constitué d'un ou plusieurs modules qui définissent la notion de SCHEMA caractérisant un sous modèle autonome. Les modules ne peuvent être ni imbriqués, ni hiérarchisés, par contre ils sont capables de communiquer et de partager des informations ou des processus de traitement de données.

Pour modéliser un métier ou une partie du monde réel, le langage EXPRESS utilise les concepts suivants :

- **entité** pour modéliser une classe d'objets possédant des propriétés communes,
- **attribut** pour modéliser les propriétés des objets,
- **relation** pour modéliser les liens entre les objets,
- **contrainte** pour modéliser le comportement des objets.

Un descriptif du langage est donné en annexe afin de permettre la compréhension des différents modèles présentés dans les chapitres III et IV.

#### 4.3.1.2. Cadre d'utilisation et outils

Le langage EXPRESS est aujourd'hui utilisé au delà du projet STEP [DIV 95] [MOL 95] et il est proposé pour la modélisation de différents processus dans divers métiers : bâtiment, pétrochimie, gestion, ... Il est, entre autre, utilisé comme outil de spécification dans le cadre du projet Esprit III PRIAM (**P**re-normative **R**equirement for **I**ntelligent **A**ctuation and **M**esure). On voit également apparaître de plus en plus de mises en oeuvre informatiques basées sur ce langage (cf. figure II.10) : éditeurs, compilateurs, traducteurs vers d'autres langages tels que le C++ ainsi que des générateurs de schémas de bases de données relationnelles et orientées objets (ONTOS, ...).

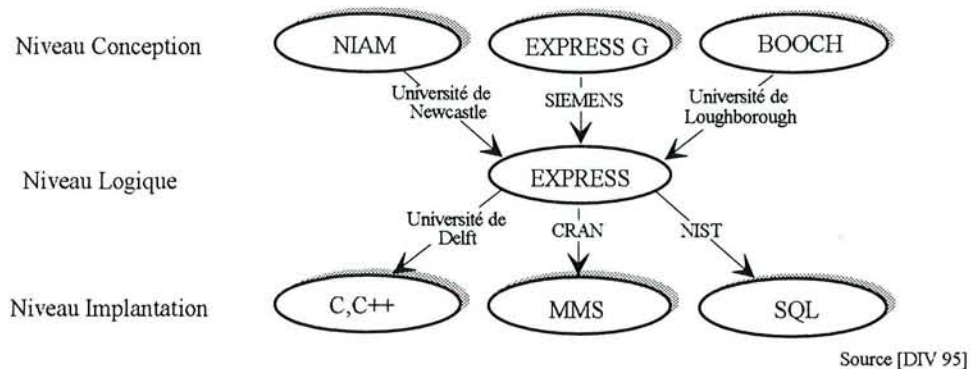


Figure II.10: EXPRESS, langage pivot.

L'attrait d'EXPRESS provient, à notre sens, des caractéristiques suivantes :

- il est normalisé ISO, ceci assurant une certaine pérennité aux travaux de recherche et développement,
- il est informatiquement utilisable par des outils tels que Lex et Yacc (outils standards de la communauté informatique) car totalement défini par une grammaire formelle décrite dans la norme,
- un certain nombre d'outils développés dans des laboratoires universitaires sont disponibles (FEDEX développé par le National Institute of Standards and Technology, XPRESS développé par l'université de Delft en Hollande, ...) et l'on assiste également à l'arrivée récente mais massive d'outils commerciaux,
- il est doté d'une puissance de modélisation importante,
- la traduction de modèles Entité-Relation Etendu Binaire vers EXPRESS a été prouvée, permettant la conservation des savoir-faire formalisés.

EXPRESS peut être utilisé à des fins de construction du modèle logique de données, indépendamment du formalisme retenu pour la modélisation au niveau conceptuel, et donc des habitudes de modélisation du concepteur. Dans notre cas, il permet de servir de base à la génération du modèle de données des produits à implanter sur les objets nomades.

#### *4.3.1.3. Les travaux en cours*

Des extensions et des langages dérivés du langage EXPRESS apparaissent. En particulier des travaux en cours tendent à définir une version 2, dans un cadre normatif, compatible avec la première version et permettant d'une part de prendre en considération la gestion des instances des entités et d'autre part d'associer un comportement à ces instances sous la forme de méthodes (au sens orienté-objet). Le langage EXPRESS-C<sup>8</sup> défini dans le projet Esprit N° 6876 PISA sert actuellement de base de travail à la spécification d'EXPRESS-V2. Aucune discussion n'est toutefois engagée afin d'enrichir la sémantique des relations inter-objets au sens des diverses méthodes d'analyse objet.

De même, un langage permettant de spécifier les opérations de passage des données d'un modèle vers un autre modèle est en cours d'étude [BAI 94]. Ce langage, EXPRESS-M [CLA 92], permet de formaliser la transcription des données d'un protocole d'applications vers un autre. Cette opération est couramment appelée conversion de STEP vers STEP [HAE 90] mais

---

<sup>8</sup> Enregistré à l'ISO sous la référence TC184/SC4/WG5/N202

peut *a priori* être utilisée pour transférer des données d'un modèle STEP vers le modèle de données particulier d'une application, à la condition de posséder une description en langage EXPRESS du modèle de ladite application. Dans notre cas particulier, EXPRESS-M offre la possibilité de formaliser la migration des occurrences des modèles utilisés en conception vers le modèle de données de l'objet nomade utilisé en production.

#### 4.3.2. L'échange par fichiers neutres 10303-21

Le format de fichier neutre STEP [ISO 2] (cf. figure II.11), contrairement à IGES, est basé sur un modèle d'information décrit en EXPRESS qui permet d'assurer la cohérence du fichier au travers de règles d'écriture. Tout comme pour le langage EXPRESS, une grammaire formelle a été définie pour rendre la syntaxe du fichier neutre non ambiguë et compilable par un outil logiciel.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(
('Piece dans une cellule'),
'An implementation level');
FILE_NAME('Demo','1992-08-31 17:23:48',
('F. Chaxel','E-mail: chaxel@cran.u-nancy.fr')
('CRAN-GGP','Faculte des sciences BP 239','54506 Vandoeuvre Cedex','France'),
'STEP Version 1.0','STEP Physical File testing','An originating system','from TNO-Bouw');
FILE_SCHEMA(('ESSAI'));
ENDSEC;
DATA;
#1=Cellule((#2,#3));
#2=Machine ('Num');
#3=Machine ('Charly Robot');
#4=Operation ((#2,#3),());
ENDSEC;
END-ISO-10303-21;
```

**Figure II.11 :** Exemple de fichier neutre STEP, instance du modèle donné figure II.9.

STEP définit également des tests de validation permettant de vérifier la conformité de l'implantation des pré et post-processeurs acceptant ou générant des fichiers neutres basés sur un protocole d'applications particulier (Partie 34 de la norme). Ce point permet aux programmeurs de valider leurs outils logiciels *a priori* mais, contrairement aux précautions prises par le Pentagone avec le langage Ada, la vérification de conformité par un organisme externe n'est aucunement prévue. Le risque de voir des outils logiciels (CAO, gammes automatiques, ...) générant des fichiers neutres, "estampillés STEP", non conformes n'est donc pas exclu.



Un fichier neutre contient une instance d'un modèle EXPRESS décrit dans un protocole application. Ce fichier neutre est généré par un pré-processeur ayant à vérifier l'intégrité des informations, exprimée sous forme de règles dans le modèle EXPRESS. Le post-processeur, en se basant sur le schéma EXPRESS, crée les occurrences du modèle de données dans le système d'information de l'application [NEU 94]. Le post-processeur est également chargé de vérifier les règles spécifiées dans le modèle. Cette démarche est illustrée par la figure II.12.

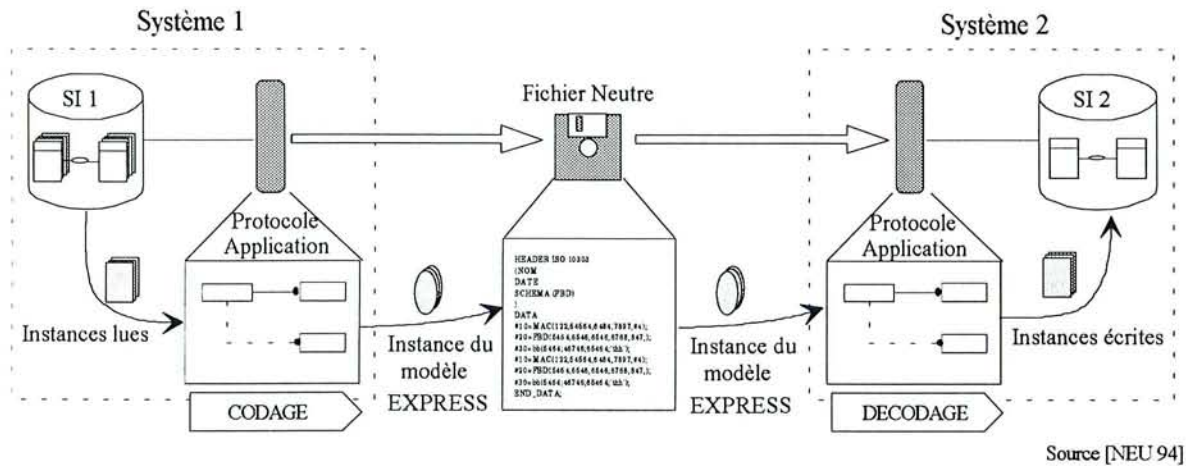


Figure II.12: Transfert de données par fichier neutre STEP.

Si les modèles EXPRESS implantés dans les deux processeurs ne sont pas identiques, la correspondance entre chaque entité peut être réalisée par un outil intermédiaire appliquant les règles de conversion définies dans un modèle en langage EXPRESS-M.

#### 4.3.3. L'interface d'accès aux données SDAI 10303-22

Les mécanismes de base pour une intégration forte, autour d'une base de données commune à l'ensemble des outils, consistent à offrir une interface ouverte permettant l'accès aux données sur un ou plusieurs supports réseaux et multi-utilisateurs.

Dans STEP, cette interface d'accès aux données d'un protocole d'applications est nommée SDAI (Simple Data Access Interface [ISO 3]). Elle permet aux développeurs de manipuler les données d'instances d'un modèle EXPRESS sans avoir à connaître la manière dont elles sont effectivement écrites ou lues sur le support informatique. Cette approche caractérise les travaux réalisés sur les gestionnaires de bases de données depuis les années soixante [GAR 85] basés sur l'indépendance données-traitements [PIC 90].

L'interface SDAI est définie hors de tout contexte d'implantation et nécessite donc l'utilisation d'une base de données relationnelle ou objet voire de fichiers neutres comme supports physiques réels de stockage de données persistantes.

La centaine d'instructions d'accès à une base de données SDAI permettent [MOR 92a]:

- l'accès et la manipulation de n'importe quelle instance du modèle,
- l'accès au dictionnaire de données (la définition des entités),
- la vérification des contraintes d'intégrité des données par rapport au modèle,
- l'accès à plusieurs supports de données par une même application.

Les reproches majeurs formulés à l'encontre de SDAI sont la non prise en considération de la gestion de la sécurité et de l'accès multi-utilisateurs à un même support de données [SCH 93]. Ce problème est, selon les auteurs, à la charge de la base de données réellement utilisée, comme cela est d'ailleurs le cas pour le langage SQL.

Mais SDAI n'est toujours pas dans un état stable et ces reproches pourraient fort bien être levés dans le futur.

#### **4.4. Vers un standard universel ?**

STEP réunit en sa faveur des atouts indiscutables tels que son langage de description EXPRESS et l'adhésion quasi générale à son développement des organismes de normalisation nationaux et internationaux. De très nombreux projets de recherche visent à contribuer à son évolution ou à son utilisation. En France, le GOSET<sup>9</sup> fondé par d'importants consommateurs de produits de CAO tels que l'Aérospatiale, la DGA<sup>10</sup>, Renault, ... ne travaille plus sur son objectif originel de promotion du standard national SET mais sur STEP. L'année 1995 voit l'arrivée en masse d'interfaces STEP sur les systèmes de CAO implantant le protocole d'applications 201 pour le dessin technique et le protocole d'applications 203 pour la géométrie surfacique B-rep, validés entre autre par Boeing dans le cadre du projet AeroSTEP. C'est le cas de CV-Doors de Computervision, de CAA de Dassault ou de Cas.Cade de Matra Datavision.

Ce succès grandissant auprès des industriels démontre que STEP peut prétendre au titre de standard des standards pour l'échange des données de définition CAO des produits. La société

---

<sup>9</sup> Groupement Opérationnel sur SET

<sup>10</sup> Direction Générale pour l'Armement

Spacial Technology parle d'ailleurs de "**bus géométrique**" entre les logiciels de CAO. Les deux protocoles d'applications STEP actuels remplacent donc IGES, SET ou VDA.

Mais l'objectif de STEP est bel et bien de **couvrir l'ensemble du cycle de vie d'un produit**. Le chemin à parcourir est donc encore long. Christophe Viel, animateur du groupe Afnor-représentation des données, fait d'ailleurs remarquer que STEP court un réel danger à cause même de son ambition. En effet, le nombre de champs d'applications qu'il vise suscite une multiplication des protocoles d'applications, qui peuvent posséder de nombreuses zones de recouvrement parfois non homogènes [VIE 94].

La question posée est "Faudra-t-il créer des interfaces entre protocoles applications - solution ayant donnée naissance à EXPRESS-M - ou même équiper les logiciels de tous les protocoles touchant le domaine qu'ils traitent, soit presque tous les protocoles pour les logiciels de CAO". C'est donc en termes de cohérence et d'inter-opérabilité des protocoles d'applications que se pose le problème. Les choix stratégiques à venir ainsi que le renforcement de l'implication des utilisateurs peuvent permettre d'éviter ces pièges.

## **5. Architectures intégrantes STEP**

---

Les travaux réalisés dans STEP définissent les modèles de données - pour l'échange entre systèmes - aux niveaux conceptuel et logique. Aucune directive n'est donnée quant à la mise en oeuvre effective de ces échanges. De nombreux projets, tant universitaires qu'industriels, ont tenté d'apporter des réponses à cette question et proposent :

- des outils logiciels reposant sur une méthode de mise en oeuvre stricte qui permettent la constitution d'un environnement de travail,
- des architectures prototypes de systèmes d'information basées sur STEP.

### **5.1. Les outils logiciels**

Dans un contexte de gestion de données techniques nous pouvons identifier les principaux besoins en [JAR 94] :

- outils de lecture et de création de fichier neutre STEP. Ces outils permettent le transfert électronique de documents entre sites,
- outils d'interfaçage SDAI; c'est à dire des bibliothèques d'accès aux supports de stockage de données,

- outils permettant de vérifier la cohérence des données, exprimée sous forme de contraintes dans le modèle EXPRESS considéré,
- outils de visualisation dépendant du modèle de données mais indépendant de la ou des applications générant ces données.

L'environnement de travail (cf. figure II.13) sera complété par des outils logiciels spécifiques effectuant du traitement sur les données issues de la CAO, de la FAO, ...

L'ensemble de ces outils est actuellement disponible auprès du NIST (National Institute of Standards and Technology), du GOSET, de l'université de Delft (Hollande), ou des entreprises CadLab, CIMIO, STEPTools (ST-DEVELOPER est sans doute l'outil le plus complet du marché), ... et un recensement complet est régulièrement fait par Peter Wilson [WIL 94]. Le nombre d'outils disponibles et la diversité de leur provenance est significatif de l'intérêt scientifique et des enjeux économiques de ces travaux.

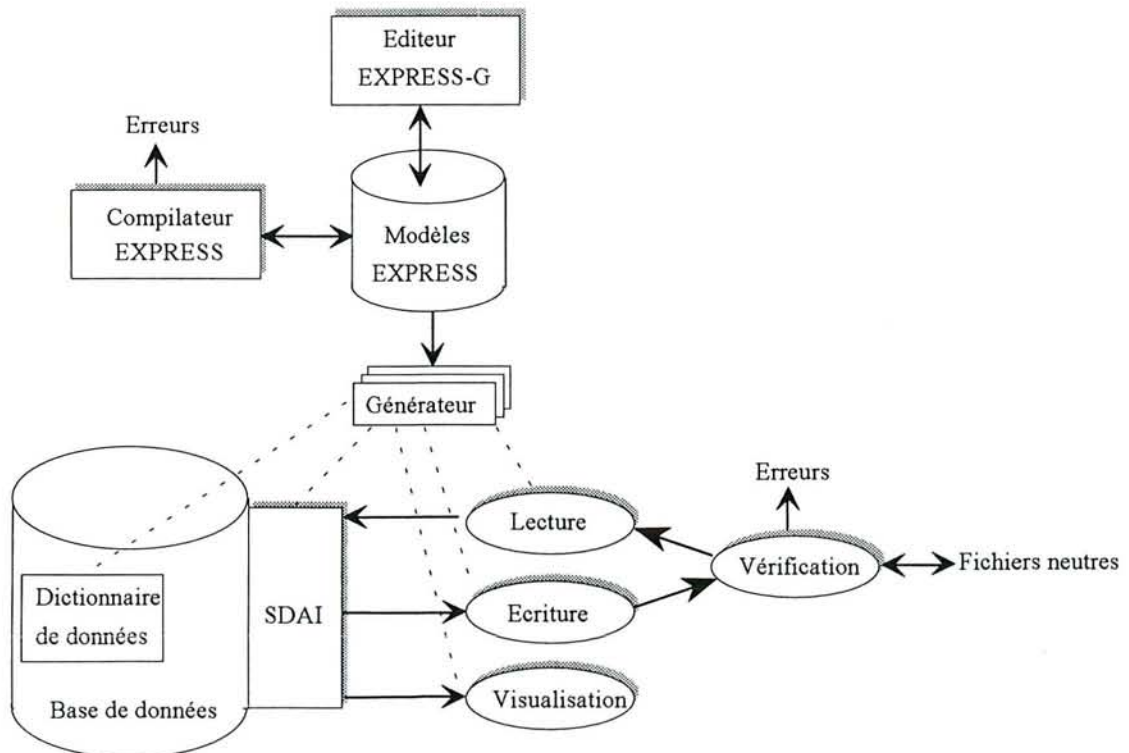


Figure II.13: Environnement de travail EXPRESS-STEP.

## 5.2. Les plates-formes prototypes

Les différentes plates-formes définies et utilisées dans différents projets Esprit constituent des environnements de travail très proches des recommandations faites par l'ECMA<sup>11</sup> visant à

<sup>11</sup> European Computer Manufacturers Association

caractériser les principaux composants d'un atelier d'ingénierie intégré [LUT 94]. On peut également comparer ces solutions avec celles proposées par PCTE [WAK 93] ou SHOOD [NGU 94], utilisé entre autre par l'équipe Conception Intégrée du laboratoire 3S [TIC 94], de par le nombre important de points communs identifiables tels que :

- ce sont des plates-formes supports pour le développement d'applications, architecturées autour d'un SGBD et permettant l'intégration de logiciels de CAO et de FAO,
- elles permettent la gestion complète des données partagées par plusieurs applications travaillant sur le même projet (interrogation, mise à jour, sauvegarde, gestion des versions, gestion automatique de la migration des données d'un schéma vers un autre) [NGU 94],

Nous noterons toutefois que SHOOD et PCTE sont des plates-formes orientées conception, c'est à dire permettant la gestion de données évolutives ayants parfois des états incohérents avec le modèle, alors que les plates-formes STEP, reposant sur les protocoles d'applications STEP, sont orientés échanges de données techniques stables et sans incohérence.

*Mais "si l'on admet qu'une Information diffère d'une Donnée par les divers degrés de transformation qu'elle est à même de subir, il est alors plus aisé de comprendre en quoi la maîtrise d'un système d'Information dépasse largement celle de gestion physique de fichiers informatiques, même à l'aide de Systèmes de Gestion de Base de Données, en impliquant le contenu sémantique des informations manipulées" [MOR 92b].*

Il faut donc garder à l'esprit que ces plates-formes ne sont que des réceptacles technologiques **nécessaires** au savoir faire "productique". Le savoir faire en soit s'exprime par la sémantique des données. L'ensemble des modèles et de leurs relations géré par la plate-forme permettent de traduire cette sémantique.

Ces plates-formes expérimentales (cf. figure II.14), inspirées pour la plupart des résultats du projet Esprit N° 2010 NEUTRABAS [BRU 91] [FER 91] semblant *a priori* cristalliser les développements, sont utilisées dans la liste non restrictive des projets :

- Esprit N° 2202 CIM-PLATO "**CIM - PL**anning **TO**olbox" [OSO 94],
- Esprit N° 2165 IMPACT "**I**ntegrated **M**odelling of **P**roducts and **P**rocesses using **A**dvanced **C**omputer **T**echnologies" [GIE 91] [MEI 91],
- Esprit N° 6040 PRODEX "**P**ROduct model **E**Xchange using STEP",
- Esprit N° 6876 PISA "**P**latform for **I**nformation-**S**haring by **C**IME **A**pplications",

- Esprit N° 7280 ATLAS "Architecture, methodology and Tools for computer-integrated of LARge Scale engineering projects",
- Esprit N° 7704 AIT "Advanced Information Technology in design and manufacturing". La très forte implication des grands groupes industriels européens dans le projet AIT est un gage de l'intérêt de telles plates-formes mais aussi un espoir quant à leur évolution des domaines de la conception vers les domaines de la production. En particulier nous pensons aux modes d'intégration possibles avec des plates-formes de production du type CCE-CNMA autour du concept de "bus logiciel" défini dans AIT [GUI 95].

De même, le projet CIM-BIOSYS [WES 94] développé à l'université de Loughborough, consistant plus particulièrement à la mise en oeuvre des concepts énoncés dans CIM-OSA, propose l'utilisation d'un ensemble d'outils logiciels STEP autour d'une infrastructure intégrante comparable à NEUTRABAS pour l'aspect gestion des données produit.

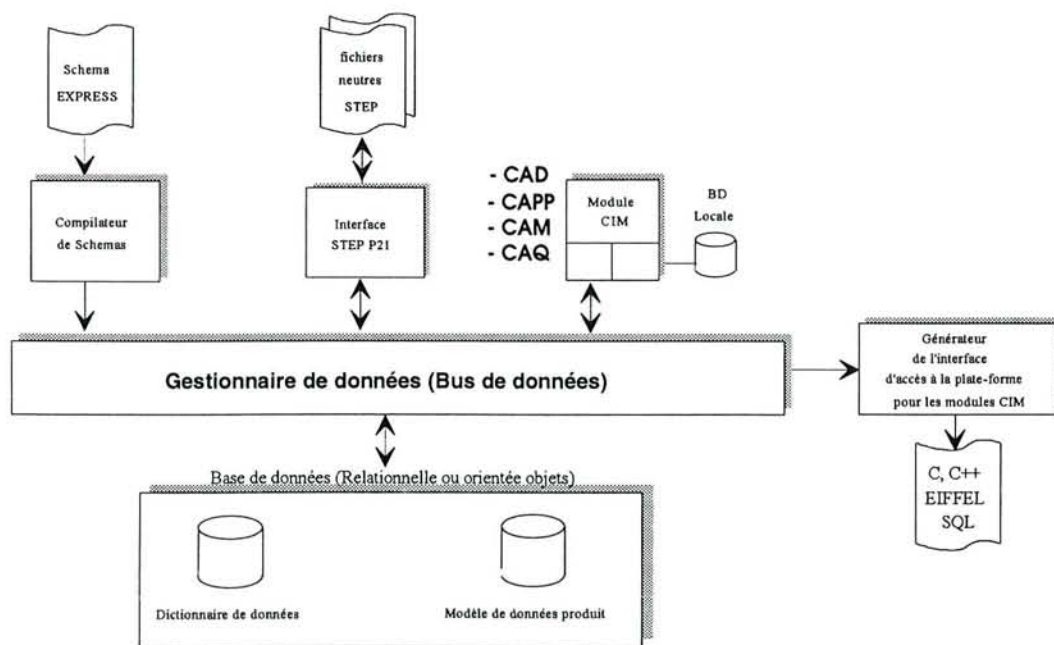


Figure II.14: Archétype d'une plate-forme STEP.

L'intégration de nos outils à ce type de plates-formes est, à notre sens, le point de départ primordial à l'intégration verticale, et un objectif prioritaire pour nous.

## 6. Conclusion

Dans ce chapitre, nous avons examiné différentes voies ouvertes dans le domaine de l'intégration des données du produit. L'objectif était de proposer une solution viable au

problème de la migration des données de conception vers un modèle informationnel afin de l'implanter sur un objet nomade de production. Il s'agissait de définir comment réunir les activités de conception, génératrices de données sur le produit, et notre proposition de gestion des données du produit en production selon le concept d'objet nomade?

Après avoir rejeté les solutions propriétaires, nous nous sommes orientés de manière plus précise vers les travaux en cours au niveau international. Ces travaux, consensuellement tournés vers STEP tendent à définir des modèles assurant la gestion des données du produit durant l'ensemble de son cycle de vie.

En complément à cet effort de normalisation, un ensemble de projets ont contribué à définir un ensemble d'outils logiciels d'assistance à la mise en oeuvre de plate-formes permettant l'intégration de logiciels métiers (CAO, Gammes, ...) autour des différents protocoles d'applications STEP existants ou à venir. Parmi ces outils d'assistance, ceux qui nous semblent essentiels sont les outils d'interfaçage SDAI, les outils de lecture et de création de fichiers neutres et les outils traitant les modèles EXPRESS afin de générer les structures de données (paragraphe 5.1.).

Ces travaux dans STEP sont actuellement centrés sur les activités de CAO mais ils s'ouvrent sous la pression d'un certain nombre d'utilisateurs soucieux de voir entrer STEP dans les ateliers de fabrication.

Les perspectives offertes par une démarche d'intégration basée sur STEP sont nombreux. En particulier, le gouvernement américain a décidé de financer à hauteur de 10 millions de dollars le projet de création d'une architecture ouverte à STEP pour les commandes numériques de machines outils. Ce projet, structuré autour d'un consortium nommé OASYS, regroupe des fabricants de commandes numériques américains, des utilisateurs tels que General Motors et la société STEPTools en tant que spécialiste des outils STEP. L'objectif est de définir les CN futures, aptes à traiter les informations produits structurées selon un protocole d'applications non encore défini. A terme, le code ISO pourrait disparaître au profit d'un protocole d'applications STEP décrivant les opérations à réaliser sans tenir compte explicitement des caractéristiques de la ou des machines, augmentant par la même la flexibilité des systèmes de production [STE 95].

Une seconde initiative de l'US Air Force avec le projet IPPI (Integrated Product Processing Initiative) va dans ce sens.

Nos travaux, visant à gérer les informations associées aux produits sur des supports d'identification automatique, ayant rapidement retenu STEP et l'ensemble des outils associés comme choix stratégique, se voient là ouvrir des perspectives d'utilisation prometteuses.

# Chapitre III

## Des produits aux objets nomades de production

*Résumé :* Nous partons de la finalité d'intégration des systèmes par étiquettes électroniques dans les systèmes de production pour remonter les éventuelles contraintes constatées au niveau de l'analyse conceptuelle. Après caractérisation de l'objet nomade de production, nous proposons les différents éléments constitutifs d'un SGBD distribué sur des étiquettes électroniques. Nous détaillons plus particulièrement les spécifications d'une interface d'accès à ce SGBD par les applications. Puis finalement nous proposons une méthode de conception de la partie commande d'un SAP utilisant l'interface d'accès à un tel système d'information discret et embarqué.

### 1. Introduction

La conception de la partie commande et du système d'information d'un SAP<sup>1</sup> requiert l'application d'une méthode de développement permettant de passer de la spécification des besoins "utilisateur" à l'implantation réelle sur site du système (cf. figure III.1).

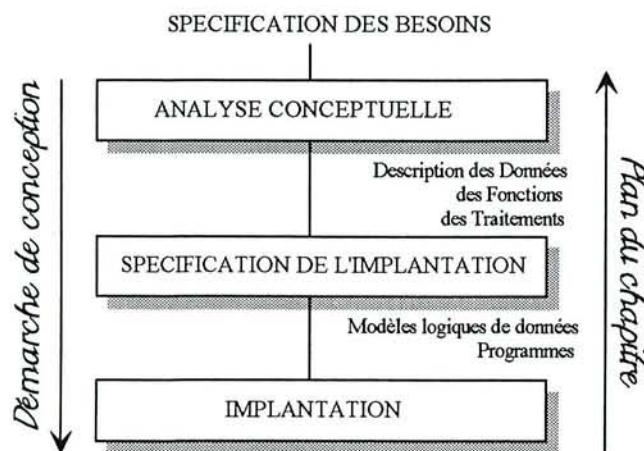


Figure III.1 : Démarche de conception.

<sup>1</sup> Système Automatisé de Production



La démarche de conception pourra être qualifiée d'opérationnelle au sens de [GRU 94] dans la mesure où elle propose un ensemble de méthodes et d'outils permettant l'utilisation de la totalité des modèles générés par une phase comme une donnée d'entrée de la phase suivante, et ce jusqu'à l'implantation physique.

Le plan du chapitre suit une démarche ascendante partant de la problématique de gestion des données du produit dans le système de production. Nous donnons d'une part une définition détaillée de l'objet nomade de production et de ses caractéristiques et d'autre part nous proposons une structure d'accueil et une méthode de mise en oeuvre d'un système d'information **distribué sur des supports d'identification automatique**.

Ensuite, considérant que *"le système de bases de données n'est pas une finalité mais un composant du système CIM"* [DI. 91], nous proposons une démarche de conception de la conduite de systèmes automatisés par objets nomades de production reposant sur la construction des trois modèles suivants :

- le modèle **fonctionnel**. Il décrit les fonctions du système avec comme exemple de méthode d'analyse utilisée SADT. Cette analyse doit permettre la décomposition hiérarchique des fonctions en sous-fonctions plus élémentaires,
- le modèle **comportemental**. Il décrit l'évolution temporelle du système à l'aide de Grafset, de réseaux de Petri, ...
- le modèle **informationnel**. Il décrit les données associées aux **objets nomades** de production et traitées par le système avec des méthodes de représentation et formalisation telles que NIAM, EXPRESS-G.

Les formalismes retenus permettront la traduction partielle ou totale des modèles en codes exécutables, facilitant la mise en oeuvre du système automatisé.

## **2. Un objet nomade particulier : l'objet informé**

---

Selon les approches multi-agents, décrites dans le chapitre I, il apparaît que chaque produit peut être considéré comme un acteur dans son processus de transformation ou d'utilisation. A partir de la classification systémique établie par K.E. Boulding puis complétée par J.L. Lemoigne, nous caractérisons le degré d'autonomie de ces agents de production [CHA 95] selon leurs capacités à (cf. figure III.2):

- fournir des informations orientant leur devenir : les objets nomades de production gèrent leur propre base de données. Ils offrent des services équivalents aux services fournis par un SGBD. L'objet nomade est dit de **Classe 1**. Le "comportement" des objets nomades de production existe ici :
  - ↳ sous la forme de fonctions (comparables aux fonctions SQL par exemple) permettant le traitement de requêtes complexes que nous désignons sous le vocable "méthodes d'accès",
  - ↳ sous la forme d'actions de type réflexes rendant compte d'un comportement autonome. Cette technique est proposée sur la plupart des SGBD, où il est possible d'implanter des algorithmes activés systématiquement lors d'opérations d'accès à la base (comparables aux triggers SQL par exemple),
- coopérer avec les autres entités participant à leur évolution. Ceci est une caractéristique de **Classe 2**. Les objets nomades emportent à la fois leurs informations mais également un ensemble de fonctions de décision et disposent de la capacité de communication avec leur environnement (approches multi-agents),
- adapter leurs règles de décision en fonction de l'expérience acquise durant les phases précédentes de leur cycle de production -**Classe 3**-. Les objets produisent des décisions se référant à un double cadre : celui de leur finalité et celui de leur histoire,
- cumuler plusieurs projets, tels l'acquisition de connaissances sur le processus exploré et la réalisation de leur finalité, nécessitant la coordination des processeurs de décision -**Classe 4**-,
- s'auto-finaliser ou imaginer, dans la mesure où l'on considère qu'un jour ce privilège ne sera plus uniquement réservé à l'homme mais qu'il pourra être étendu aux systèmes artificiels -**Classe 5**-.

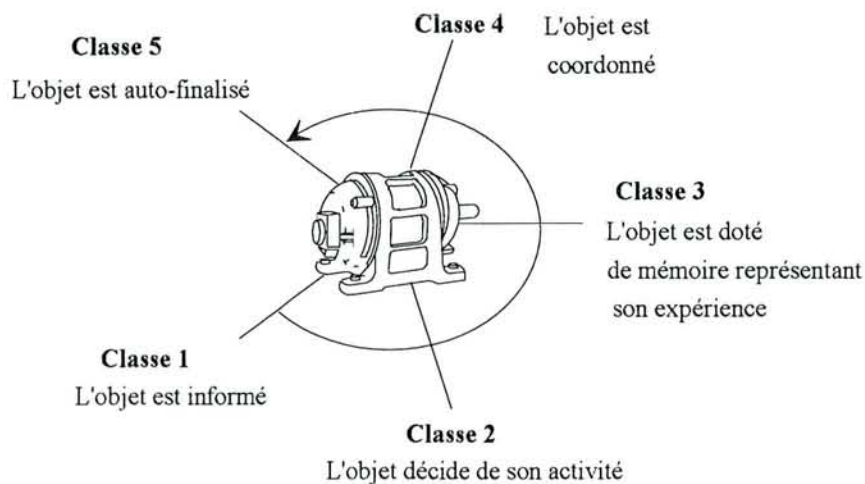


Figure III.2: Classification des objets nomades de production.

Nos propositions consistent en l'**élaboration d'un système d'information nomade de classe 1**, avec les méthodes et les outils associés, où l'initiative du dialogue produit-processus est prise non pas par le produit, serveur de ses informations, mais par les machines, clientes des produits. Cette approche, plus pragmatique que les "produits 'intelligents' et autonomes", offre plus de possibilités d'intégration dans des structures de production existantes car elle ne remet pas en cause les modèles de pilotage de systèmes couramment utilisés [DER 95], comme le font les approches multi-agents. Le système automatisé de production reste malgré tout piloté par les produits puisque la connaissance qu'ils emportent est l'événement déclenchant les activités de production.

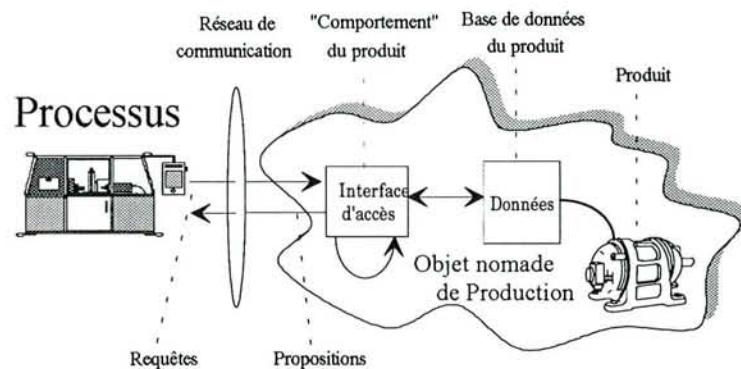


Figure III.3: L'objet serveur de ses informations [BAJ 92].

La constitution et la gestion du système d'information distribué sur étiquettes électroniques selon cette approche client/serveur (cf. figure III.3) allant au delà de l'utilisation d'octets reste, à notre sens, **la base d'une mise en oeuvre** de concepts ouvrant des perspectives nouvelles à la distribution des fonctions de décisions, soit donc à l'implantation d'objets nomades de classes 2 et 3, voire 4.

Les deux aspects présentés figure III.3, soit la base de données du produit et l'interface d'accès, constituent les deux points caractérisant l'objet nomade de production. Nous allons successivement étudier chacun de ces points et proposer des solutions permettant de passer de cette définition à des techniques de mise en oeuvre.

## 2.1. Les données du produit

### 2.1.1. Formalisme de modélisation des données

Le point d'entrée d'un SGBD est couramment un modèle logique de données. Ce modèle, directement interprétable par une machine, est une traduction manuelle ou semi-automatique

du modèle conceptuel de données retenu. Il permet la création des structures physiques de stockage des données ainsi que la configuration de l'interface d'accès à la base de données [PIC 90]. Le choix et l'utilisation d'un modèle logique de données ont des répercussions très importantes sur la nature des applications que peut supporter un SGBD, ainsi que sur la façon dont ces applications sont réalisées. Les applications utilisent les possibilités qu'offre ce modèle tout en supportant ses contraintes.

Dans le domaine de l'identification automatique, où il est encore d'usage de parler d'octets pour quantifier et appréhender les données manipulées, les différents formalismes retenus sont :

- la modélisation sous la forme d'un ensemble de **blocs mémoires ou de fichiers structurés ou non**. Cette approche, proposée dans le cadre du projet Esprit MONOLIN [MON 94], fait partie des possibilités offertes par un nombre croissant de fabricants d'étiquettes électroniques.

Ce mode de structuration des données, correspond typiquement aux services offerts par un système d'exploitation type MS-DOS ou UNIX. La dépendance physique des données, vis à vis des supports, a justifié l'ensemble des travaux entrepris dans le domaine des SGBD [GAR 85] [PIC 90],

- la modélisation sous la forme d'une **liste de variables** possédant un nom et un type (entier, réel, chaîne de caractères, structure, ...).

Dans ce cas, l'accès aux attributs se fait par lecture ou écriture de variables. La correspondance entre la variable et sa position physique de stockage sur l'étiquette peut être réalisée par la partie commande (lecture des octets 50 et 51 pour accéder par exemple à la couleur du produit) ou par l'intermédiaire d'une fonction effectuant la conversion entre le niveau logique et le niveau physique. Ce mode de modélisation est celui retenu par Pereyrol [PER 93] à l'Université de Montpellier. Nos premières expérimentations nous ont montré que cette approche, faiblement structurée, pose problème dès lors que le nombre de variables nécessaires pour modéliser les produits devient important,

- le **modèle relationnel**.

C'est la solution ayant donné naissance au langage CQL<sup>2</sup>, un SQL adapté à de petites structures embarquées sur des cartes à puce, qui vise, entre autre, le marché de la carte médicale personnelle. Bien que fort séduisante, cette solution est mal adaptée dans notre cas, car le traitement de requêtes nécessite la réalisation de jointures entre tables [GAR 85]. Ces opérations sont coûteuses en

---

<sup>2</sup> Card Query Language, © Société Gemplus

temps et *a priori* incompatibles avec les contraintes technologiques liées à l'emploi de systèmes d'identification automatique. Le problème majeur provient du faible débit d'accès aux étiquettes électroniques alors que nous ne disposons - ou estimons ne disposer - que de fenêtres temporelles souvent extrêmement étroites.

Notre travail se place dans une démarche d'intégration des informations, et cherche à couvrir l'ensemble du cycle de vie du produit afin d'assurer une compatibilité avec la série de normes STEP (cf. chapitre II), notamment avec les activités de conception soit par transfert de fichiers neutres STEP [ISO 2] soit par accès à une base de données SDAI. A cette fin, nous proposons l'utilisation d'un **modèle orienté-objet** au travers d'une description dans le langage EXPRESS [ISO 1] dont les avantages identifiés, en plus d'être intégré dans la démarche STEP, sont :

- en tant que modèle logique de données, EXPRESS permet de s'affranchir du formalisme de modélisation retenu par le concepteur car il existe à présent des traducteurs de modèles conceptuels NIAM, Booch, ... vers EXPRESS (cf. chapitre II, § 4.3.1.2). Ceci assure la disponibilité d'outils et de méthodes de conception qui permettent d'aller, par une démarche continue, de la phase de modélisation conceptuelle à la phase d'implantation du système d'information,
- EXPRESS constitue un choix cohérent en accord avec les orientations historiques prises dans les domaines des systèmes d'information (octets, fichiers, modèles relationnels et modèles orienté-objet). Nos travaux sont ainsi complémentaires aux travaux tendant à offrir des services de gestion et d'exploitation des étiquettes plus proches des besoins des applications (entre autre par MONOLIN au travers des fichiers structurés),
- finalement, nous constaterons que ce choix offre de bonnes performances temporelles, tout du moins, dans le cas particulier de notre expérimentation (Chapitre IV - § 3.5).

## 2.1.2. Distribution des données

### 2.1.2.1. Nécessité de la distribution

Au vu des résultats probants obtenus, en terme de flexibilité des structures de production, par l'utilisation des premiers dispositifs à étiquettes électroniques [HEN 87] [UDO 92], les utilisateurs ont entraîné les fabricants à augmenter au fur et à mesure la capacité mémoire

des étiquettes électroniques sans pour autant en modifier le coût. Néanmoins, les besoins exprimés par le groupe d'utilisateurs du projet MONOLIN montrent que ces étiquettes électroniques n'offriront sans doute jamais une capacité de stockage suffisante à la gestion totale des informations associées à un véhicule automobile, dont une estimation basse est de l'ordre de grandeur donnée figure III.4. C'est donc comme des outils d'implantation partielle du système d'information qu'il faut les considérer.

Nous pouvons toutefois noter que les valeurs présentées ici ne prennent pas en compte les besoins futurs tels que les présentations de type multimédia à destination des opérateurs sur les chaînes de montage. De plus, cette capacité, en octets, ne tient compte que des données brutes nécessaires pour assurer les fonctions de production, maintenance, ... et non pas des données de structure : la mémoire occupée par un fichier, une table relationnelle ou un objet au sens informatique du terme, est plus importante que la somme des éléments qu'il ou qu'elle contient.

Domaine d'application	Capacité requise
Fabrication	32.000 Octets
Qualité	5.000 Octets
Identification du véhicule	200 Octets
Maintenance	40.000 Octets <
Traçabilité	10.000 Octets <
Antivol	1.000 Octets
Vérifications périodiques	1.000 Octets
Identification du propriétaire	200 Octets
<b>Total</b>	<b>89.400 Octets</b>

pouvant atteindre  
plusieurs Megaoctets

Source MONOLIN User Group

**Figure III.4:** Capacité de stockage estimée pour un véhicule [MON 94].

Il apparaît de cette première approximation que le stockage de la totalité des informations caractérisant un produit sur une étiquette électronique n'est pas toujours possible. Qui plus est, le stockage total sur des supports individualisés n'est pas forcément voulu et surtout consiste à dupliquer systématiquement les mêmes informations, communes à un groupe ou type de produits, sur chaque produit: quel pourrait être l'intérêt économique de la redondance massive dans le système d'information d'un dessin de définition ou d'une notice de montage, et à plus forte raison lorsque ceux-ci n'ont qu'une faible probabilité d'être consultés?

Les étiquettes électroniques sont des outils créés afin de gérer les informations associées aux produits en évitant les connexions avec une base de données via un réseau de communication. Il semble donc paradoxal de vouloir maintenir cette connexion mais, au vu des besoins exprimés, elle peut quelquefois être souhaitable ou être nécessaire sans toutefois être obligatoire. Ces deux accès ne sont pas antagonistes mais complémentaires.

Nous allons étudier les caractéristiques des informations associées aux produits afin de déterminer dans quelle mesure nous pouvons en stocker une partie sur une base de données fixe sans toutefois perdre de vue les spécificités qui justifient l'emploi des étiquettes électroniques programmables dans les structures de production (cf. chapitre I, § 4).

### 2.1.2.2. Critères de distribution

Chaque produit possède un certain nombre d'attributs tels la couleur, la masse ou la gamme, manipulés par le processus. L'étude des informations relatives aux produits en cours de production permet de dégager deux grandes classes, ou catégories, d'attributs [HEN 87] impliquant des modes de gestion différents [NOU 94]:

- la classe "**attribut statique**". Ces éléments, communs à un groupe ou type de produits, ne changent pas de valeur au cours de la vie du produit et peuvent être par exemple:
  - ↳ la gamme décrivant le nombre et le nom, l'ordre et les temps opératoires de chaque opération,
  - ↳ l'identification de la pièce,
  - ↳ ...,
- la classe "**attribut dynamique**". Ces éléments, dont la valeur est spécifique à chaque produit, évoluent en fonction du contexte, parmi lesquels nous pouvons citer:
  - ↳ l'opération en cours,
  - ↳ l'état (brut, en attente, en cours, produit fini),
  - ↳ ...

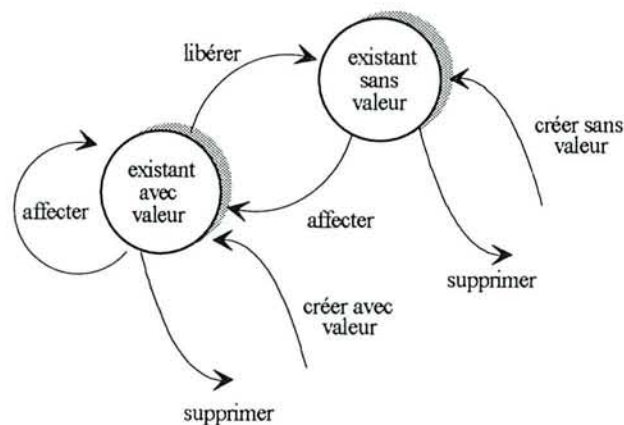


Figure III.5: Etat des attributs du produit.

Durant la phase de production, les attributs statiques, partagés par plusieurs produits, ne peuvent ni être détruits, ni être affectés. Cela reviendrait, par exemple à détruire la gamme de fabrication ou à modifier les caractéristiques de cette gamme. Par contre les attributs dynamiques, motivant l'emploi des étiquettes électroniques programmables, peuvent être affectés, créés avec ou sans valeur initiale, libérés et détruits selon le graphe donné figure III.5.

Notre objectif étant ici la constitution d'un système d'information porté par des étiquettes électroniques programmables, nous avons choisi de poser la contrainte suivante, formalisée figure III.6 :

*"Les attributs statiques pourront être stockés sur une base de données ou sur l'étiquette alors que les attributs dynamiques seront obligatoirement stockés sur l'étiquette. Les opérations d'écriture sur la base de données ne sont donc pas permises".*

Cela pose le problème de la détermination des attributs significatifs pertinents du produit vis à vis de la structure de production. Une autre réflexion, suscitée par cette règle, concerne l'adéquation nécessaire entre la capacité de stockage du support mobile et le nombre d'attributs dynamiques générés pour une application donnée. Ce problème, non complètement indépendant du précédent, fera l'objet d'un développement dans le paragraphe 3.2.

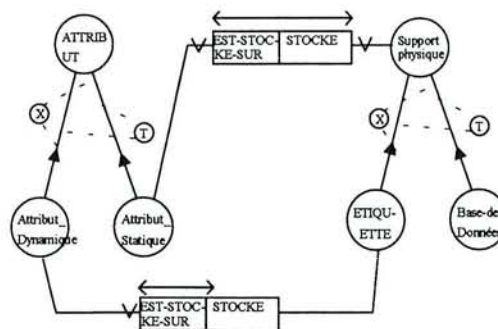


Figure III.6: Stockage des attributs du produit (modèle NIAM).

Un des problèmes posés par les SGBD distribués provient de la duplication possible des données. Dans un tel cas, il est nécessaire de s'assurer que la valeur des attributs est la même dans le temps et dans l'espace. Afin de s'affranchir de ce problème, qui fait encore l'objet de nombreux développements scientifiques, nous proposons le postulat suivant : les informations caractérisant un objet nomade production ne sont nécessaires que lorsque le produit physique est à proximité d'une unité de communication du système d'identification automatique. Cela nous permet d'énoncer la contrainte suivante :



*"Un attribut statique du produit sera stocké soit sur l'étiquette soit sur la base de données, à l'exception des attributs permettant l'identification du produit sans ambiguïté (le ou les identifiants du produits), qui eux seront stockés sur les deux supports".*

La valeur de ce ou ces derniers attributs statiques constitue la clef unique pour la base de données au même titre que l'identifiant utilisé dans les applications à code à barres [LUX 94]. Cette contrainte permet d'assurer la cohérence spatio-temporelle des attributs du produit en limitant au maximum la duplication des données.

## 2.2. Proposition d'une interface d'accès à l'objet nomade de production

### 2.2.1. Structure d'accueil

Nous avons vu précédemment que les données associées à un produit peuvent éventuellement être distribuées sur deux supports distincts: l'étiquette électronique et une base de données connectée sur un réseau de communication.

L'approche naturelle dans le cadre des SGBD répartis [GAR 85] consiste à rendre cette distribution transparente pour l'utilisateur, qu'il soit une machine de production ou un opérateur utilisant un terminal de dialogue. Cela impose l'utilisation d'une interface d'accès aux données, apte à rechercher l'information sans aucune intervention extérieure. Le rôle de cette interface, placée entre les machines de production et le support physique de données du produit, est de gérer la cohérence, l'accessibilité et de masquer la complexité de l'implantation physique des données pour l'application. Dans la terminologie MONOLIN cette interface est appelée A.P.I. (**A**pplication **P**rogramme **I**nterface), acronyme que nous conservons dans ce document.

Au vu de l'hétérogénéité tant des matériels que des logiciels composant les cellules de production, il nous semble nécessaire que cette API puisse être **implantée sur les unités de contrôle** du système d'identification automatique, connectables à un réseau de communication, comme cela est présenté sur la structure d'accueil figure III.7. Cela est justifié par le fait que les programmes informatiques nécessaires à la réalisation des traitements sur les données des produits ne peuvent, d'un point de vue réaliste, être programmés sur l'ensemble des matériels hétérogènes composant une structure de production, et donc dans une multitude de langages de programmation différents.

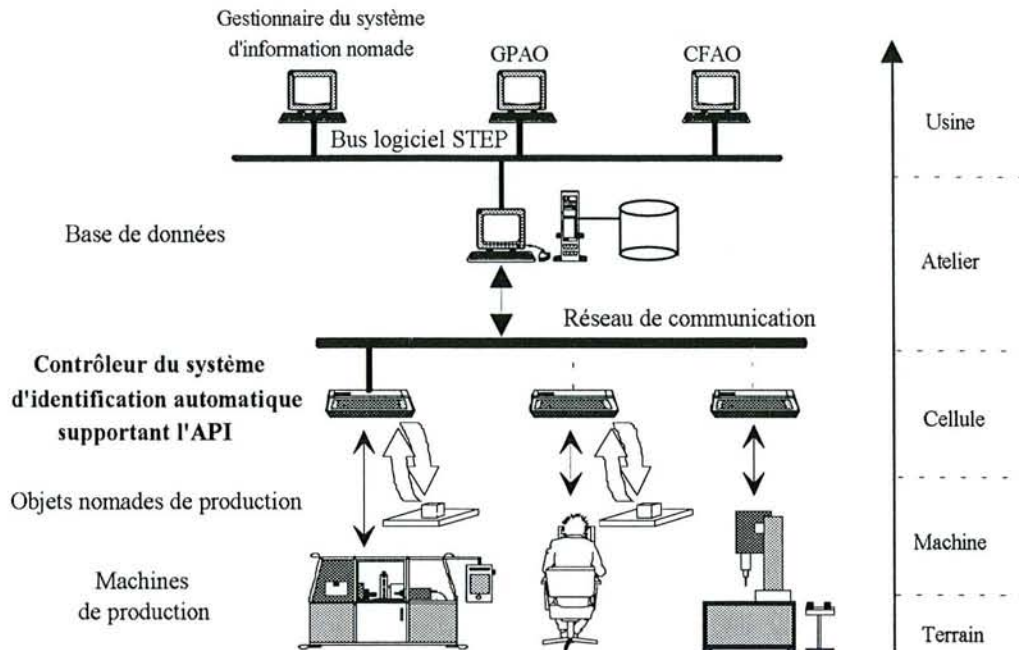


Figure III.7: Structure d'accueil du système d'information nomade [CHA 91b].

La structure d'accueil que nous proposons est bâtie autour de quatre types d'éléments :

- un "**gestionnaire du système d'information nomade**" en liaison avec la gestion de production et la CFAO, sur un bus logiciel STEP type NEUTRABAS [FER 91] ou AIT [GUI 95]. Il permet le traitement des données de définition des produits et leur migration soit vers la base de données de production soit vers les étiquettes électroniques. Son rôle principal est de configurer le système d'information.
- **une base de données multi-utilisateurs en réseau** afin de déporter les fonctions de mémorisation des informations statiques non stockées sur les étiquettes électroniques, par exemple les programmes commande-numérique, les procédures de montage, ...
- **les contrôleurs de systèmes d'identification automatique**, supportant l'API. Leur rôle est d'assurer, après réception de requêtes en provenance des postes de travail, la fourniture des informations concernant les produits. Ils doivent à ce titre assurer d'une part la gestion physique des données sur les étiquettes électroniques, et d'autre part la transparence à la localisation des données lorsque la base de données en réseau est utilisée. Ils sont connectés :
  - ↳ à la base de données, si besoin est,
  - ↳ aux machines via une liaison bipoint ou réseau, voire par un bus fond de panier dans le cas des automates.

16

- les **étiquettes électroniques** portant les informations associées aux produits et assurant une synchronisation naturelle du flux physique et du flux informationnel.

Nous allons détailler, dans le paragraphe 2.2.2., l'élément central de cette structure, soit le contrôleur de systèmes d'identification automatique supportant l'interface d'accès aux données portées par les produits.

## 2.2.2. Spécifications du module supportant l'interface

### 2.2.2.1. Modèle en couches

La décomposition en couches informelle du module supportant l'API, que nous proposons figure III.8, permet de satisfaire à l'ensemble des objectifs précédents. Cette décomposition en couches est motivée par les résultats obtenus, en termes de modularité et d'extensibilité, dans la définition tant du modèle de communication OSI [LEC 94] que des architectures de bases de données [GAR 85] [PIC 90].

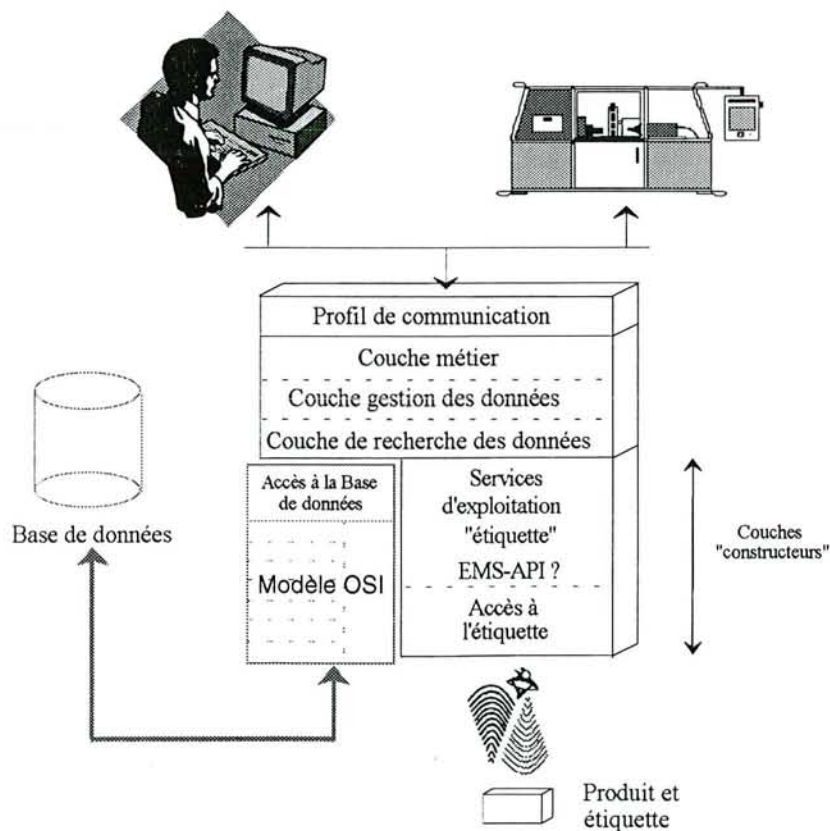


Figure III.8: Décomposition proposée du module supportant l'API [CHA 94c].

L'accès aux structures physiques de stockage de données est réalisé au travers de couches "constructeurs" tant pour la base de données que pour l'étiquette. Dans ce cadre, il nous semble intéressant que l'accès aux étiquettes électroniques puisse être fait au travers d'une interface d'accès aux étiquettes performante, standardisée ou standardisable, tel que l'EMS-API définie dans MONOLIN. Son rôle est d'offrir des services d'accès de haut niveau (du type fichiers) au stockage physique des données sur le support.

Au même titre que dans des SGBD distribués, une couche fonctionnelle réalise la recherche des données sur les deux supports distincts. Elle traite les demandes de lecture ou d'écriture en provenance de la couche de gestion des données.

Cette couche de gestion des données reçoit des requêtes en provenance de la couche "métier". Le rôle de la couche "métier" est de traduire les demandes émises par l'application (*Operation\_Suivante*, *Operation\_finie*, ...), en un ensemble de fonctions élémentaires contenues dans une chaîne, ou unité de traitements, permettant la navigation dans le modèle de données.

Comme nous le montrerons dans le paragraphe 2.2.3, la couche de gestion des données ainsi que la couche "métier" seront générées automatiquement à partir des spécifications fournies par le concepteur: soit un modèle logique de données exprimé dans le langage EXPRESS et des fragments de code traduisant les spécifications exprimées dans le modèle de traitements.

#### 2.2.2.2. *Analyse fonctionnelle du module*

Les spécifications statiques du module supportant l'A.P.I. peuvent être modélisées à l'aide de la méthode SADT permettant d'une part d'identifier les flux entre les éléments du modèle en couches présenté précédemment, et d'autre part de clarifier les différents aspects relatifs à la gestion et au traitement des données. Les figures III.9 et III.10 représentent ce modèle dont nous détaillons ci-dessous quelques activités:

- l'activité "**Gérer la communication avec le poste**" consiste en la réception, et l'émission de messages de et vers le poste, au travers d'un profil réseau. Cette activité de communication ne peut être décomposée plus avant sans effectuer le choix du réseau, mais nous considérons que FIP, PROFIBUS ou Ethernet TCP-IP/MMS sont à ce jour tout à fait adaptés à nos besoins,

- l'activité "Traiter les informations du Produit", implantée dans la couche "métier", réalise l'interprétation du message reçu, sa conversion en requêtes de navigation dans le modèle de données. Nous détaillerons le contenu de cette activité dans le paragraphe 2.2.2.5,
- l'activité "Gérer la base de données", implantée dans la couche de gestion des données, réalise la conversion de l'ensemble des requêtes d'accès au modèle logique de données en requêtes élémentaires d'accès aux occurrences et attributs des entités EXPRESS indépendamment de leur support de stockage. Cette activité est détaillée dans le paragraphe 2.2.2.4,
- l'activité "Accéder aux données du produit", implantée dans la couche "recherche des données", offre les moyens d'accès directs aux attributs des occurrences d'entités et ceci indépendamment du support de stockage physique. Cette activité, dont la décomposition est donnée sur la figure III.10 suivante, laisse apparaître les fonctions d'accès à l'étiquette électronique et à la base de données en réseau.

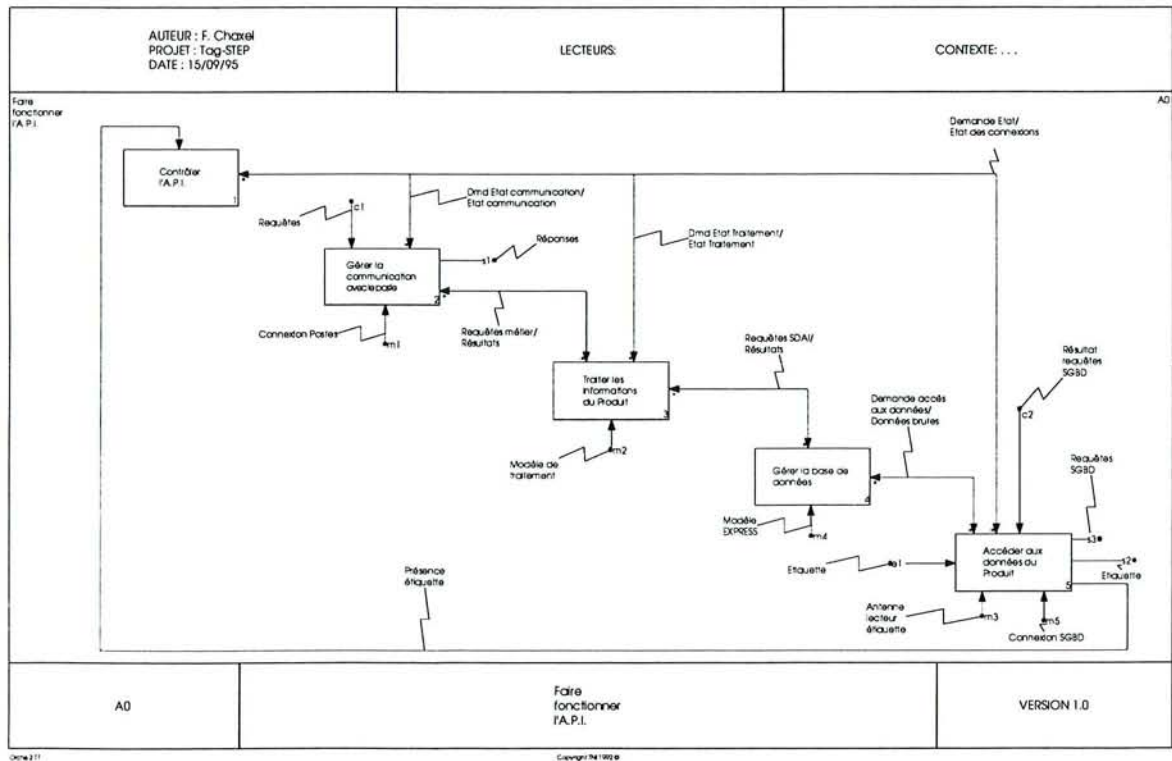


Figure III.9: Modèle fonctionnel du module supportant l'A.P.I. - Diagramme A0.

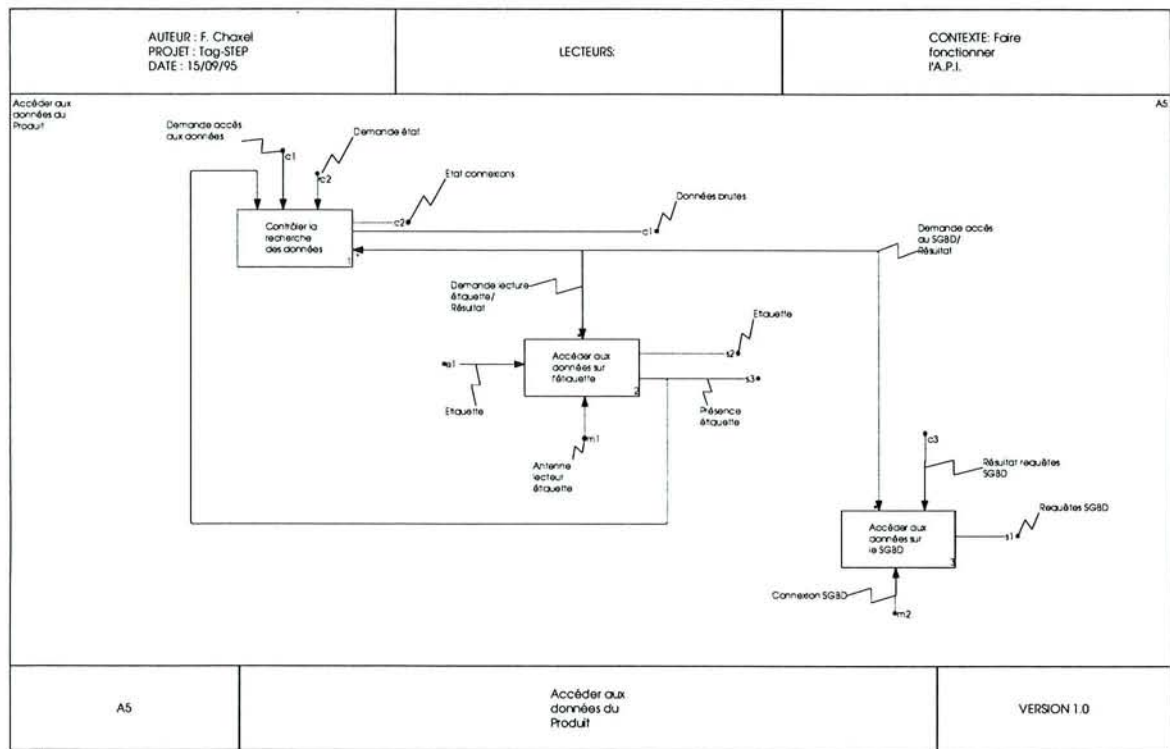


Figure III.10: Accès aux données sur les différents supports - Diagramme A5.

### 2.2.2.3. Accès aux données distribuées

La distribution des données sur plusieurs supports de stockage, proposée sur différents SGBD du marché tels Ingres ou Sybase, fait encore l'objet de nombreux travaux de recherche dans les domaines de l'informatique [BEN 94b].

Toutefois les solutions employées (bases de données fédérées, multi-bases de données, ...) ne sont pas transposables à notre cas spécifique car elles ne prennent pas en compte la mobilité des supports, les faibles débits d'accès ainsi que les faibles capacités mémoire qui caractérisent actuellement les systèmes à étiquettes électroniques. Mais plus encore, les techniques proposées permettent essentiellement la gestion du partitionnement du schéma global en sous-schémas chacun répartis sur un SGBD distinct (par exemple une base de données gère l'ensemble des occurrences du sous-schéma "facturation" et une autre les occurrences du sous-schéma "client"). Dans notre cas il s'agit d'un partitionnement vertical où chaque base de données gère la totalité du schéma mais un nombre limité occurrences. Ce nombre réduit d'occurrences doit correspondre à l'ensemble des attributs caractérisant le produit présent devant l'unité de contrôle.

2.2.2.3.1. Techniques d'accès éligibles

Afin d'accéder aux données du produit non présentes sur l'étiquette électronique, indépendamment du modèle retenu pour modéliser les données de l'objet nomade de production, nous avons identifié deux techniques différentes:

- la première consiste à ajouter sur l'étiquette un ensemble de données de référence permettant la formulation instantanée des requêtes à émettre vers le SGBD sur le réseau afin d'accéder aux données du produit,
- la seconde consiste à lire un ou des identifiants clés, portés par le produit, et à les utiliser comme clés d'accès aux données stockées sur le SGBD en réseau. Le module supportant l'interface d'accès aux données, se charge de construire dynamiquement les requêtes d'interrogation de la base, en fonction de la structure dans laquelle il est immergé. Ces clés consistent au moins en un identifiant unique du produit dans la structure de production, mais peuvent également contenir les caractéristiques de la base de données en réseau afin de donner la possibilité au module d'établir la connexion (par exemple l'adresse réseau du serveur de base de données, le nom de la base de données, ...).

2.2.2.3.2. Méthode retenue

Nous avons retenu la seconde solution proposée, car la capacité mémoire nécessaire de l'étiquette est indépendante du nombre d'occurrences d'entités EXPRESS stockées sur la base de données associée aux produits. Le produit peut donc porter une masse de données nettement plus importante que la capacité physique de l'étiquette, disposant ainsi d'un espace mémoire virtuel limité à la seule capacité du SGBD sur le réseau.

L'accès à un attribut d'une entité EXPRESS est réalisé, après réception d'une requête en provenance de la couche d'accès aux données [CHA 97], selon la procédure suivante (cf. figure III.11):

- l'entité est sur l'étiquette (2): le message est traité en local (6),
- l'entité n'est pas sur l'étiquette:
  - ↳ le produit est identifié par lecture de ses caractéristiques (identifiant du type de produit, identifiant du produit dans son type) (3),
  - ↳ les attributs de l'objet sont extraits du SGBD connecté sur le réseau (4),
  - ↳ puis émis au demandeur (7) ou un message d'erreur est généré (5).

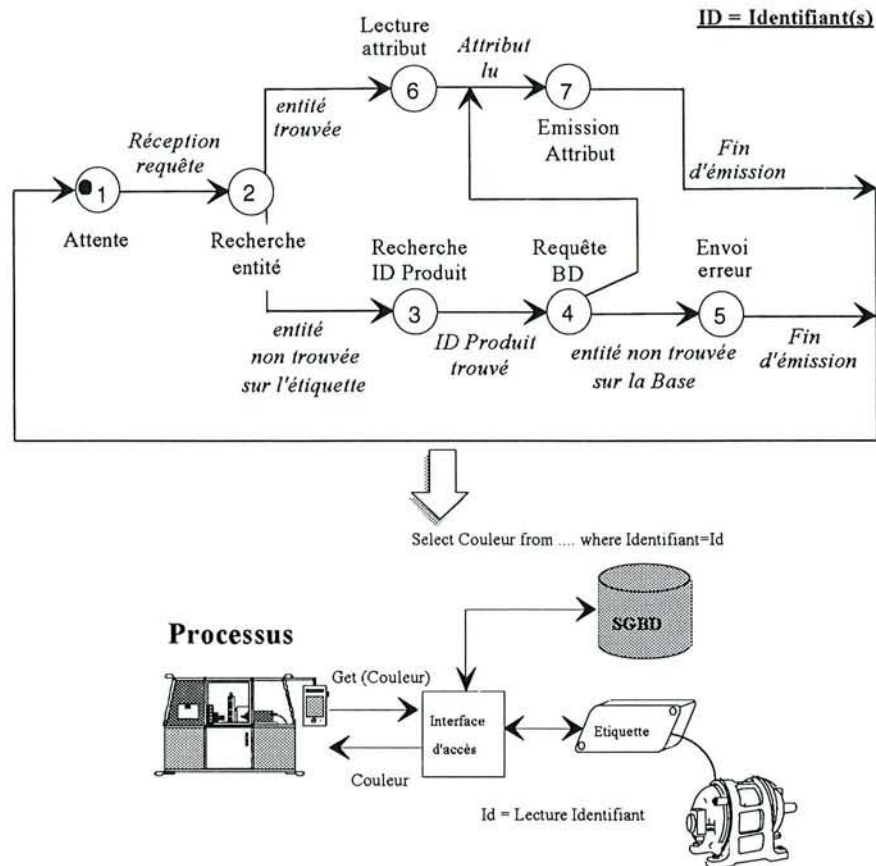


Figure III.11: Spécification sous forme d'un graphe d'état de l'accès aux attributs.

#### 2.2.2.4. La gestion des données

L'accès aux occurrences d'un modèle EXPRESS, stockées sur une base de données, est, dans la série de normes STEP, réalisé selon les spécifications de l'interface SDAI [ISO 3].

Au même titre que le langage SQL d'interrogation des bases de données relationnelles, SDAI fournit les mécanismes de base nécessaires à la lecture et à l'écriture d'occurrences d'entités par l'intermédiaire d'un ensemble de fonctions permettant la navigation dans le modèle de données. SDAI permet également de vérifier la cohérence des données vis à vis du modèle.

L'avantage de SDAI est d'être dans sa phase finale de normalisation, à l'inverse des langages propriétaires d'interrogation de bases de données orientées objets [BOU 95]. Ceci assure une certaine pérennité aux travaux effectués. La gestion et la navigation dans le modèle de données, seront donc réalisés au travers de cette interface, comme le présente la figure III.12 suivante:



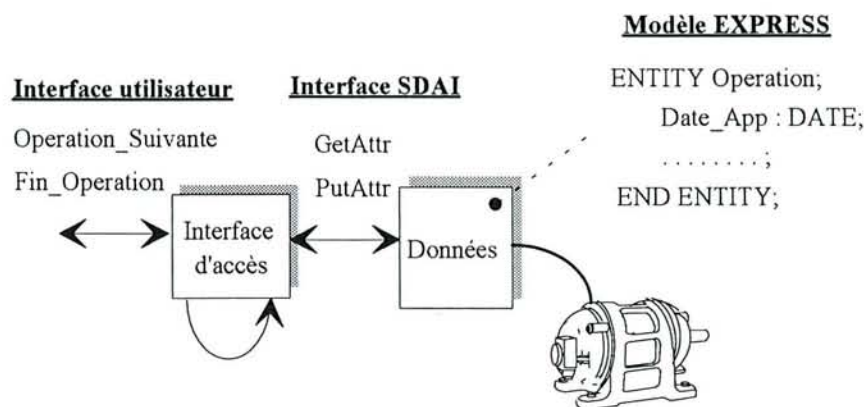


Figure III.12: Modèle informel de l'objet nomade de production [CHA 93].

Comme l'a prévu l'ISO, et comme l'ont démontré les partenaires du projet Esprit NEUTRABAS [BRU 91] [FER 91], cette interface SDAI peut être générée ou configurée de façon automatique à partir du modèle logique de données en EXPRESS (cf. chapitre II, § 5.2). Ce mécanisme, particulièrement intéressant, nous permettra de passer directement du modèle logique de données au codage automatique du module supportant l'interface d'accès.

#### 2.2.2.5. La couche "métier"

La communication entre le processus et l'API est réalisée par l'intermédiaire de requêtes ou transactions de niveau application :

*"Les transactions sur les bases de données sont les éléments utilisés pour décrire les unités de traitements sur le système, finalement perçues par l'utilisateur comme des opérations uniques et atomiques"* (traduit de [JOR 90]).

L'implantation de ces transactions directement sur le système de base de données permet d'éviter la programmation des traitements dans une syntaxe contraignante sur des matériels hétérogènes (de l'assembleur aux langages évolués). Cette approche, proposée sur la majeure partie des SGBD commerciaux, consiste à permettre la programmation d'algorithmes sur le serveur de base de données. Ces algorithmes, programmés dans le langage du SGBD, sont exécutables à distance au travers du langage d'interrogation de la base de données et permettent la réalisation de traitements sur les données.

Les requêtes complexes générées par les différentes machines sont du type *Opération\_suivante*, *Opération\_Finie*, *Qualité\_Produit*, *Historique*, ... Elles sont

décomposées, dans des unités de traitements, en une succession de **requêtes SDAI** (cf. figure III.12 précédente) afin d'être interprétées par la couche d'accès aux données.

Cette interprétation est réalisée par des fragments de programmes implantés dans la couche que nous appelons "métier", car elle est générée grâce aux spécifications des besoins des utilisateurs (cf. § 3.1.3.2.) Elle propose des services orientés vers l'utilisateur, donc métier, complétant les fonctions élémentaires de lecture ou d'écriture sur le modèle du produit.

Ces fragments de programmes ainsi que le modèle logique de données en langage EXPRESS constituent les deux points d'entrée pour la mise en oeuvre du système d'information nomade au même titre qu'un script de création de tables SQL et qu'un ensemble de procédures "utilisateur" complexes SQL pour une base de données relationnelle.

### 2.2.3. Génération de l'interface physique

La mise en oeuvre du système d'information passe par la programmation du module supportant l'API à partir des éléments conçus précédemment : le modèle logique de données en EXPRESS et les fragments de programmes support de la couche "métier". Nous proposons la génération automatique du code de l'API, dans un langage cible, à partir de ces éléments selon la figure III.13.

Cette figure III.13 montre les transformations successives que doit subir le modèle logique de données ainsi que les fragments de programmes correspondants à la transcription des services applicatifs en requêtes SDAI :

- ils sont utilisés afin de **générer le code de l'interface applicative** dans un langage procédural. Cette opération, permettant l'obtention de l'API, est réalisé par un outil que nous avons développé : **Tag-STEP**. Le code généré, à partir du modèle EXPRESS, est compilé simultanément avec les fragments de programmes effectuant la traduction des requêtes métiers en requêtes SDAI. L'ensemble est finalement lié au code réalisant la recherche des données sur les différents supports et aux bibliothèques propriétaires permettant l'accès aux ressources physique: les étiquettes et le réseau. Le programme exécutable généré est à transférer sur les unités de contrôle d'étiquettes,

- ils servent également à **générer le schéma de la base de données** connectée sur le réseau permettant la gestion d'une partie des données du produit. Cette base de données peut être du type relationnel ou objet, voire même de simples fichiers.

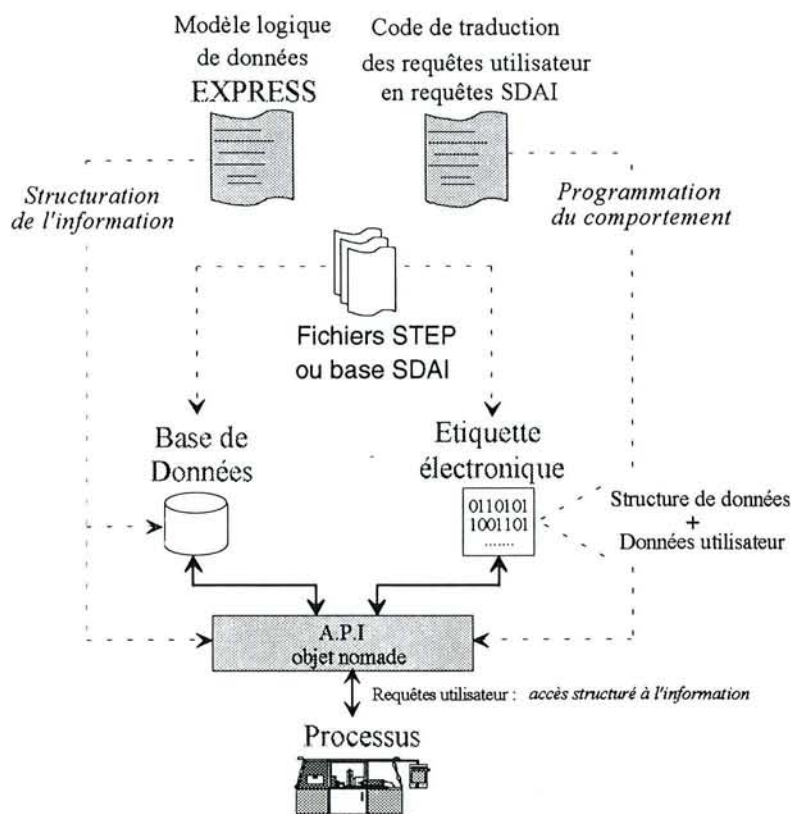


Figure III.13: Génération de la structure [CHA 94b].

Les données de définition du produit, provenant de la CFAO sont quant à elles employées pour :

- **créer l'image mémoire de l'étiquette électronique**, qui sera par la suite associée à un produit mis en circulation dans l'atelier. L'organisation physique des données en termes de blocs, de fichiers et de codage doit être en accord avec les méthodes d'accès implantées dans l'API. Le modèle physique revêt un intérêt crucial car de lui dépendent les possibilités de recouvrement d'erreurs ainsi que les temps d'accès aux données. De plus, la capacité des étiquettes étant limitée, il doit être optimisé en taille. Cette double optimisation en temps et en taille est d'ailleurs antinomique,
- générer une occurrence du schéma de la **base de données** connectées sur le réseau dont le schéma a été créé précédemment. La gestion physique des disques de stockage est assurée par le SGBD choisi.

Nous ne disposons pas actuellement de modèles aptes à optimiser, selon un critère particulier, les choix de **répartition** des données d'un produit sur la base de données ou sur l'étiquette. Le problème ne se pose toutefois pas pour les données dites dynamiques qui ne peuvent être stockées que sur l'étiquette associée au produit.

De ce fait ce choix est fait manuellement par un opérateur et nous espérons, à terme, pouvoir formaliser les règles de décision qu'il sera amené à prendre, afin de les relier avec les résultats obtenus (temps d'accès, accessibilité des données, ...).

Les techniques de mise en oeuvre, que nous avons développé, telles que les transformations à faire subir au modèle EXPRESS ou les règles de codage de données, relèvent à notre sens du cadre de l'application et sont à ce titre décrites dans le paragraphe 2 du chapitre IV.

### **3. Proposition d'une démarche de conception**

---

Le formalisme de modélisation logique des données du produit, la définition de l'interface applicative et de sa structure d'accueil constituent les outils élémentaires à la mise en oeuvre du système d'information au même titre que, par exemple, une base de données relationnelle et le langage SQL.

Un cadre méthodologique ainsi qu'un ensemble d'outils restent toutefois nécessaires afin de guider l'ingénieur dans sa démarche de conception et de développement d'applications exploitant le concept d'objet nomade de production.

#### **3.1. Conception de l'application**

La conception d'un système de contrôle-commande d'un processus requiert la conception d'un modèle unifié couvrant les aspects fonctionnel, informationnel et organisationnel; modèle qu'il faut, par la suite, dériver afin d'en obtenir une réalisation concrète. Notre objectif ici n'est pas de proposer une méthode nouvelle de conception, s'ajoutant aux nombreux travaux scientifiques réalisés dans ce domaine [CIM 94] [GRU 94] [LHO 94] [LES 94] [BRU 94a] ..., mais de montrer que les différents modèles nécessaires à la création de l'API constituent le résultat d'une démarche de conception classique.

Les différents aspects présentés ici, et appliqués dans le chapitre IV, reposent sur l'expérience acquise et sur les outils développés au CRAN mais la similitude avec la méthode proposée dans le cadre de CIM-OSA laisse supposer des perspectives plus vastes.

Nous allons détailler, dans les paragraphes 3.1.1, 3.1.2 et 3.1.3, la démarche globale de modélisation présentée sur la figure III.14 précédente qui nécessite la conception :

- d'un modèle fonctionnel,
- d'un modèle comportemental, décrivant la dynamique du système associée aux fonctions,
- d'un modèle informationnel. Il est décomposable en deux sous-modèles :
  - ↳ le modèle organisationnel de données (MOD), décrivant les données manipulées par l'application,
  - ↳ le modèle organisationnel de traitements (MOT), spécifiant d'une part les modes traitements logiques d'accès aux données, et d'autre part la dynamique de ces données lors d'accès en modification.

Le terme **organisationnel**, utilisé au détriment du terme conceptuel, est justifié ici par le **domaine** visé qui ne couvre pas l'ensemble des informations manipulées par le système de production et par la notion de **site** d'implantation: nous ne traitons que les informations associées aux produits et implantées sur des étiquettes électroniques.

Nous noterons toutefois que les relations inter-modèles proposées ici de façon informelle, donc d'interprétation libre, nécessiteraient une description formelle par exemple par méta-modélisation comme cela peut être proposé par J.J. Lesage [LES 94]. Le méta-modèle ainsi obtenu permettrait la réalisation d'un outil informatique assurant la cohérence entre les différents modèles fonctionnel, informationnel et comportemental.

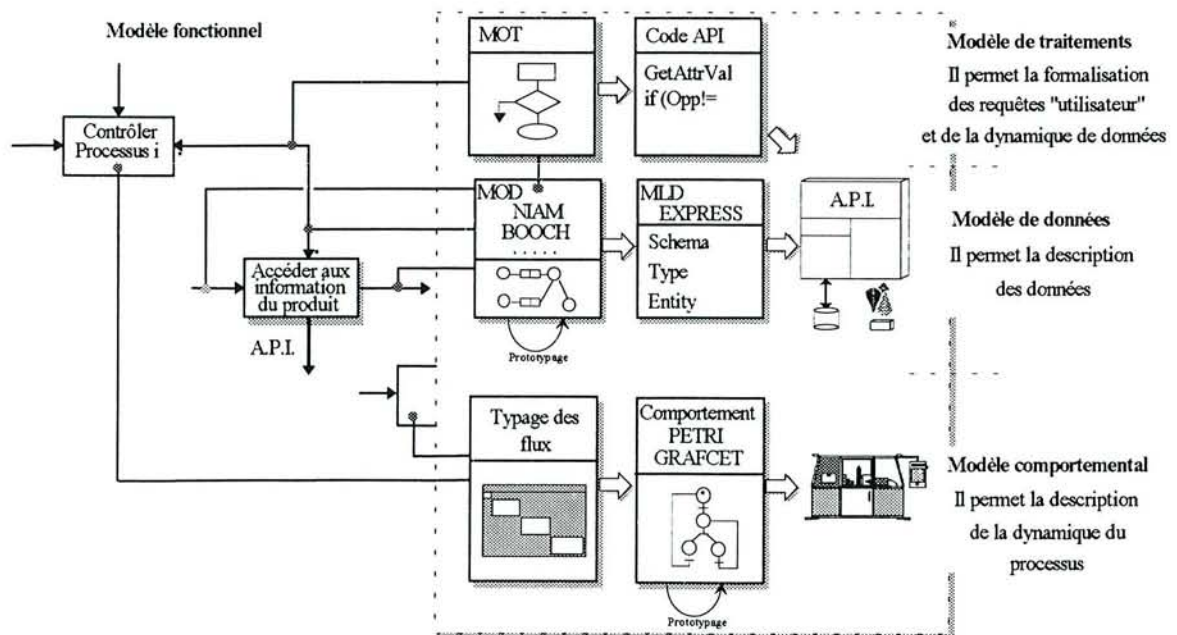


Figure III.14: Démarche globale de modélisation.

### 3.1.1. Modèle fonctionnel

La modélisation fonctionnelle du système de production à concevoir est réalisée selon une approche empruntée à P. Lhoste [LHO 85] et H. Panetto [PAN 91]. Cette approche maintenant validée, consiste en une décomposition, en accord avec la méthode SADT [IGL 89], des processus suivant une structure de contrôle-commande. Chaque processus est décomposé en sous-processus et en une fonction de contrôle nécessaire à la coordination et/ou la mise en coopération des sous-processus identifiés.

Ces derniers, également décomposables de manière similaire, réalisent des fonctions de transformation sur les flux de produits ou d'informations entrants, sous contrôle des flux d'informations provenant de la fonction de contrôle (flux bidirectionnels du type ordre/compte rendu ou demande/réponse).

Le concepteur décomposera les processus réclamant des données associées aux produits, donc en relation avec une API d'accès au système d'information nomade, en faisant apparaître un sous-processus "Accéder aux informations du produit" (cf. figure III.15). Ce sous-processus d'accès aux informations du produit est **entièrement réalisé** par le module supportant l'API précédemment défini. Il n'est plus décomposable sans changer de point de vue.

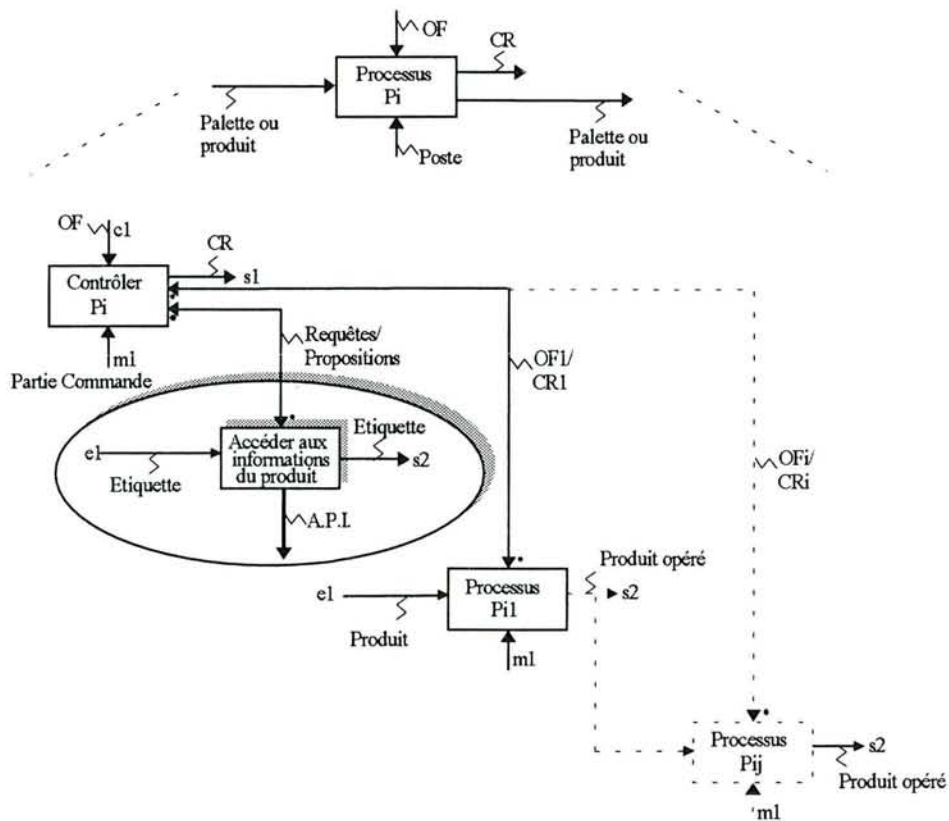


Figure III.15: Décomposition d'un processus en relation avec une API.

Cette modélisation fonctionnelle est purement statique et ne permet pas d'apprécier le comportement dynamique des modèles réalisés. De ce fait, les inadéquations entre le modèle obtenu et les besoins exprimés ne sont appréciables qu'au travers d'un cycle auteur-lecteurs préconisé dans la méthode SADT.

### 3.1.2. Modèle dynamique

Quelques unes des techniques employées pour adjoindre un comportement dynamique à une spécification SADT à des fins de vérification sont [LHO 94]:

- la transformation de la spécification en Réseaux de Petri, justifiée par un besoin de vérification de la synchronisation des processus asynchrones et la vérification de l'allocation de ressources physiques à des processus [CHA 91a],
- l'association de comportements à la spécification fonctionnelle. Ces comportements peuvent être décrits dans les formalismes Grafcet, réseaux de Petri, langages procéduraux, schémas à relais, ...

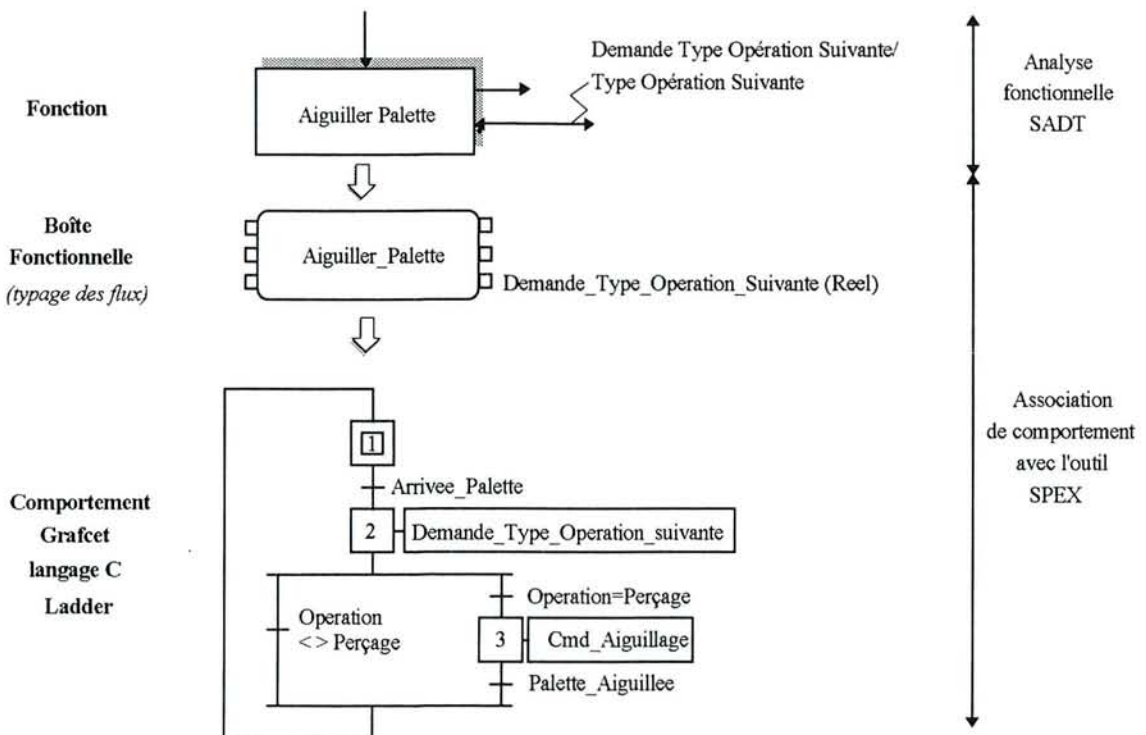


Figure III.16: De la fonction au comportement.

L'environnement SPEX<sup>3</sup> permet l'utilisation de la technique par association de comportements pour la validation du modèle par simulation. Chaque fonction SADT est transformée en boîte fonctionnelle SPEX puis chaque boîte fonctionnelle, correspondant au dernier niveau de décomposition SADT, reçoit une description dynamique comme cela est illustré figure III.16. Les flux identifiés lors de l'analyse fonctionnelle sont soit typés soit considérés comme des groupes de variables plus élémentaires (décomposition des flux).

Sur SPEX, la vérification des modèles est réalisée par simulation synchrone des boîtes fonctionnelles à l'aide de pupitres opérateurs ou de scénarii de test écrits en langage C.

### 3.1.3. Modèle informationnel

La construction du modèle informationnel peut être réalisée en suivant soit une démarche ascendante, soit une démarche descendante ou plus couramment en une combinaison implicite ou explicite des deux : *"Les objets du domaine particulier sont soit recopiés du modèle partiel, soit personnalisés, soit développés pour répondre à un besoin particulier que le modèle partiel ne recouvrait pas"* [CIM 94].

#### 3.1.3.1. Le Modèle organisationnel de données

Dans le cadre d'une démarche ascendante, lorsqu'un modèle de référence du système d'information existe préalablement, il sert de base à la modélisation des vues fonctionnelles et comportementales. Ce modèle existant pourrait être un protocole d'applications de STEP.

Dans le cadre d'une démarche de modélisation descendante du système d'information produit, les différents flux identifiés de et vers l'API vont permettre de concevoir le modèle organisationnel de données (cf. figure III.15) selon le formalisme retenu (Entités-Relations-Attributs de CIM-OSA, NIAM, EXPRESS-G, orienté objets, ...).

En ce sens nous sommes en accord avec R. Gaches [CIM 94] qui énonce en synthèse à la réalisation de l'atelier flexible de l'Aérospatiale-Bourges *"L'analyse opérationnelle a conduit à l'identification des vues d'objets (référencées en entrée ou sortie d'activités). L'analyse informationnelle consiste à en définir le contenu en termes d'éléments d'information, puis à bâtir les objets et définir leurs relations"*.

---

<sup>3</sup> © CRAN-TNI-Spie Trindel



La vérification de complétude de ce modèle organisationnel de données peut être partiellement effectuée par prototypage: nous citerons en exemple l'outil MEGA<sup>4</sup>, supportant la méthode MERISE, qui permet de prototyper le modèle de données par génération de formulaires d'accès à une base de données virtuelle et réduite correspondante au modèle.

### 3.1.3.2. Le Modèle organisationnel de traitements

Le modèle organisationnel de données est conçu à partir des flux de communication, ou requêtes, tel que le présente le paragraphe précédent. L'ensemble des requêtes constitue l'interface "métier" de l'objet nomade. Chacune de ces requêtes d'accès "métier" doit, *in fine*, être transformée en appels aux fonctions SDAI de navigation dans le modèle EXPRESS comme cela est exposé paragraphe 2.2.2.5.

La spécification de ces opérations de transformation, ainsi que la dynamique des données, est couramment appelée modèle organisationnel de traitements ou MOT [PIC 90] (cf. figure III.17). Elle peut être réalisée à l'aide des méthodes d'analyses logicielles permettant l'obtention progressive du code dans le langage cible de programmation de l'API.

Dans ce cadre nous pouvons citer M\*, en grande partie utilisée dans CIM-OSA, qui construit le MOT en utilisant le concept de transactions et de modèle de comportement dynamique des données à l'aide de réseaux de Petri [DI. 91].

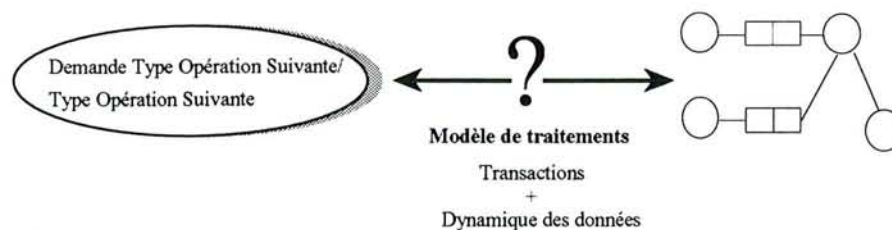


Figure III.17 : Rôle du modèle de traitements.

## 3.2. Vérification des modèles

Un des problèmes posés par la démarche réside dans la vérification séparée par prototypage des deux vues, fonction et information, d'un seul et même objet: le système de

<sup>4</sup> © Gamma International

production. Le système d'information, qu'il soit distribué sur des supports d'identification ou non, existe pour être utilisé par les différentes machines composant l'atelier de fabrication. Or bien souvent les vérifications des modèles par simulation sont réalisées séparément.

Qui plus est, comme nous l'avons présenté, nous ne disposons pas de modèles nous permettant de faire les choix judicieux quant à la répartition des données des produits manipulées par l'application. La vérification de cette distribution peut être réalisée en mettant le système d'information dans sa configuration d'utilisation, soit donc en réalisant une simulation croisée (cf. figure III.18).

De ce fait, le prototypage par simulation du système d'information n'est plus réalisé à l'aide de pupitres opérateur mais grâce aux requêtes émises par la spécification exécutable du système. Réciproquement, le comportement dynamique du système évolue, tout du moins partiellement, en fonction des états fournis par le prototype du système d'information.

Ce type d'approche fait également l'objet de développements par le Polytecnico di Torino [BRU 94a] à des fins de mise en oeuvre des concepts énoncés dans CIM-OSA [CIM 91]: la vue information est supportée par l'outil Quid (Entité-relation) et la vue fonction par l'outil Artifex (Réseaux de Petri).

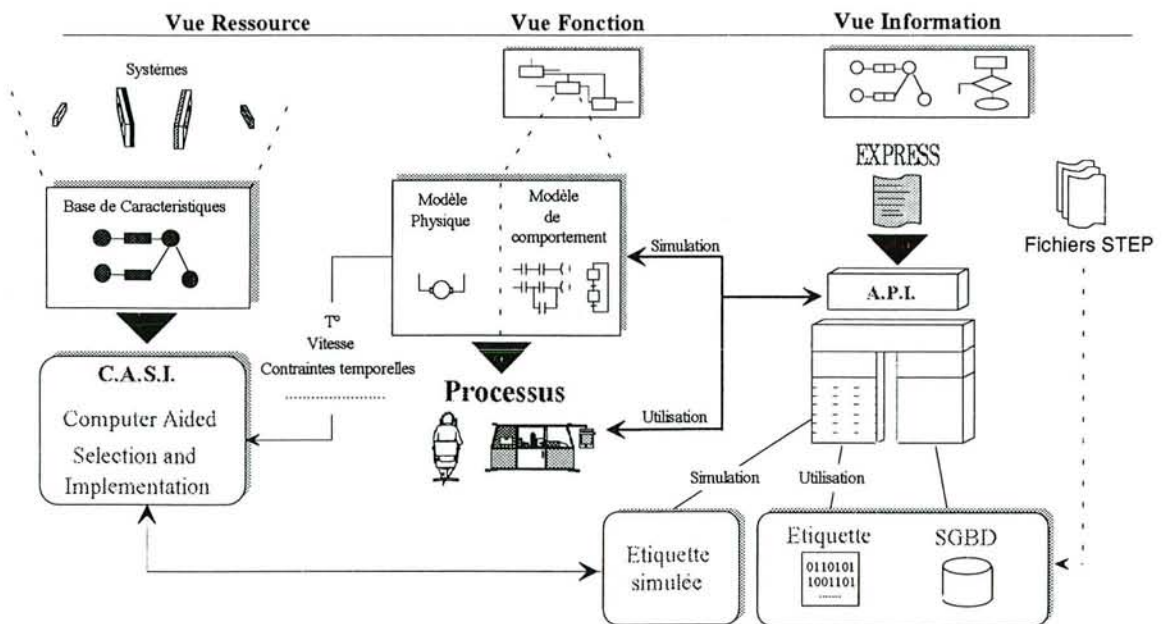


Figure III.18: Prototypage par simulation des vues fonctionnelles et informationnelles.

Les résultats de simulation permettent de vérifier que les différents modèles répondent bien aux besoins, mais également d'obtenir des valeurs quantitatives et qualitatives quant aux sollicitations du système d'identification automatique.

Les **caractéristiques physiques** du processus (température, vitesse de défilement des produits, ...) ainsi que les paramètres associés à chaque transaction (lectures, écritures) sont utilisés par l'**outil CASI** (Computer Aided Selection and Implementation), en cours de développement par nos collègues Allemands de l'IFK dans le cadre du projet PROCOPE 94064, afin de réaliser le choix du système d'identification le mieux adapté. Cet outil, basé sur une formalisation des caractéristiques réels des systèmes d'identification automatique [ARN 95] devra permettre, par une analyse experte, l'extraction des dispositifs du marché répondant aux différents critères demandés.

Afin de rester cohérent avec l'ensemble de nos choix, les données (relatives aux caractéristiques des transactions) transmises à l'outil CASI sont émises dans un fichier neutre STEP, occurrence d'un modèle EXPRESS défini conjointement par le CRAN et l'IFK.

#### 4. Conclusion

---

La constitution et la gestion d'un système d'information sont les éléments majeurs dans la définition de l'ouverture, de la flexibilité et de la réactivité d'un système de production. Les techniques nouvelles de gestion et d'échange de données, issues de l'identification automatique, ouvrent de nouvelles perspectives pour la circulation et la gestion des informations liées à la production. Mais à l'heure actuelle, les systèmes d'identification automatique sont encore pauvres en méthodes de gestion des données, et les structures de conduite des SAP ne profitent pas totalement de ces techniques par inadaptation fonctionnelle. Nos travaux tendent à concilier ces deux parties en définissant d'une part une méthode de construction du système d'information réparti sur les étiquettes électroniques, avec les outils de mise en oeuvre associés, et d'autre part en proposant une structure d'accueil de cette méthode pour les SAP.

Les principales caractéristiques attendues résultant de la mise en oeuvre de ce concept sont :

- une **meilleure distribution et uniformisation des données** dans l'ensemble de la chaîne de fabrication permettant un suivi temps réel des produits. Chaque produit est un élément d'une base de données distribuée permettant une consultation quasi-immédiate par l'ensemble des intervenants (systèmes automatisés ou opérateurs humains),
- une **simplification de la structure et des logiciels des machines** de production. Une interface générique permet de définir le lien entre tous les utilisateurs et les

informations portées par les produits. Cela apportant un gain non négligeable en termes de modularité et de capacité de modification (ajout et ou suppression de machines par exemple),

- une **robustesse accrue du système** d'information due à la distribution des bases de données non dépendantes d'un réseau de communication,
- une **traçabilité effective des produits**, offerte par une structure apte à acquérir l'ensemble des paramètres associés à chaque opération (dates, temps opératoires, machines, ...).

Les travaux dans le domaine des architectures de contrôle-commande classiques et multi-agents montrent la nécessaire gestion des données du produit, auquel nous apportons une réponse.

L'autonomie du produit au sens de l'objet nomade de classe 2 et 3 (paragraphe 2) consiste en la description et l'implantation de fonctions de décision évoluées basées sur l'état du produit et l'interprétation des messages en provenance des agents-machines. Il nous apparaît que ces traitements décisionnels pourraient être implantés dans une sur-couche à la couche "métier" (paragraphe 2.2.2.5). Cela nécessiterait la modélisation de cette sur-couche à l'aide de formalismes adaptés à la description de tels traitements (par exemple les réseaux de Petri).

Un des intérêts apporté par un travail sur les objets nomades de classe 2 et 3 serait à notre avis de fournir les mécanismes permettant une validation expérimentale de projets tel que "Holonc Manufacturing System" [HEI 95] dans des contextes industriels de production de masse.

Un tel travail serait complémentaire et non contradictoire avec nos propositions et avec l'outil Tag-STEP que nous allons décrire.

# Chapitre IV

## Mise en oeuvre et application du concept

---

***Résumé :** Nous montrons comment nous sommes passé du concept d'objet nomade à sa mise en oeuvre en productive. Nous présentons l'outil Tag-STEP que nous avons développé afin de traduire des modèles décrits dans le langage EXPRESS en schéma spécifique à la base de données relationnelle utilisée, ainsi que la manière dont ces modèles sont traités pour générer automatiquement une interface d'accès SDAI aux données. Nous appliquons la démarche de modélisation des aspects fonctionnels, comportementaux et informationnels proposée dans le chapitre III afin de prototyper la cellule flexible d'assemblage à palettes de l'AIP Lorrain. Les résultats de simulation permettent de valider simultanément les stratégies de contrôle-commande de la cellule et le modèle de données du produit retenu. Ils donnent également des informations essentielles permettant le choix du système d'identification automatique.*

### 1. Introduction

---

Selon Samuel Kant "*La pratique sans théorie est aveugle et la théorie sans pratique est absurde*". Les modèles développés dans le chapitre III se doivent d'être validés par une phase d'expérimentation et donc de mise en oeuvre effective du concept d'objet nomade. Pour ce faire nous avons développé un outil nécessaire à la configuration de la structure de gestion des modèles de données des applications de contrôle-commande, par traitement des modèles logiques décrits en EXPRESS, quels qu'ils puissent être.

La réalisation de cet outil a été menée à son terme à partir d'un ensemble de choix techniques pris *a priori*:

- la norme SDAI [ISO 3] n'étant toujours pas dans un état totalement stabilisé de spécification, la transmission de données, entre le niveau conception et nos outils, est donc réalisée au travers de fichiers neutres STEP. Les différents logiciels développés l'ont toutefois été en gardant à l'esprit la nécessaire évolution vers SDAI,

- la base de données, sur laquelle viennent se connecter les A.P.I. (Application Programme Interface), est la base de données relationnelle réseau ORACLE 7.0<sup>1</sup> de l'AIP Lorrain<sup>2</sup> caractéristique des moyens technologiques usuels à la disposition des entreprises.

L'expérimentation en elle-même, consiste en l'utilisation de cet outil et de la méthode de conception associée, sur une application cible particulièrement représentative des systèmes flexibles actuels en termes de complexité de gestion de flux et en nombre de produits variés réalisables : la cellule flexible d'assemblage de l'AIP Lorrain.

## 2. Tag-STEP - Outil d'implantation

Afin de supporter les concepts exposés dans le chapitre III, nous avons développé un outil d'implantation permettant de générer, à partir d'un modèle logique EXPRESS et de données associées sous la forme de fichiers neutres P21 :

- les modèles de données sur les deux types de supports : modèle logique sur la base de données relationnelle et modèle physique sur les étiquettes électroniques,
- l'interface d'accès aux données du système d'information nomade.

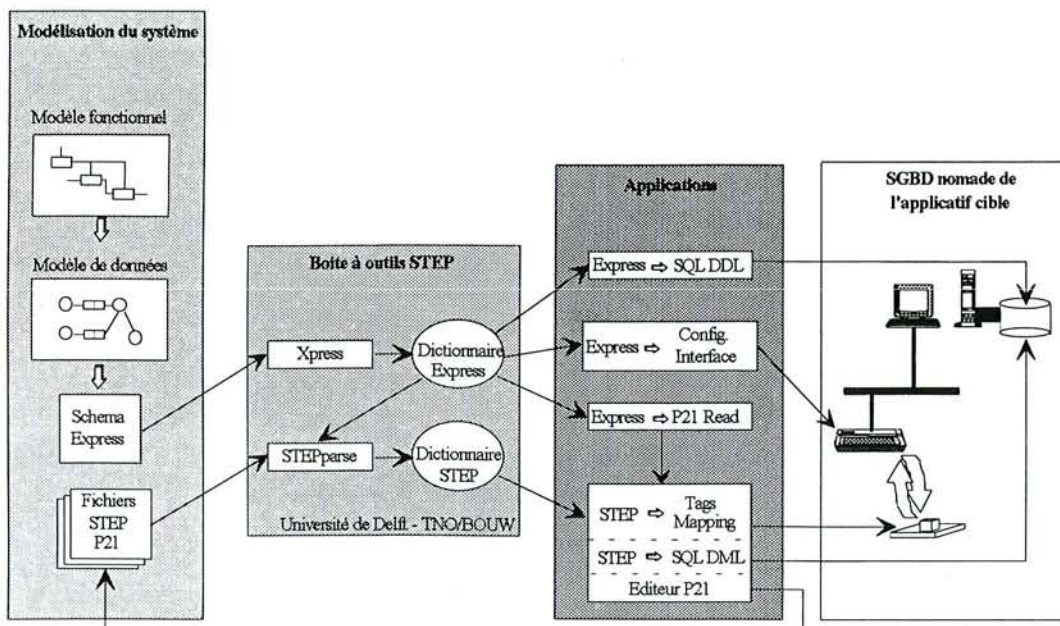


Figure IV.1: Vue globale de l'outil [CHA 95].

<sup>1</sup> © ORACLE Corporation

<sup>2</sup> Atelier Inter-établissements de Productique.

Il est pratique courante de construire des outils par assemblage d'outils réalisant des opérations unitaires indépendantes et sérialisables. Cette démarche, prise en compte pour la programmation de l'outil Tag-STEP, a nécessité la réalisation d'un ensemble de logiciels remplissant chacun une tâche bien particulière. La remise en cause de choix initiaux, la base de données ORACLE par exemple, est d'autant plus simple que les différents programmes sont indépendants. L'ensemble des opérations que nous allons détailler est schématisé sur la figure IV.1 précédente. Leur utilisation permet de valider le concept d'objet nomade par traitement automatique du modèle de connaissance du produit.

L'environnement de travail ou "Boîte à Outils" présenté figure IV.2, assure la couverture de la phase de conception-validation des modèles de contrôle-commande. Il est composé :

- de l'outil Tag-STEP (environ 2 Mo de code exécutable) pour la génération automatique du système d'information à objets nomades de production,
- d'un éditeur de modèles de données selon le formalisme EXPRESS-G,
- de l'outil d'analyse fonctionnelle ORCHIS,
- de l'outil de spécifications exécutables SPEX,
- de l'outil permettant la définition des gammes d'assemblages sur la cellule AIP,
- de l'interface applicative pour Windows (API) générée automatiquement,
- de l'outil Fedex du NIST permettant une analyse redondante des modèles EXPRESS afin de vérifier la validité des informations fournies par notre analyseur.

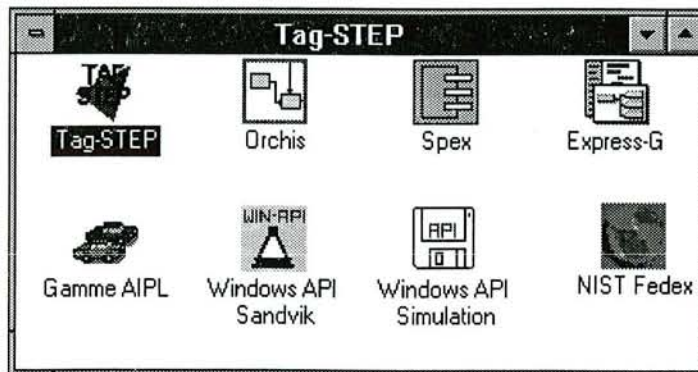


Figure IV.2: L'environnement de Travail.

## 2.1. Analyse lexico-grammaticale (STEP Toolkit)

### 2.1.1. Outil Xpress

L'outil **Xpress** effectue une analyse lexico-grammaticale et vérifie la cohérence des modèles de données EXPRESS selon les règles décrites dans la norme. Cette analyse permet :

- de détecter les erreurs et incomplétudes du modèle,
- de stocker le modèle dans une base de données accessible à l'ensemble de nos outils, au travers de primitives informatiques.

Cet outil prototype, développé dans le cadre du projet Esprit COMBINE à l'université hollandaise de Delft, constitue la base de notre plate-forme de développement.

L'analyse des modèles étant réalisée grâce à des générateurs de compilateurs - Lex et Yacc - du monde UNIX, on parle souvent, par abus de langage, de compilation du modèle de données.

### 2.1.2. Outil STEPparse

Tout comme l'outil Xpress, l'outil **STEPparse** (Université de Delft) effectue une analyse lexicale et grammaticale des fichiers neutres STEP.

Associé aux résultats de l'analyse du modèle EXPRESS, il permet de détecter partiellement les incohérences des données par rapport à leur modèle, et donc réduit le risque d'introduction d'erreurs dans le système. Certains outils commerciaux effectuent une vérification totale de la cohérence de données, différenciant le prototype de l'outil finalisé.

Ces deux outils offrent un accès à la fois au modèle et aux données par l'intermédiaire de primitives appelées depuis un programme écrit soit en langage C soit en C++. L'ensemble des primitives pourrait simplement être remplacé par des appels à des fonctions SDAI. Dans cette configuration, les programmes que nous avons développé peuvent s'intégrer sur une plate-forme du type Esprit NEUTRABAS [FER 91].

## 2.2. Traitement du modèle de données

### 2.2.1. Génération de la base de données relationnelle

L'étude de la génération d'un modèle relationnel apte à stocker des instances d'un modèle EXPRESS a été l'une des premières réalisations du NIST [EGG 88] dans le projet "PDES Testbed" dans le cadre du programme CALS. Cette mise en oeuvre a entre autres été faite par les partenaires des projets NEUTRABAS [FER 91] et CIM-BIOSYS [WES 94].

Les bases de données relationnelles étant les supports privilégiés de données utilisés dans l'industrie, il était en effet essentiel de valider l'hypothèse selon laquelle la mise en oeuvre des modèles STEP ne remettait pas en cause les systèmes actuels.



Depuis, d'autres outils ont été développés, afin d'utiliser des bases de données objets telles que MATISSE ou ONTOS [GOH 94]. Nous pouvons par ailleurs remarquer qu'à l'heure actuelle aucune base de données spécifiquement EXPRESS-SDAI n'est commercialisée, ni même proposée à l'état de prototype, ce qui prouve finalement que les SGBD actuels permettent de remplir cette tâche sans difficultés majeures.

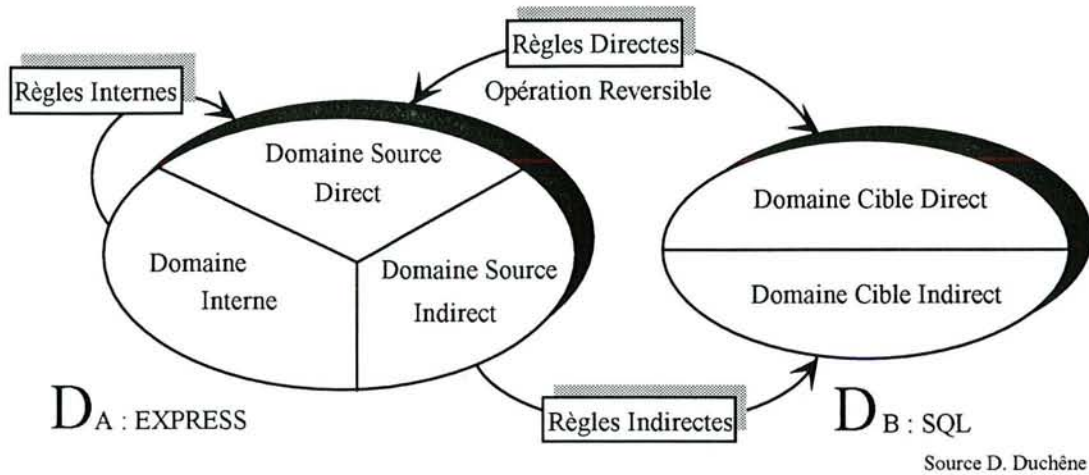


Figure IV.3: Méthode de traduction de modèles.

La traduction des modèles exprimés dans un formalisme en des modèles équivalents dans un autre formalisme nécessite la formalisation de trois types de règles de conversion, ou règles de méta-traduction : *directe*, *interne* et *indirecte* (cf. figure IV.3) qui vont définir ainsi les trois domaines *direct*, *interne* et *indirect* [DUC 94]:

- les règles directes permettent le passage du domaine source vers le domaine cible sans perte de sémantique. Il s'agit d'exprimer les mêmes concepts dans les deux langages, en transformant un groupe grammatical source en un groupe grammatical cible équivalent. Dans notre cas, on peut citer "*Toute entité EXPRESS est traduite sous forme d'une table contenant au moins une colonne OID (Object Identifier)*",
- les règles internes sont des règles imposées au sein du domaine source. Ces règles réalisent une ré-écriture simplifiée des modèles dans le langage source. L'objectif est de pouvoir appliquer les règles directes de traduction. Dans un tel cas, il y a presque toujours une perte de sémantique. On peut citer "*Tout attribut dont le type est redéfini dans une entité conserve son typage initial*",
- les règles indirectes sont des algorithmes de translation réalisant le passage du domaine indirect-source au domaine indirect-cible. Ces règles permettent d'exprimer une sémantique ne pouvant être traitée dans le domaine direct-source. On peut citer "*Toute entité ayant des sous-types est représentée sous la forme d'une vue relationnelle construite par projection de l'ensemble de ses sous-types et union*".

EXPRESS	Code SQL
<pre>ENTITY Cellule;   Id_Cellule: INTEGER;   Nom: STRING(20); END_ENTITY;</pre>	<pre>CREATE TABLE Cellule ( OID INTEGER PRIMARY KEY,   Id_Cellule INTEGER NOT NULL,   Nom CHAR(20) NOT NULL, )</pre>
<pre>ENTITY Machine;   Id_Machine: INTEGER;   Nom: STRING(20);   Localisation: Cellule; END_ENTITY;</pre>	<pre>CREATE TABLE Machine ( OID INTEGER PRIMARY KEY,   Id_Machine INTEGER NOT NULL;   Nom CHAR(20) NOT NULL,   Localisation: INTEGER NOT NULL )</pre>

*Figure IV.4: Exemple de traduction d'EXPRESS vers SQL.*

Dans le cadre d'une traduction d'EXPRESS vers l'algèbre relationnel SQL, l'existence de règles indirectes implique que l'accès transparent à la base de données ne peut être réalisé qu'au travers d'une interface, développée dans un langage procédural, faisant office de couche de remise en forme du modèle. Ce type de contraintes identifié au CRAN par [COU 91] existe également lors de la traduction d'un modèle entités-relations étendu utilisant le sous-typage vers son modèle relationnel équivalent.

Des outils commerciaux, mettant en oeuvre tout ou partie de l'ensemble des règles de conversion, sont disponibles (Sociétés STEPTools, Siemens Nixdorf, ...). Quant à l'outil prototype que nous avons développé, il n'applique que partiellement l'ensemble des règles de conversion proposées par le NIST, imposant ainsi des restrictions sur le modèle de données.

Une fois généré, le script de création du modèle de la base de données (cf. figure IV.4) doit être transféré sur une base respectant la norme SQL (en l'occurrence ORACLE 7.0).

### 2.2.2. Interface SDAI

Conformément aux spécifications exposées dans le chapitre III, l'interface applicative permet un accès à l'ensemble des attributs caractérisant un produit particulier, que ceux-ci soient stockés sur la base de données ou sur une étiquette électronique.

L'interface SDAI développée, par traitement automatique du modèle EXPRESS, offre les sept services élémentaires définis dans le document de pré-normalisation [ISO 3]:

- accès en lecture à la valeur des attributs <GetAttr>,
- accès en écriture à la valeur des attributs <PutAttr>,

- désactivation de la valeur des attributs `<UnsetAttr>`,
- accès en lecture à la valeur des attributs du type tableau ou liste `<IndexedGet>`,
- accès en écriture à la valeur des attributs du type tableau ou liste `<IndexedPut>`,
- désactivation de la valeur des attributs du type tableau ou liste `<IndexedUnset>`,
- accès au dictionnaire de données `<GetEntityDefinition>`.

Associés à ces sept primitives d'accès aux données, d'autres services plus spécifiques à notre application permettent :

- la vérification d'existence d'une instance particulière du modèle de données `<ExistInstanceProductModel>`,
- le téléchargement depuis la base de données de l'image d'une étiquette vierge associée à un produit `<InitTag>`,

Il est à noter toutefois que le code généré souffre de quelques manques :

- la traduction automatique des attributs dérivés n'est pas réalisée (résultat d'un calcul simple ou paramètre de retour d'une fonction EXPRESS),
- la vérification des règles de cohérence n'est pas mise en oeuvre. Ces deux derniers points nécessitent la traduction en langage C des règles écrites en EXPRESS, travail relativement ardu mais non impossible,
- il ne permet pas la création dynamique d'instances d'entités contenues dans le modèle de données.

Pour illustrer la difficulté de la tâche et surtout l'investissement que le développement d'un traducteur exhaustif de code EXPRESS vers le C ou le C++ suppose, nous prendrons en exemple l'outil ECCO développé par l'*Institut für Rechneranwendung in Planung und Konstruktion (RPK)* de l'Université de Karlsruhe. Cet outil traitant plus particulièrement du code EXPRESS-C défini dans le cadre du projet ESPRIT PISA [BEE 93], rendu public depuis Juillet 1995, a nécessité un travail de programmation estimé à deux années-homme.

Le code résultant du traitement du modèle de données (cf. figure IV.5) doit être lié à une librairie offrant les services de communication avec les postes (accès à la couche réseau application), les services de recherche du support de stockage et de lecture/écriture de données sur ces différents supports. Plus précisément, cette ou ces librairies procurent un accès à l'ensemble des fonctions que l'on retrouve dans la figure IV.5 :

- <TAGS\_Int>, <TAGS\_Str>, <TAGS\_EntRef>, ... : lecture des attributs typés,
- <Send\_Value>, <Send\_Error> : émission du résultat d'une requête vers le poste,
- ainsi que les fonctions de réception et de pré-traitement des requêtes : activation des fonctions de lecture d'une instance d'une entité telles que <TAGSREAD\_Machine>, <TAGSWRITE\_Machine>, ...

**EXPRESS;**

```
ENTITY Machine;
    Id_Machine : INTEGER;
    Name : STRING (20);
    Localisation : Cellule;
END_ENTITY;
```

**Interface IDTAGS (Messages GetAttr et IndexedGet):**

```
void TAGSREAD_Machine (id_Entity, id_Inst, Attribut)
int id_Entity, id_Inst, Attribut;

{/* Physical position of attribut in data block */
int      P_Id_Machine=0, P_Name=2, P_Localisation=23;
/* Attributs Value */
int      Id_MachineVal;
char     NameVal[21];
int      LocalisationVal;

if (LockInst (id_Entity, id_Inst)==TRUE)
switch (Attribut)
{
    case 1 :
        TAGS_Int (P_Id_Machine, &Id_MachineVal);
        Send_Value (2, &Id_MachineVal);
        break;
    case 2 :
        TAGS_Str (P_Name, NameVal);
        Send_Value(strlen(NameVal), NameVal);
        break;
    case 3 :
        TAGS_EntRef(P_Localisation, &LocalisationVal);
        Send_Value(2,&LocalisationVal);
        break;
    default :
        Send_Error (ATTRIBUT_NOT_DEFINED);
};
else Send_Error (ENTITY_INST_NOT_FOUND);
}
```

*Figure IV.5: D'EXPRESS vers l'interface applicative [CHA 9X].*

Cette interface est utilisée comme support de l'interface métier sur les contrôleurs ouverts d'étiquettes (cf. figure IV.6).

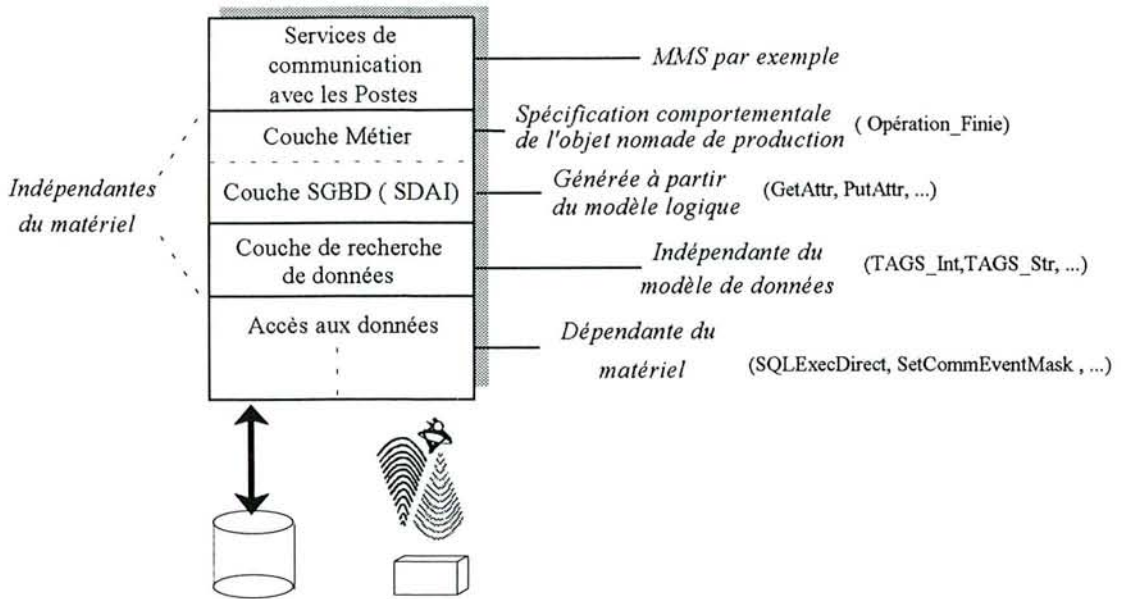


Figure IV.6: Détail de la structure en couches du module supportant l'A.P.I.

Ce code est également utilisé dans un but de prototypage de l'application cible en lien avec l'outil de spécifications comportementales SPEX fonctionnant dans l'environnement Windows. Sous Windows, la couche d'accès aux données est réalisée par des appels aux fonctions ODBC (Open DataBase Connectivity de Microsoft) pour accéder à la base de données SQL et par lectures/écritures sur des fichiers pour les étiquettes électroniques simulées.

On notera que cette interface pourrait être utilisée avec d'autres outils de simulation. Nous pensons notamment aux outils permettant le dimensionnement des ateliers par simulation de flux, tel que SIMAN. Le principe général restant identique à celui présenté sur la figure IV.7.

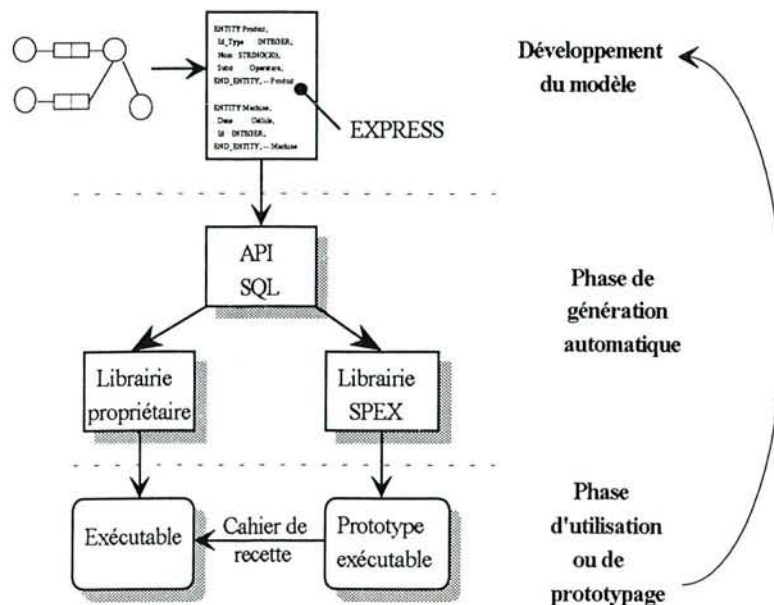


Figure IV.7: Du modèle EXPRESS au prototype puis à l'API.

### 2.2.3. Les convertisseurs de fichiers neutres

Les fichiers neutres STEP, instances d'un modèle EXPRESS donné, doivent être traités afin de pouvoir convertir les données qu'ils contiennent en formats plus spécifiques de la base de données et d'images binaires d'étiquettes électroniques.

Cette conversion nécessite de posséder à la fois le modèle de données de définition, le fichier de données et les règles de conversion. La solution préconisée par le NIST pour ce type d'opérations consiste à créer des programmes exécutables spécifiques au modèle de données plutôt que d'utiliser un programme totalement générique apte à convertir tout fichier de données, contenant des instances de n'importe quel modèle. Ceci se justifie par la taille largement plus réduite des fichiers exécutables générés.

Le modèle de données est donc converti sous la forme de programmes en langage C, aptes à faire migrer des données d'un fichier neutre vers la base de données et vers l'étiquette.

## 2.3. Traitement des données du modèle

### 2.3.1. Répartition des données

Ne disposant pas, à l'heure actuelle, de modèles aptes à guider nos choix quant à la meilleure solution en terme de distribution des données, cette opération est réalisée manuellement.

L'opérateur doit toutefois tenir compte des trois points suivants, que nous présentons comme les règles élémentaires de distribution :

- les étiquettes électroniques peuvent contenir des données à la fois statiques et dynamiques,
- la base de données ne peut contenir que des données statiques,
- une instance d'une entité EXPRESS est en totalité sur un des deux supports permis. Autrement dit, il est impossible de stocker un attribut sur l'étiquette et un autre sur la base de données. Il s'agit là toutefois d'un choix arbitraire lié à la simplification des différents logiciels créés ou générés (API, convertisseur EXPRESS->SQL, ...).

Qui plus est, avant le transfert de l'image de l'étiquette sur la base de données, il convient de vérifier que celle-ci n'excède pas la capacité réelle de l'étiquette utilisée.

### 2.3.2. Base de données relationnelle

Les instances d'entités, après vérification de cohérence, sont transmises à la base ORACLE sous la forme d'un fichier SQL. L'exemple suivant (cf. figure IV.8) présente une telle migration en considérant un fichier neutre instance du modèle présenté figure IV.4.

Fichier Neutre STEP	Code SQL
#4=Cellule (12,"Cellule AIP");	INSERT INTO Cellule VALUES (4,12,'Cellule AIP');
#5=Machine (1,"Machine 1",#4);	INSERT INTO Machine VALUES (5,1,'Machine 1',4);
#6=Machine (2,"Machine 2",#4);	INSERT INTO Machine VALUES (6,2,'Machine 2',4);

*Figure IV.8: Migration de données d'un fichier neutre STEP vers un SGBD SQL.*

### 2.3.3. Modèle physique de données sur les étiquettes électroniques

Contrairement au problème posé par la base de données relationnelle, il n'existe pas de modèle de structuration de données sur les systèmes à base d'étiquettes électroniques. Cela nous laisse donc libre de définir la structure la plus adéquate, en fonction du formalisme du modèle logique de données retenu, soit EXPRESS. La difficulté est essentiellement liée au fait que nous sommes dans l'obligation de descendre au niveau de l'octet, comme cela a d'ailleurs été le cas pour bon nombre de générations d'automaticiens (ceci est toujours le cas pour mettre en oeuvre des systèmes d'identification automatique par étiquettes électroniques).

La structure choisie (cf. figure IV.9) est le résultat de plusieurs expérimentations. Elle offre des performances acceptables, mais ne peut être considérée comme optimisée selon un critère précis, si ce n'est celui de la simplicité (il ne s'agit ici que d'un prototype).

La génération de l'image de l'étiquette selon ce modèle physique de données est réalisée en deux passes successives de compilation. La première passe permet de générer les zones mémoires Fichier\_Identification (les données de l'unique instance de l'entité Produit), Fichier\_Données (les données des autres instances d'entités) et Fichier\_Liste (permet le stockage des attributs de types LIST, SET, BAG, ARRAY). La seconde passe permet de créer la zone Fichier\_Allocation équivalente à la FAT<sup>3</sup> (Files Allocation Table) de MS-DOS.

L'utilisation d'étiquettes de tailles supérieures à 64 Ko nécessite la modification de cette structure mémoire car les pointeurs utilisés sont codés sur 16 bits.

<sup>3</sup> Bloc mémoire particulier permettant d'accéder rapidement aux fichiers contenus sur un disque ou une disquette

```

SCHEMA Mapping_Etiquette;

ENTITY Memoire_Etiquette;
    Fichier_Identification : T_Identification;    -- Instance de l'entité PRODUIT
    Fichier_Allocation    : T_Allocation;        -- Table d'allocation des instances d'entités
    Fichier_Donnees       : T_Donnees;          -- Valeur des instances des entités
    Fichier_Listes        : T_Listes;           -- Cas particulier des attributs LIST, SET, ...
END_ENTITY;

ENTITY T_Identification;
    Id_Type_Produit      : INTEGER(2);    --
    Id_Produit           : INTEGER(2);    -- L'entité Produit possède obligatoirement ces attributs
    Set_Unset            : INTEGER(2);    -- Permet de gérer l'état de 16 attributs scalaires
    Valeur_Attributs     : OPTIONAL BINARY; -- Valeur de tous les autres attributs sauf les listes
END_ENTITY;

ENTITY T_Allocation;
    Nb_Instances_Entites : LIST [0:?] OF INTEGER(2);    -- Nb d'instances de chaque ENTITY
    Instances             : LIST [0:?] OF Une_Instance;
END_ENTITY;

ENTITY T_Donnees;
    Set_Unset            : INTEGER(2);    -- Permet de gérer l'état de 16 attributs pas entité
    Donnees_Instance    : LIST[0:?] OF OPTIONAL BINARY; -- Données brutes pointées
                                                                -- dans "Une_Instance"
END_ENTITY;

ENTITY T_Listes;
    Donnees              : LIST [0:?] OF Une_Liste;    -- L'ensemble des listes
END_ENTITY;

ENTITY Une_Instance;
    Identifiant          : INTEGER(2);    -- Identifiant dans le fichier P21
    Pointeur_Donnees     : INTEGER(2);    -- Pointeur dans la zone instance de "T_Donnees"
    UNIQUE
        Identifiant;
        Pointeur_Donnees;
END_ENTITY;

ENTITY Une_Liste;
    Nombre_Elements     : INTEGER(1);    -- Nombre d'éléments dans la liste
    Set_Unset           : INTEGER(4);    -- Permet de gérer l'état de 32 éléments
    Donnees             : OPTIONAL BINARY; -- Données brutes
END_ENTITY;

END_SCHEMA;

```

*Figure IV.9: Modèle physique de données stockées sur les étiquettes.*

### 3. Une Application particulière: la cellule de l'AIP Lorrain

Afin de valider les concepts et les outils définis et développés dans le cadre de ces travaux, nous avons choisi la cellule d'assemblage de l'AIP Lorrain comme application cible. Cette cellule n'est toutefois pas encore opérationnelle. De même les derniers choix concernant la partie commande et les réseaux de communication ne sont pas actuellement arrêtés. Notre

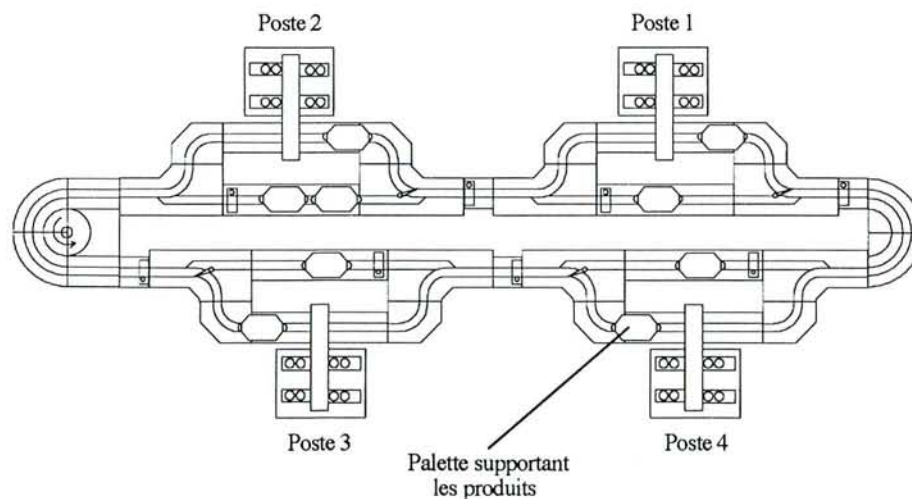


étude ne peut en conséquence aller jusqu'à la phase d'implémentation réelle avec les difficultés que cela laisse supposer. Toutefois, nous pouvons signaler à titre indicatif que l'API a été portée, en moins d'une journée, sur un système d'identification d'outils coupants Sandvik-Coromant (256 octets).

### 3.1. Description de la cellule flexible d'assemblage

#### 3.1.1. Partie Opérative de la cellule

La cellule flexible d'assemblage de l'AIP Lorrain est constituée de quatre postes identiques réalisant des assemblages de produits à partir de pièces élémentaires s'emboîtant les unes sur les autres. La figure IV.10 donne une vue schématique de cette cellule, hormis le poste de déchargement manuel, où chaque poste est positionné autour d'un anneau central permettant ainsi la distribution des palettes de et vers tout autre poste.



*Figure IV.10: Cellule d'assemblage de l'AIP Lorrain.*

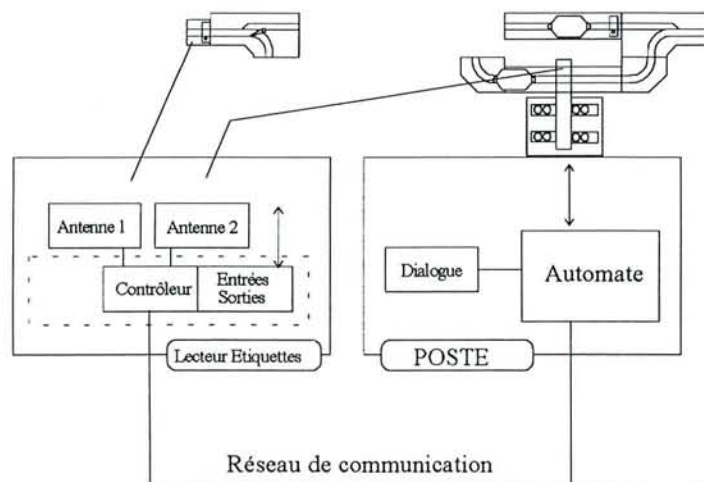
Pour des raisons d'ordre pédagogique, les quatre postes constituant la cellule sont tous identiques (partie opérative et partie commande semblables).

De telles configurations physiques imposent l'utilisation de moyens d'identification automatique des produits. Sans ces moyens, il est en effet nécessaire de réaliser un suivi de la position des produits grâce à un modèle complexe et rigide pouvant être composé de files d'attentes.

### 3.1.2. Partie Commande de la cellule

La partie commande de la cellule sera constituée d'un automate par poste (cf. figure IV.11). Cet automate sera connecté via un réseau de communication (FIP ou PROFIBUS) à l'ensemble des autres postes composant la cellule ainsi qu'à un système d'identification automatique. Ce système d'identification, utilisant deux antennes de communication, permettra de faire la lecture des étiquettes :

- en entrée du poste pour identifier les palettes à détourner,
- sur le poste même afin d'écrire les caractéristiques de l'opération réalisée (paramètres, nature, contraintes, date et heure, ...).



*Figure IV.11: Partie commande d'un poste d'assemblage.*

Afin de permettre les travaux pratiques d'automatisme, le système d'identification possédera un automate interne permettant de détourner les palettes sans intervention du poste.

Des ordinateurs individuels du type PC seront connectés à la fois sur le réseau de communication cellule et sur le réseau de communication de l'atelier (Ethernet) afin de réaliser les liaisons entre la gestion de production, le SGBD ORACLE et le pilotage cellule.

Vu les caractéristiques d'ouverture des systèmes actuels d'identification automatique, l'API devra être implantée sur un ou plusieurs de ces PC, engendrant par là-même une surcharge de communication. Lorsqu'un automate voudra émettre une requête vers le produit, il devra l'envoyer vers ce PC qui ensuite émettra des requêtes en lecture ou écriture vers le système d'IdA puis ré-émettra la réponse vers l'automate. L'objectif d'intégration de l'API sur les contrôleurs d'identification automatique ne pourra être atteint sans une collaboration étroite avec un fabricant comme cela a été le cas dans le cadre du projet Esprit MONOLIN.

### 3.1.3. Pièces et Produits assemblés

Les pièces, éléments de base de la réalisation des produits, sont de quatre types différents. Afin d'obtenir une grande diversité des produits réalisables sur la cellule et une non redondance totale des quatre postes, nous associons également un attribut de couleur à chacune de ces pièces. De par la conception mécanique du manipulateur d'assemblage des postes il n'est pas possible d'assembler plus de quatre pièces les unes sur les autres.

La cellule a donc la possibilité de fabriquer 25 variantes d'assemblage, dont quelques exemples sont présentés sur la figure IV.12 suivante. Le nombre de produits possibles, dû au choix de différentes couleurs, est donné par le polynôme  $NVP = \frac{x + 15x^2 + 15x^3 + 19x^4}{2}$ , où la variable  $x$  correspond au nombre de couleurs. Pour trois couleurs, le nombre permis de produits différents NVP est de 1041.

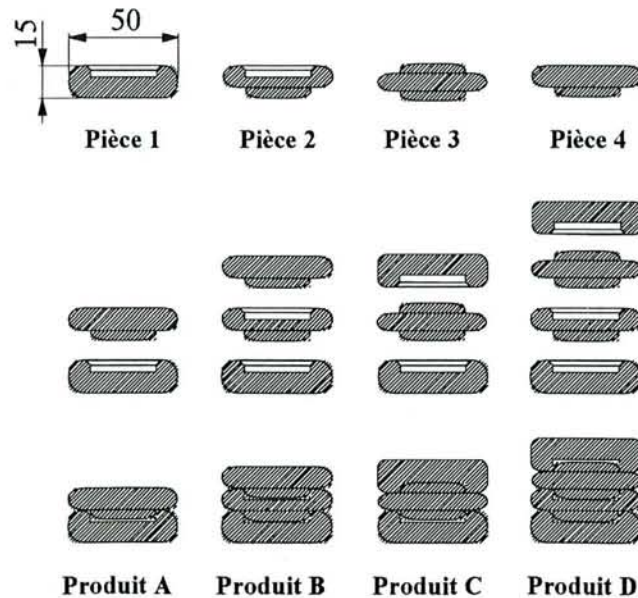


Figure IV.12: Pièces et Produits

### 3.2. Modélisation fonctionnelle de la cellule

Comme support informatique de la méthode SADT, permettant l'analyse fonctionnelle de la cellule, nous utilisons l'outil informatisé ORCHIS<sup>4</sup>. Les quatre postes présents sur la cellule sont modélisés de manière identique (cf. figure IV.13).

<sup>4</sup> © TNI Industries

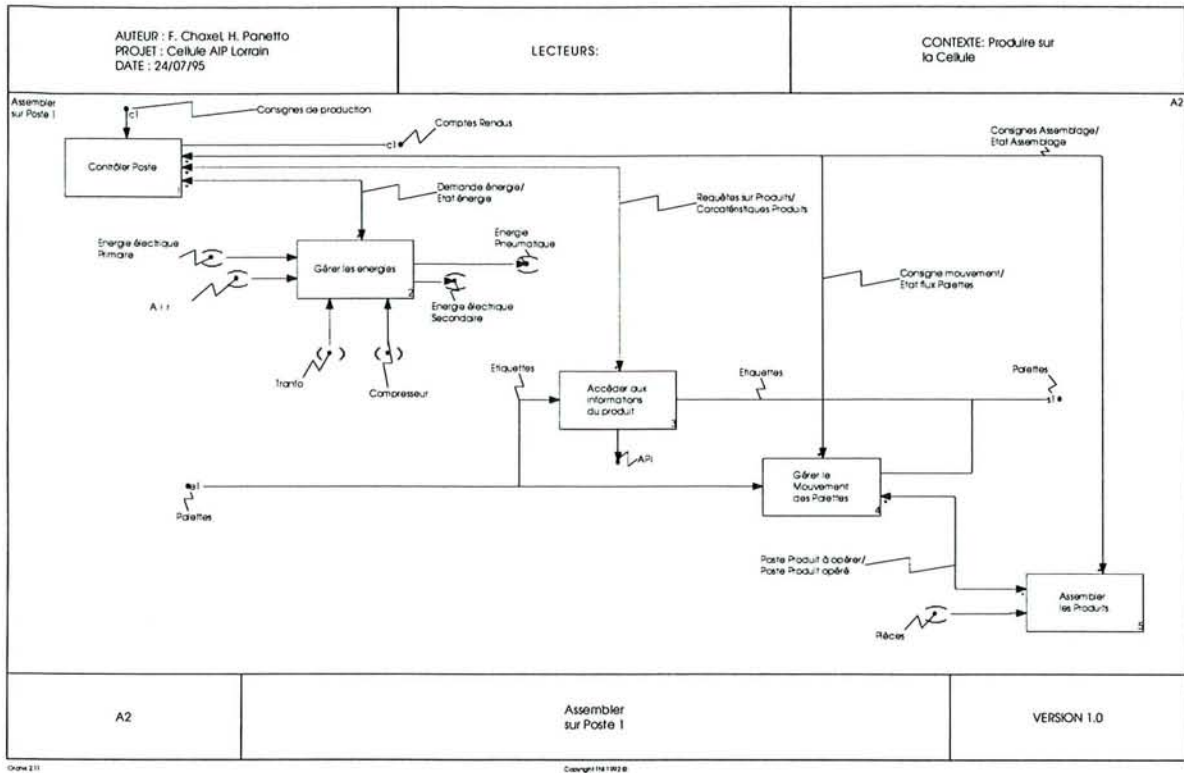


Figure IV.13: Modélisation fonctionnelle d'un poste.

Par affinages successifs de chacune des fonctions (cf. figure IV.14), nous obtenons un modèle fonctionnel complet de la cellule, statiquement validé par un cycle auteur-lecteurs.

*"Dans un deuxième temps, que la décomposition précédente soit terminée ou non, chaque fonction de contrôle identifiée est projetée sur le modèle de référence proposé par P. Lhoste servant ainsi de canevas de réflexion systématique pour la définition des fonctions de contrôle-commande élémentaires" [MOR 92b] sur l'outil GRILLE réalisé par H. Panetto [PAN 91].*

Chaque fonction de contrôle est ainsi décomposée en trois fonctions elles-mêmes re-décomposées, exprimant ainsi le modèle de référence particulier de contrôle-commande :

- composante "PILOTER" décomposée en "SUIVRE", "OPTIMISER", "REGLER",
- composante "CONDUIRE" décomposée en "PERCEVOIR", "REFLECHIR", "AGIR",
- composante "COORDONNER" décomposée en "SURVEILLER", "REAGIR", "COMMANDER".

Les fonctions terminales, non décomposables selon le modèle de référence, sont destinées à supporter du comportement. Ce comportement est spécifié, dans notre cas, en utilisant le formalisme Grafcet pour l'aspect séquentiel, le langage C pour l'aspect algorithmique et les schémas à relais pour l'aspect combinatoire.

La saisie de ces comportements est réalisée grâce à l'outil SPEX après transfert automatique des spécifications fonctionnelles de GRILLE vers SPEX. La vérification finale est réalisée par simulation synchrone du modèle à l'aide de pupitres opérateurs et/ou à l'aide de scénarii décrits en langage C.

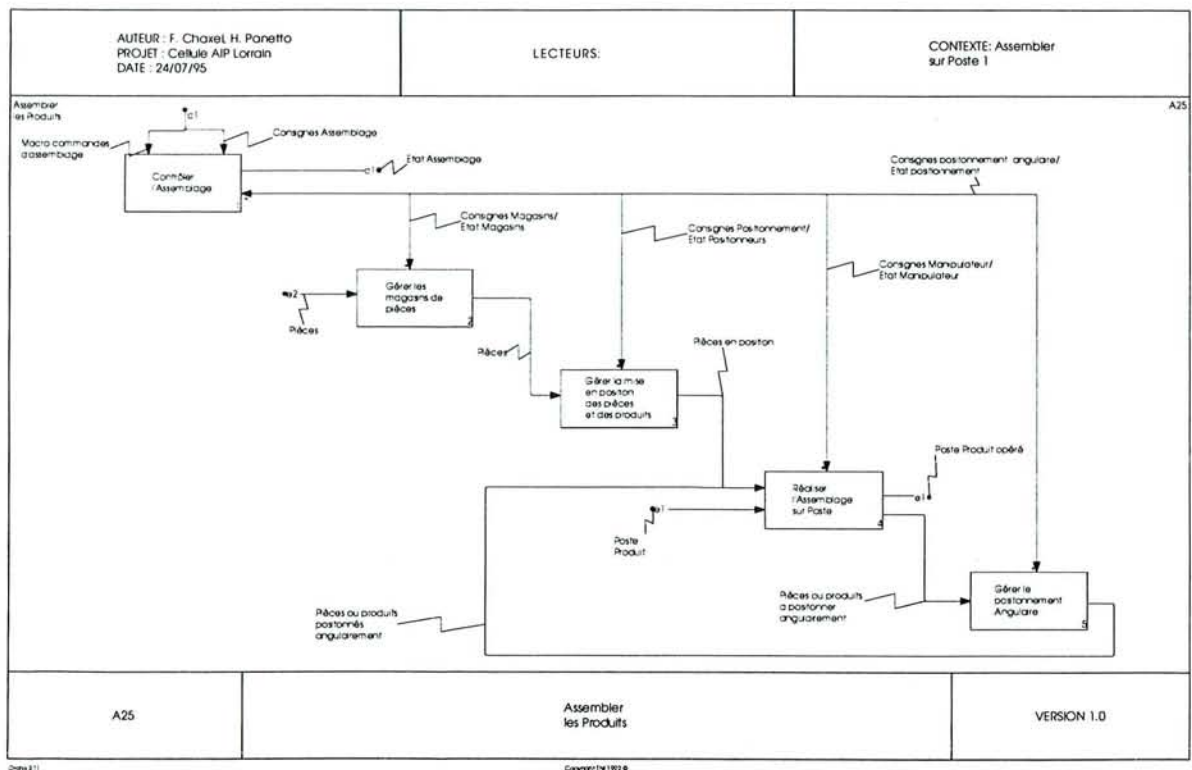


Figure IV.14: Décomposition de la fonction "Assembler sur Poste"

### 3.3. Modélisation des données associées aux produits

Le modèle de données du produit (cf. figure IV.15), bâti à partir des besoins identifiés en informations des fonctions de contrôle-commande de chaque poste, a été réalisé sur l'outil EXPREME<sup>5</sup>. Cet éditeur EXPRESS-G permet la saisie, la vérification ainsi que la génération

<sup>5</sup> © CadLab/Siemens-Nixdorf

automatique de code EXPRESS à partir du modèle. Ce modèle contient obligatoirement une entité "Produit" possédant les attributs 'Id\_Type" et "Identifiant" servant de point d'entrée du modèle nécessaire à l'outil Tag-STEP.

Les opérations subies par chaque produit sont modélisées sous la forme d'une entité sous-typée (opération d'assemblage de qualification ou de transport) et en relation avec des entités permettant de mémoriser la date et la machine effectuant ladite opération. Cela peut permettre, en fin de production de constituer la base de données de traçabilité (non réalisée).

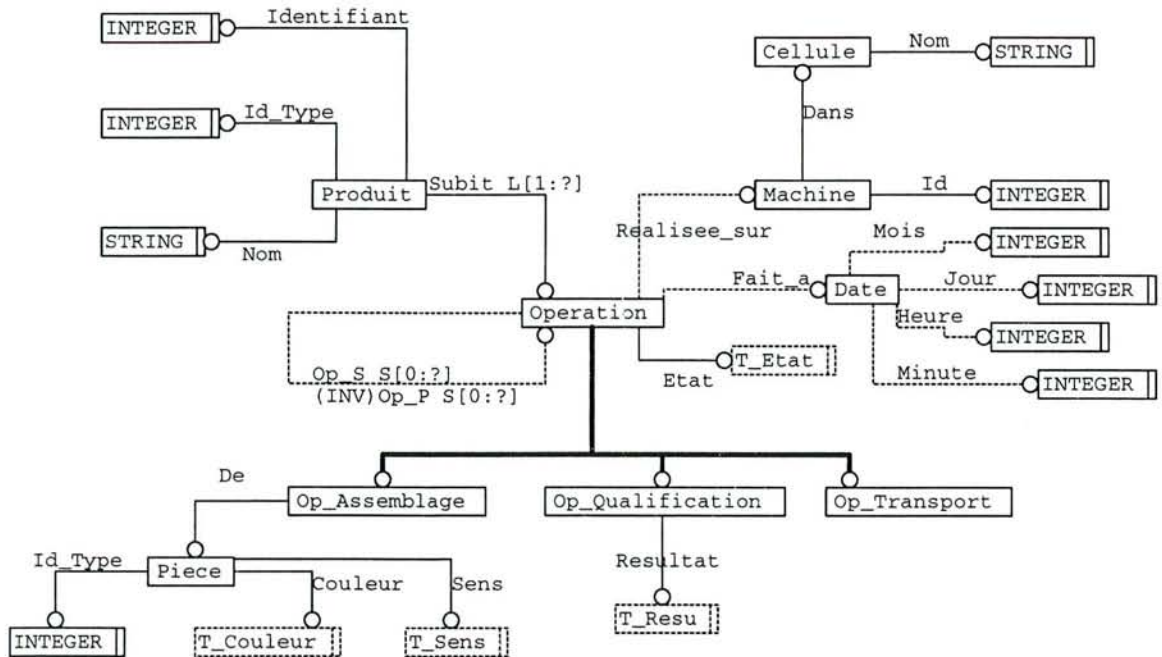


Figure IV.15: Modèle EXPRESS-G, particulier à la cellule, des données du produit.

La création d'instances à ce modèle (cf. figure IV.16) est faite, non pas par utilisation d'outils génériques tels qu'une CAO et un générateur de gammes d'assemblages, mais grâce à un programme spécifique développé sous MS-Windows (cf. figure IV.17). Cependant les travaux consistant en la définition de modèles STEP pour la fabrication [STE 95] (entités d'usinage, entités d'assemblage) permettrons l'utilisation de logiciels génériques comme source de données.

#1=Produit(2,'Produit 2',(#21,#22,#23));	#23=Op_Assemblage(\$,.PAS_FAIT.,(),#103,#33);
#2=Cellule('Cellule AIP Lorrain');	#31=Date(\$,\$,\$,\$);
#3=Machine(#2,1);	#32=Date(\$,\$,\$,\$);
#4=Machine(#2,2);	#33=Date(\$,\$,\$,\$);
#5=Machine(#2,3);	#101=Piece(2,.Vert.,.Envers.);
#6=Machine(#2,4);	#102=Piece(2,.Jaune.,.Envers.);
#21=Op_Assemblage(\$,.PAS_FAIT.,(#22),#101,#31);	#103=Piece(1,.Vert.,.Envers.);
#22=Op_Assemblage(\$,.PAS_FAIT.,(#23),#102,#32);	

Figure IV.16: Instance particulière générée par l'outil de définition de gammes.

Le modèle de données ainsi que les fichiers de définition des différents produits sont traités par l'outil Tag-STEP.

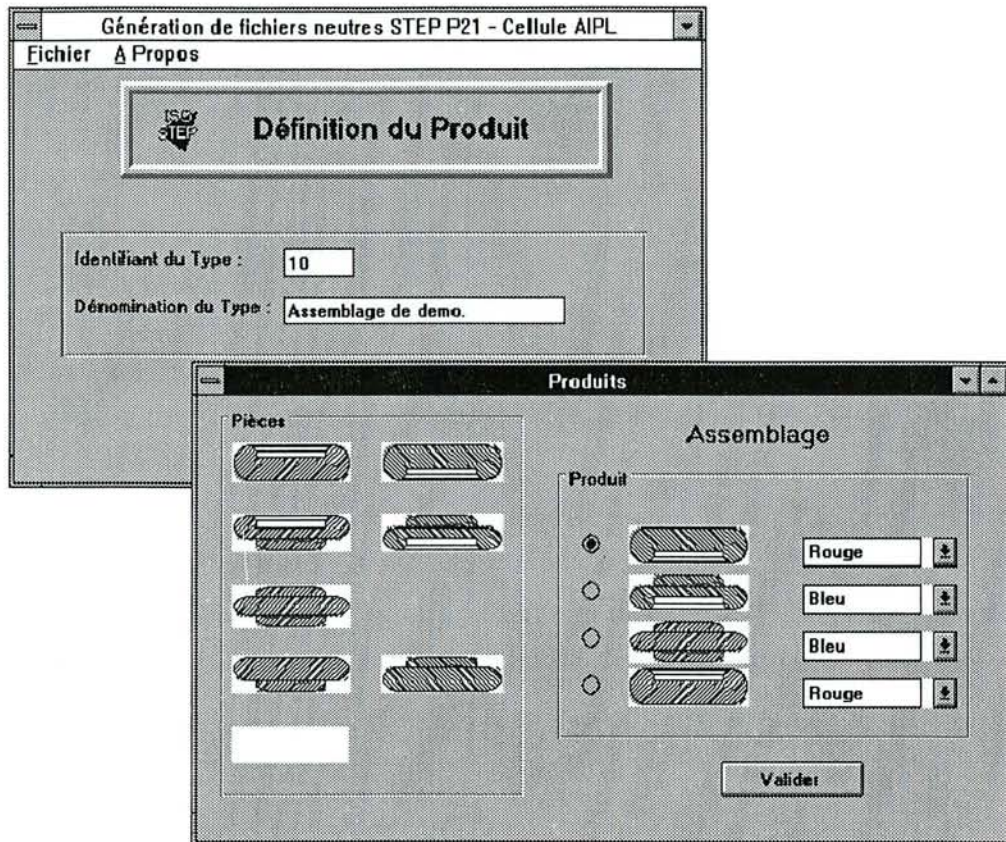


Figure IV.17: Edition de la gamme d'assemblage.

### 3.4. Formalisation des requêtes utilisateurs

Les besoins exprimés lors de l'analyse fonctionnelle ont mis en évidence la nécessité pour l'objet nomade d'offrir les services suivants :

- la fourniture du type de pièce à assembler pour l'opération courante afin que la partie commande puisse aiguiller la palette,
- la fourniture à la partie commande du sens dans lequel doit être assemblée la pièce pour l'opération courante,
- l'acquisition de l'information de terminaison de l'opération courante, permettant une mise à jour de l'état dans la gamme et l'horodatage de l'opération.

La formalisation du passage de ces fonctions utilisateur de haut niveau en un ensemble de fonctions élémentaires SDAI est réalisée par l'intermédiaire d'un diagramme FAST dont un exemple partiel est donné figure IV.18.

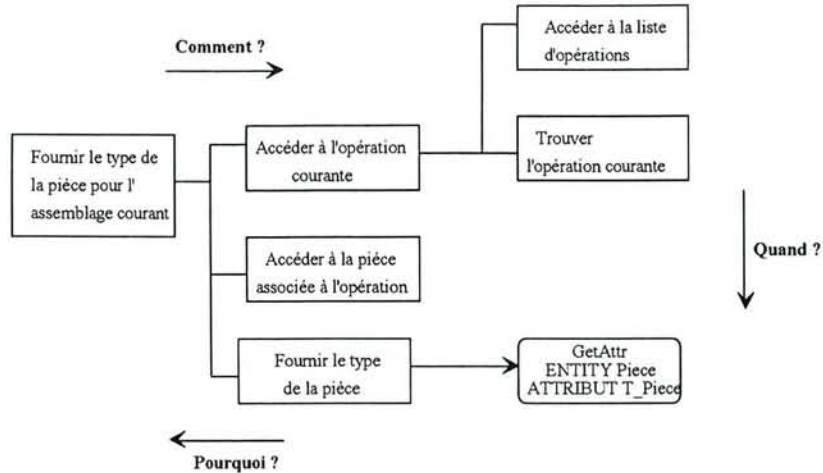


Figure IV.18: Formalisation du comportement de l'objet nomade

Les fonctions programmées en langage C des spécifications permettent de donner un comportement à l'objet nomade. Ces fonctions, interface d'accès à l'objet nomade de production, sont compilées, puis liées au code offrant les primitives d'accès SDAI. La figure IV.19 illustre la traduction de la spécification exprimée figure IV.18 permettant à l'objet nomade de répondre à la demande exprimée par la partie commande à des fins d'aiguillage de la palette.

```

/* Primitive d'accès au type de la pièce à assembler pour l'opération courante */
int Type_Piece_Next_Opp (int *Type_Piece)
{ int resu, Next_Opp, Id_Piece;
  /* recherche de la prochaine opération d'assemblage */
  resu=Next_Operation_ID(ID,&Next_Opp);
  if (resu!=0)
    return (ERREUR);
  if (Next_Opp==0)      /* Gamme d'assemblage finie */
  {
    &Type_Piece=-1;
    return (0);
  }
  /* Accéder à la pièce de l'opération courante */
  resu=GetAttrValue("Operation",(int)Next_Opp,"De",(void *)&Id_Piece);
  if (resu==ERREUR)    /* Pas de pièce associée à l'opération !!? */
    return (ERREUR);
  /* Accéder à l'attribut Id_Type de l'entité Pièce */
  resu=GetAttrValue("Piece",(int)Id_Piece,"Id_Type",(void *)&Type_Piece);
  if (resu!=0)
    return (ERREUR);
  else
    return (0);
}

```

Figure IV.19: Traduction des spécifications.



### 3.5. Prototypage de l'application par simulation sur SPEX

Une fois l'ensemble des modèles générés, nous pouvons passer à l'étape de la simulation sur l'outil SPEX. L'accès à l'API est réalisé dans les boîtes en langage C proposées par SPEX. Le code contenu dans ces boîtes, traité de façon synchrone, utilise les fonctions procurées par l'API telle `<Type_Piece_Next_Opp>`, basées sur l'ensemble réduit de fonctions SDAI retenues : `<GetAttr>`, `<PutAttr>`, `<Unsetattr>`, `<IndexedGet>`, `<IndexedPut>`, `<IndexedUnset>`, `<ExistInstanceProductModel>` et `<IniTag>`.

L'outil SPEX ne traitant que les types de données simples (entier, réel, booléen) il n'est pas possible de vérifier l'ensemble du modèle de données mais tout au moins une grande partie.

De plus, afin de limiter les développements, la base ORACLE est également utilisée par l'API de simulation, imposant donc de disposer d'un ensemble d'outils conséquent. Nous pouvons toutefois noter que l'interface ODBC permet de rendre transparent à l'application le changement de base de données (nous pourrions utiliser n'importe quelle base de données fonctionnant sous M.S. Windows voire des fichiers). Un écran de simulation est présenté sur la figure IV.20 suivante.

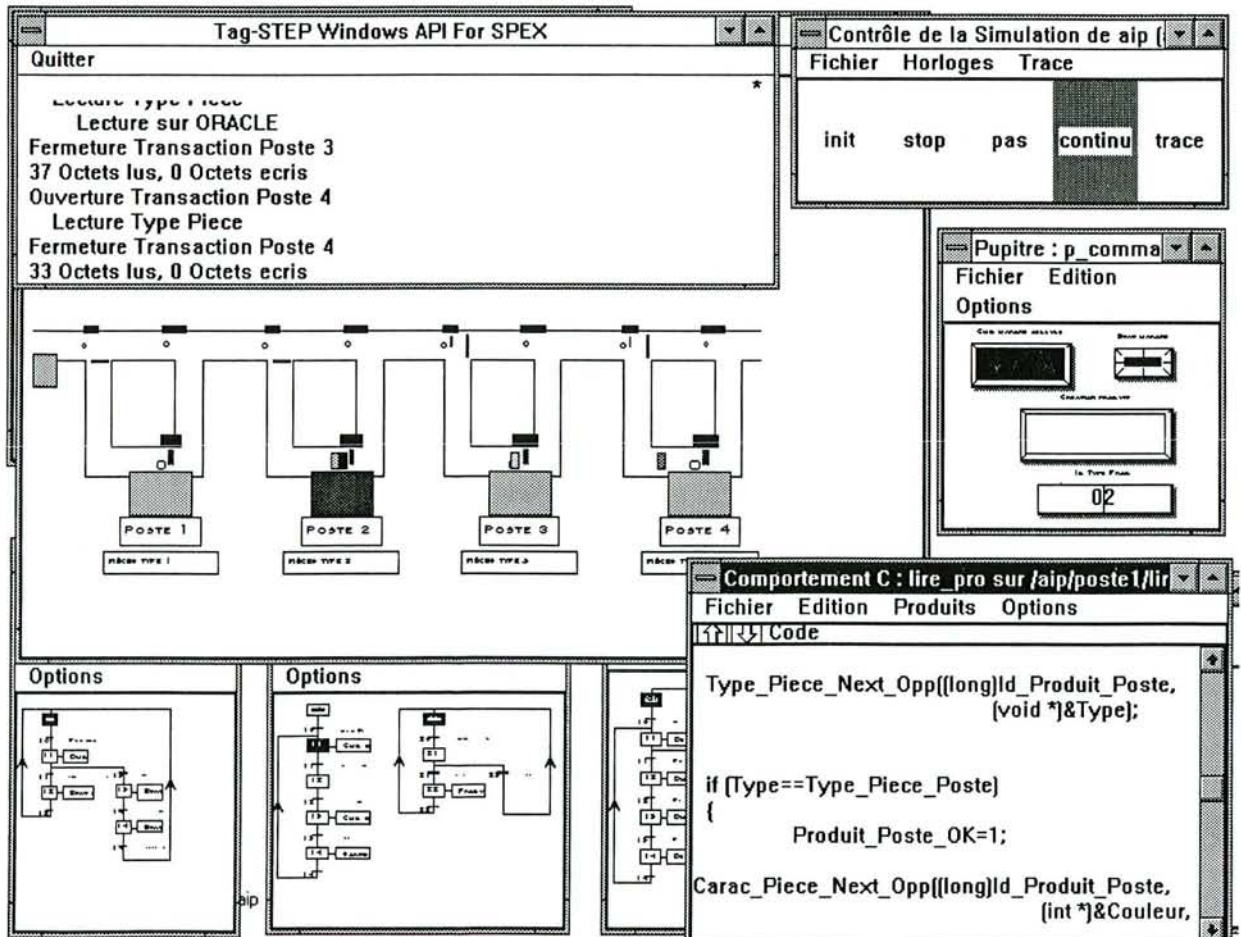
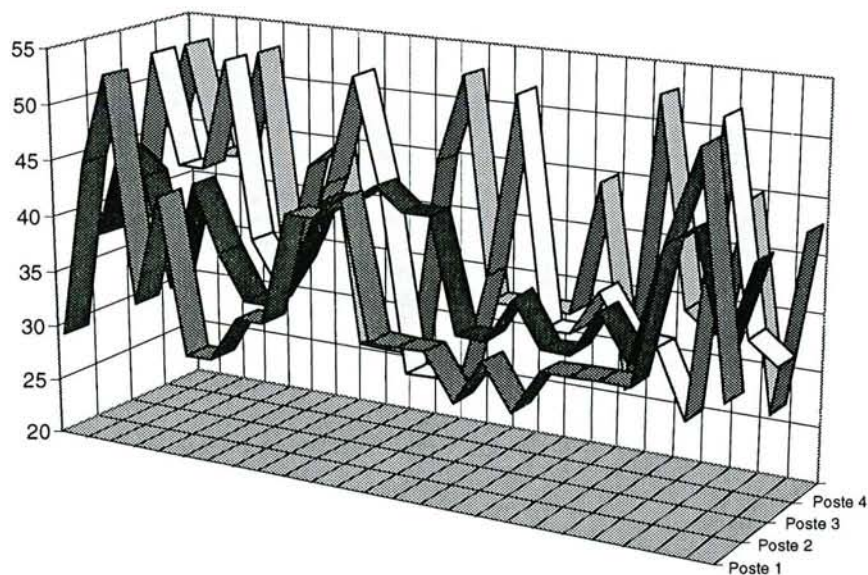


Figure IV.20: Copie d'écran de simulation sur SPEX.

Les résultats de simulations, présentés figure IV.21, sont caractéristiques du nombre d'octets lus à l'entrée de chaque poste. Les postes émettent des requêtes vers l'API pour prendre connaissance des caractéristiques des pièces à assembler. Cette information est, dans notre modèle, la seule nécessaire afin de détourner la palette. Sur l'outil SPEX, nous avons lancé différents produits ayant des gammes d'assemblage différentes. Nous pouvons constater que sur cet exemple le nombre maximum d'octets à lire est de 53. Cette valeur va nous permettre d'effectuer le choix du système d'identification, satisfaisant ce besoin, en fonction de la vitesse de défilement du tapis grâce à l'outil CASI développé par l'Institut Für Fordertechnik de l'Université de Karlsruhe (cf. chapitre III, § 3.2).

**Nombre d'octets lus par transaction**



*Figure IV.21: Accès physique aux données.*

Etant donné les caractéristiques des systèmes d'identification, l'ensemble des valeurs obtenues en nombre de lectures et écritures ne nous semble pas incompatible avec les fenêtres temporelles dont bon nombre de processus disposent. Cela prouve que la structuration des données du produit, sur des supports étiquettes électroniques, sous forme d'octets peut être dépassée, y compris dans le cadre des systèmes automatisés de production.

#### 4. Conclusion

Nous avons montré comment nous sommes passés du concept d'objets nomades de production à une mise en oeuvre particulière. La réalisation nécessaire des outils prototypes et

l'application stricte de la méthode de développement d'automatisation de la cellule ont permis de démontrer que la tâche, bien que lourde, n'est pas utopique. Le gain en flexibilité des logiciels de contrôle-commande, travaillant sur un modèle produit plutôt que sur les données des produits, permet d'envisager la production unitaire d'éléments dans des cellules, des ateliers voire inter-usines.

L'intégration et la vérification de l'adéquation des modèles et outils développés à des modèles et méthodes de conception de systèmes automatisés bâtis sur un paradigme multi-agents reste toutefois une tâche à laquelle nous commençons à nous atteler sous la forme de propositions de travaux en DEA Production Automatisée.

Cette approche remet toutefois en cause la manière d'appréhender le problème, acquise par les intégrateurs de systèmes d'identification automatique. Il ne s'agit plus seulement de maîtriser les langages de programmation métiers afin d'accéder à de l'information faiblement structurée mais de maîtriser les notions de modèles de données et de modes d'accès structurés.

De plus, bon nombre de tenants et aboutissants de cette évolution sont entre les mains des fabricants de systèmes d'identification, essentiellement des PMI-PME, ne disposant pas forcément des moyens nécessaires à la réalisation d'outils logiciels complexes.

Nous pensons néanmoins que ce changement est inéluctable.

# Conclusions et Perspectives

---

---

## 1. Conclusion

---

Le concept de produit acteur dans son processus de transformation ou d'utilisation est proposé par la communauté scientifique internationale comme une ouverture possible vers des systèmes de production plus flexibles et plus autonomes.

L'association de comportements complexes aux objets, qui circulent dans les structures de production, ne peut être atteinte sans l'utilisation de modèles de données détaillés caractérisant tant le processus que les produits. L'ensemble de ces informations technologiques est géré dans le système d'informations de la cellule, de l'atelier, ... afin d'être mis à disposition des acteurs de la production. Le problème d'échelle posé par cette nécessité de gestion d'une masse importante d'information peut être partiellement résolu par l'emploi du concept d'objets autonomes de production. Chaque objet gère l'ensemble de ses informations et les met à disposition des composants du processus considéré.

D'un point de vue technologique, le maintien de la relation bi-univoque entre l'objet physique et son modèle informatique, comportement et information, peut être assuré grâce aux technologies de l'identification automatique.

Nos travaux ont consistés à démontrer la faisabilité d'une telle approche et à proposer les outils assurant la mise en oeuvre d'un système d'information basé à la fois sur ce paradigme et sur cette technologie.

Pour ce faire, nous avons développé les mécanismes permettant d'une part d'assurer l'intégration des activités de conception et de fabrication du produit au travers des normes STEP, et d'autre part de prendre en compte ce concept d'objet nomade de production en phases de spécification-conception-réalisation du système de production. La démarche de conception proposée et appliquée à la cellule flexible d'assemblage de l'AIP Lorrain, n'étant toutefois le reflet que d'une application particulière, nous a permis d'enrichir notre savoir faire et surtout nous laisse à penser que l'API (Application Programme Interface) doit pouvoir

servir de support à l'implantation de fonctions de communication et de décision évoluées associées aux objets nomades au sens des approches multi-agents en productique.

Notre travail permet de démontrer la faisabilité technique d'utilisation des étiquettes électroniques comme de réelles bases de données mobiles, au même titre qu'une base de données en réseau. L'intégration de cette technologie au système d'information de production offre de nouvelles perspectives d'ouverture dans le domaine des échanges de données informatisés (EDI) aussi bien internes qu'externes à l'entreprise, car elle s'affranchit de toute connexion permanente à un réseau de communication.

Les propositions tendant à tirer un lien entre les activités de conception et les activités de production au travers des données du produit [STE 95] ont pour objectif de rapprocher des éléments souvent considérés comme des îlots d'automatisation et contribuent ainsi à la réduction du célèbre "time to market" ou temps de réaction de l'entreprise vis à vis du marché.

Nous considérons nos propositions comme une contribution à l'établissement de ce lien en permettant la gestion et la structuration des données des produits selon des modèles et des méthodes proches des modes de représentation et de pensée habituels dans les phases de conception des produits.

La distinction claire entre le QUOI faire sur les produits et le COMMENT faire, propriétés des machines, est une des bases à la mise en oeuvre de structures de production flexibles, c'est à dire aptes à fabriquer des produits diversifiés. En permettant une implantation séparée de ces deux aspects, nos travaux contribuent à la réalisation de systèmes flexibles et proposent des outils qui permettent une description détaillée du modèle du produit, pouvant aller jusqu'à l'entité d'usinage ou d'assemblage. Il est donc possible d'envisager la composition dynamique de la gamme sur l'outil de production et l'acquisition aisée des informations nécessaires à la traçabilité.

## **2. Perspectives**

---

D'un point de vue général, il nous paraît nécessaire d'envisager à long terme la définition des méthodes et des outils permettant la gestion intégrée des données du produit dans des contextes ouverts comme illustré figure C.1.

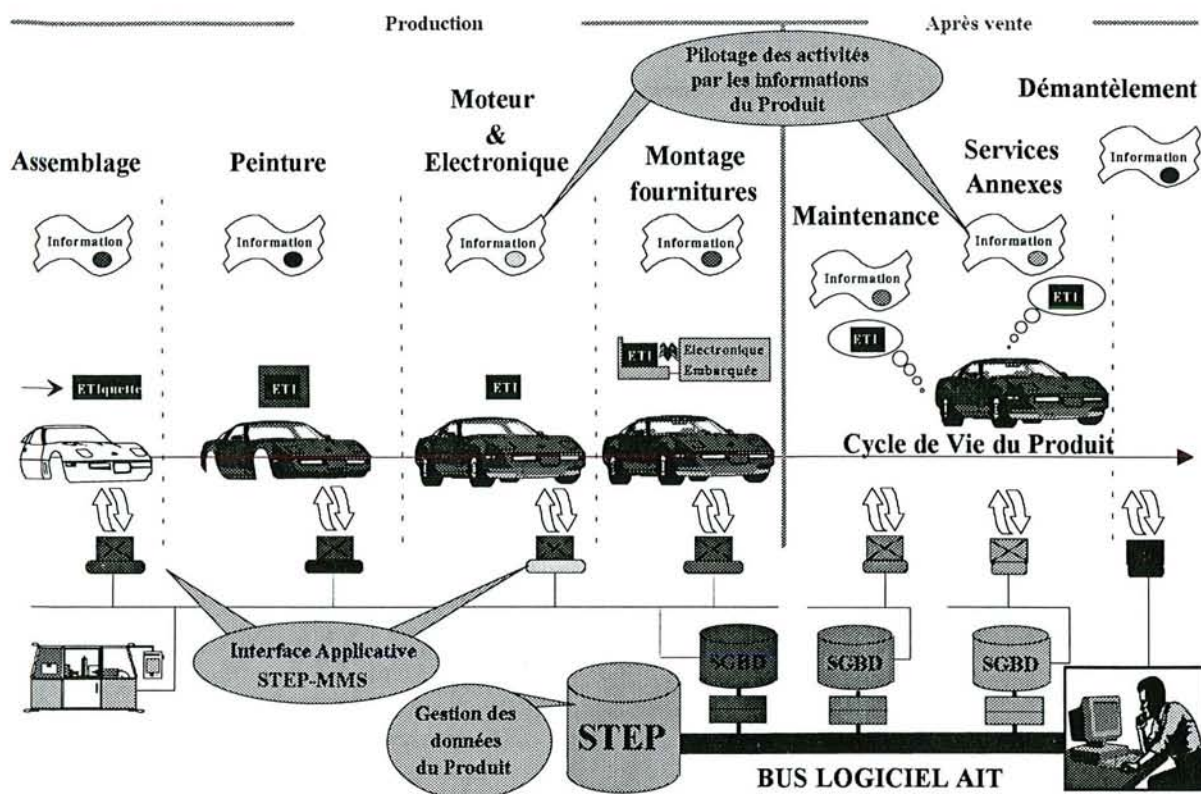


Figure C.1 : Couverture du cycle de vie d'un produit manufacturier.

En particulier, et de manière non restrictive, il nous semble prioritaire d'étudier les aspects suivants :

- la gestion multi-modèles évoluant au cours du temps afin de gérer les données du produit depuis l'ensemble des phases de fabrication jusqu'à son démantèlement. Les multiples sous-modèles ou vues du modèle global correspondent chacun à un domaine d'application particulier tels le pilotage ou la gestion de la qualité (cf. figure C.3). L'évolution du schéma de données, avec migration de tout ou partie des données de l'ancien schéma, permet de passer de la phase de fabrication à la phase d'après vente ou de faire évoluer le schéma en après vente: par exemple la durée de vie d'un véhicule, supérieure à cinq ans, nécessite sans aucun doute l'évolution du modèle de données du produit en fonction des adaptations apportées au système d'information de l'entreprise. La gestion des versions du schéma pose un problème majeur du fait de la 'disparition' des éléments de la base de données pour des durées indéterminées (quelques jours, quelques mois, quelques années),

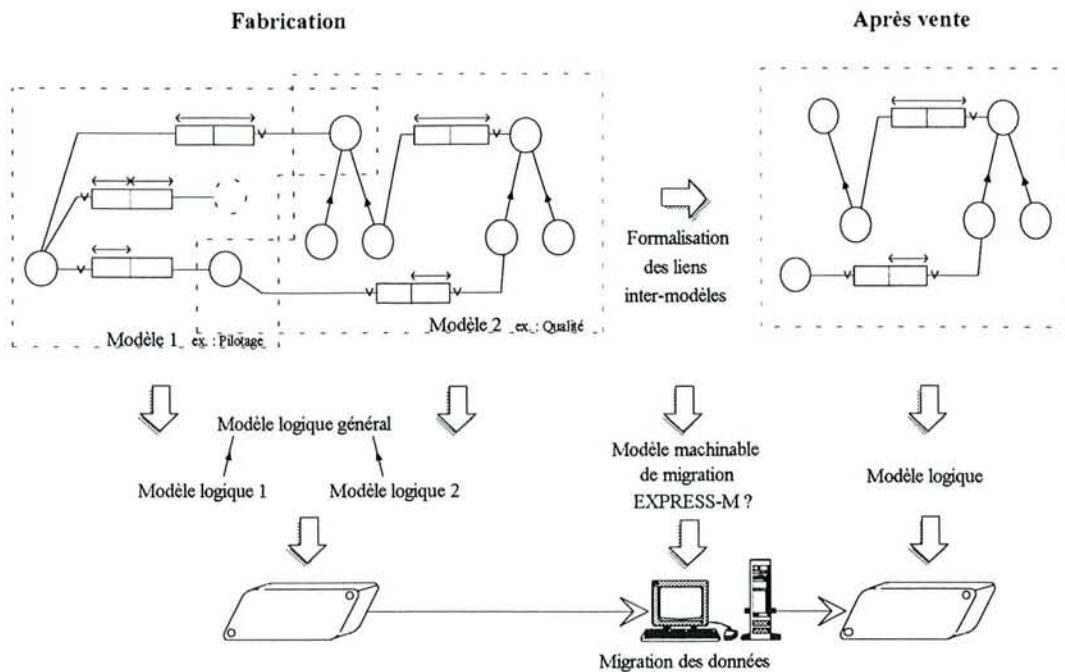


Figure C.2 : Représentation multi-modèles et évolution temporelle du modèle.

- dans le domaine automobile, la connexion des étiquettes électroniques à l'électronique embarquée des véhicules afin d'éviter les problèmes de recyclage des étiquettes en fin de lignes de fabrication et d'abaisser les coûts de cette technologie par effet de masse. De façon plus prospective, l'association d'une étiquette à tout produit, comme base d'information pour l'usage, l'historique, la gestion, la maintenance, ... La **normalisation des technologies** (fréquences, protocole de communication entre l'unité de contrôle et l'étiquette) **et des modèles de données** (modèle physique des données sur l'étiquette, modèle(s) conceptuel(s) de données) constitue dans ce cas un aspect incontournable,
- la connexion de l'interface applicative sur un profil réseau industriel normalisé à hautes performances tel que MMS. Considérant le produit comme une entité communicante, nous pensons au développement d'un "VMD Produit" bâti à partir du modèle de données EXPRESS (cf. figure C.2).
- l'intégration de l'ensemble autour d'une architecture de type AIT couvrant aussi bien les aspects temps réels en production que les aspects temps différés ou multimédia pour l'après vente.

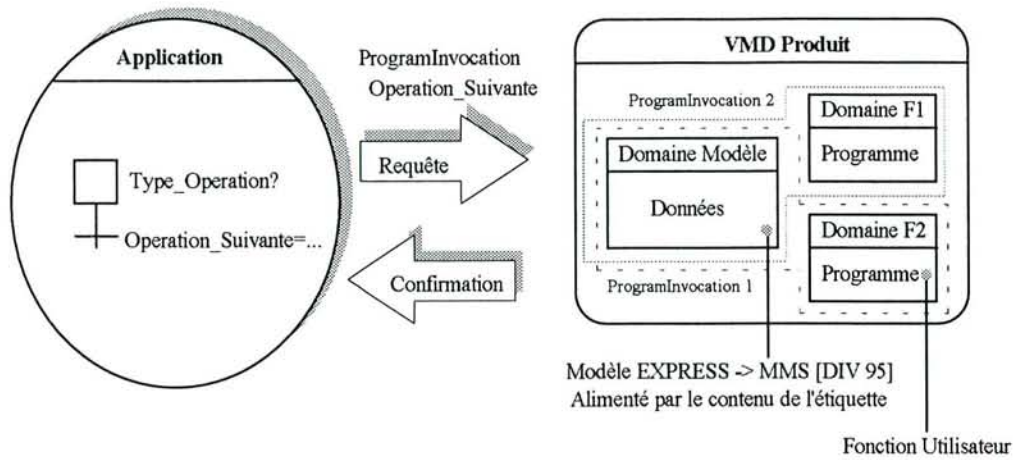


Figure C.2 : VMD Produit.

Ces travaux vont se poursuivre par un projet de recherche européen, dans le cadre d'Esprit IV, préparé en collaboration avec des constructeurs automobiles européens, des entreprises du secteur de l'identification automatique et un universitaire européen. La mise en place d'un tel projet offrira un cadre favorable à la poursuite et à l'amélioration de nos travaux pour aboutir, à terme, à une validation industrielle voire à une phase de pré-normalisation.



---

# Bibliographie

---

- [AFN 94] **AFNOR**  
*"STEP : Concepts fondamentaux"*  
Collectif d'auteurs  
Editions AFNOR, ISBN 2-12-486912-4, 1994, 175 pages.
- [ARN 95] **D. Arnold, J. Richard, E. Bajic, G. Reuter, F. Chaxel, P. Fery, J. Nowack**  
*"Electronic tags integration aspects in manufacturing production system"*  
2ème conférence internationale sur l'automatisation industrielle  
Nancy, France, 7-9 Juin 1995, pp 293-297, Vol 1
- [BAI 94] **I. Bailey, G.S. Birring, M. Mead**  
*"Latest Thinking in EXPRESS-M"*  
4th Annual EXPRESS User Group International Conference  
Greenville SC, USA, 13-14 Octobre 1994
- [BAJ 92] **E. Bajic, F. Chaxel, J. Richard**  
*"Intelligent-Parts Management in CIM"*  
1st Japanese-French Congress of Mechatronics  
Besançon, France, 20-22 Octobre 1992
- [BAJ 95] **E. Bajic**  
*"Gestion de qualité en productique par la conduite des processus et le système d'information"*  
Thèse d'habilitation à diriger des recherches, Université H. Poincaré - Nancy I, 1995
- [BEE 93] **D. Beeckman**  
*"Modélisation de l'information dans les systèmes industriels"*  
Ingénierie des systèmes d'information, ISSN 1247-0317, Vol 1, N° 3, pp 305-323, 1993
- [BEN 94a] **L. Bensaïd**  
*"Méthodologie d'intégration d'un système d'information sur support réparti de type étiquettes électroniques"*  
Rapport de Recherche de DEA Production Automatisée, CRAN-EACN, Université H. Poincaré - Nancy I, 1994
- [BEN 94b] **L. Bensaïd**  
*"Les bases de données distribuées en productique"*  
Rapport Bibliographique de DEA Production Automatisée, CRAN-EACN, Université H. Poincaré - Nancy I, 1994
- [BOH 92] **H.M. Bohms, F.P. Tolman**  
*"A Methodology for Expressing CIM Reference Architectures"*  
Robotics and Computer Integrated Manufacturing, ISSN 0736-5845, Vol 10, N° 1-2, pp 131-140, 1992

- [BON 92] **A. H. Bond, L.G. Gasser**  
*"A subject-indexed bibliography of distributed artificial intelligence"*  
IEEE Transaction on Systems, Man, and Cybernetics, ISSN 0018-9472, Vol 22, N° 6, pp 1260-1281, 1992
- [BON 95] **L. Bongaerts, P. Valckenaers, H. Van Brussel, J. Wyns**  
*"Schedule Execution for a Holonic Shop Floor Control System"*  
Preprints Advanced Summer Institute '95, ASI'95  
Lisbonne, Espagne, 25-28 Juin 1995
- [BOU 95] **M. Bouzeghoub, G. Gardarin, P. Valduriez**  
*"Du C++ à Merise Objet"*  
Editions EYROLLES, ISBN 2-212-08822-1, 344 pages, 1995
- [BRO 94] **J.Y. Bron, M. Lombard-Grégori, G. Morel**  
*"Gestion technique en productique : application à la gestion des outils d'un centre d'usinage"*  
Revue d'Automatique et de Productique Appliquée, ISSN 0990-7009, Vol 7, N°5, pp 553-562, 1994
- [BRU 91] **P. Brun, F. Fernandez, M. Lehne, A. McClelland, H. Nowacki**  
*"A neutral product database for large multifunctional systems"*  
Proceeding of the Seventh CIM-Europe annual conference - CEC DG XIII  
Turin, Italie, 29-31 mai 1991, pp 87-97
- [BRU 94a] **G. Bruno, R. Agarwal, C. Reyneri, B. Chiavola, M. Varani**  
*"Making CIMOSA operational"*  
European workshop on Integrated Manufacturing Systems Engineering, IMSE  
Grenoble, France, 12-14 Décembre 1994, pp 97-106
- [BRU 94b] **J.M. Brun**  
*"Un modèle produit : pour quoi, pour qui ?"*  
Journée PRIMECA "Modélisation Géométrique et Technologique en Conception de Produits"  
Grenoble, France, 14 Octobre 1994, pp 77-82
- [BUR 94] **J. Burnell**  
*"Beer, pigeons or boxes, RF/ID helps distribution"*  
Automatic ID News Europe, Vol 3, N° 4, pp 20-22, 1994
- [CHA 91a] **P. Charpentier, O. Garro, L. Lossent, P. Martin**  
*"Démarche de conception de systèmes automatisés de production : Application à la transitique"*  
Revue d'Automatique et de Productique Appliquée, ISSN 0990-7009, Vol 4, N° 2, pp 141-153, 1991
- [CHA 91b] **F. Chaxel, E. Bajic, J. Richard**  
*"Commande et suivi de production : proposition d'un architecture reposant sur l'identification automatique"*  
Rapport fin de contrat, CRAN-EACN/DS Innovation, Université H. Poincaré - Nancy I, Septembre 1991
- [CHA 93] **F. Chaxel, E. Bajic, J. Richard**  
*"Intelligent-parts management in a fully distributed architecture"*  
International Conference on Industrial Engineering and Production Management IEPM'93  
Mons, Belgique, 2-4 Juin 1993, Vol 2, pp 846-855
- [CHA 94a] **F. Chaxel, F. Duchscher, E. Bajic, J. Richard**  
*"Gestion des flux d'information produits dans une structure CIM"*  
Revue d'Automatique et de Productique Appliquée, ISSN 0990-7009, Vol 7, N°4, pp 401-414, 1994

- [CHA 94b] **F. Chaxel, E. Bajic, L. Bensaïd, J. Richard, A. Arnold**  
*"Distributed manufacturing Information Management for Process Control Using Electronic Data Carriers"*  
International Symposium on Automotive Technologies and Automation, ISATA 27th  
Aaren, Allemagne, 31 Octobre-4 Novembre 1994, pp 197-204
- [CHA 94c] **F. Chaxel, E. Bajic, J. Richard**  
*"From STEP Product Modelling to Product Manufacturing : an Approach using Identification Tags"*  
European workshop on Integrated Manufacturing Systems Engineering, IMSE  
Grenoble, France, 12-14 Décembre 1994, pp 529-535
- [CHA 95] **F. Chaxel, E. Bajic, J. Richard**  
*"Distributed Database Management Using Identification Systems : a STEP Based Approach"*  
5th International Conference on Flexible Automation & Intelligent Manufacturing, FAIM'95  
Stuttgart, Allemagne, 28-30 Juin 1995, pp 178-187
- [CHA 97] **F. Chaxel, E. Bajic, J. Richard**  
*"Mobile Database Nodes for Manufacturing Information Management : a STEP Based Approach"*  
International Journal of Advanced Manufacturing Technology, ISSN 0268-3768, A Paraitre en 1997.
- [CIM 91] **Esprit Consortium AMICE**  
*"CIMOSA : Open System Architecture for CIM"*  
Editions Springer-Verlag, ISBN 3-540-56256-7, 234 pages, 1991
- [CIM 94] **Collectif d'auteurs**  
*"La modélisation en entreprise. CIM-OSA et ingénierie simultanée"*  
Editions Hermès, ISBN 2-86601-419-7, 215 pages, 1994
- [CLA 92] **S.N. Clark**  
*"Transformr : A Prototype STEP Exchange File Migration Tool"*  
NISTIR 4944, National Institute of Standards and Technology, Gaithersburg, Maryland, 1992
- [COU 91] **J.F. Couturier**  
*"Auto contrôle d'une cellule flexible d'usinage. Conception et réalisation du système d'information"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1991
- [CZE 94] **A.S. Czewinski, K. Sivayoganathan**  
*"Development of CIM Applications from PDES/STEP Information Models"*  
Concurrent Engineering : Research and Application, ISSN 1063-293X, Vol 2, pp 133-136, 1994
- [DER 95] **I. Derancourt**  
*"Conduite des systèmes automatisés de production : définitions, rôles, structures et méthodes"*  
Rapport bibliographique de DEA Production Automatisée, CRAN-GGP, Université H. Poincaré - Nancy I, 1995
- [DEW 90] **M.F. Dewries**  
*"Integration of EDM within a CIM Environment"*  
Annals of the CIRP Manufacturing Technology, ISBN 3-905-277-14-X, Vol 39, No 2, 1990
- [DI 91] **A. Di. Leva, P. Giolito, F. Vernadat**  
*"M\* : méthodologie d'intervention pour l'analyse et la conception des systèmes d'information de productique"*  
Congrès Afcet : Autour et à l'entour de Merise: Les méthodes de conception en perspective  
Sophia-Antipolis, France, 17-19 Avril 1991, pp 395-408

- [DIL 91] **D.M. Dilts, N.P. Boyd, H.H. Whorms**  
*"The Evolution of Control Architectures for Automated Manufacturing systems"*  
Journal of Manufacturing Systems, ISSN 0278-6125, Vol 10, N° 1, pp 63-79, 1991
- [DIV 95] **T. Divoux, D. Duchêne, E. Rondeau**  
*"Virtual Manufacturing Devices design by means of EXPRESS analysis"*  
International Conference on Industrial Engineering and Production Management  
Marrakech, Maroc, 4-7 Avril 1995, pp 156-165
- [DOI 92] **C. Doiteaux**  
*"Auto-contrôle de cellule flexible d'usinage : étude et réalisation du système de pilotage"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1992
- [DUC 94] **D. Duchêne**  
*"Spécification d'applications en langage EXPRESS et traduction automatique en code MMS"*  
Rapport de Recherche de DEA Production Automatisée, CRAN-EACN, Université H. Poincaré  
- Nancy I, 1994
- [DUF 86] **N. A. Duffie, R. S. Piper**  
*"Nonhierarchical Control of Manufacturing Systems"*  
Journal of Manufacturing Systems, ISSN 0278-6125, Vol 5, N° 2, pp 137-139, 1986
- [DUF 89] **N.A. Duffie, R. Chitturi, J.I. Mou**  
*"Fault tolerant Heterarchical Control of Manufacturing Systems Entities"*  
Journal of Manufacturing Systems, ISSN 0278-6125, Vol 7, N° 4, pp 315-327, 1989
- [DUF 90] **N.A. Duffie**  
*"Synthesis of Heterarchical Manufacturing Systems"*  
Computer in Industry, ISSN 0166-3615, Vol 14, pp 167-174, 1990
- [DUN 95] **D. Dunlap**  
*"Automatic Manufacturing Systems with Code Data Capture"*  
5th International Conference on Flexible Automation & Intelligent Manufacturing, FAIM'95  
Stuttgart, Allemagne, 28-30 Juin 1995, pp 458-473
- [EGG 88] **J.A. Eggers**  
*"Implementing EXPRESS in SQL"*  
ISO TC184/SC4/WG1 Document N292  
National Institute of Standards and Technology, Gaithersburg, Maryland, 1988
- [FER 91] **F. Fernandez-Gonzalez**  
*"Neutral Product Definition Database for Large Multifunctional Systems - NEUTRABAS -"*  
Annual Esprit Conference  
Bruxelle, Belgique, 25-29 Novembre 1991, pp 578-592
- [FER 93] **A. Ferrarini, L. Couvreur, D. Brun-Picard**  
*"A New Decentralized Approach for F.M.S. Control"*  
7th Annual European Computer Conference, COMP EURO'93  
Paris-Evry, France, 24-27 Mai 1993, pp 410-416
- [FER 95] **A. Ferrarini, D. Brun-Picard**  
*"Model and Implementation of a Totally Decentralized Control System for Flexible Manufacturing Systems"*  
2ème conférence internationale sur l'automatisation industrielle  
Nancy, France, 7-9 Juin 1995, Vol 2, pp 603-608
- [GAR 85] **Gardarin G., Valduriez P.**  
*"Bases de données relationnelles, analyse et comparaison des systèmes"*  
Editions Eyrolles, 1985

- [GAR 91] **Y. Gardan**  
*"La CFAO : Introduction, techniques et mise en oeuvre"*  
Editions Hermès, ISBN 2-86601-293-3, 418 pages, 1991
- [GAR 92] **O. Garro**  
*"Conception d'éléments physiques de systèmes de production. Application aux machines outils à architectures parallèle"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1992
- [GIE 91] **W.F. Gieling, W.J. de Bruijn, H.M. Bohms, S. Suhm, J. Bassan**  
*"An Open Architecture for Information Integration of CIM Modules"*  
Computer Applications in Production and Engineering  
Bordeaux, France, 10-12 Septembre 1991, pp 739-748
- [GOH 94] **A. Goh, S.C. Hui, B. Song, F.Y. Wang**  
*"A study of SDAI implementation on object-oriented databases"*  
Computer Standards & Interfaces, ISSN 0920-5489, Vol 16, pp 33-43, 1994
- [GRO 95] **M. Grolee**  
*"Identification automatique : techniques et normalisations"*  
RobAut, ISSN 1249-8998, Vol 9, pp 30-35, 1995
- [GRU 94] **E. Grünenberger**  
*"PC CoDe : une méthodologie opérationnelle pour concevoir les systèmes de conduite des cellules de production"*  
Thèse de doctorat de l'Université de Tours, 1994
- [GUI 95] **G. Guilbert**  
*"Advanced Information Technology for design and manufacturing : A Model for the European Industry"*  
Preprints Esprit Advanced Summer Institute '95, ASI'95  
Lisbonne, Espagne, 25-28 Juin 1995
- [HAE 90] **J. Haenisch**  
*"CAD Exchange : Towards a First Step Implementation"*  
16th Annual Conference of IEEE Industrial Electronics Society - IECON'90  
Pacific Grove CA, USA, 27-30 Novembre 1990, pp 734-739
- [HEI 95] **T. Heikkilä, M. Järviluoma, J.M. Hasemann**  
*"Holonc Control of a Manufacturing Cell"*  
5th International Conference on Flexible Automation & Intelligent Manufacturing, FAIM'95  
Stuttgart, Allemagne, 23-30 Juin 1995, pp 861-871
- [HEN 87] **B. T. Henoch**  
*"Information architecture in CIM as related to product identification and monitoring"*  
International Symposium on Automotive Technologies and Automation, ISATA 17th  
Munich, Allemagne, Octobre 1987, pp 87104/1-8
- [HOU 92] **F.J.A.M. Van Houten**  
*"Manufacturing Interface"*  
Annals of the CIRP, ISBN 3-905-277-18-2, Vol 41, N° 2, pp 699-710, 1992
- [IGL 89] **IGL Technologie**  
*"SADT : un langage pour communiquer les idées"*  
Editions Eyrolles, 1989
- [ISO 1] **ISO 10303-11, Product Data Representation and Exchange**  
*"Description Methods : The EXPRESS Language Reference Manual"*  
International Standard, ISO TC184/SC4, 1994

- [ISO 2] **ISO 10303-21, Product Data Representation and Exchange**  
*"Clear Text Encoding of the Exchange Structure"*  
International Standard, ISO TC184/SC4, 1994
- [ISO 3] **ISO 10303-22, Product Data Representation and Exchange**  
*"Standard Data Access Interface Specification"*  
Draft International Standard, ISO TC184/SC4, 1994
- [JAR 94] **R. Jardim, M. Martins, M. Tavares, A. Steiger**  
*"SIP - a STEP-based Integration Platform. Application in Industrial Environments"*  
Advanced Summer Institute'94 in Computer Integrated Manufacturing & Industrial Automation"  
Rion Patras, Grèce, 26 Juin - 1 Juillet 1994, Vol 1, pp 45-52
- [JOR 90] **H. Jorysz, F. Vernadat**  
*"CIM-OSA Part 2 : information view"*  
International Journal of Manufacturing Systems, ISSN 0951-192X, Vol 3, N° 3 et 4, pp 157-167, 1990
- [KLA 93] **F.L. Krause et al.**  
*"Product Modelling"*  
Annals of the CIRP, ISBN 3-905-277-20-4, Vol 42, N° 2, pp 695-706, 1993
- [LEC 93] **J. Lecuire., F. Simonot, J.P. Thomesse**  
*"CIM Applications Modelling and 'Traceability': from Requirements to a Real Time Distributed Solution"*  
Congrès IEEE / SMC 93 "Systems Engineering in the service of Humans"  
Le Touquet, France, 17-20 Octobre 1993, Vol 1, pp 548-554
- [LEC 94] **V. Lecuire**  
*"La gestion des systèmes et des réseaux industriels"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1994
- [LES 94] **J.J. Lesage**  
*"Contribution à la formalisation des modèles et méthodes de conception des systèmes de production"*  
Thèse d'habilitation à diriger des recherches, Université de Nancy I, 1994
- [LHO 85] **P. Lhoste**  
*"EXploitation Assistée par Ordinateur, EXAO : Proposition d'une approche méthodologique et d'outils d'assistance"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1985
- [LHO 94] **P. Lhoste**  
*"Contribution au Génie Automatique : Concepts, Modèles, Méthodes et Outils"*  
Thèse d'habilitation à diriger des recherches, Université de Nancy I, 1994
- [LOM 94] **M. Lombard-Gregori**  
*"Contribution au génie productique : prototypage d'une architecture d'ingénierie concourante des systèmes intégrés de fabrication manufacturière"*  
Thèse de doctorat, Université H. Poincaré - Nancy I, 1994
- [LUT 94] **E. Lutherer, M. Martinez, B. Querenet, J. Favrel**  
*"Toward A CIM Enginneering Workbench"*  
European workshop on integrated manufacturing systems engineering, IMSE  
Grenoble, France, 12-14 Décembre 1994, pp 79-85

- [LUX 94] **J.T. Luxhoj, J. Ribeiro, G. Zinn**  
*"Development of a Material Tracking Strategy for an Advanced Manufacturing System"*  
International Journal of Flexible Automation and Integrated Manufacturing, ISSN 1064-6345,  
Vol 2, N° 3, pp 273-283, 1994
- [MAL 88] **J.G. Maley**  
*"Managing the Flow of Intelligent Parts"*  
Robotics & Computer-Integrated Manufacturing, ISSN 0736-5845, Vol 4, N° 3/4, pp 525-530,  
1988
- [MEI 91] **A. Meier**  
*"Advantages of Using Features to Integrate Product and Process Modelling - Results of  
IMPACT (ESPRIT 2165)"*  
Seventh CIM-Europe annual conference - CEC DG XIII  
Turin, Italie, 29-31 Mai 1991, pp 69-77
- [MIL 95] **B. Mills**  
*"Auto-ID in distribution : the Tesco experience"*  
Automatic ID News Europe, Vol 4, N° 2, pp 15-16, 1995
- [MOL 95] **A. Molina, T. Ellis, R. Young, R. Bell**  
*"Modelling Manufacturing Capability to Support Concurrent Engineering"*  
Concurrent Engineering : Research and Application, ISSN 1063-293X, Vol 3, N° 1, pp 29-42,  
1995
- [MON 94] **ESPRIT Project MONOLIN 6936**  
*"Deliverable 06 - MONOLIN Implementation Guide : Implementation Context and General  
Overview"*  
15 Mars 1994
- [MOR 92a] **K.C. Morris, M. Mitchell**  
*"Database management systems in engineering"*  
NISTIR 4987, National Institute of Standards and Technology, Gaithersburg, Maryland, 1992
- [MOR 92b] **G. Morel**  
*"Contribution à l'Automatisation et à l'Ingénierie des Systèmes Intégrés de Production"*  
Thèse d'habilitation à diriger des recherches, Université H. Poincaré - Nancy I, 1993
- [MRT 91] **J. Richard, E. Bajic, T. Divoux, R. Stepoutjine, D. Gien, G. Coeurdeuil**  
*"Auto-contrôle de cellules flexibles d'usinage - Intégration dans un ensemble CFAO"*  
MRT décision d'aide N° 88.P.0718  
Rapport de fin de recherche financée par le Ministère de la Recherche et de la Technologie,  
Juin 1991.
- [NEU 94] **E. Neunreuther**  
*"Intégration et interconnexion d'outils : Modèles et Langages d'échanges normalisés"*  
Rapport de recherche de DEA Production Automatisée, CRAN EACN, Université H. Poincaré -  
Nancy I, 1994
- [NGU 94] **G.T. Nguyen**  
*"SHOOD : Plate-forme pour la conception assistée"*  
Journée PRIMECA "Modélisation Géométrique et Technologique en Conception de Produits"  
Grenoble, France, 14 Octobre 1994, pp 95-110
- [NOU 94] **J.F. Noubissie-Tchako**  
*"Contribution à la conception d'un système de pilotage distribué pour les systèmes automatisés  
de production"*  
Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambresis, 1994

- [OKI 94] **N. Okino**  
*"Bionic Manufacturing System"*  
Journal of Manufacturing Systems, ISSN 0278-6125, Vol 23, N° 3, pp 175-187, 1994
- [OSO 94] **A.L. Osorio, L.M. Camarinha-Matos**  
*"Information based control architecture for CIM"*  
IFIP Transactions B [Applications in Technology], ISSN 0926-5481, Vol B-17, pp 243-257, 1994
- [PAN 91] **H. Panetto**  
*"Une contribution au génie automatique : le prototypage des machines et systèmes automatisés de production"*  
Thèse de Doctorat, Université H. Poincaré - Nancy I, 1991
- [PAT 95] **V. Patrìti, O. Garro, P. Charpentier**  
*"Application des SMA à la fabrication : cas de SHIVA, machine à architecture parallèle"*  
Actes des Troisièmes Journées Francophones sur l'Intelligence Artificielle Distribuée & les Systèmes Multi-Agents  
St Baldoph, France, 15-17 Mars 1995, pp 66-78
- [PTA 92] **Norme AFNOR expérimentale Z 68-901 : Modèle conceptuel Base PTA**  
*"Représentation des systèmes de contrôle et de commande des Systèmes Automatisés de Production"*  
Septembre 1992
- [PER 93] **F. Pereyrol**  
*"Interaction produit-procédé. Extension de la modélisation de la commande avec suivi automatique du produit"*  
Thèse de Doctorat, Université de Montpellier II, Septembre 1993
- [PIC 90] **E. Pichat, R. Bodin**  
*"Ingénierie des données : bases de données, systèmes d'information, modèles et langage"*  
Editions Masson, ISBN 2-225-82266-2, 428 pages, 1990
- [SCH 93] **E. Schmidt, V. Jagganathan, R. Karinihi, G. Trapp, M. Shiye Qiu**  
*"The CERC Information Sharing System (ISS) and the STEP Standard Data Access Interface (SDAI)"*  
CE & CALS Conference  
Washington DC, USA, 28 Juin - 1 Juillet 1993, pp 75-98
- [SMI 80] **R.G. Smith**  
*"The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solving"*  
IEEE Transactions on Computers, ISSN 0018-9340, Vol c-29, N° 12, pp 1104-1113, 1980
- [STE 95] **The Newsletter of STEPTools Inc.**  
*"STEP Machine Tool Controllers"*  
Volume 4, N°1, Avril 1995, pp 1 et 3, 1995
- [STI 95] **S. Stinckwich**  
*"Agents logiciels pour les systèmes d'informations coopératifs et nomades"*  
Actes des Troisièmes Journées Francophones sur l'Intelligence Artificielle Distribuée & les Systèmes Multi-Agents  
St Baldoph, France, 15-17 Mars 1995, pp 245-252
- [THI 95] **H. Thimbleby, I. Witten, D. Pullinger**  
*"Concepts of Cooperation in Artificial Life"*  
IEEE Transactions on Systems, Man, and Cybernetics, ISSN 0018-9472, Vol 25, N° 7, pp 1166-1171, 1995



- [TIC 94] **S. Tichkiewitch**  
*"De la CFAO à la conception intégrée"*  
Revue internationale de CFAO et d'infographie, ISSN 0298-0924, Vol 9, N° 5, pp 609-621, 1994
- [TIC 95] **S. Tichkiewitch, E. Chapa, P. Belloy**  
*"Un modèle produit multi-vues pour la conception intégrée"*  
Congrès international de génie industriel de Montréal  
Montréal, Octobre 1995, Vol III, pp 1986-1998
- [UDO 92] **S. J. Udoka**  
*"The role of automatic identification in the CIM architecture"*  
Computer & Industrial Engineering, ISSN 0360-8352, Vol 23, N° 1-4, pp 1-5, 1992
- [UED 94] **K. Ueda**  
*"Biological-oriented Paradigm for Artifactual Systems"*  
Japan-USA Symposium on Flexible Automation  
Kobe, Japon, 11-18 Juillet 1994, pp 1263-1266
- [UPT 92] **D.M. Upton**  
*"A Flexible Structure for Computer-Controlled Manufacturing Systems"*  
Manufacturing Review, Vol 5, N° 1, pp 58-74, 1992
- [VEE 93] **D. Veeramani, B. Bhargava, M.M. Barash**  
*"Information system architecture for heterarchical control of large FMSs"*  
Computer Integrated Manufacturing Systems, ISSN 0951-5240, Vol 6, N° 2, pp 76-92, 1993
- [VER 94] **F. Vernadat**  
*"A Language for CIM System Specification"*  
10th ISPE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future  
Ottawa, Canada, 21-24 Aout 1994, pp 283-288
- [VIE 94] **C. Viel**  
*"STEP : un standard de modèle produit"*  
Journée PRIMECA "Modélisation Géométrique et Technologique en Conception de Produits"  
Grenoble, France, 14 Octobre 1994, pp 77-82
- [WAK 93] **L. Wakeman, J. Jowett**  
*"PCTE : the Standard for Open Repositories"*  
Editions Prentice Hall, ISBN 0-13-065566-X, 262 pages, 1993
- [WES 94] **R.K. Weston, I.S. Murgatroyd**  
*"Information modelling methods and tools for manufacturing systems"*  
International Symposium on Automotive Technologies and Automation, ISATA 27th  
Aaren, Allemagne, 31 Octobre-4 Novembre, pp 227-235
- [WIL 92] **P.R. Wilson**  
*"A View of STEP"*  
IFIP Transactions B [Applications in Technology], ISSN 0926-5481, Vol B-8, pp 267-296, 1992
- [WIL 94] **P.R. Wilson**  
*"EXPRESS TOOLS and SERVICES"*  
4th Annual EXPRESS User Group International Conference  
Greenville SC, USA, 13-14 Octobre 1994

---

## Acronymes

---

<b>API</b>	<b>A</b> pplication <b>P</b> rogramme <b>I</b> nterface interface de niveau programme d'application masquant la complexité
<b>CALS</b>	<b>C</b> omputer <b>A</b> ided and <b>L</b> ogistics <b>S</b> upport
<b>CAO</b>	<b>C</b> onception <b>A</b> ssitée par <b>O</b> rdinateur
<b>CIM</b>	<b>C</b> omputer <b>I</b> ntegrated <b>M</b> anufacturing
<b>CIM-OSA</b>	<b>C</b> omputer <b>I</b> ntegrated <b>M</b> anufacturing - <b>O</b> pen <b>S</b> ystems <b>A</b> rchitecture
<b>EDI</b>	<b>E</b> changes de <b>D</b> onnées <b>I</b> nformatisés
<b>EMS-API</b>	<b>E</b> scort <b>M</b> emory <b>S</b> ystem - <b>A</b> pplication <b>P</b> rogrammable <b>I</b> nterface interface d'accès à des systèmes d'identification, définie dans le projet ESPRIT 6936 MONOLIN
<b>ESPRIT</b>	<b>E</b> uropean <b>S</b> trategic <b>P</b> rogram on <b>R</b> esearch in <b>I</b> nformation <b>T</b> echnologies programme cadre de recherche européen dans le secteur des technologies de l'information
<b>EXPRESS</b>	Langage textuel de modélisation de données normalisé définit par une grammaire formelle
<b>EXPRESS-G</b>	Langage graphique de modélisation de données normalisé
<b>FAO</b>	<b>F</b> abrication <b>A</b> ssistée par <b>O</b> rdinateur
<b>IdA</b>	<b>I</b> dentification <b>A</b> utomatique ensemble des moyens mis en oeuvre pour saisir une information en réduisant, voire supprimant, la part de l'intervention humaine dans cette opération de saisie
<b>MMS</b>	<b>M</b> anufacturing <b>M</b> essages <b>S</b> pecification - <b>I</b> SO 9506 protocole de niveau application d'échange de messages, définit à l'origine dans le projet de réseau MAP
<b>MONOLIN</b>	<b>M</b> obile <b>N</b> ode <b>L</b> ogistics and <b>I</b> ndustrial <b>N</b> etworks projet ESPRIT 6936 portant sur le développement des technologies de l'information visant l'intégration des systèmes d'identification automatiques en production et logistique du transport Partenaires : Renault, Volvo, Redar, Baumer Ident, RIT, EPFL
<b>NIAM</b>	<b>N</b> ijsen <b>I</b> nformation <b>A</b> nalysis <b>M</b> ethod méthode d'analyse et de formalisation de données

---

<b>NIST</b>	<b>National Institute of Standards and Technology</b>
<b>PCTE</b>	<b>Portable Common Tools Environnement - ISO 13719</b> environnement informatique de travail
<b>SADT</b>	<b>Structured Analysis and Design Technic</b> technique d'analyse fonctionnelle structurée
<b>SAP</b>	<b>Système Automatisé de Production</b>
<b>SDAI</b>	<b>Simple Data Access Interface</b> ensemble de primitives d'accès à une base de données STEP - ISO DIS 10303-22
<b>SGBD</b>	<b>Système de Gestion de Bases de Données</b> logiciel qui permet d'interagir avec une base de données. Il permet à un utilisateur de définir des données, d'insérer, de modifier et de rechercher des données.
<b>SPEX</b>	<b>SPécifications EXécutables</b> outil logiciel de spécification et de simulation de modèles comportementaux (© TNI - SPIE TRINDEL - CRAN)
<b>SQL</b>	<b>Structured Query Language - ISO 9075</b> langage normalisé de modélisation logique de données et d'interrogation des base de données relationnelles
<b>STEP</b>	<b>Standard for Transfert and Exchange of Product data model - ISO 10303</b> ensemble de normes visant à permettre l'échange des données de définition des produits

# Annexe

# Le langage EXPRESS

---

## 1. Structure d'un modèle

---

Un modèle EXPRESS est constitué d'un ou plusieurs modules non imbriqués définis à l'aide du mot clé SCHEMA. Un schéma est composé d'une collection d'entités, d'algorithmes de calculs, de déclarations et de contraintes sur les instances d'une ou plusieurs entités.

Syntaxe :

```
SCHEMA <identificateur>; ... END_SCHEMA;
```

Un schéma peut utiliser ou référencer, à l'aide de USE FROM ou REFERENCE FROM les objets déclarés dans un autre schéma. Ces deux concepts permettent de spécifier une interface entre les différents schémas d'un même modèle.

## 2. Entité

---

En déclarant une entité EXPRESS, on crée un type de données référencé par l'utilisateur de cette entité. Ceci permet de définir une classe d'objets aux propriétés communes, où chacun de ces objets est considéré comme une instance de l'entité déclarée.

Syntaxe:

```
ENTITY <identificateur>; ... END_ENTITY;
```

### 2.1. Héritage

Au même titre que dans le modèle entités-relations étendu ou dans un modèle orienté objets, EXPRESS définit le concept d'héritage à l'aide de SUPERTYPE OF ou SUBTYPE OF. Si une entité est sous-type d'une autre entité, alors en plus de ses propriétés propres elle va hériter de toutes les propriétés de ses super-types.

Exemple:

```
ENTITY Operation;  
...  
END_ENTITY;  
  
ENTITY Operation_Usinage SUBTYPE OF (Operation);  
...  
END_ENTITY;
```

Un sous type d'une entité peut redéfinir un attribut hérité de trois manières différentes :

- en affaiblissant le type de données de l'attribut, par exemple passer d'un réel à un entier,
- en rendant un attribut obligatoire s'il était optionnel,
- en le transformant en attribut dérivé s'il était explicite.

Une entité peut hériter de plusieurs entités à la fois, dans ce cas on parle d'héritage multiple. Les problèmes posés par l'héritage multiple sont résolus syntaxiquement dans EXPRESS. On peut citer le cas d'une entité E1 sous type de E2 et E3, définissant chacun un attribut de même nom A, que E1 héritera une première fois de E2 et une seconde fois de E3. Pour éviter toute confusion, le langage EXPRESS impose, lors de l'utilisation de A, de référencer le super-type en question. Cette solution est également valable dans le cas où E2 et E3 hérite de A d'un super-type commun E4.

## 2.2. Entité abstraite

Un entité déclarée ABSTRACT SUPERTYPE ne peut être utilisée que comme réservoir d'information pour ses sous-types. Cette entité ne peut avoir d'instances que par l'intermédiaire de ses sous-types.

## 3. Attributs

---

Les attributs représentent les propriétés caractérisant une entité. Chaque attribut est identifié par un nom, unique pour l'entité.

Syntaxe:

```
<Nom Attribut> : <Type attribut>;
```

### 3.1. Attributs explicites

Les attributs explicites déclarés dans l'entité possèdent un type et peuvent être optionnel par utilisation du mot clé OPTIONAL.

Exemple:

```
ENTITY Point;
    x,y : REAL;
    z : OPTIONAL REAL;
END_ENTITY;
```

### 3.2. Attributs dérivés

Les attributs dérivés sont des attributs dont la valeur est issue d'un calcul soit direct soit par appel d'une fonction déclarée dans le schéma.

Exemple:

```
ENTITY Cercle;
    centre : Point;
    rayon : REAL;
    DERIVE
        perimetre : REAL :=2.0*PI*rayon;
END_ENTITY;
```

*ou*

```
FUNCTION Calcul_Perimetre (rayon : REAL) : REAL;
    RETURN(2.0*PI*rayon);
END_FUNCTION;
ENTITY Cercle;
    centre : Point;
    rayon : REAL;
    DERIVE
        perimetre : Calcul_Perimetre(rayon);
END_ENTITY;
```

### 3.3. Attributs inverses

Un attribut explicite de type E1 déclaré dans une entité E2 engendre une relation R entre E2 et E1. Par défaut la relation  $R^{-1}$  entre E1 et E2 a une cardinalité (0,n) au sens du modèle entités-relations. On peut contraindre cette relation en déclarant un attribut INVERSE dans E1.

Exemple:

*Un Cercle possède un Point centre, un Point est utilisé comme centre dans 0, 1, 2, n Cercle(s).*

```
ENTITY Cercle;  
  centre : Point;  
  rayon : REAL;  
  DERIVE  
    perimetre : REAL :=2.0*PI*rayon;  
END_ENTITY;  
  
ENTITY Point;  
  x,y : REAL;  
  INVERSE  
    reference_cercle : SET[0:?] OF Cercle FOR centre;  
END_ENTITY;
```

## 4. Types de données

---

Le langage EXPRESS définit un ensemble élémentaire de types de données initiaux, auquel le concepteur peut ajouter ses propres types.

### 4.1. Types simples

- INTEGER désigne l'ensemble des entiers relatifs,
- REAL désigne l'ensemble des réels,
- NUMBER utilisé pour tout type de données numériques,
- BINARY pour définir des séquences binaires. La taille peut être fixe ou variable,
- BOOLEAN pour définir des valeurs de l'ensemble {true,false},
- LOGICAL pour définir des valeurs de l'ensemble {true, unknow, false},
- STRING permet la définition de chaînes de caractères de tailles fixes ou variables.

### 4.2. Types collections

- ARRAY pour déclarer une collection de taille fixe,
- LIST pour déclarer une séquence d'éléments d'une liste dont la taille maximale peut être fixée à la déclaration,
- BAG pour déclarer une collection d'éléments de même type, où l'ordre est sans importance,
- SET cas particulier de BAG mais ne permettant pas la duplication d'éléments.

### 4.3. Constructeurs de Types

L'instruction TYPE permet de créer ses propres types

Exemple:

```
TYPE Temperature = REAL; END_TYPE;
```

L'instruction SELECT permet de créer des listes de type de données. Le domaine de valeurs d'un type SELECT est l'union des domaines de valeurs des types constituant la liste de types

ENUMERATION permet de construire un type de données dont le domaine de valeur est une liste d'identificateurs. Chaque identificateur de la liste représente une valeur possible du type de donnée déclaré.

Exemple:

```
TYPE RESULTAT_MESURE = SELECT (bon, mauvais); END_TYPE
```

## 5. Relations entre entités

---

Les relations entre occurrences d'entités EXPRESS s'expriment à l'aide d'attributs dans les entités. Une relation possède un sens de lecture dit direct ou inverse. Une relation inverse non explicitement déclarée correspond à une relation de cardinalité (0,n)

En EXPRESS il n'existe aucune façon pour préciser si la relation est une relation dite d'instances ou une relation de composition : par exemple, 'une cellule est composée de machines' et un 'opérateur conduit une machine' n'ont pas le même sens mais vont pourtant s'écrire de façon identique.

Exemples:

*Une Opération a été réalisée sur 0 ou 1 machine. Implicitement 1 machine a réalisé de 0 à n opérations.*

```
ENTITY Machine;
END_ENTITY;
ENTITY Operation;
  Realise_sur : OPTIONAL Machine;
END_ENTITY;
```

*Une cellule est composée de 1 à n machines et 1 machine n'appartient qu'à une cellule.*

```
ENTITY Cellule;
  Composee_de : SET[1:?] OF Machine;
END_ENTITY;
ENTITY Machine;
  INVERSE Appartient : Cellule FOR Composee_de;
END_ENTITY;
```



## 6. Contraintes de cohérence

---

### 6.1. Règles locales

Les règles locales sont des contraintes sur les instances d'une entité. On trouve des règles d'unicité qui permettent de contrôler l'unicité des valeurs des attributs parmi toutes les instances d'une entité. Il y a aussi les règles sur le domaine pour introduire des contraintes sur les valeurs des attributs d'une seule instance.

Exemple:

*L'identifiant de la machine est unique*  
 ENTITY Machine;  
     Identifiant : INTEGER;  
                 UNIQUE ur: Identifiant;  
 END\_ENTITY;

*L'heure d'une date est compris entre 1 et 24*  
 ENTITY Date;  
     Heure : INTEGER;  
     .....  
     WHERE wh : (heure<=24) and (heure>=1);  
 END\_ENTITY;

### 6.2. Règles globales

Les règles locales permettent d'introduire des contraintes sur les instances d'un seul type d'entité. Lorsqu'il s'agit de contrôler les instances de plusieurs types d'entités à la fois, les règles globales s'appliquent.

Exemple:

*Considérons que le nombre d'instances de l'entité entreprise d'un modèle ne puisse être que de 1.*  
 RULE instance\_unique\_entreprise FOR (Entreprise);  
     SIZEOF(QUERY(\* Entreprise))=1;  
 END\_RULE  
 ENTITY Entreprise;  
     .....  
 END\_ENTITY;

Nom: CHAXEL

Prénom: Frédéric

DOCTORAT de l'UNIVERSITE HENRI POINCARÉ, NANCY-I

en PRODUCTION AUTOMATISEE

VU, APPROUVÉ ET PERMIS D'IMPRIMER

Nancy, le 22 JAN. 1996 U4P n° 544

Le Président de l'Université



## Résumé :

Les travaux présentés dans ce mémoire constituent une contribution à la définition et à la gestion de systèmes d'information répartis dans les structures de production. L'approche proposée repose sur le produit, considéré comme un vecteur d'information, qui au cours de son élaboration se comporte comme un agent mettant à disposition et gérant ses informations à destination du processus considéré. Le produit se caractérise en tant qu'acteur de son évolution, par le biais des informations qu'il supporte et qu'il fournit aux prestataires de services qui opèrent les modifications sur son état. Nos travaux de recherche, en exploitant les techniques de l'identification automatique, ont ainsi avancé le concept d'objet nomade de production ou ONP.

Après un état de l'art des travaux dans le domaine et une analyse de la problématique liée à la définition et à la gestion des données d'un produit durant son cycle de vie, en particulier au travers de STEP, nous détaillons le concept d'ONP. A des fins de mise en oeuvre du concept, nous avons spécifié une structure d'accueil appuyée par une méthodologie de réalisation d'un système d'information métier distribué sur des supports mobiles d'identification automatique. Les spécifications d'un modèle en couches de l'interface d'accès aux données du produit sont présentées puis analysées. Une méthode de génération de cette interface, reposant sur un modèle logique de données exprimé dans le langage STEP-EXPRESS, est proposée. Enfin une démarche de conception de la partie commande d'un système de production utilisant le concept d'ONP est présentée.

Ces propositions sont supportées par un outil Tag-STEP, que nous avons développé et dont nous détaillons les fonctionnalités. Il permet la génération des différents supports de données par traduction du modèle EXPRESS, la distribution des données, ainsi que la configuration automatique de l'interface d'accès aux informations emportées par les produits. Nous vérifions les propositions avancées en prototypant, par simulation, la cellule flexible d'assemblage de l'AIP Lorrain.

Mots clés: Identification automatique, Etiquettes électroniques, Langage EXPRESS, STEP, Système d'information, Bases de données mobiles.

## Abstract :

The work presented is a contribution towards defining and managing manufacturing systems. The proposed approach is based around the product, taken as an information vector. Each product behaves like an agent which manages and offers necessary information to given machines. The product, an actor in its development process, is characterised by its knowledge and its capacity to co-operate with the manufacturing processes which change its state. Our research work, by taking advantages of automatic identification technologies, proposes the nomadic object concept in production.

After a state of the art in the corresponding fields, and a problem analysis concerning the definition and the data management of a product through its life cycle (in particular through the STEP methodology), we detail the nomadic object concept. A framework architecture and a realisation methodology are specified which match the concept on mobile data carriers. Specifications of a layered data access interface model are discussed and analysed. A method on how to generate this interface is also given. It is based on a logical data model using the EXPRESS modelling language. Finally, we present and discuss a conception method for production systems taken into account the nomadic object concept.

These propositions are supported by a tool detailed in this thesis, namely Tag-STEP. Tag-STEP allows us to: generate diverse data supports by translating EXPRESS models; distribute information; and automatically configure the data access interface embedded on the product. A demonstrator has been built to validate the proposed methodology and is based upon a simulation of the flexible assembly cell of the AIP Lorrain.

Keywords : Automatic identification, electronic tags, EXPRESS language, STEP, Information system, Mobile databases.



## Résumé :

Les travaux présentés dans ce mémoire constituent une contribution à la définition et à la gestion de systèmes d'information répartis dans les structures de production. L'approche proposée repose sur le produit, considéré comme un vecteur d'information, qui au cours de son élaboration se comporte comme un agent mettant à disposition et gérant ses informations à destination du processus considéré. Le produit se caractérise en tant qu'acteur de son évolution, par le biais des informations qu'il supporte et qu'il fournit aux prestataires de services qui opèrent les modifications sur son état. Nos travaux de recherche, en exploitant les techniques de l'identification automatique, ont ainsi avancé le concept d'objet nomade de production ou ONP.

Après un état de l'art des travaux dans le domaine et une analyse de la problématique liée à la définition et à la gestion des données d'un produit durant son cycle de vie, en particulier au travers de STEP, nous détaillons le concept d'ONP. A des fins de mise en oeuvre du concept, nous avons spécifié une structure d'accueil appuyée par une méthodologie de réalisation d'un système d'information métier distribué sur des supports mobiles d'identification automatique. Les spécifications d'un modèle en couches de l'interface d'accès aux données du produit sont présentées puis analysées. Une méthode de génération de cette interface, reposant sur un modèle logique de données exprimé dans le langage STEP-EXPRESS, est proposée. Enfin une démarche de conception de la partie commande d'un système de production utilisant le concept d'ONP est présentée.

Ces propositions sont supportées par un outil Tag-STEP, que nous avons développé et dont nous détaillons les fonctionnalités. Il permet la génération des différents supports de données par traduction du modèle EXPRESS, la distribution des données, ainsi que la configuration automatique de l'interface d'accès aux informations emportées par les produits. Nous vérifions les propositions avancées en prototypant, par simulation, la cellule flexible d'assemblage de l'AIP Lorrain.

Mots clés: Identification automatique, Etiquettes électroniques, Langage EXPRESS, STEP, Système d'information, Bases de données mobiles.

## Abstract :

The work presented is a contribution towards defining and managing manufacturing systems. The proposed approach is based around the product, taken as an information vector. Each product behaves like an agent which manages and offers necessary information to given machines. The product, an actor in its development process, is characterised by its knowledge and its capacity to co-operate with the manufacturing processes which change its state. Our research work, by taking advantages of automatic identification technologies, proposes the nomadic object concept in production.

After a state of the art in the corresponding fields, and a problem analysis concerning the definition and the data management of a product through its life cycle (in particular through the STEP methodology), we detail the nomadic object concept. A framework architecture and a realisation methodology are specified which match the concept on mobile data carriers. Specifications of a layered data access interface model are discussed and analysed. A method on how to generate this interface is also given. It is based on a logical data model using the EXPRESS modelling language. Finally, we present and discuss a conception method for production systems taken into account the nomadic object concept.

These propositions are supported by a tool detailed in this thesis, namely Tag-STEP. Tag-STEP allows us to: generate diverse data supports by translating EXPRESS models; distribute information; and automatically configure the data access interface embedded on the product. A demonstrator has been built to validate the proposed methodology and is based upon a simulation of the flexible assembly cell of the AIP Lorrain.

Keywords : Automatic identification, electronic tags, EXPRESS language, STEP, Information system, Mobile databases.