

N° d'ordre : 398

50376
1989
125-1

50376
1989
125-1

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

en

**PRODUCTIQUE, AUTOMATIQUE ET INFORMATIQUE
INDUSTRIELLE**

par

Hervé PERRIEN
Ingénieur IDN



MODELE DE STRUCTURE DE DONNEES DITE APPLICATION
POUR UN POSTE DE TRAVAIL DE CONCEPTION
DE REALISATION ET D'EXPLOITATION D'EQUIPEMENTS
DE PRODUCTION AUTOMATISEE

soutenue le 7 Juillet 1989 devant la Commission d'Examen

Membres du Jury :

M. FRACHET

M. PRUNET

M. STAROSWIECKI

M. GENTINA

M. BOUTEILLE

M. BRARD

M. THUREL

Rapporteur

Rapporteur

Rapporteur

Examineur,

Directeur de Thèse

Invité

Invité

Invité

SCD LILLE 1



D 030 320103 9

10
10
10
10
10

50376
1989
125-1

50376
1989
125-1

R E M E R C I E M E N T S

Je tiens avant tout à remercier ici Monsieur le Professeur Jean-Claude GENTINA, directeur de l'IDN qui m' a dirigé, conseillé et soutenu dans la préparation de cette thèse.

Je suis également très heureux de la présence dans ce jury, en tant que rapporteur, de Monsieur Jean-Paul FRACHET, directeur du laboratoire de Génie Automatique de l'ISMCM et responsable du projet BASEPTA.

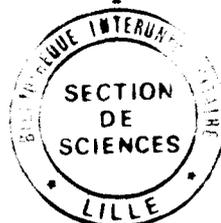
Je remercie également Monsieur PRUNET, Professeur au Laboratoire d'Automatique et de Micro-électronique de Montpellier et Monsieur STAROSWIECKI, Professeur à l'Université des Sciences et Techniques de LILLE d'avoir accepté d'être rapporteurs de cette thèse.

Je suis particulièrement reconnaissant à Monsieur THUREL, Directeur de recherche Automatique et Informatique Industrielle de TELEMECANIQUE, pour la confiance et le soutien qu'il m'a témoignés au cours de ce travail.

Je suis aussi très heureux de la présence dans ce jury de Monsieur BRARD qui a enrichi mon travail de sa grande expérience et de ses précieux conseils.

Je tiens également à remercier Monsieur Noël BOUTEILLE de sa présence dans ce jury et, à travers lui, l'ensemble de l'équipe technique BASEPTA: Madame Françoise DIVERCHY et Messieurs Thierry DELARBRE et Ali SAIDI. Le modèle présenté dans ce mémoire est avant tout le résultat du travail de cette équipe.

Enfin, je tiens à témoigner ma reconnaissance à mes collègues de TELEMECANIQUE qui m'ont apporté leur soutien. Je remercie tout particulièrement Messieurs BOUX, BUISSON et LAGOUGE de la Division Applications et Systèmes qui m'ont aidé à constituer l'exemple d'illustration.



S O M M A I R E

SOMMAIRE

REMERCIEMENTS

SOMMAIRE

INTRODUCTION

PARTIE I - EXPOSE DU PROBLEME

Introduction.....	I- 2
1. Définitions et vocabulaire.....	I- 3
11. Automatisme industriel	
12. Industrie manufacturière - industrie du process	
13. Partie commande - Partie opérative	
14. Application automatisée	
15. Notion de cycle de vie - les différentes étapes	
16. Les trois volets de la fourniture d'un BEA	
2. La cible de l'étude : l'application automatisée.....	I- 8
21. Précision sur la notion d'équipement automatisé	
22. Caractéristiques d'une application automatisée	
3. Présentation d'un exemple d'illustration.....	I-11
4. Etude du dossier d'affaire.....	I-13
41. Définitions	
42. Caractéristiques d'un dossier d'affaire	
43. Composition d'un dossier d'affaire	
431. Structure d'un dossier d'affaire TELEMECANIQUE	
432. Composition d'un dossier conformément à la norme CNOMO E03-03-205N	
433. Composition d'un dossier d'automatisme programmé réalisé par EDILOG	
434. Composition du dossier de l'exemple	
5. Etat de l'art - étude des outils logiciels existants.....	I-20
51. Outils internes TELEMECANIQUE	
511. La CSAO (Conception de Schémas Assistée par Ordinateur)	
5111. Descriptif sommaire	
5112. Analyse de l'outil CSAO	
5113. Conclusion	
512. La CDAO Pneumatique (Conception et Dessin Assistés par Ordinateur)	
5121. Description sommaire	
5122. Analyse de la CDAO	
5123. Conclusion	

SOMMAIRE

513.	Le logiciel PL7-3	
	5131.	Description sommaire
	5132.	Analyse de PL7-3
	5133.	Conclusion
	514.	Conclusion sur les outils TELEMECANIQUE
52.	Autres outils disponibles	
	521.	Les outils de conception de logiciel
		5211. OMEGA
		5212. CADEPA
		5213. GRAL
		5214. Autre produits logiciels
	522.	Les outils de schématique électrique
		5221. XELEC
		5222. DESCARTES
		5223. Conclusion sur les outils de schématique
	523.	Conclusion sur l'étude des outils logiciels
6.	Le Poste de Travail de l'Automaticien.....I-44	
61.	Synthèse sur l'étude des outils logiciels	
62.	Le problème de la cohérence	
	621.	La cohérence de l'application au cours de la prestation d'automatisation
	622.	La cohérence du dossier
63.	Définition du besoin: le Poste de Travail de l'Automaticien	
	631.	Généralités
	632.	Proposition d'une architecture pour un PTA
7.	Etat des travaux.....I-58	
71.	Introduction	
72.	Liste des projets	
Conclusion.....	I-61	
 PARTIE II - UN MODELE POUR LA STRUCTURE DE DONNEES APPLICATION D'UNE CAO D'AUTOMATISMES		
Introduction.....	II- 2	
1.	Définir un modèle conceptuel pour une structure de données particulière: la base de données application.....II- 3	
11.	La Base Application: la solution au problème de la cohérence.	
12.	La définition d'un modèle conceptuel	
13.	Le travail du projet BASEPTA	
14.	Objectifs de la présentation du modèle conceptuel dans ce mémoire	
15.	Contenu de la présentation du modèle conceptuel dans ce mémoire	

SOMMAIRE

- 2. Les principes du modèle de données.....II- 7
 - 21. La cible du modèle: l'application en cours de traitement
 - 22. Modéliser l'application et non sa représentation
 - 23. Un modèle "methodology-free"
 - 24. Les futures implémentations du modèle de données
- 3. L'outil de modélisation.....II-11
 - 31. Introduction
 - 32. Les concepts de base de l'outil de modélisation
 - 321. L'entité
 - 322. L'association
 - 323. Cardinalités d'une association
 - 324. Attributs d'une association
 - 33. Un premier enrichissement du modèle Entités-Associations: la relation d'agrégation
 - 34. Un deuxième enrichissement du modèle Entités-Associations: la relation de généralisation
 - 35. Contraintes sur les associations
 - 351. Contraintes d'exclusion
 - 352. Autres contraintes
 - 36. Conclusion
- 4. Les grandes familles de données du modèle.....II-27
 - 41. Introduction
 - 42. La fourniture
 - 43. Les données de structuration
 - 44. Les données annexes
 - 45. Liens entre les différents aspects
- 5. La fourniture: les données décrivant le matériel.....II-30
 - 51. Introduction
 - 52. L'assemblage des constituants matériels de l'application
 - 521. Le constituant. Aspect logique
 - 522. Assemblage logique de constituant
 - 523. Aspect logique d'une liaison réelle entre constituants
 - 53. Les aspects réels et physiques des constituants et liaisons matériels de l'application.
 - 531. Aspect réel du constituant
 - 532. Localisation physique d'un constituant réel
 - 533. Les deux types de constituants réels
 - 534. Aspect réel d'une connexion entre constituants
 - 535. Localisation physique d'une liaison réelle
 - 54. L'aspect approvisionnement des constituants et des liaisons de l'application
 - 541. Approvisionnement d'un constituant ou d'une connexion

SOMMAIRE

- 542. Notion de Borne disponible
- 543. Les différents types d'articles d'approvisionnement
- 55. Caractéristiques technologiques des constituants de l'application
 - 551. La classe fonctionnelle d'un constituant application
 - 552. Un catalogue "application" des constituants
 - 553. Caractéristiques techniques de bornes
- 56. Conclusion: vue générale de l'aspect fourniture matérielle

- 6. La fourniture. Les données décrivant le logiciel.....II-75**
 - 61. Introduction
 - 62. La configuration du logiciel
 - 63. La ressource système utilisée
 - 64. L'aspect bibliothèque de logiciels
 - 65. Le logiciel application
 - 651. La structuration utilisateur du logiciel
 - 652. Réalisation d'un élément de logiciel
 - 66. L'implantation du logiciel
 - 661. Notion de logiciel implanté
 - 662. Notion de système d'exploitation
 - 67. Regroupements particuliers de logiciels
 - 68. Relations logiciel application-système
 - 681. Relations variable logicielle - variable système
 - 682. Relations élément de logiciel - ressource système de type logiciel
 - 683. Données de description logicielle
 - 69. Conclusion: Vue générale de l'aspect fourniture logicielle

- 7. La fourniture. Les liens entre le logiciel et le matériel.....II-101**
 - 71. Implantation d'un logiciel sur un constituant programmable
 - 72. L'Association variable système - Voie automate

- 8. Les Données de structuration.....II-104**
 - 81. Introduction
 - 82. Objectifs et contraintes du système automatisé: le lien avec le cahier des charges fonctionnel
 - 83. Les structurations de l'application
 - 831. Description d'une structure
 - 832. Relations inter-structures
 - 833. Choix de conception: liens objectifs- structurations
 - 834. La modélisation comportementale
 - 84. Relations entre la structuration de l'application et le logiciel
 - 841. Vision structurante du logiciel
 - 842. Association variable logicielle - variable de comportement

SOMMAIRE

85. Relations entre la structuration de l'application et le matériel	
851. Vision structurante du matériel	
852. Variable de comportement d'un constituant matériel	
86. Conclusion: Vue générale des données de structuration et de leur relations avec la fourniture	
9. Les Données annexes.....	II-124
91. Introduction	
92. L'identification de l'application automatisée traitée	
93. L'association d'un ensemble de données annexes à un élément de l'application	
94. Référence externe d'un ensemble de données annexes	
95. Description de l'ensemble de données annexes	
96. Les entités génériques	
961. Le corps	
962. La caractéristique	
963. La donnée	
97. Conclusion vue générale des données annexes du modèle	
Conclusion.....	II-131
 PARTIE III - EXPERIMENTATION DU MODELE DE STRUCTURE DE DONNEES	
Introduction.....	III- 2
1. Traitement d'un exemple d'application automatisée.....	III- 3
11. Structure de données de l'exemple "Chaine automatique d'emballage de rouleaux de papier"	
12. Analyse des résultats	
2. Implantation de la structure de données application d'un d'un exemple grace à un SGBD relationnel.....	III-15
21. Objectifs	
22. Mécanismes de transcription du schéma conceptuel	
221. Introduction	
222. Présentation sommaire d'un système relationnel	
223. Dérivation d'une entité non contrainte	
224. Dérivation d'une association "simple"	
225. Dérivation d'une agrégation	
226. Traduction des contraintes	
227. Conclusion	
 PARTIE IV - PERSPECTIVES D'UTILISATION DU MODELE DE STRUCTURE DE DONNEES	
Introduction.....	IV- 2

SOMMAIRE

1. Réalisation d'un SGBD supportant le formalisme Entité - Association enrichi.....	IV- 3
11. Quel type de SGBD pour implémenter le schéma conceptuel de la structure de données?	
12. Spécifications générales du LMD d'un SGBD Entité - Association enrichi.	
2. La Base Documentation.....	IV- 6
3. La Base Constituants.....	IV- 7
4. Produits, outils et atelier PTA.....	IV- 9

CONCLUSION

BIBLIOGRAPHIE (CLASSEMENT PAR ORDRE ALPHABETIQUE)

BIBLIOGRAPHIE (CLASSEMENT PAR THEMES)

LISTE DES ABREVIATIONS USUELLES

ANNEXE

DOSSIER D'AFFAIRE : CHAINE D'EMBALLAGE AUTOMATIQUE DE ROULEAUX DE PAPIER

I N T R O D U C T I O N

INTRODUCTION

La croyance populaire a depuis longtemps décrété que les cordonniers étaient les plus mal chaussés. Si les automaticiens étaient aussi courants que les cordonniers, elle pourrait tout aussi bien affirmer aujourd'hui que les automaticiens sont les personnes dont le travail est le plus mal automatisé. En effet, alors que mécaniciens, électroniciens, gestionnaires disposent déjà d'outils de CAO performants, l'automaticien reste très démuné. Tout au plus dispose-t-il de logiciels, d'ailleurs très récents, qui l'assistent de manière ponctuelle et limitée à certaines étapes de sa prestation.

Cette carence en outils informatiques qui provient sans doute pour une grande partie de la complexité et de la diversité des métiers de l'automaticien, est de plus en plus vivement ressentie. Elle s'accompagne d'une insuffisance d'outils méthodes universellement reconnus, par exemple dans l'expression de la commande d'un automatisme industriel ou dans sa spécification. L'automatique étant une discipline relativement récente, beaucoup de méthodologies restent encore à découvrir.

Afin de pallier ces manques, industriels et universitaires ont entamé une série de travaux dans le cadre d'une nouvelle discipline baptisée d'ores et déjà "Génie Automatique" [FRA 87A]. A l'heure où les machines automatiques deviennent de plus en plus complexes, où l'on ne raisonne plus au niveau de la machine mais au niveau de la cellule et de l'atelier, où l'on ne vise plus à résoudre seulement les problèmes d'automatique mais aussi ceux plus complexes encore de la productique, il est stratégique que l'automaticien dispose d'outils performants. C'est dans cette optique de génie automatique que se situe résolument ce travail.

L'ensemble des futurs outils de CAO pour automaticiens "harmonieusement" intégrés, nous verrons dans cette thèse comment, constitue le Poste de Travail de l'Automaticien ou PTA [PTA 87A] ou encore Poste de Travail pour l'Automatisation. L'une des parties fondamentales de ce futur PTA est la description des données caractérisant l'automatisme industriel, objet de la prestation de l'automaticien. Ces informations seront organisées en un ensemble structuré qui sera enrichi, modifié et exploité par les divers traitements de CAO. Cette thèse a pour objet d'étudier les caractéristiques d'une telle structure de données et d'en proposer un modèle de description.

INTRODUCTION

Pour parvenir à la définition de ce modèle, nous avons tout d'abord réalisé une analyse des outils logiciels existants chez TELEMECANIQUE mais aussi plus généralement chez les concepteurs d'automatismes. L'accent a été porté sur les structures de données de ces outils. Les besoins en matière de CAO ont également été étudiés. Ces analyses ont été poursuivies dans le cadre du projet national BASEPTA [BAS 87A] auquel nous avons participé en tant que membre de l'équipe technique.

Dans une première partie, nous nous efforcerons de situer cette thèse dans le cadre de tous les travaux en cours en génie automatique. Après avoir délimité le champ de l'étude, nous essaierons de montrer les motivations qui ont conduit à la définition d'une structure de données décrivant l'automatisme.

La seconde partie présentera de manière fine le modèle élaboré par BASEPTA, modèle à la définition duquel nous avons participé. Nous tenterons de montrer les faiblesses et les forces de ce modèle et également la manière de le détailler et de le "personnaliser" en vue d'une implémentation.

Dans une troisième partie, nous présenterons les résultats d'une expérimentation du modèle et nous préciserons les mécanismes de son implantation avec un système de gestion de base de données relationnel.

Enfin, dans une dernière partie nous énoncerons différents axes de recherche à mener à bien dans l'avenir pour aboutir à la réalisation de produits logiciels PTA.

PARTIE I

EXPOSE DU PROBLEME

INTRODUCTION

Le but de cette première partie est de poser correctement le problème d'une CAO pour automaticien et de la structure de données d'une telle CAO.

Pour cela, il nous a paru indispensable de définir préalablement le vocabulaire utilisé et le contexte de travail. Puis nous délimiterons précisément le champ de l'étude. Afin d'illustrer notre propos tout au long de cette thèse, nous présenterons ensuite un exemple type d'automatisme industriel.

Nous pourrons alors, à travers une étude du dossier d'affaire et des outils logiciels existants, énoncer les grandes lignes du problème de la CAO pour automaticien.

Nous terminerons cette partie par un tour d'horizon des travaux en cours, ce qui nous permettra de situer de manière encore plus précise le champ de notre étude.

1. DEFINITIONS ET VOCABULAIRE

Tout au long du projet BASEPTA qui réunissait l'essentiel des industriels du monde de l'automatisme, la terminologie a été une difficulté constante. Il semble en effet qu'il n'y ait pas un vocabulaire reconnu et adopté par l'ensemble de la profession. La signification des termes varie d'une entreprise à l'autre, d'un bureau d'étude à l'autre voire d'un ingénieur à l'autre. A titre d'exemple, nous avons pu constater que l'expression "tête de filerie" était comprise de manière différente par deux personnes du même bureau d'études de RENAULT, l'un s'occupant de l'automatisation des machines de manutention, le second d'assemblage carrosserie. C'est pourquoi nous proposons ci-après une définition des principaux termes. Dans un souci de cohérence nécessaire à la compréhension de ce mémoire, nous nous conformerons à ces définitions.

11. Automatisme industriel

On appelle automatisme industriel ou équipement de production automatisé l'ensemble des équipements matériels et des logiciels permettant l'automatisation, c'est à dire le fonctionnement avec une intervention humaine réduite, d'un équipement de production industriel. Le terme équipement de production devra être pris dans un sens élargi: Il pourra désigner un équipement qui apporte effectivement une valeur ajoutée à un produit tel une machine automatique de soudure, mais aussi par exemple, un système de transport transférant les produits d'un endroit de l'atelier à un autre.

12. Industrie manufacturière - industrie du process

Nous serons conduits plus loin, à plusieurs reprises, à distinguer deux types d'automatismes: ceux de l'industrie manufacturière élaborant des produits en quantités discrètes (ex: l'industrie automobile) et ceux de l'industrie du process (ex: l'industrie pétrolière). Traditionnellement, l'industrie manufacturière est le domaine de l'automatisme séquentiel, l'industrie du process celui de l'automatisme continu. Mais cette distinction bien théorique tend à s'estomper avec notamment l'introduction de boucles de régulation dans le manufacturier ou l'existence de procédés batch (ou mixtes) dans l'industrie du process.

13. Partie Commande - Partie Opérative

Habituellement, on distingue dans un automatisme industriel deux parties:

- la Partie Opérative (PO) qui opère sur le processus automatisé par l'intermédiaire des actionneurs,
- la Partie Commande (PC) qui coordonne les actions de la Partie Opérative en recevant des informations de celle-ci et en lui envoyant des ordres qu'elle élabore.

Cette distinction soulève de nombreuses questions que nous ne développerons pas ici. En particulier se pose le problème de la frontière PO / PC bien difficile à fixer et qui a conduit dans certains travaux à l'abandon de ces notions de PC et de PO [FRA 86A].

Remarquons que le classement même des constituants matériels en capteurs / actionneurs / pré-actionneurs... est difficile. A titre d'exemple, le variateur de vitesse comprend une partie commande, une partie distribution, une partie pré-actionneur et une partie dialogue homme-machine.

14. Application automatisée

Dans ce qui suit, nous utiliserons au minimum les notions de PO et de PC et nous envisagerons l'automatisme industriel globalement en fonction de la prestation demandée au Bureau d'Etudes en Automatismes (BEA). Nous désignerons par application automatisée l'ensemble des moyens matériels et logiciels faisant l'objet de la prestation du BEA (étude et/ou réalisation). Cette prestation pourra s'étendre plus ou moins en direction de la Partie Opérative. Ainsi dans l'optique PEUGEOT, c'est le mécanicien qui détermine et approvisionne les fins de course et la boutonnerie. Le Service des Travaux Neufs qui rassemble les automaticiens proprement dits, est essentiellement intéressé par la Partie Commande. Chez d'autres, le BEA définira capteurs et actionneurs.

15. Notion de cycle de vie - les différentes étapes

Le terme cycle de vie désigne l'ensemble des étapes par lesquelles passe l'application automatisée depuis la définition du cahier des charges de l'automatisme jusqu'à sa disparition de l'atelier.

Pour fixer les idées, on peut distinguer dans le cycle de vie les étapes suivantes:

- La rédaction du pré-cahier des charges permet d'exprimer en termes de besoins, les fonctions à remplir par l'automatisme.

- L'élaboration d'un devis fournit déjà un certain nombre de réponses au cahier des charges telles que pré-nomenclatures de constituants, propositions de prix et parfois même plans d'implantation face avant des armoires.

- La rédaction du cahier des charges permet ensuite de détailler plus finement les objectifs à remplir, les contraintes à respecter par exemple en matière de sûreté et de disponibilité. Le cahier des charges décrit en général le principe de fonctionnement et les différents modes de marche. Des travaux [SOU 87A] [SOU 89A] ont montré que le cahier des charges n'est jamais purement fonctionnel et qu'il contient déjà des éléments de solutions, soit en termes de procédés à appliquer, soit en termes de moyens à mettre en oeuvre. Très souvent, le cahier des charges n'est pas seulement une expression des besoins mais déjà une spécification. Ainsi l'architecture de l'automatisme y est parfois décrite.

- La conception est la phase la plus complexe. Elle consiste à élaborer la structure de la commande. Sécurité, modes de marche et dialogue opérateur sont traités. Les pupitres opérateurs sont définis. Le choix de technologie (programmée, électro-mécanique, pneumatique...) étant fait, les constituants matériels sont déterminés. Le logiciel est décrit de manière détaillée. Le dossier est élaboré. Il contient notamment l'ensemble des schémas de principe et de réalisation, le dossier du logiciel, de principe (organigrammes, analyse fonctionnelle...) et de réalisation (dossier programme). Nous verrons plus loin de manière détaillée les éléments composant un dossier.

PARTIE I - EXPOSE DU PROBLEME

- La réalisation effective consiste en l'implémentation des logiciels sur les constituants programmables, le montage des constituants en armoire et leur cablage, la réalisation des pupitres... Elle peut être suivie d'une étape de tests, l'installation étant mise au point en plate-forme d'essai.

- La mise en oeuvre se fait sur le site d'implantation. Les capteurs sont réglés. Les marches de vérification sont effectuées. Un certain nombre de test est réalisé.

- L'exploitation est la phase la plus longue dans la vie de l'automatisme. Elle correspond à l'utilisation de l'équipement de production automatisée.

- La maintenance comprend toutes les interventions sur l'automatisme en cas d'incident détecté en exploitation. Après cette détection, une phase de diagnostic permet de déterminer la nature de l'intervention nécessaire (réglage, modification à apporter à l'automatisme...). Une fois l'incident corrigé, il faut procéder à la remise en marche souvent appelée remise en cycle de l'équipement de production.

Pour conclure sur cette notion de cycle de vie, il convient d'ajouter que cette présentation en étapes reste assez théorique. De nombreuses nuances apparaissent dans la réalité. Ainsi la conception est généralement réparties sur deux bureaux d'études: un bureau d'étude PC et un bureau d'étude PO qui doivent s'échanger des informations. Il faudra donc procéder à l'intégration PO-PC avant de passer à la réalisation. D'autre part il existe pratiquement toujours des phénomènes de "rebouclages", c'est à dire de retour sur les phases précédentes. Ainsi la mise en oeuvre, voire même malheureusement la maintenance, peut entraîner une reprise partielle de la conception. En fait, un grand nombre d'informations est repris d'une étape du cycle de vie sur l'autre. Pour la cohérence globale de l'application, il est indispensable que les données récupérées le soient de manière complète et cohérente. Ainsi que nous le verrons, c'est un des points essentiels de l'étude.

Notons pour terminer que la prestation d'un BEA peut se limiter à quelques étapes seulement du cycle de vie (le plus souvent la conception).

16. Les trois volets de la fourniture d'un BEA

Nous désignerons par fourniture d'un bureau d'études en automatismes:

- le matériel, c'est à dire l'ensemble des constituants matériels que comprend une application automatisée,
- le logiciel, c'est à dire l'ensemble des programmes informatiques concernant l'application étudiée. Cette définition exclut par exemple l'interpréteur Grafcet d'un automate puisqu'il s'agit d'un logiciel non spécifique fourni par le constructeur de l'automate au concepteur de l'automatisme,
- le dossier qui décrit de manière conventionnelle les deux aspects précédents.

2. LA CIBLE DE L'ETUDE : L'APPLICATION AUTOMATISEE

21. Précision sur la notion d'équipement automatisé

Nous avons défini précédemment l'application automatisée comme étant l'objet de l'étude et/ou de la réalisation d'un équipement de production automatisée par un BEA. Il subsiste cependant une ambiguïté sur le terme équipement de production que nous allons essayer de lever.

En effet, dans une optique productique, il est usuel de distinguer les niveaux suivants:

CENTRE DE PRODUCTION

ATELIER

LIGNE OU SECTEUR DE FABRICATION

CELLULE OU ILOT DE FABRICATION

SYSTEMES DE TRANSPORT / MACHINES / ROBOTS

CAPTEURS / ACTIONNEURS

De plus, le vocabulaire est fonction du type d'installation automatisée. En tolèrie, on parlera d'ilôt pour désigner l'équivalent de la cellule en mécanique ou du chantier en fonderie. Chez CITROEN [CIT 87A], le terme machine peut s'appliquer tout aussi bien à une station, c'est à dire à une zone correspondant au pas d'une pièce qu'à un module c'est à dire une entité autonome réalisant une fonction donnée. Il convient cependant de situer sur cette échelle les équipements de production auxquels nous nous intéresserons.

Afin d'éviter toute ambiguïté, nous désignerons à présent par équipement de production automatisée une entité unitaire, c'est à dire réalisée par une même équipe d'automaticiens au cours d'une même période. Suivant le cas, il pourra s'agir d'une machine, d'un poste, d'une ligne ou d'une cellule. Nous reconnaissons cependant que les problèmes de coordination dans une cellule sont relativement complexes et n'ont pas été particulièrement étudiés. Il s'agit donc ici d'un cas limite. De même, la cible visée est plutôt l'équipement automatisé de l'industrie manufacturière. Nous verrons cependant que les résultats s'appliquent aussi pour la plupart d'entre eux à l'industrie du process.

22. Caractéristiques d'une application automatisée

Une application automatisée possède un certain nombre de caractéristiques bien identifiées:

En premier lieu, il s'agit généralement d'une réalisation unique ou quasi-unique. On ne parle en effet plus d'"application automatisée" dès que l'on aborde un équipement de production réalisé en série. Notre préoccupation ici n'est pas le produit mais bien le "système" réalisé en quelques exemplaires au plus. Ce caractère unitaire a comme conséquence la part importante du coût de conception dans le coût total du système. La réduction de ce coût de conception contribue donc à une amélioration de la productivité de manière plus sensible que dans le cas d'un équipement de série où la conception ne représente qu'une petite partie du coût total.

Ces applications automatisées sont essentiellement conçues à partir de composants standards du commerce. C'est l'assemblage de ces constituants qui est original pour chaque application. Bien sur, il peut arriver que le bureau d'études fasse développer un produit spécifique comme une carte d'axe ou une carte régulation pour un système donné mais il s'agit là de cas particuliers.

Une autre caractéristique des applications automatisées est la réutilisation de sous-ensembles déjà étudiés et réalisés pour d'autres applications. Il est fréquent que ces éléments de solutions soient parfaitement identifiés et formalisés. Ils constituent alors des "standards maison" et sont le reflet du savoir-faire de l'entreprise. C'est ainsi que l'on a pu recenser chez TELEMÉCANIQUE environ une vingtaine de manières de réaliser une fonction "départ-moteur". Ces standards internes sont d'ailleurs progressivement intégrés dans les CAO par exemple électriques, et permettent de personnaliser les CAO de chaque entreprise.

Il convient de noter que les concepteurs d'automatismes s'efforcent de procéder avec le logiciel comme avec le matériel, par réutilisation de sous-ensembles déjà réalisés. C'est ce dont témoignent les efforts importants des constructeurs de consoles automate pour mettre à la disposition du programmeur tout un ensemble de bibliothèques de logiciels constructeur et utilisateur. On peut même affirmer que l'automaticien souhaite avoir de moins en moins de logiciel à écrire effectivement et

PARTIE I - EXPOSE DU PROBLEME

procède de plus en plus par paramétrage et assemblage de modules existants, connus et maîtrisés ce qui en fait un gage de sécurité.

Il apparait alors clairement que dans l'avenir le concepteur d'applications automatisées devra pouvoir réutiliser des sous-ensembles intégrant à la fois du matériel et du logiciel. C'est l'un des services que l'on est en droit d'attendre des futures CAO pour automaticien.

Enfin pour terminer, signalons que les applications automatisées mettent toujours en oeuvre plusieurs technologies (électromécanique, hydraulique, pneumatique, logique programmée...). Ainsi tout système à base d'automate programmable contient une partie câblée ne serait-ce que pour traiter les sécurités conformément à la législation.

3. PRESENTATION D'UN EXEMPLE D'ILLUSTRATION

Afin d'illustrer notre propos et de servir de base au maquettage, nous avons choisi de présenter une application automatisée. Il s'agit d'un exemple tiré d'une application réelle que nous avons modifié afin d'une part de le simplifier et d'autre part de le rendre plus significatif en y intégrant le maximum de technologies différentes.

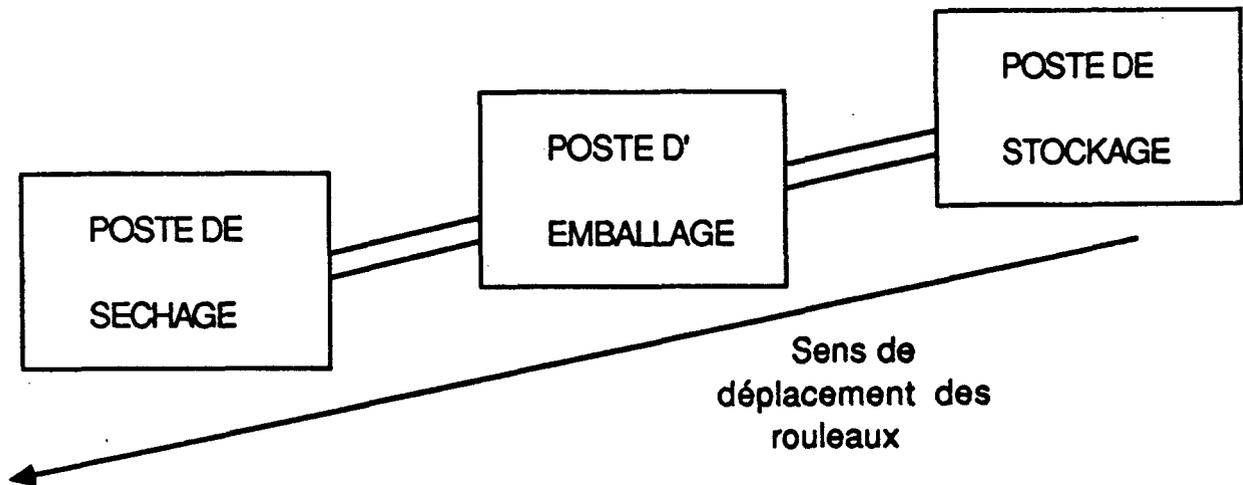


Fig 1: Schéma général de l'équipement

La fonction de cet équipement est d'emballer des rouleaux de papier. Les rouleaux à traiter sont stockés en amont de l'équipement. Ils sont ensuite acheminés un par un en roulant sur un plan incliné vers un poste d'emballage. On enroule alors sur chaque rouleau deux tours de papier que l'on encolle. Puis le rouleau emballé est envoyé à travers un aiguillage vers l'un des deux postes de séchage dans lequel il restera le temps nécessaire pour que la colle sèche.

Le cahier des charges de cette application ainsi que l'ensemble du dossier technique figurent en ANNEXE.

PARTIE I - EXPOSE DU PROBLEME

Cette application intègre plusieurs technologies. La partie séquentielle est réalisée à l'aide d'un automate, les actionneurs sont essentiellement pneumatiques à l'exception des moteurs qui donnent lieu à tout un appareillage électromécanique. Le dossier est très riche et nous le détaillerons plus loin.

4. ETUDE DU DOSSIER D'AFFAIRE

Ainsi que nous l'avons vu, l'un des trois volets de la prestation d'un BEA est la production du dossier d'affaire. Ce dossier constitue en fait la seule image disponible de l'application, le moyen unique de communication entre les différents intervenants sur le système automatisé. Il est donc essentiel de préciser ce qu'est un dossier d'affaire, ce qui le compose et ce qui le caractérise.

4.1. Définitions

Au cours de la réalisation d'un automatisme industriel, un grand nombre de documents est produit. Ces documents de nature différente se rapportant à la même application constituent une entité: le dossier d'affaire.

Un dossier d'affaire est une entité parfaitement déterminée, identifiée par une référence, un intitulé, un nom de client, un responsable, un code marketing... C'est un objet évolutif qui peut être en cours, terminé ou archivé et auquel est rattaché un certain nombre de dates (date de création, date de clôture, date de réouverture...).

Habituellement, on distingue plusieurs types de dossiers:

- Le **cahier des charges** est un document émis par un demandeur et destiné à un ou plusieurs concepteurs-réalisateur explicitant les spécifications d'une prestation et ayant pour objet des produits et/ou des systèmes et/ou des services [SOU 87A]. Ici, nous n'envisagerons que des cahiers des charges de systèmes.
- Le **dossier de conception ou de fabrication** contient l'ensemble des documents résultant de la conception, les informations concernant le choix du matériel, son implantation et la description du logiciel.
- Le **dossier d'exploitation-maintenance** contient les informations nécessaires à l'exploitation et à la maintenance. Il peut s'agir d'informations spécifiques à ces activités (par exemple procédures de maintenance) ou d'informations présentes dans le dossier de conception mais présentées ici d'une manière qui facilite le travail de l'opérateur ou de l'agent de maintenance (par exemple liste de câblage ou carnet de câbles).

Ici encore, il convient de préciser qu'il s'agit d'une dichotomie purement théorique. Ainsi JC BOSSY [AUT 87A] estime que la séparation cahier des charges / dossier technique est un peu arbitraire et que l'on retrouve en réalité dans le dossier d'affaire, les deux aspects plus ou moins importants suivant le stade du cycle de vie.

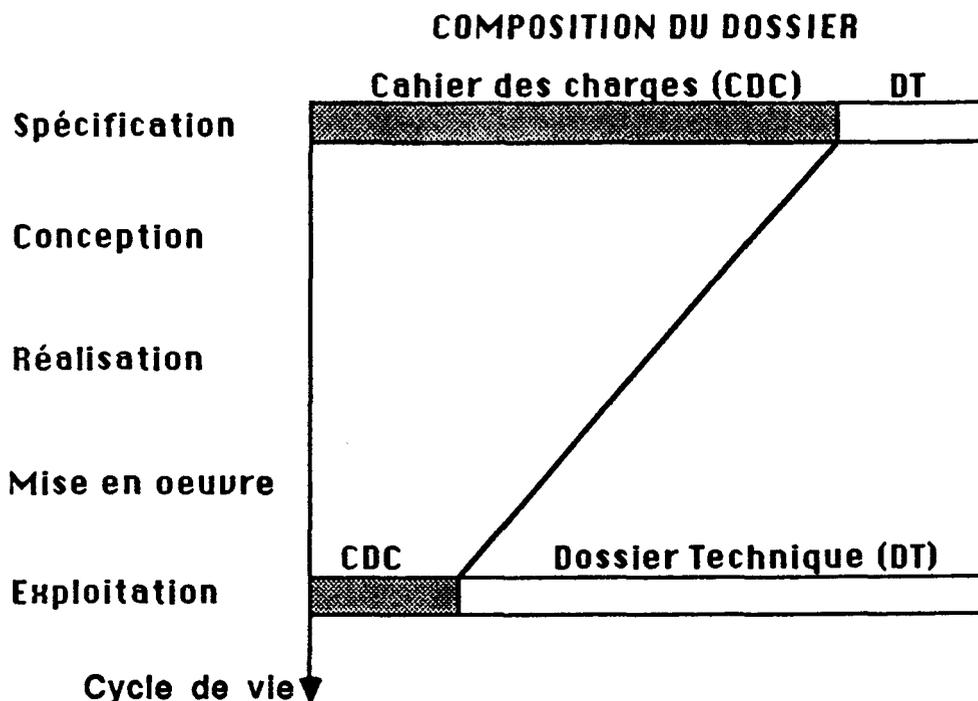


Fig 2

Ainsi en exploitation, le manuel opérateur peut être d'une certaine manière envisagé comme le cahier des charges de ce que l'on demande à l'opérateur.

42. Caractéristiques d'un dossier d'affaire.

La caractéristique principale des dossiers d'affaire est qu'il n'en existe pas deux identiques, dans leur contenu bien évidemment, mais aussi dans leur forme. Le type de document composant un dossier varie d'une entreprise à l'autre. Par exemple, certains dossiers contiennent des nomenclatures de pièces de rechange, d'autres pas, certains le schéma unifilaire et le schéma multifilaire correspondant, d'autres se contentent de l'unifilaire. La structure même des documents varie fortement. Les normes

et les règles de représentation diffèrent. Les constructeurs automobiles utilisent la norme CNOMO et TELEMECANIQUE préconise la norme CEI. Bien que dans ce cas précis, l'ouverture du marché européen en 1993 apportera des solutions, il demeure un énorme travail d'unification et de standardisation qui pourrait aller de pair avec l'apparition de CAO d'automatismes performantes.

Il faut remarquer également que la taille d'un dossier d'affaire est très variable, souvent importante. Le dossier peut contenir de l'ordre du millier de folios et cette taille n'aura certainement pas tendance à diminuer tant le besoin en documentation nouvelle se fait plus sentir chaque jour.

De plus, à l'image des applications qu'il représente, le dossier est multi-technologies. Schémas électromécaniques, hydrauliques, pneumatiques, plans mécaniques, et programmes automate s'y cotoient avec parfois aussi des documents électroniques.

L'organisation du dossier est également une composante variable. La découpe du dossier en sous-dossiers est généralement une découpe en sous-ensembles topo-fonctionnels qui fait apparaître les lignes de fabrication, les lignes, les machines, les stations, les modules... Mais cette division peut aussi résulter des compétences des services qui le réalisent (ainsi chez TELEMECANIQUE: service CSAO pour la partie électromécanique, division pneumatique pour le pneumatique).

43. Composition d'un dossier d'affaire

Comme nous l'avons dit précédemment, la composition d'un dossier d'affaire est très variable. Elle dépend des habitudes de chaque bureau d'études mais aussi des performances des outils logiciels d'aide à la réalisation de ces dossiers. Plutôt que de décrire de manière exhaustive tous les documents manipulés en conception d'automatismes industriels, nous proposons à titre d'exemple la composition d'un certain nombre de dossiers.

431 Structure d'un dossier d'affaire TELEMECANIQUE
[BRA 87A]

Documents généraux:

Folio en-tête spécifiant les intitulés et les formats des folios constituant le dossier technique,

Liste des évolutions,

Spécifications fonctionnelles du cahier des charges,

Spécifications générales et particulières concernant la fabrication, le cablage et les repérages,

Spécifications d'essais et de recettes.

Documents d'analyse et de réalisation:

Notices techniques,

Synoptique explicatif, architecture générale du système,

Structuration et hiérarchie,

Sécurités et surveillances,

Etude fonctionnelle (organigrammes, Grafcet),

Etude opérationnelle (modes de marches, GEMMA),

Schémas de puissance,

Schémas de contrôle renseignés (équipotentiels, références croisées, renvois résolus),

Légende des symboles,

Légendes fonctionnelles,

Schémas des borniers,

Raccordements extérieurs, plans de filerie,

Nomenclature du matériel et ses caractéristiques mécaniques et électriques,

Implantation du matériel,

Plans mécaniques d'usinage et d'habillage,

Plans des pupitres, des châssis, charpentes et armoires,

PARTIE I - EXPOSE DU PROBLEME

Programmes,
Tableau des entrées/sorties,
Feuilles de calcul,
Notices de fonctionnement,
Notices d'entretien et de dépannage,
Consignes de réglage.

432. Composition d'un dossier conformément à la norme CNOMO E03-03-205N [LAM 87A]

Descriptif général de l'installation,
Schéma fonctionnel,
Schéma logique,
Schéma des circuits,
Diagramme de séquences,
Notices de fonctionnement,
Nomenclature du matériel,
Disposition de l'appareillage,
Schéma des connexions extérieures,
Plan d'installation,
Instructions d'entretien,
Articles d'usure ou de sécurité (dispositions pour
l'approvisionnement),
Liste des articles d'usure,
Description du logiciel.

433. Composition d'un dossier d'automatisme programmé réalisé par EDILOG [LAM 87A] :

Cartes d'entrées/sorties,
Répertoire des entrées/sorties,

PARTIE I - EXPOSE DU PROBLEME

Implantation des entrées/sorties,
Répertoire des bits non mémorisés,
Répertoire des relais,
Répertoire des temporisateurs et compteurs,
Répertoire des séquences,
Séquences logiques,
Répertoire des branchements à une séquence,
Répertoire des appels calculs,
Répertoire des mots mémorisés,
Lignes de calcul,
Variables à recycler.

434. Composition du dossier de l'exemple

Afin de montrer comment un exemple de dossier, bien que construit à des fins "pédagogiques" peut s'écarter des compositions types, nous donnons ci-après la composition du dossier de l'exemple "Chaine d'emballage de rouleaux de papier" :

- Cahier des charges de l'application comprenant en particulier:

- * les plans de la machine,
- * la description fonctionnelle du mode automatique,
- * une première description des modes de marches,
- * une première définition de pupitre,
- * la description de diverses contraintes notamment liées à l'environnement...

- Etude des modes de marches (GEMMA)

- Dossier de la partie matérielle comprenant:

- * Schémas développés,
- * Schémas de borniers,
- * Légende fonctionnelle,
- * Nomenclature...

PARTIE I - EXPOSE DU PROBLEME

- Dossier du logiciel comprenant:

- * Configuration automate,
- * Cablage des entrées,
- * Cablage des sorties,
- * Bits et mots utilisés,
- * Temporisateurs utilisés,
- * Programme proprement dit.

5. ETAT DE L'ART - ETUDE DES OUTILS LOGICIELS EXISTANTS

La plupart des dossiers d'affaires que nous venons de passer en revue sont élaborés de façon manuelle. Certains cependant sont générés par des outils logiciels qui sont en quelque sorte les premières CAO d'automatismes. Quoique fort parcellaires et assez "rustiques" pour l'essentiel d'entre elles, elles constituent pourtant des ébauches de solution au problème du Poste de Travail de l'Automaticien. A ce stade, il nous est apparu nécessaire de faire l'état de l'art en matière d'outils de CAO.

51. Outils internes TELEMECANIQUE

Dans un premier temps, nous allons étudier un certain nombre d'outils logiciels développés entièrement ou partiellement par TELEMECANIQUE, en particulier pour les besoins internes de la société. Il existe également des outils TELEMECANIQUE conçus pour des applications spécifiques, soit matériels (ISP 70 en pesage, dosage, régulation), soit logiciels (SYNOPTEL pour la surveillance et la conduite de process) dont nous ne parlerons pas ici.

511. La CSAO (Conception de Schémas Assistée par Ordinateur)

5111. Descriptif Sommaire

Généralités

Le logiciel CSAO est un outil utilisé par la division Applications et Systèmes (A&S) de TELEMECANIQUE et certains Centres Techniques Régionaux (CTR) de la société.

Son but est la réalisation du dossier électrique de fabrication de l'automatisme depuis la création de schémas de principe jusqu'à l'implantation en armoire.

Conçu sur la base du système de CAO SAFIRS de la société ASSIGRAPH, il bénéficie des fonctionnalités de dessin du logiciel, mais sa richesse et son originalité proviennent

PARTIE I - EXPOSE DU PROBLEME

des développements spécifiques réalisés conjointement par TELEMECANIQUE et ASSIGRAPH.

Les machines sur lesquelles est implanté le logiciel sont:

- un VAX 780 avec des consoles graphiques pour A&S,
- des MICRO-VAX II pour les CTR.

Création de schémas électromécaniques.

A l'aide des bibliothèques graphiques contenant des symboles normalisés, des fonds de plans (ou grilles), des éléments de décors et des standards de schémas, l'opérateur dessine ses schémas développés de contrôle et de puissance. Ces schémas ne sont pas de véritables schémas de principe puisque l'opérateur y positionne des bornes de borniers.

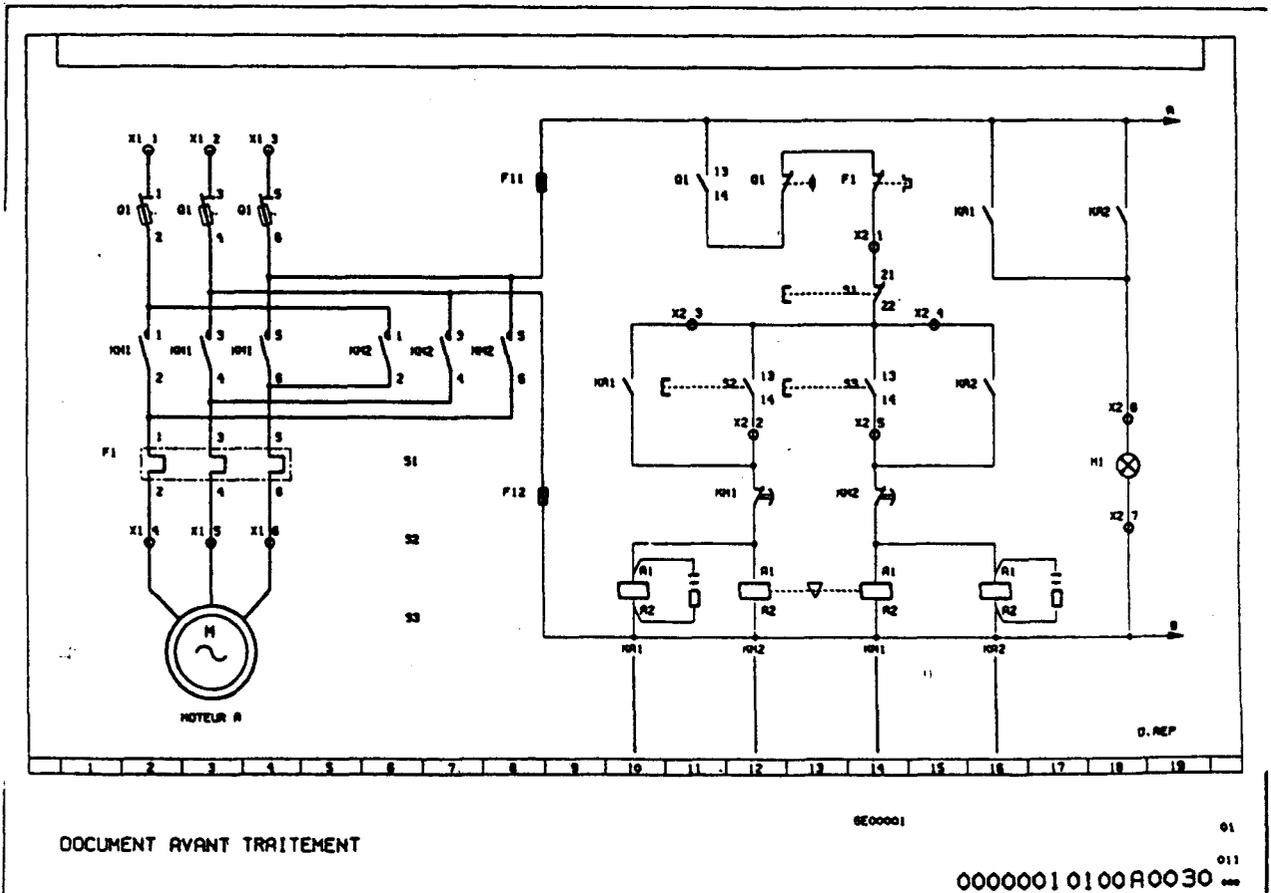


Fig 3: Schéma électrique développé avant traitement.

PARTIE I - EXPOSE DU PROBLEME

L'utilisateur bénéficie des primitives de DAO classiques mais peut également, en interactif, effectuer des opérations simples sur la "logique" du schéma telles que le contrôle électrique des connexions ou la numérotation des équipotentielles.

Traitement des schémas

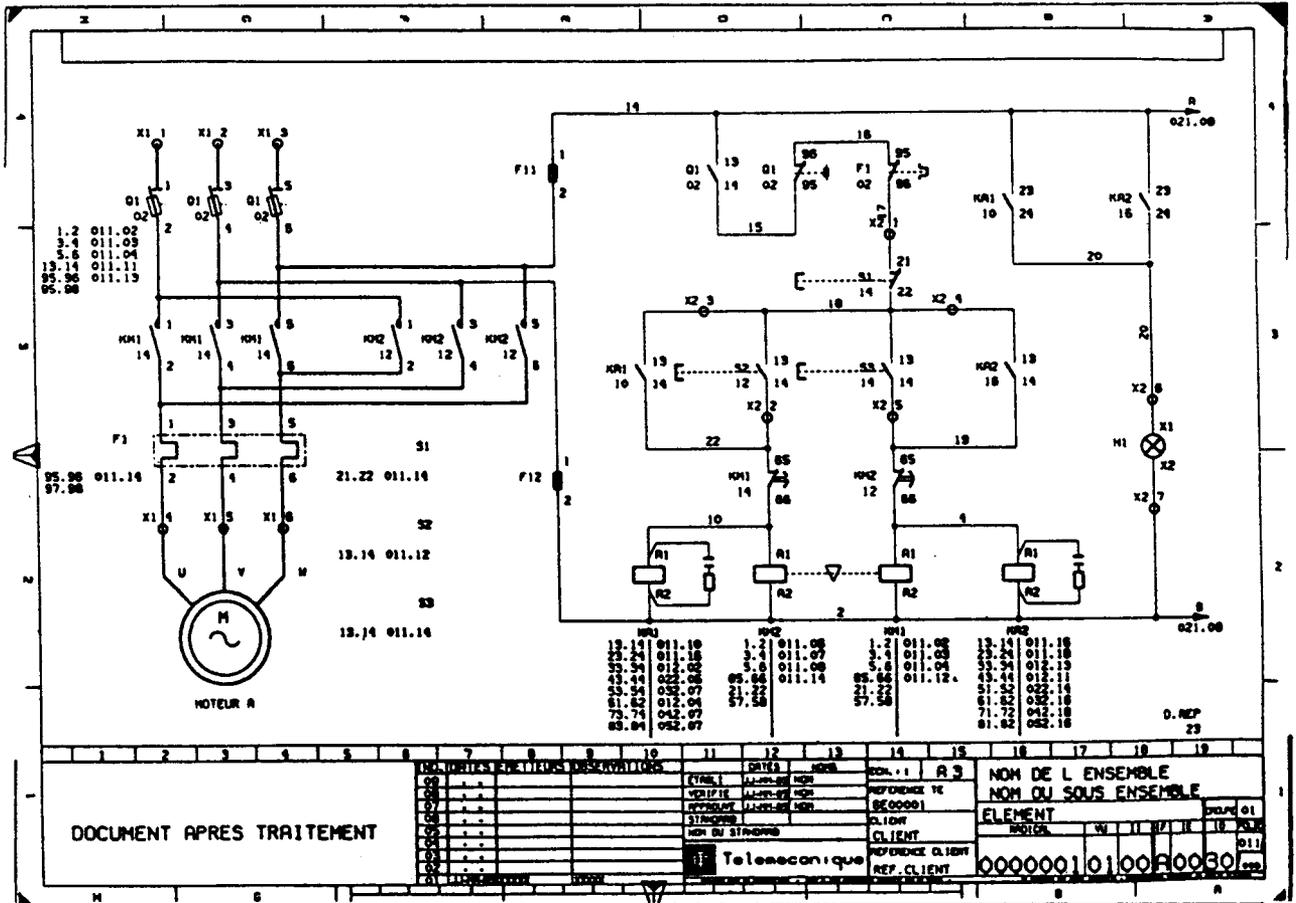


Fig 4: Schéma électrique renseigné après traitement

Les schémas constitués des différents folios et intégrés dans une affaire sont soumis à un traitement batch. Le logiciel exploite une base de données de matériels pour attribuer des matériels en fonction de contraintes introduites par l'opérateur (ex: la famille de composants pré-déterminés). Il choisit ainsi les contacts additifs à monter sur le contacteur en fonction du nombre de contacts commandés par une bobine donnée. Il peut alors

PARTIE I - EXPOSE DU PROBLEME

renseigner les folios des schémas en venant "réinjecter" les informations convenables aux emplacements réservés a cet effet (ainsi les repères des contacts). Le folio comporte également la localisation de la bobine maître sous chaque contact et des contacts commandés sous chaque bobine (références croisées) ainsi qu'une numérotation d'équipotentiellles.

Le système peut également renseigner des folios de bornes préparés à cet effet:

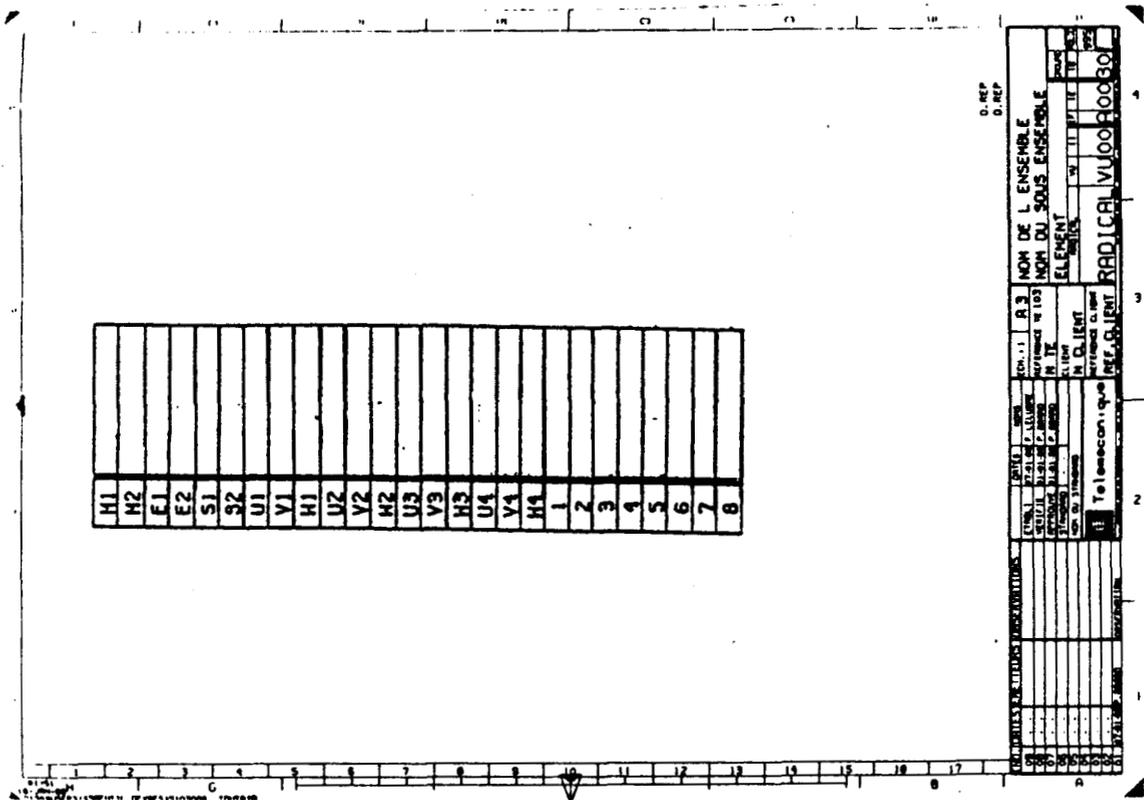


Fig 5: Schéma de bornier avant traitement

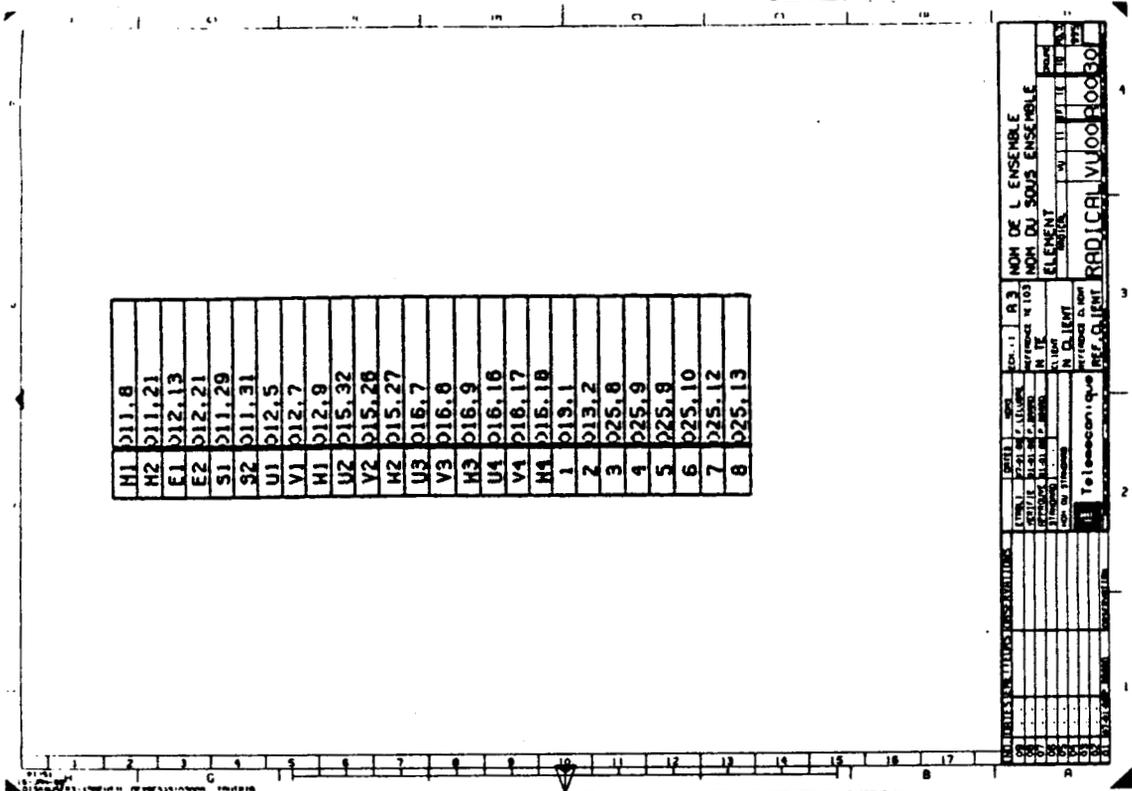


Fig 6: Schéma de bornier après traitement

Le logiciel élabore de la même manière des folios de légende fonctionnelle et de nomenclature. Il produit aussi un ensemble de documents de type listings informatiques:

- liste des documents
- liste des bornes
- liste des erreurs
- liste des équipotentiels
- références croisées
- liste de matériel attribué
- nomenclature

Implantation du matériel

La réalisation des folios d'implantation est faite de manière semi-automatique. Après avoir défini son armoire et ses profilés sur le folio d'implantation, l'opérateur introduit le nom de légende fonctionnelle de l'élément à implanter et son point d'implantation. Le système connaissant le matériel correspondant, exploite une base de données matériel pour déterminer si le montage est

possible, rajoute éventuellement du matériel associé pour permettre le montage (ex: platine de montage pour profilé DIN entraxe 120) et dessine une vignette graphique représentant l'encombrement du constituant en utilisant une autre bibliothèque. Puis le logiciel génère une nomenclature additionnelle des matériels qu'il a été amené à introduire.

Autres documents

L'opérateur peut utiliser les facilités graphiques de la CSAO pour générer d'autres documents purement graphiques: schéma général de l'installation, dessins de pupitres, grafcet de programmes...

5112. Analyse de l'outil CSAO

La principale limitation de l'outil CSAO est qu'il ne traite qu'une technologie précise: l'électromécanique. S'il est possible de dessiner des grafcets de programmes, il est en revanche exclu de faire un traitement logique sur ces grafcets ou, par exemple, d'assurer une cohérence entre les variables d'entrées/sorties automate figurant dans les actions et les receptivités de ces grafcets et les bornes d'entrées/sorties apparaissant sur les schémas électriques. Le grafcet n'est donc en fin de compte ici qu'un simple dessin.

L'avantage essentiel de la CSAO est de fournir un dossier électrique cohérent où le même constituant électromécanique est représenté de façon différente sur le schéma développé, la nomenclature et les plans d'implantation, la cohérence entre ces différentes vues étant assurée par le nom fonctionnel du constituant. La démarche CSAO intègre une partie importante du cycle de conception, même s'il n'y a pas, comme nous le verrons plus loin pour la CDAO pneumatique, de description fonctionnelle des constituants. Le choix d'un constituant est en effet réalisé grâce au nombre de symboles qui le constituent et aux contraintes imposées par l'opérateur (nom du fournisseur, type de matériel...) et non suivant la fonctionnalité à remplir. Cependant, dans le domaine de l'électromécanique, c'est à notre connaissance le seul logiciel allant jusqu'à la réalisation des plans d'implantation et le plus performant dans le choix du matériel tant sur le plan de la détermination du matériel additif que du matériel associé.

La CSAO produit des dossiers qui sont d'une grande qualité graphique avec des représentations normalisées (cartouches, symboles conformes notamment aux normes CEI). Le logiciel décharge l'opérateur de tâches assez systématiques telles que numérotation des équipotentielles, schémas de borniers, établissement de nomenclatures et fournit une aide à la conception appréciable au niveau choix du matériel mais aussi dessin en permettant la réutilisation de standards ou d'éléments déjà réalisés. L'apport de la CSAO se situe donc sur la plan de la qualité et de l'homogénéité de la documentation de l'aspect électrique de l'affaire, mais aussi sur le temps d'étude et bien sûr de dessin.

Il convient cependant de remarquer que la CSAO est un outil de conception dédié certes à l'électromécanique mais surtout à TELEMECANIQUE. Il dépend en grande partie de l'organisation même des produits et du catalogue de la société.

D'autre part, le procédé de conception ne peut être repris partiellement. Le choix du matériel est réalisé en temps différé. Une modification sur un contact par exemple nécessite la relance complète du traitement et non du seul processus qui a conduit au choix du contacteur commandant le contact. En fait, le contacteur n'est pas stocké dans une base de données où on pourrait venir le modifier. La notion de "base application" n'existe pas: le traitement exploite des schémas pour faire d'autres documents et non pour constituer des fichiers représentant l'application ré-exploitable par la suite.

5113. Conclusion

Conscients de ces insuffisances, mais désireux de conserver le mieux possible tout l'acquis de l'outil, les responsables du développement CSAO ont spécifié une nouvelle version du logiciel qui devrait introduire de manière plus importante la notion de base application, en liaison éventuelle avec les travaux de BASEPTA. Dans l'optique d'une communication facilitée avec les futures consoles automates TELEMECANIQUE et par là même de s'engager sur la voie d'un PTA, le choix s'est porté sur un développement PS.

512. La CDAO Pneumatique (Conception et Dessin Assistée par Ordinateur pur les automatismes à commande Pneumatique)

5121. Description sommaire

Généralités

La CDAO Pneumatique est un outil logiciel de CAO pour les automatismes à commande pneumatique. Il est implanté sur la T607, console de programmation pour les automates TELEMECANIQUE de haut de gamme TSX67 et TSX87 et utilise l'éditeur Grafcet de cette console.

Conçue entièrement par TELEMECANIQUE, la CDAO Pneumatique est à l'heure actuelle utilisée dans les CTR. Le logiciel permet la réalisation de l'ensemble du dossier technique des automatismes à commande pneumatique, c'est à dire essentiellement des petits automatismes de 6 à 10 vérins et moins de 40 entrées/sorties.

Utilisation

Le logiciel comprend quatre phases:

- établissement du grafcet de fonctionnement,
- spécifications technologiques,
- modes de marche et d'arrêt,
- implantation.

Le traitement d'une affaire nouvelle s'effectue selon ces quatre phases. Pour modifier une affaire déjà existante, il est possible de reprendre le traitement au niveau de l'une de ces phases, le choix du point de reprise étant fonction de la nature de la modification:

- si la modification concerne le cycle, il faut reprendre le traitement au niveau Grafcet,
- si la modification concerne le type de capteur choisi, il faut reprendre au niveau spécifications technologiques...

Dans tous les cas, les phases n'étant pas indépendantes, il faut reprendre toutes les phases se situant en aval de la phase de reprise. Mais il est possible à l'issue de chaque phase d'interrompre le traitement d'une affaire pour sauvegarder les données en vue d'une reprise ultérieure ou d'une édition partielle.

Le traitement d'une affaire se déroule comme suit:

- Saisie des données sur l'application. Ces données (référence client, référence affaire...) concernent l'ensemble des documents. On les retrouve notamment dans les cartouches.

- Grafcet de fonctionnement. L'acquisition du grafcet décrivant les modes de marches "séquentiels" utilise un éditeur Grafcet pour la console T607. Cette opération se déroule en deux phases:

* acquisition de la structure grafcet. La saisie est guidée par des menus dynamiques éliminant a priori les choix impossibles. En fin de saisie, le système s'assure que la structure est syntaxiquement correcte.

* acquisition des réceptivités des transitions et des actions associées aux étapes suivant une syntaxe bien précise.

- Spécifications technologiques. Les spécifications technologiques permettent d'explicitier les variables utilisées au niveau des réceptivités et actions du grafcet. Elles précisent notamment pour chaque mouvement le signal de pilotage, le distributeur, le nom et le type du capteur.

- Modes de marches et d'arrêts. La saisie se fait en décrivant un GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts) simplifié. L'ensemble des cases du GEMMA compatible avec le grafcet précédemment décrit est proposé à l'opérateur qui précise les cases qu'il désire conserver. Les liaisons imposées sont automatiquement tracées. Les autres le sont en fonction des réponses aux questions posées.

Implantation

Le système propose à l'opérateur une implantation optimale sur la plus petite platine respectant l'alignement grafcet, un entraxe des profilés donné, la séparation logique d'entrée - logique de sortie. L'opérateur peut modifier cette implantation en changeant la place des éléments, la taille de la platine, l'alignement grafcet ou l'entraxe des profilés.

Edition du dossier

Une fois ces opérations réalisées, il est possible d'éditer partiellement ou totalement le dossier.

PARTIE I - EXPOSE DU PROBLEME

Le dossier complet comprend:

- le grafcet,
- le GEMMA ainsi que des renseignements complémentaires sur certains modes de marche et d'arrêt,
- la description des entrées/sorties séquenceurs ou spécifications technologiques,
- la nomenclature de l'ensemble des équipements entrant dans la composition du coffret de commande plus éventuellement la platine et le coffret si une platine standard a été choisie,
- le synoptique d'implantation,
- la liste de montage donnant les références des produits à monter,
- la liste de câblage,
- le schéma de principe séquenceur,
- le schéma de principe relais,
- le schéma de principe modes de marches et d'arrêts,
- le schéma de principe borniers,
- le schéma de principe pupitre,
- le schéma de principe distributeur.

5122. Analyse de la CDAO

Comme la CSAO, la CDAO Pneumatique est limitée à une seule technologie: le pneumatique pour la commande et les actionneurs. Il est ainsi impossible par exemple d'intégrer des aspects logiciels.

Le principe est différent de celui de la CSAO, en ce sens que les schémas sont générés de manière automatique par le logiciel et ne constituent pas les informations de saisie. De plus, le système intègre un aspect fonctionnel. En effet, les constituants matériels sont choisis suivant la fonctionnalité à remplir et les schémas sont ensuite tracés. Cependant cette fonctionnalité est toujours de bas niveau (par ex: distributeur bistable). Le système n'intègre pas de

fonctionnalité de plus haut niveau type "macro-constituant" et ne permet pas la réutilisation de sous-ensembles déjà réalisés. Le problème est de plus très simplifié car, compte-tenu des spécifications technologiques qu'introduit l'opérateur, le composant adéquat est unique. Le choix est déterministe.

Il faut également noter que le logiciel CDAO Pneumatique se situe relativement en aval dans la conception. En effet, la synthèse de la commande doit déjà être réalisée puisque le grafcet introduit par l'opérateur n'est pas un grafcet de spécifications fonctionnelles mais un grafcet de spécifications technologiques [GRA 85A] qui implique, par exemple que la manière dont est réalisée la distribution (avec des monostables ou des bistables) a été choisie. De façon schématique, on pourrait dire que la CDAO est un outil de choix de matériel dans un catalogue et de réalisation de schémas de mise en oeuvre et de maintenance à partir d'une description de la commande. Cette description est faite au moyen d'outils formels (Grafcet et GEMMA) qui, bien que simplifiés ici, permettent un certain niveau d'abstraction, notamment par l'utilisation de variables symboliques. Le logiciel exploite toute une connaissance sur la manière de passer du grafcet aux séquenceurs et sur la façon de réaliser les modes de marche, connaissance que l'on trouve d'ailleurs formalisée dans le manuel de la commande pneumatique TELEMECANIQUE [TEL 82A].

Sur le plan de la qualité de la documentation générée, la CDAO ne souffre aucune comparaison avec la CSAO compte-tenu de la différence des moyens matériels (traceur électrostatique pour la CSAO, petite imprimante graphique pour la CDAO). Cependant, comme pour celui de la CSAO, le dossier issu de la CDAO présente l'avantage d'être cohérent et homogène dans sa présentation.

5123. Conclusion

Afin de pallier le principal inconvénient de la CDAO Pneumatique qui est de fonctionner sur une console spécifique TELEMECANIQUE, il est prévu de transférer le logiciel sur PC/PS. Le système comprendra de plus un module de calcul de devis et un début de "base application". En effet, la logique des schémas sera mémorisée indépendamment de leur représentation.

513. Le logiciel PL7-3

5131. Description sommaire

Généralités

Le logiciel PL7-3 est un ensemble d'outils de développement et de "mise au point" des logiciels pour les automates TSX67 et TXS87 de TELEMECANIQUE. Il est actuellement installé sur la console haut de gamme T607. Il permet l'écriture des programmes et offre une documentation complète du logiciel de l'application. Il facilite les tests, la mise au point et le réglage.

Il est essentiel de noter que le point d'entrée de PL7-3 n'est pas la description de la commande de l'application (description qui serait indépendante de la réalisation de cette commande) mais une description du logiciel de l'application qui dépend notamment de la structure de programmation proposée par l'automate cible. Cette distinction essentielle donne un éclairage sur la structure de données BASEPTA ainsi que nous le verrons dans la partie II.

Utilisation

Les deux caractéristiques principales du logiciel PL7-3 sont ses aspects multi-langages et multi-tâches.

Le superviseur PL7-3 gère automatiquement l'enchaînement et l'exécution des différentes tâches. Ceci facilite le travail du programmeur qui utilise la structure de tâches fournie par PL7-3 pour programmer de manière indépendante ses tâches sans avoir à programmer le système de gestion de tâches. Les tâches fournies par PL7-3 comprennent un programme principal et des sous-programmes associés. Certaines sont asynchrones (tâche interruption), d'autres sont périodiques (tâche rapide, tâche maître, tâches auxiliaires). La tâche maître destinée au traitement séquentiel est composée de trois modules:

- le préliminaire permettant de traiter les modes de marches, la logique d'entrée, les initialisations et les reprises secteur,
- le séquentiel structuré grâce aux macro-étapes,

- le postérieur élaborant les variables de sortie destinées aux actionneurs et tenant compte des sécurités.

PL7-3 est également multi-langage, ce qui permet au programmeur de choisir le langage convenant le mieux à son type d'application:

- le Grafcet qui, outre son mode de saisie graphique, possède l'avantage d'être normalisé sur le plan international [IEC 848],

- le langage à contacts dont la saisie est également graphique et qui, présentant une similitude avec le traitement en relais familier aux électro-mécaniciens, est bien adapté aux traitements logiques de type combinatoire,

- le littéral, langage évolué de type algorithmique, qui est bien adapté à la programmation de fonctions telles que calculs arithmétique, manipulations de tableaux, gestion de messages...

De plus, PL7-3 met à la disposition du programmeur un certain nombre de ressources logicielles utilisables en littéral, et langage à contacts: les blocs fonctions tels que temporisateurs, compteurs...

Configuration

L'opérateur commence par saisir la configuration de son application. Il indique la mémoire, la composition des cartes d'entrées/sorties et paramètre son application (configuration des tâches, des blocs fonctions, des constantes, du grafcet...). Toutes ces informations permettent des contrôles en temps réel lors de la saisie du programme. Ainsi le logiciel vérifie que les variables d'entrées/sorties utilisées dans le programme correspondent bien à des ressources configurées.

Ecriture du programme

La saisie d'un programme littéral est alphanumérique, celle d'un programme grafcet ou langage à contacts est graphique. Elle est guidée par des menus dynamiques ne proposant que les choix a priori possibles dans le contexte courant (ex: respect de l'alternance étape-transition). A la fin de la saisie d'une ligne de

littéral, d'un réseau de contacts, d'une page de grafct, le logiciel controle que la syntaxe a été respectée et enregistre la saisie sous une forme exécutable par l'automate.

Mise au point

PL7-3 permet à l'utilisateur:

- visualisation et animation du grafct et des réseaux de contacts,
- suivi en temps réel de l'évolution des variables,
- visualisation de l'historique d'un bit, d'un mot...,
- recherche de bits forcés ou bloqués,
- modification des valeurs des variables en cours d'exécution,
- activation ou désactivation forcée d'étapes,
- gel ou dégel de l'évolution du grafct.

Edition de la documentation logicielle de l'application

Les TSX67 et 87 contiennent en mémoire les programmes mais aussi les données complémentaires (commentaires associés, configuration matérielle et logicielle). L'édition d'un dossier complet du logiciel à partir de l'automate est donc possible grâce à une console T607 quelconque. Ce dossier contient:

- les caractéristiques de l'application (date, nom, tâches, sous-programmes),
- la configuration matérielle (nombre de racks, type des modules d'entrées/sorties, configuration mémoire),
- la configuration des tâches (nature et périodicité, nombre de sous-programmes)
- la configuration du grafct (nombre d'étapes et de macro-étapes),
- la configuration des blocs fonctions,
- la valeur des constantes,
- les listings de programmes avec les commentaires associés.

5132. Analyse de PL7-3

Le problème traité par PL7-3 est limité exclusivement au cas des automatismes programmables. Aucune interface n'est possible d'autant plus que le logiciel est spécifique à un matériel: la T607.

PL7-3 se situe très en aval dans la démarche de conception puisque la synthèse du programme est déjà réalisée. Cependant, en offrant des ressources du type blocs fonctions, mécanisme de tâches, le logiciel assiste beaucoup le concepteur dans cette synthèse.

Contrairement à la CSAO et à la CDAO Pneumatique, PL7-3 dépasse le stade de la conception et peut être utilisé en exploitation et au cours de la mise en oeuvre. Il offre en effet la possibilité de visualiser en temps réel le déroulement du programme, soit en suivant, l'évolution des données, soit en animant le grafset. En connectant à l'automate, non pas les entrées/sorties réelles du process mais des entrées/sorties simulées, l'opérateur pourra effectuer une simulation temps réel de l'automatisme. D'autre part, même s'il existe chez TELEMECANIQUE des consoles spécifiques pour l'exploitation (terminaux XBT), il est toujours possible d'utiliser les possibilités de visualisation du déroulement du programme sur la T607.

La documentation générée, si elle n'est pas d'une grande qualité graphique, compte-tenu des moyens d'impression bas de gamme, présente cependant l'avantage d'être l'image fidèle des informations stockées dans l'automate. Une certaine cohérence du dossier de l'automatisme programmable est assurée en ce sens que la compatibilité du programme avec la configuration matérielle de l'automate est vérifiée.

5133. Conclusion

PL7-3, malgré ses qualités et son apparition récente sur le marché, résulte d'une conception déjà ancienne de la console automate. Dans l'optique d'une console nouvelle génération, TELEMECANIQUE va bientôt proposer un PL7-3 fonctionnant sur PC/PS avec des modifications importantes telles que la possibilité d'utiliser des variables symboliques dans le programme (augmentation du niveau d'abstraction), l'extension des blocs fonctions constructeurs (intégration de fonctions plus évoluées type régulation ou diagnostic process) et la possibilité pour le programmeur de se définir ses propres blocs fonctions utilisateurs (réutilisation de solutions déjà élaborées). PL7-3 nouvelle version sera certainement plus ouvert et à la demande de certains clients, il est envisagé de pouvoir échanger avec une CAO électrique la table des symboles, la configuration des entrées/sorties et les constantes.

514. Conclusion sur les outils TELEMECANIQUE

Les trois outils logiciels que nous venons d'étudier sont très différents et dans l'état actuel ne communiquent pas. De manière simplifiée, nous avons choisi de les situer sur un graphisme montrant leur spécificité respective.

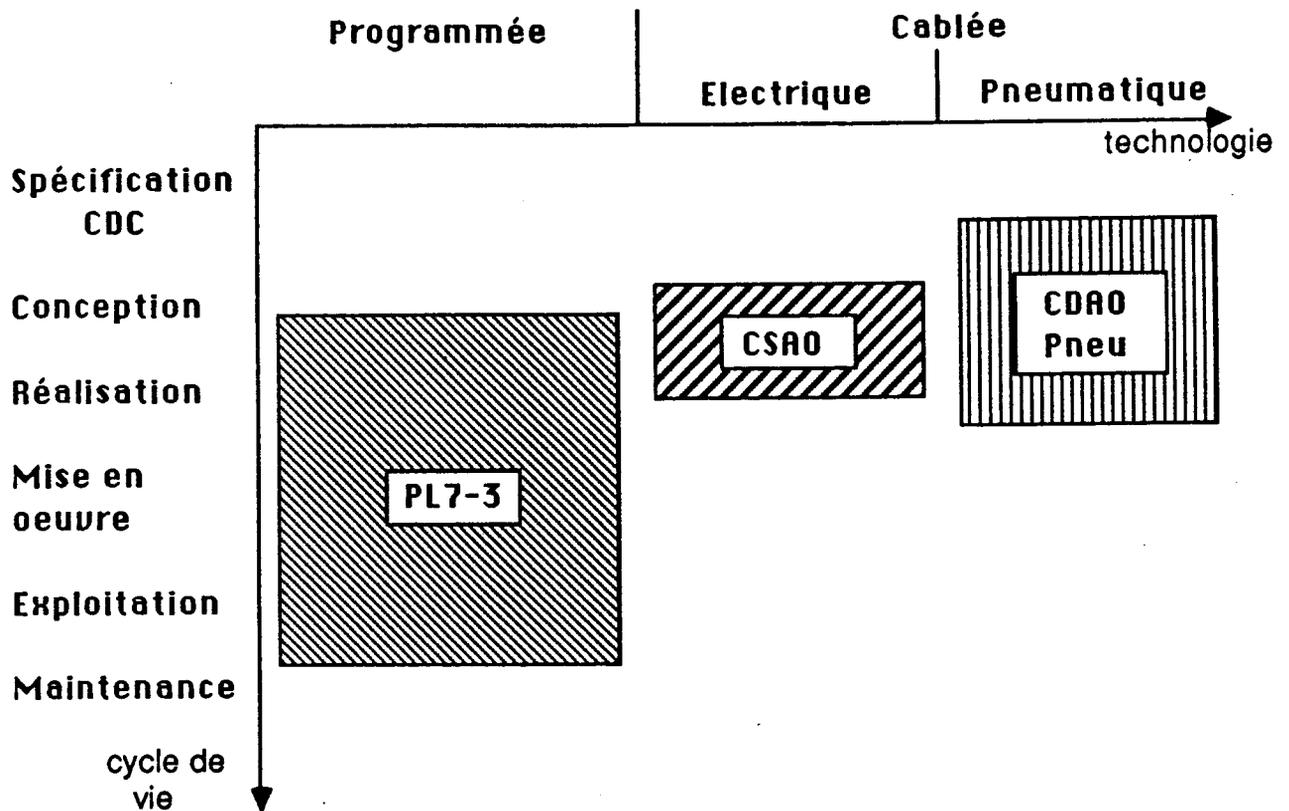


Fig 7

Cependant, ces logiciels vont être disponibles sous peu sur le même type de machine informatique. La connexion ou l'interfacage de ces logiciels va donc être simplifiée. Elle pourrait se faire grâce à la couche supérieure que constitue la Base Application.

52. Autres outils disponibles

Nous allons maintenant passer en revue de manière

beaucoup plus sommaire les principaux outils logiciels pour automaticiens disponibles sur le marché. Nous distinguerons les outils de schématique des outils de conception de logiciel, ce qui est possible car aucun outil actuel n'intègre les deux aspects.

521. Les outils de conception de logiciel

5211. OMEGA

Conçu par la société 3IP, OMEGA n'est pas à proprement parler un outil de conception de logiciel. Il s'agit d'un éditeur graphique très performant de grafcet et de GEMMA. OMEGA peut très bien être employé, dans certaines de ses fonctionnalités, par un utilisateur de grafcet visant un domaine d'application différent de celui des automatismes.

Le logiciel propose une approche structurée. L'application peut être décomposée en modules grafcet hiérarchisés et pour chaque module, il est possible de définir des blocs combinatoires d'entrée et combinatoires de sortie en langage à contacts ou équations logiques ou à l'aide de blocs fonctions pour des fonctions complexes. OMEGA intègre la notion de macro-étape mais aussi, et c'est une de ses particularités, la notion de tâche grafcet. Les grafcets peuvent être simulés de manière logique ou temporisés. Un certain nombre de facilités est offert pour la simulation: mode pas à pas, mode continu, nombre de cycles variable, temporisation de chaque étape, production de chronogrammes, vérification de synchro entre grafcets différents de l'application. OMEGA fournit également un module GEMMA pour spécifier les modes de marches et d'arrêts. L'utilisateur peut se définir son propre standard de GEMMA (élimination de cases correspondant à des modes jugés peu intéressants). Enfin, la documentation générée est très riche, l'utilisateur pouvant associer grâce à un éditeur de textes des commentaires sur les actions, les transitions, les modes de marche...

En définitive, OMEGA est sans doute parmi les plus performants éditeurs grafcet disponibles sur le marché. Il présente l'avantage de pouvoir décrire une spécification grafcet de manière hiérarchisée et modulaire et d'intégrer les notions de modes de marches et d'arrêts, quoique les liens grafcet et GEMMA soient très faibles. Il faut cependant remarquer qu'OMEGA se

situé en amont dans le cycle de vie de l'automatisme. C'est essentiellement un outil de spécification. Il n'y a pas de génération de programme automate (sauf développements à l'usage spécifique de certains clients) et, en fin de compte, actions et réceptivités sont des commentaires. Leur syntaxe est d'ailleurs très libre. OMEGA permet de décrire aussi bien du grafcet de spécifications fonctionnelles que technologiques.

5212. CADEPA

Développé initialement à l'ADEPA, CADEPA est à l'heure actuelle commercialisé par SGN. Il s'agit en quelque sorte d'une console automate universelle qui à partir d'une spécification de programme en grafcet génère une documentation homogène et indépendante de l'automate, du "code automate" pour un certain nombre d'automates du marché et permet le téléchargement.

Le point d'entrée de CADEPA est la spécification grafcet de niveau technologique. Il suppose par conséquent la commande entièrement conçue. La saisie se fait de manière graphique avec une ergonomie assez rustique ou en mode littéral, ce qui permet de "récupérer" des grafcets élaborés par d'autres CAO. C'est ainsi que pour pallier les insuffisances ergonomiques de CADEPA d'une part, et l'absence de génération de code automate chez OMEGA d'autre part, 3IP et VALORIS proposent une solution où les grafcets élaborés sur OMEGA (en respectant toutefois une syntaxe plus rigoureuse) peuvent être communiqués en entrée à CADEPA.

A notre avis, le seul intérêt de CADEPA est de permettre une spécification décrite intégralement en grafcet, ceci de manière indépendante de l'automate cible. Cependant CADEPA, comme d'ailleurs le Grafcet lui-même, ne comporte pas de méthodologie particulière d'élaboration de la commande. Ceci est conçu par les concepteurs d'automatismes comme une faiblesse de CADEPA par rapport à des outils comme GRAL que nous proposons d'aborder maintenant.

5213. GRAL

Développé par la société SYGRAL, GRAL est un outil de conception et de codage des programmes pour automates. GRAL permet à l'opérateur de structurer son application en tâches d'après une décomposition fonctionnelle préalable. Chacune des tâches comprend un corps, une liste de variables à renseigner et une structure de contrôle. La programmation des corps de tâches se fait en utilisant le langage le plus adapté au problème à résoudre: diagramme de séquences pour les aspects séquentiels, schémas à relais avec boîtes fonctionnelles pour le combinatoire et littéral de l'automate cible. GRAL n'oblige à renseigner que les variables d'entrées/sorties associées à chaque tâche mais peut affecter automatiquement les autres ressources bits ou mots de l'automate. Enfin la structure de contrôle de chaque tâche définit le contexte d'exploitation de la tâche et en particulier ses conditions d'activité.

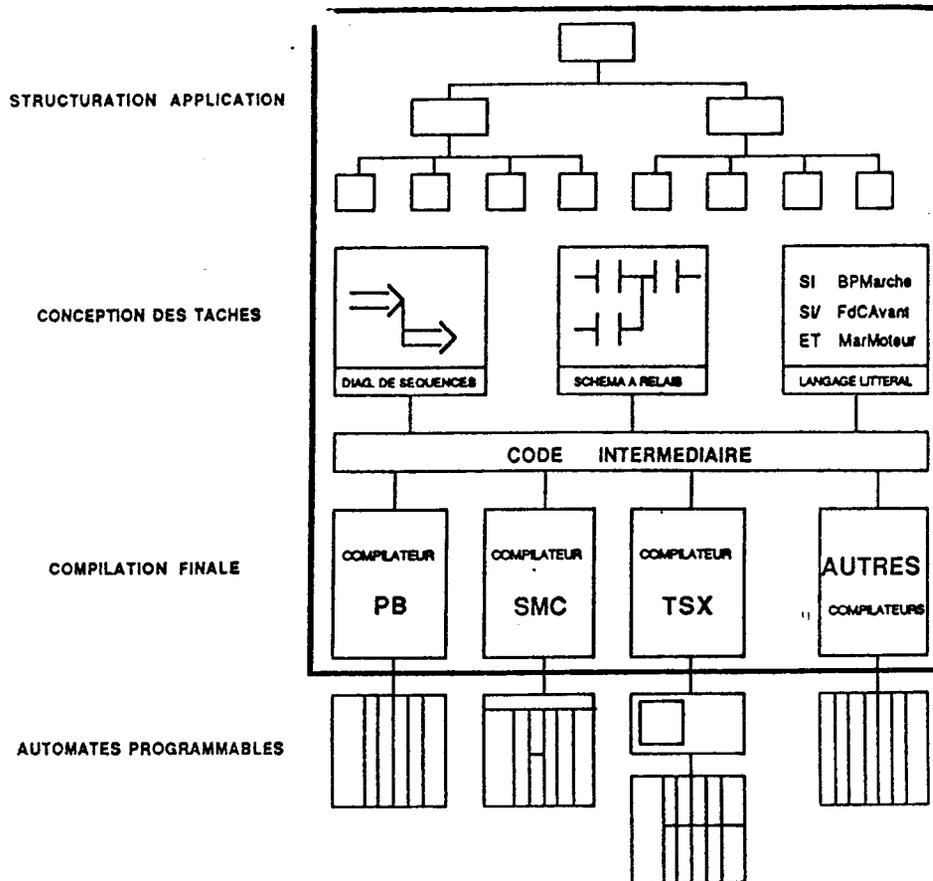


Fig 8: Organisation de GRAL [GRA 87A]

GRAL génère du code compréhensible par l'automate (cas des PB) ou par la console de programmation (cas des TSX). Il fournit une aide à la mise en service (ex visualisation dynamique) et un dossier complet comprenant la partie conception (structure de l'application en tâches et contenu de chaque tâche: corps, variables, structures de contrôle) et la partie réalisation (listing du programme automate résultant et références croisées). Il fournit également toute une structure de bibliothèques pour les applications (possibilité de réutilisation de solutions) et les boîtes fonctionnelles paramétrables.

Comme CADEPA, GRAL se veut un produit multi-automate cible mais son ergonomie est plus soignée que celle du produit de VALORIS. Il introduit surtout une méthode de conception dite "méthode GRAL": L'application est décomposée en tâches (analyse fonctionnelle) puis développée (écriture des tâches et de leur structure de contrôle) et enfin compilée pour aboutir au langage compréhensible par l'automate ou la console. Cette méthode, assez voisine de celle proposée par les consoles TELEMECANIQUE, se situe cependant à un niveau d'abstraction plus élevé. Son but est de déboucher sur une standardisation des règles de conception applicable à des automates différents.

Enfin, il faut noter la volonté affichée de SYGRAL de faire de GRAL un produit ouvert, interfaçable avec des CAO électriques. Le choix du diagramme de séquences de préférences au grafcet, quoique discutable, montre que l'outil est avant tout destiné à l'homme de process.

5214. Autres produits logiciels

Dans ce survol des outils de conception de logiciels, nous pouvons également citer CALIN étudié par le laboratoire d'automatique et d'informatique de Marseille et commercialisé par CYBERNETIX. C'est un concurrent direct de CADEPA auquel il ressemble beaucoup. Sa cible n'est cependant pas uniquement l'automate classique mais aussi les cartes micro-processeurs standards.

CJGRAF, produit de la société CJB AUTOMATION, est également un produit dont le point d'entrée est la description du programme de commande en grafcet (la saisie est littérale) mais il est plutôt destiné aux intégrateurs en informatique industrielle devant réaliser des applications complexes sur une carte spécifique

développé autour d'un micro type INTEL ou MOTOROLA. CJGRAF traite la partie séquentielle, souvent réduite, de ces automatismes.

Enfin pour terminer, notons que dans cette catégorie d'outils logiciels figurent également toutes les consoles automates. Citons aussi deux outils très particuliers développés par des utilisateurs: MULTIDOC (CITROEN INDUSTRIE) et EDILOG (RENAULT) qui sont des outils de documentation à partir de la réalisation. Ces outils lisent le programme dans l'automate et génère à partir de là une documentation avec listing et références croisées.

522. Les outils de schématique électrique

Si l'on exclut les logiciels de DAO pour lesquels le schéma n'est envisagé que de manière graphique et non sur le plan connectique, le nombre des outils de schématique électrique est très réduit. En plus de SAFIRS d'ASSIGRAPH que nous avons déjà étudié, nous en présenterons deux: XELEC et DESCARTES.

5221. XELEC

En liaison avec le laboratoire CPAO de l'ENSTA à Palaiseau, la société XAO a développé un logiciel de schématique électrique sur station de travail. D'une ergonomie très soignée, le logiciel bénéficie de la richesse de son support informatique de haut niveau, tant sur le plan de la qualité des documents que de la rapidité de traitement, ce qui lui a valu d'être retenu par certains constructeurs automobiles. Mais son intérêt majeur provient des concepts riches qu'il intègre sur le plan de la structure de données. En effet, les schémas sont mémorisés en dissociant l'aspect graphique ou représentation de l'aspect connectique ou assemblage de symboles. Ainsi, contrairement à la CSAO dans son état actuel, un traitement sur la logique du schéma (ex vérification des fils en l'air) ne nécessite pas un parcours graphique du document mais une simple analyse des liaisons logiques. Il faut également préciser que le modèle conceptuel de la structure de données de XELEC est assez voisin du modèle de données du standard d'échange SET schématique dont nous parlerons plus loin.

XELEC se compose de plusieurs modules:

- XSCHM qui réalise l'aide à la conception des schémas,
- XVALID qui traite ces schémas,
- XNOMEN qui élabore les nomenclatures,
- XCABLE qui permet d'établir les interconnexions entre les enveloppes (armoires).

Un dernier module XPLAN, actuellement en développement, fournit une aide à l'implantation en armoire proche de celle de la CSAO. Dans l'état présent XELEC n'est pas lié à un système de conception de logiciel.

5222. DESCARTES

Conçu par la société WHITEK créée par des transfuges d'ASSIGRAPH, DESCARTES est un logiciel de CAO électrique sur PC. Il permet l'élaboration des schémas de principe, le traitement de ces schémas (contrôles électriques, numérotation d'équipotentiels...), le choix du matériel, l'édition de nomenclatures et la préparation des folios d'implantation en armoire.

Comme XELEC, il intègre le caractère logique d'un schéma. Il est ainsi possible d'afficher en interactif les équipotentiels sans avoir comme dans la CSAO actuelle, besoin de lancer un traitement important sur l'ensemble du folio. De plus, DESCARTES possède une ébauche de Base Application. Il comprend effectivement une base de données temps réel dans laquelle sont stockés tous les noms fonctionnels de l'affaire en cours de traitement avec, pour chaque nom, la nature des symboles qu'il rassemble et les folios sur lesquels on trouve ces symboles.

exemple d'information disponible à tout instant:

KM1	BOBINE	Folio 011
	CONTACT A FERMETURE	Folio 011
	CONTACT A FERMETURE	Folio 012
	GROUPE DE 3 POLES	Folio 021

DESCARTES possède une grande richesse dans l'organisation des données (cf Figures 9 et 10), richesse qui à l'heure actuelle n'est pas pleinement exploitée. Il s'agit d'un produit très ouvert.

5223. Conclusion sur les outils de schématique

Jusqu'à présent, les outils de schématique n'offrent pas de liaison avec les outils de conception de logiciel. De

plus la représentation des données des schémas est spécifique à chaque système (base de données spécifique pour DESCARTES, base de données ORACLE probablement pour les prochaines versions de SAFIRS). Les choix au niveau graphique sont différents (GKS pour DESCARTES, WINDOWS pour le futur SAFIRS). Les outils de schématisation ne peuvent donc pas communiquer entre eux et s'échanger des documents. C'est ce problème précis qui a motivé les travaux sur SET schématique.

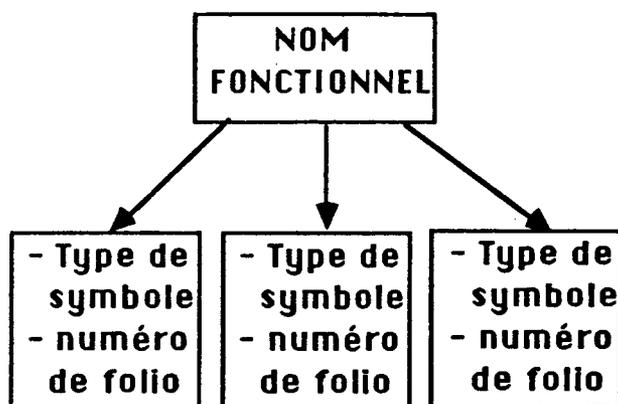


Fig 9

Base de données
"application"
dans DESCARTES

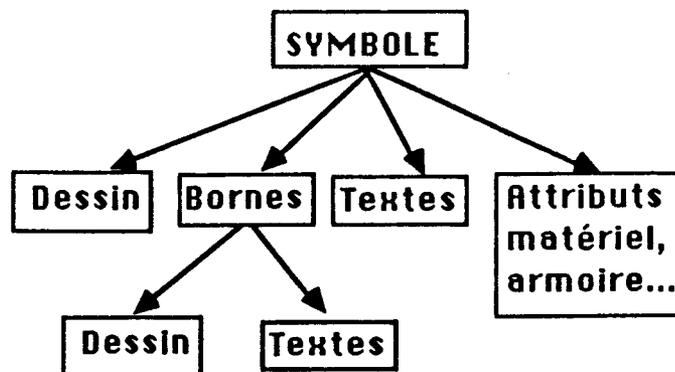


Fig 10

Description d'un symbole
dans DESCARTES

523. Conclusion sur l'étude des outils logiciels

Le dessin de la figure 11 montre une situation analogue à celle rencontrée pour les outils TELEMÉCANIQUE. Il n'existe pas actuellement de communication entre les outils de CAO électriques et les outils de conception de logiciel pour automates, même si les fabricants de ces outils affirment leur volonté de communiquer.

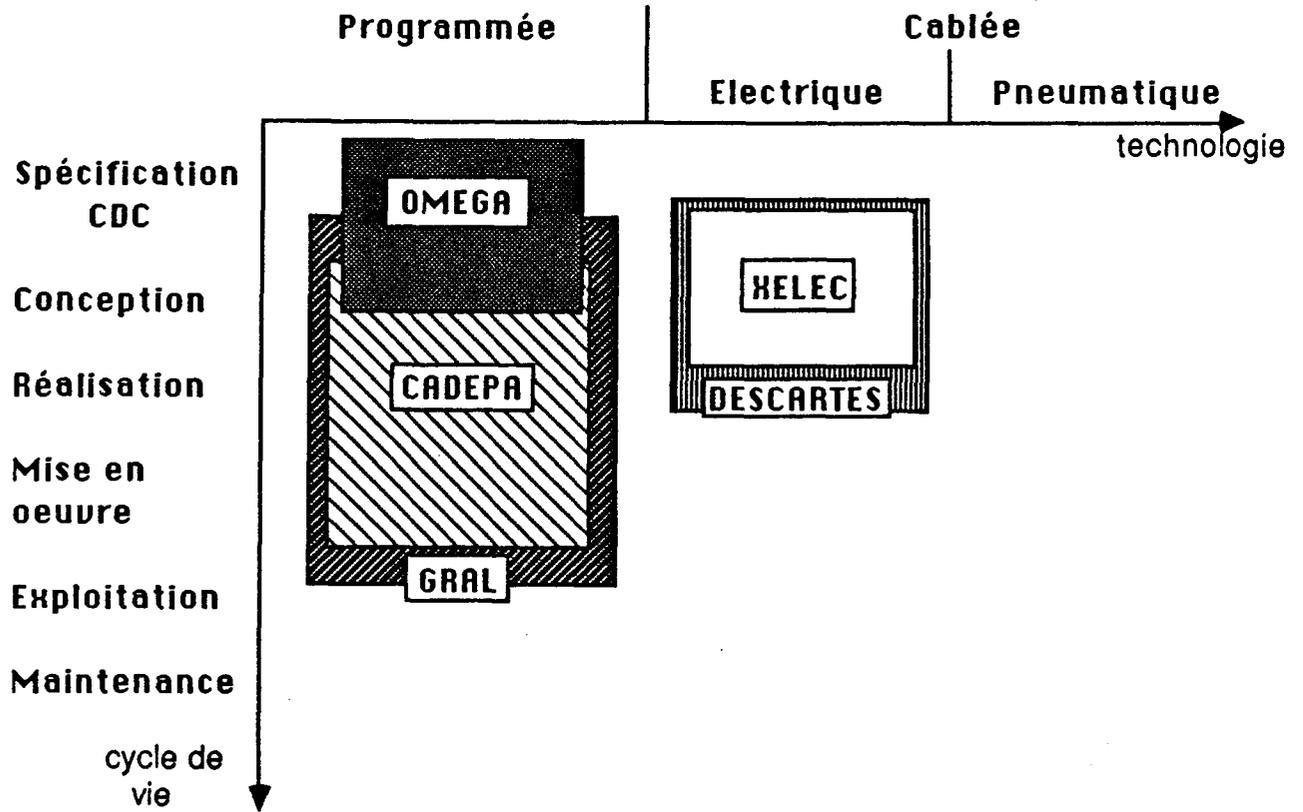


Fig 11

6. LE POSTE DE TRAVAIL DE L'AUTOMATICIEN

61. Synthèse sur l'étude des outils logiciels

Les outils logiciels présentés précédemment sont tous monotecnologies. Il en est de même pour la quasi-totalité des autres logiciels disponibles actuellement.

Tous ces outils d'aide à la conception d'automatismes industriels possèdent un logiciel de base spécifique ainsi qu'une ou plusieurs bases de données spécifiques dont l'organisation, tant informatique que logique est fonction des traitements à effectuer. De ce fait, les données produites par une CAO ne sont pas ré-exploitable pour d'autres opérations. Le schéma de la figure 12 présente une CAO classique. Notons que la notion de Base de données composants doit être ici envisagée au sens large. Il peut s'agir, par exemple, d'une simple bibliothèque de symboles pour une CAO rudimentaire.

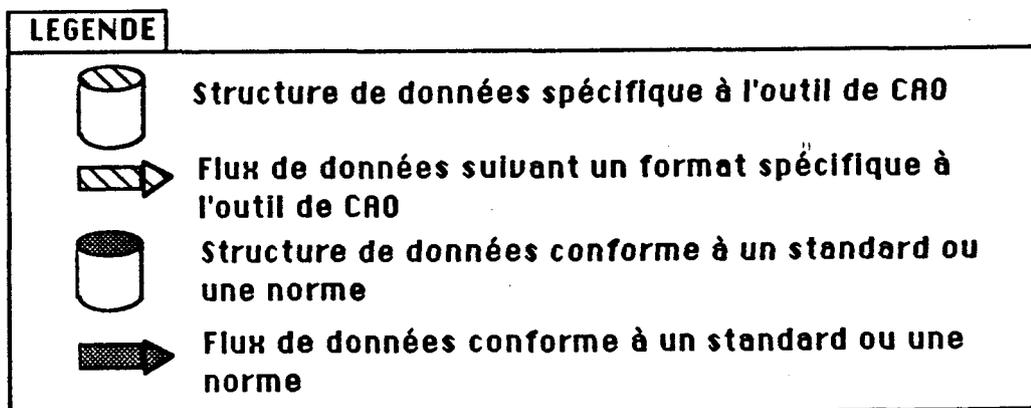
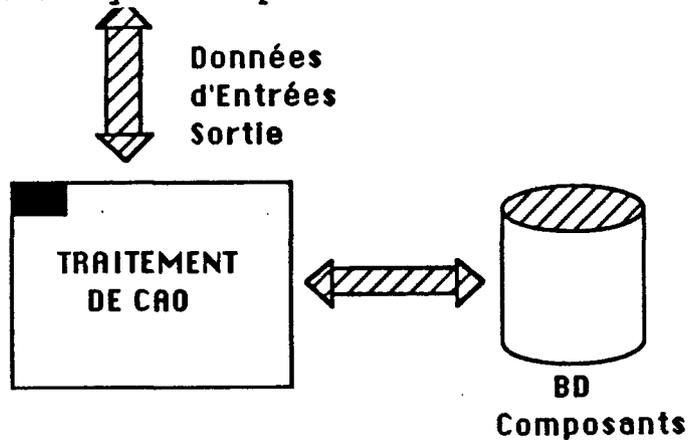


Fig 12 :CAO classique sans structure de données "application"

De plus, comme nous l'avons vu précédemment, il est rare que les CAO conservent en mémoire les données caractéristiques de l'application (et non de sa représentation) pour pouvoir les réutiliser dans des reprises de traitement. Seules certaines CAO possèdent une "Base Application" selon le schéma de la figure 13.

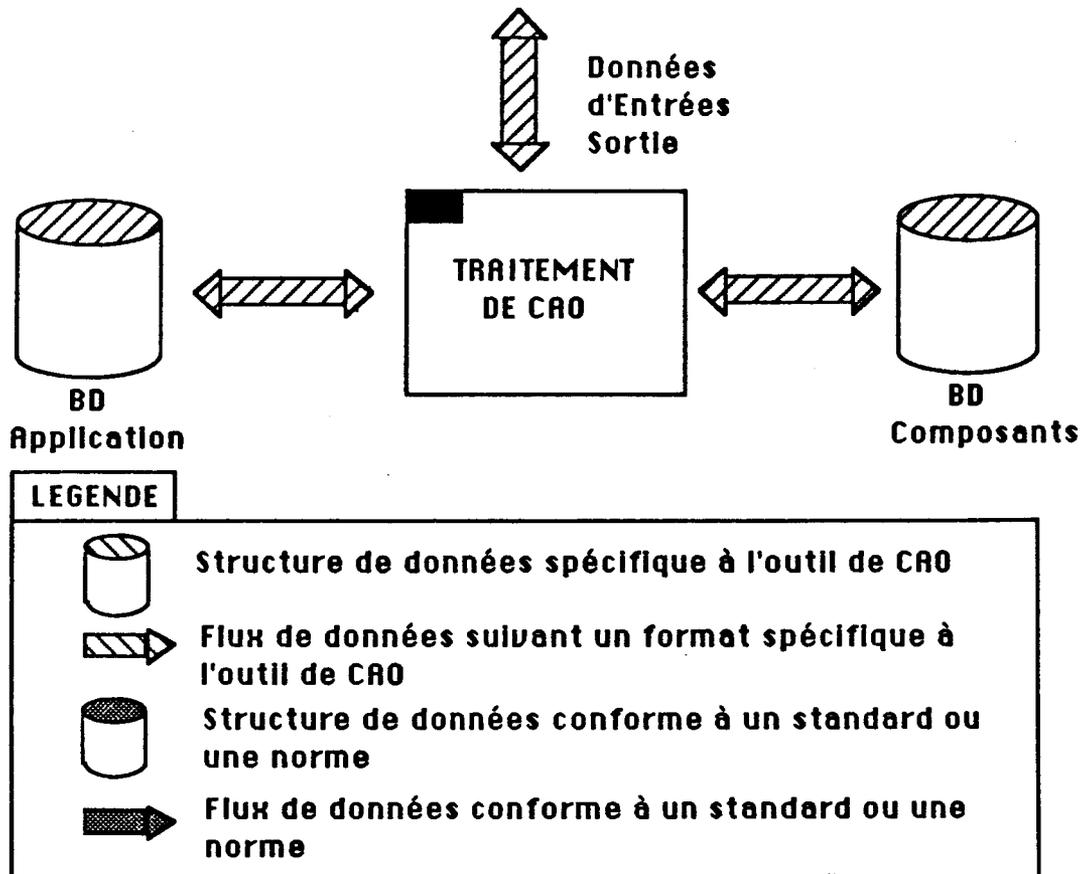


Fig 13: CAO avec structure de données "application"

Dans les outils logiciels existants, une seule partie de l'étude est prise en compte (tracé des schémas, programmation, simulation...) et la notion de cycle de vie est absente. Il en résulte que les dossiers d'affaire produits par ces logiciels sont toujours incomplets et difficilement cohérents.

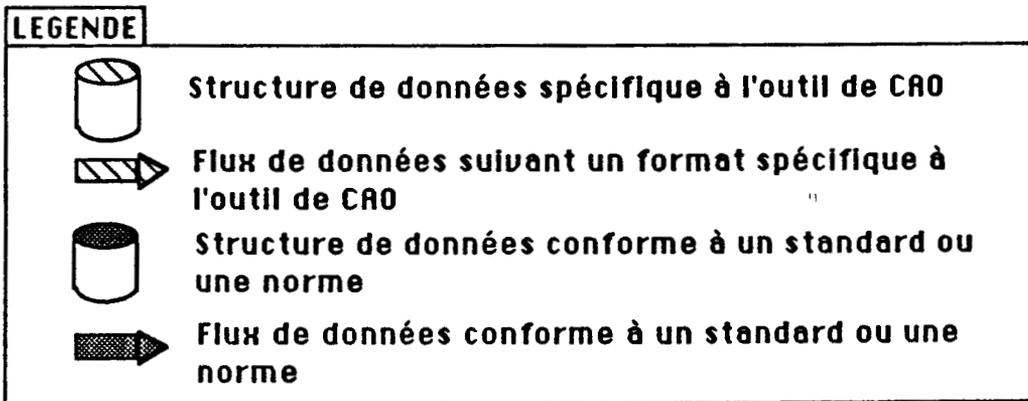
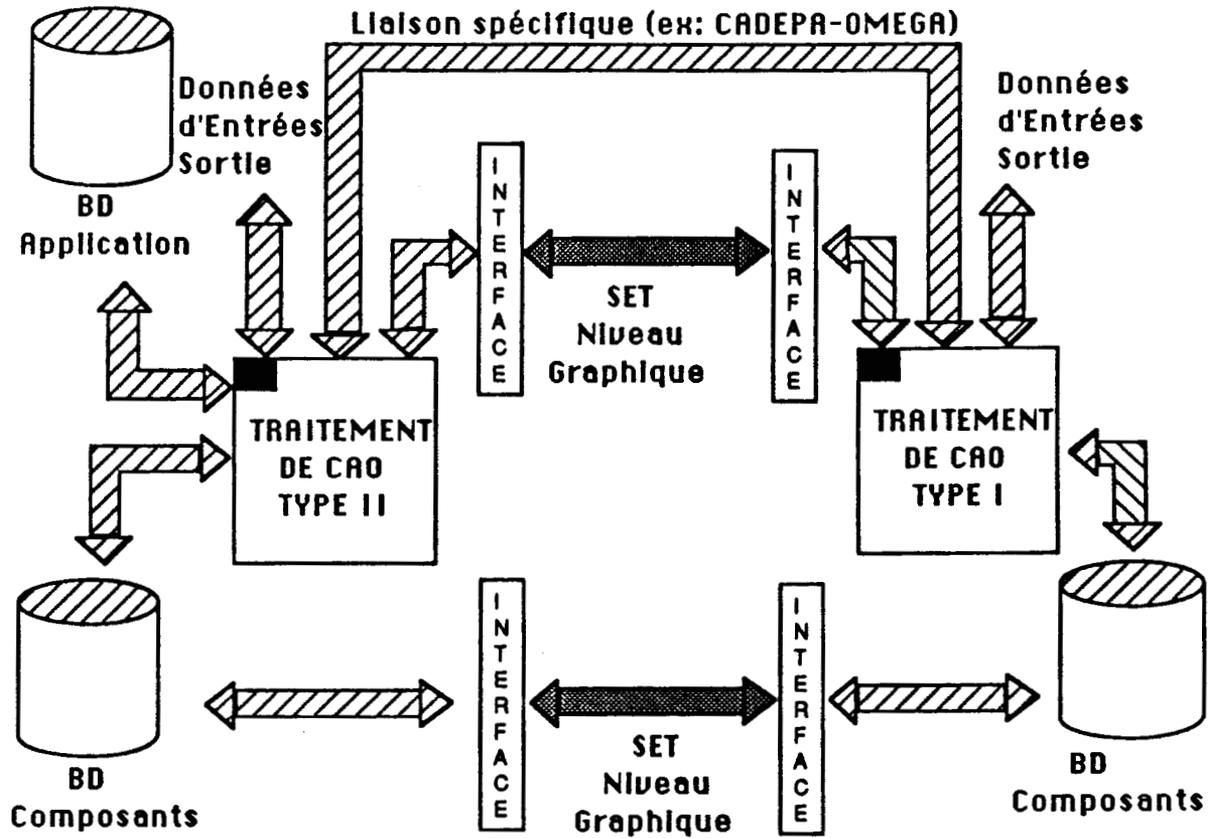


Fig 14: Echange de données entre des systèmes de CAO classiques

Compte-tenu de la spécificité des structures de données de chaque outil logiciel, de la diversité des matériels utilisés (micro-ordinateurs, mini-calculateurs, consoles automate) et de l'absence de standards d'échange, les systèmes sont fermés et incompatibles. Les possibilités d'échange de données sont donc très limitées (cf figure 14).

A l'exclusion de liaisons spécifiques (type OMEGA-CADEPA), les seuls standards d'échange disponibles jusqu'à présent (type SET) ne permettent d'échanger que des informations de nature graphique.

Autrement dit, il existe un besoin de standard analogue à la norme SET en CFAO mécanique. Rappelons que les deux objectifs visés par SET étaient "l'échange d'informations entre des systèmes différents" et "l'archivage d'informations homogènes et cohérentes au sein d'une même base de données centrale pour une réutilisation immédiate ou à long terme" [SET 85A]. Ce sont les mêmes préoccupations que l'on retrouve en CAO d'automatismes.

En conclusion sur l'étude des outils logiciels, on peut dire qu'il n'existe pas d'outil d'aide à la conception intégrant Partie Commande et Partie Opérative, couvrant l'ensemble du cycle de vie de l'application et générant une documentation logicielle et matérielle de conception et d'exploitation-maintenance.

62. Le problème de la cohérence

Afin d'assurer une bonne qualité des applications, le problème de la cohérence doit être résolu. Nous distinguerons deux types de cohérence:

621. La cohérence de l'application au cours de la prestation d'automatisation.

Il est essentiel d'assurer la cohérence de l'application à travers les différentes étapes du cycle de vie. Ceci suppose que les données créées au cours d'une phase (par exemple conception) soient reprises dans les autres phases (par exemple simulation). Il est donc souhaitable que les données qui caractérisent l'application et qui sont utilisées depuis la spécification jusqu'à la maintenance, ne soient pas propres à chaque traitement

mais proviennent d'une base de données commune enrichie, exploitée, modifiée tout au long de la prestation d'automatisation (cf figure 15).

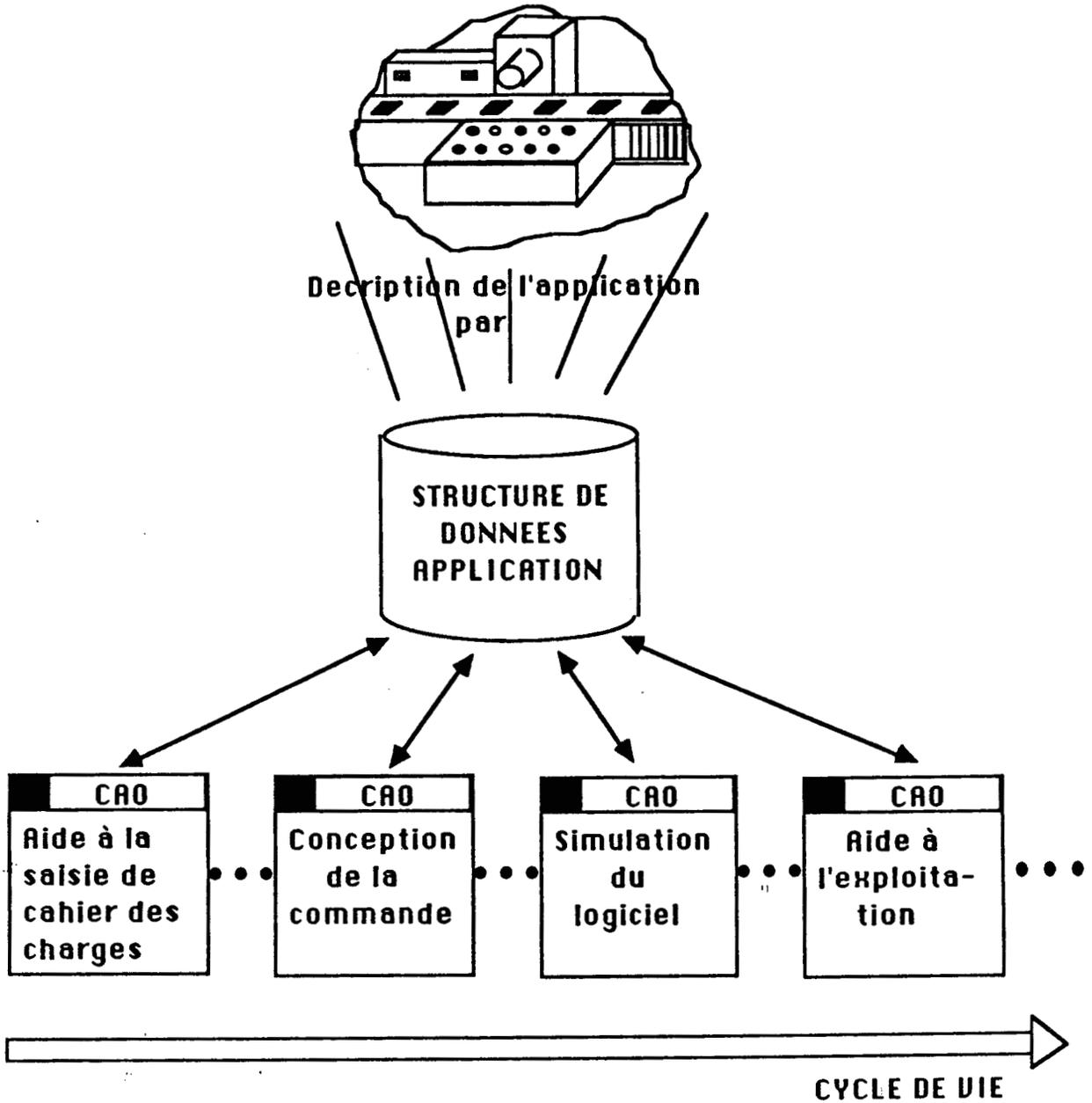


Fig 15

622. La cohérence du dossier

Le dossier d'une affaire est un élément essentiel de la prestation d'automatisation. Sa qualité est très souvent révélatrice de la qualité de l'application automatisée (cf figure 16).

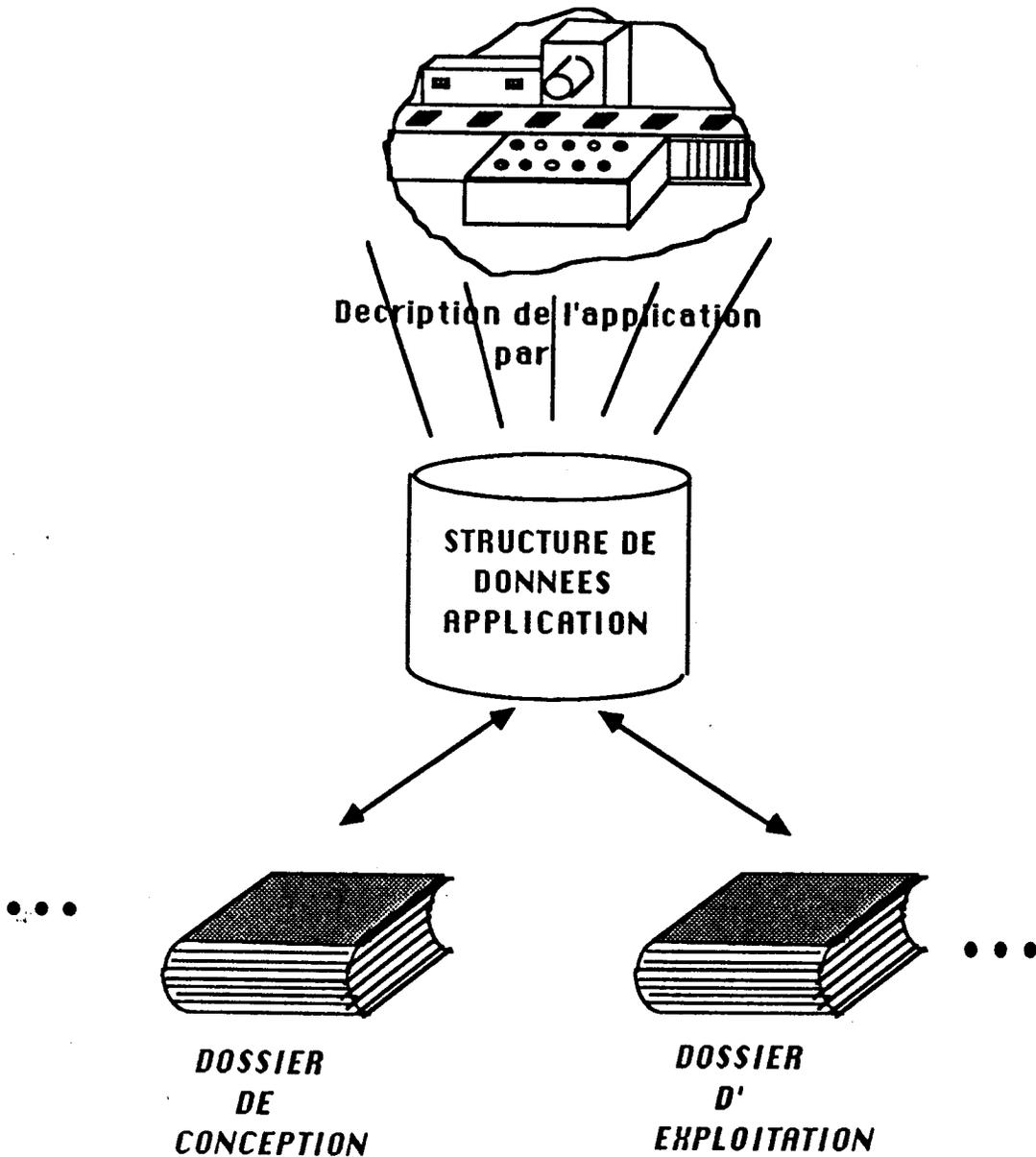


Fig 16

Comme nous l'avons vu précédemment, les dossiers d'affaire sont généralement complexes et variables. La plupart des documents peuvent être considérés comme des aspects ou "points de vue" différents sur les éléments matériels et logiciels constituant l'application. Par exemple, le même contacteur est présent sur le schéma de contrôle, le schéma de puissance, la nomenclature, le schéma d'installation... Le moyen le plus élémentaire d'assurer cette cohérence est de nommer toutes les différentes représentations d'un même constituant sur des vues différentes de la même façon par un identificateur unique, par exemple le nom fonctionnel. Cet identifiant permettra la gestion des modifications sur le constituant au cours de la vie du dossier. Ceci nous conduit également à préciser que la structure de données application doit comporter des informations caractéristiques de l'application (c'est à dire par exemple du constituant) et non de la représentation de l'application (c'est à dire par exemple des symboles qui sont les représentations du constituant suivant un certain point de vue schématique). La structure de données application est donc une garantie de la cohérence du dossier, comme elle est une garantie de la cohérence de l'application.

Comme nous l'avons vu précédemment, les dossiers issus des logiciels de CAO disponibles sont incomplets. Une certaine cohérence est en général assurée, mais elle ne concerne qu'une partie du dossier (sous-dossier électrique, sous-dossier pneumatique...). Actuellement, les seuls dossiers complets et homogènes sont réalisés en manuel. C'est donc l'ingénieur d'affaire, compétent dans les diverses technologies, qui est chargé de la cohérence. C'est lui en particulier qui est responsable des nomenclatures et donc des demandes d'achats.

Ce problème de cohérence du dossier est amplifié par le fait que, compte-tenu des besoins, un certain nombre de documents sont recréés manuellement. Ainsi, l'agent de maintenance gèrera sur un carnet un condensé de documents du dossier qui regroupe les procédures courantes, les pannes courantes... La création manuelle de ces documents témoigne d'une insuffisance en documentation. Elle est génératrice d'erreurs, soit par une mauvaise reprise des informations des "documents pères", soit par une évolution non simultanée des "documents pères" et des "documents fils"

En résumé, les futurs outils logiciels devront permettre de réaliser des dossiers cohérents et de maintenir cette

cohérence durant la vie des documents. Pour se convaincre de l'importance de tout ceci, il suffit de savoir que RENAULT estime à deux millions le nombre de plans "vivants" de machines automatiques à la régie!!

63. Définition du besoin: le Poste de Travail de l'Automaticien

631. Généralités

L'ensemble des problèmes et le constat que nous venons d'effectuer a permis de mettre en évidence le besoin d'un Poste de Travail pour l'Automatisation ou [PTA 87A]. Il s'agit d'un poste de travail de CAO à l'usage des concepteurs d'installations automatisées mais devant également être utilisé par les services de maintenance.

Plus rigoureusement, le concept de PTA désigne un ensemble de traitements de CAO travaillant sur des structures de données communes permettant une intégration cohérente et homogène. Il s'agit en quelque sorte d'un poste de travail "logique" dont les implémentations "physiques" pourraient être très variées:

- poste de conception en bureau d'études,
- poste d'exploitation/ maintenance en atelier,
- poste de consultation de documents,
- poste de saisie de cahiers des charges...

Le PTA doit assister l'automaticien tout au long du cycle de vie de l'application en assurant la cohérence entre les différentes étapes. Il doit permettre de définir une structure de commande, de produire le logiciel de commande et de générer le dossier d'affaire complet et cohérent. En particulier, les documents d'exploitation-maintenance ne seront pas recréés mais extraits des documents de conception afin de limiter les erreurs. De cette manière le PTA doit permettre une réduction des coûts d'études, une diminution sensible des erreurs et une amélioration très importante des fonctions de maintenance et d'exploitation.

Les structures de données du PTA assurant des liens cohérents entre les divers modules logiciels, le développement du PTA doit provoquer l'apparition de traitements de CAO plus performants et supportant des méthodologies plus riches. Il est donc raisonnable d'estimer que la dynamique des travaux autour du PTA

permettra de combler les carences actuelles, non seulement des outils logiciels, mais aussi des outils méthodes. De même, on peut espérer une homogénéisation des concepts et du vocabulaire comme cela s'est produit en matière de CAO électrique où les notions de calque et de symboles de réinjection, par exemple, sont maintenant bien comprises par tous les utilisateurs.

Afin d'être facilement adopté, le PTA doit être un outil multi-langage et multi-machine cible. Il doit être ouvert, c'est à dire apte à accepter des modules développés ultérieurement.

632. Proposition d'une architecture pour un PTA

Nous avons décrit précédemment les deux types de CAO existant aujourd'hui et nous avons montré les problèmes que posait l'échange des données entre différentes CAO.

Ces problèmes d'échange d'informations seront résolus par une standardisation des données manipulées par les traitements. Il s'agira dans un premier temps de décrire sur le plan sémantique ces données et les liens entre elles pour aboutir à un modèle.

Ce modèle pourra être implémenté directement dans les structures de données de la CAO (standardisation au niveau des bases de données). C'est ce que montre le schéma de la figure 17.

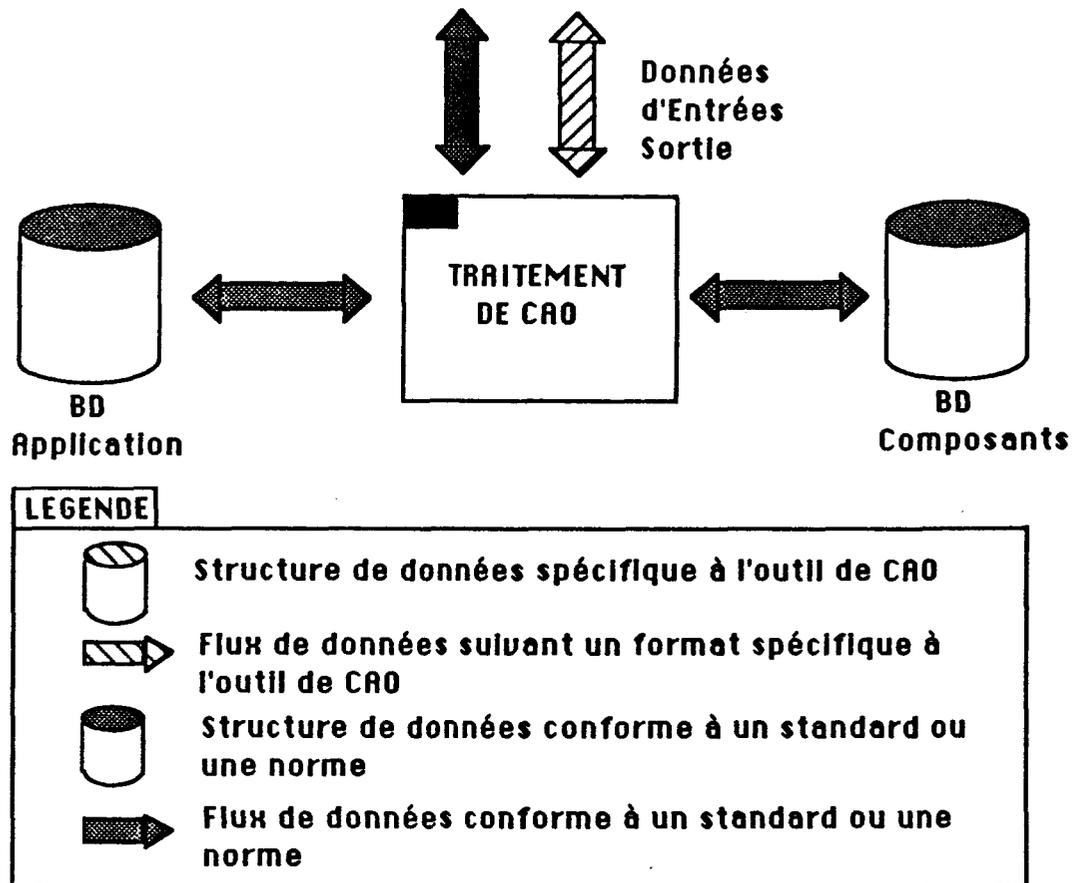


Fig 17 : CAO future possédant des bases de données conformes à un standard

Le modèle pourra également servir de modèle conceptuel à des futurs standards d'échange permettant dans une étape intermédiaire de faire dialoguer les futures CAO avec les CAO classiques actuelles (standardisation au niveau échange des données). Le schéma de la figure 18 illustre ce point de vue.

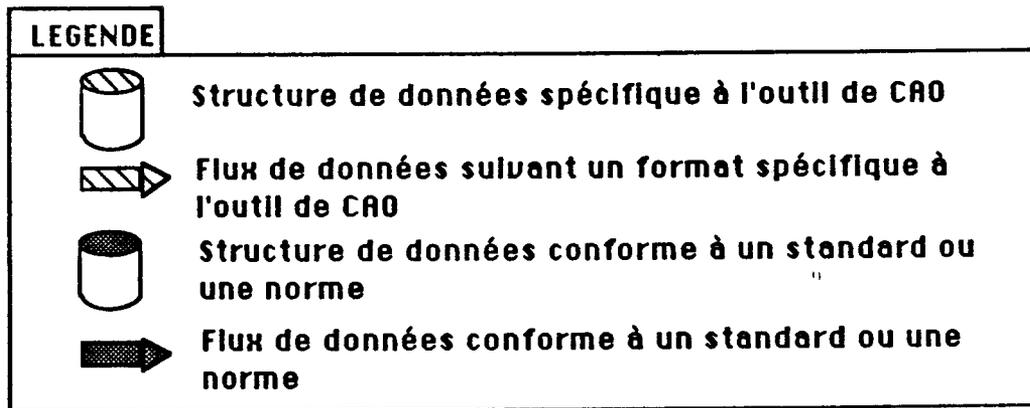
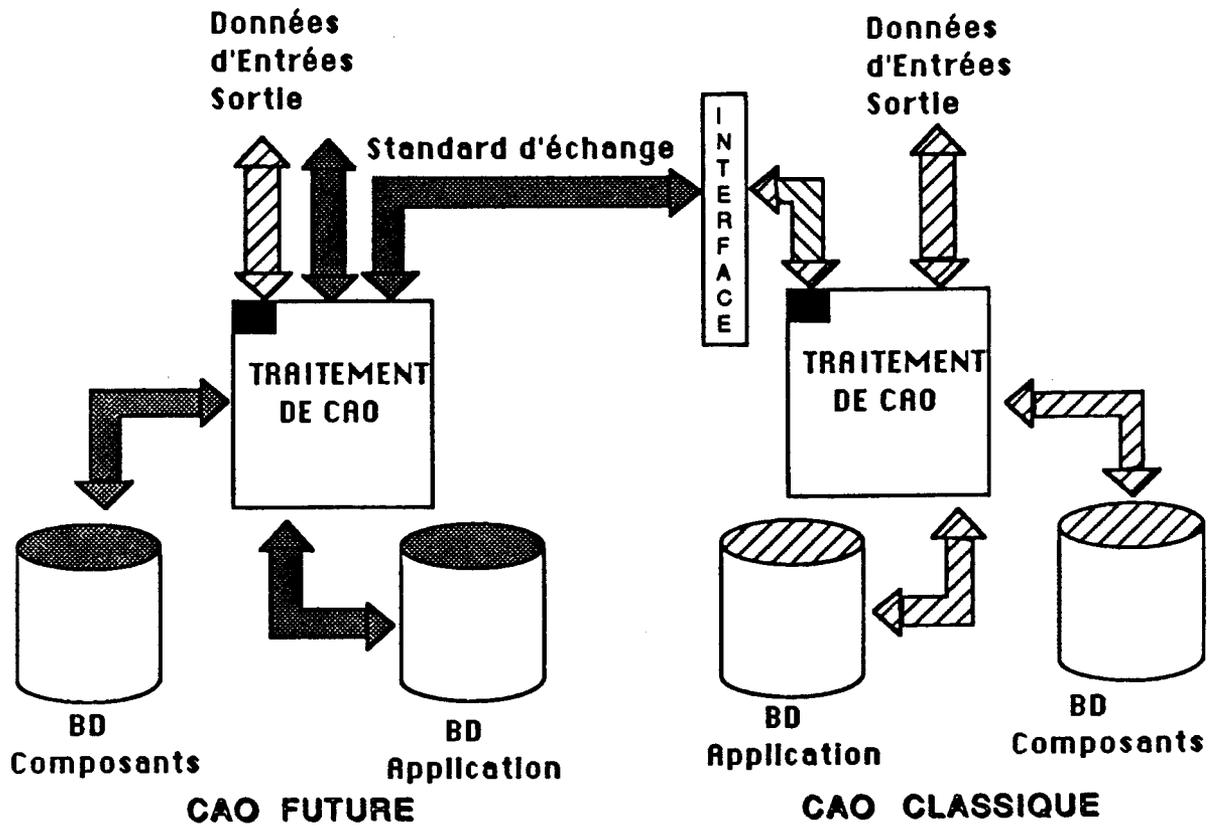


Fig 18: Echange de données entre une CAO future et une CAO classique par l'intermédiaire du standard d'échange.

Remarque :

Sur le schéma "Echange de données entre CAO classiques" (cf figure 14), nous avons montré que les échanges de données pouvaient concerner soit les données d'entrées/sorties, soit les bases de données. Dans le schéma "Echange de données entre CAO futures et CAO classiques par l'intermédiaire d'un standard d'échange" (cf figure 18) et les schémas suivants, nous ne représentons que les échanges au niveau des entrées/sorties. Le même type d'échange peut bien évidemment être considéré au niveau des bases de données.

Il apparaît assez logique, une fois les structures de données "standardisées" de faire travailler plusieurs traitements sur les mêmes bases de données. En intégrant tous les divers traitements dans une optique atelier logiciel, on peut aboutir au Poste de Travail de l'Automaticien tel que nous l'avons défini. Notons que parmi les divers traitements du PTA, il en est un fondamental qui est le traitement de la documentation de l'application. En effet, si chaque traitement génère sa propre documentation sur l'aspect qu'il prend en compte, il est essentiel que la documentation globale de l'automatisme soit traitée au niveau général du PTA.

Le schéma de la figure 19 présente une architecture générale d'un PTA. Nous avons rajouté une base de données dite "Base de Savoir-faire" [FRA 87A] qui existe d'ailleurs à l'état embryonnaire dans certaines CAO actuelles (cf section I-5). Elle contient des éléments de solutions déjà utilisés dans des applications précédentes et que les traitements pourront réutiliser.

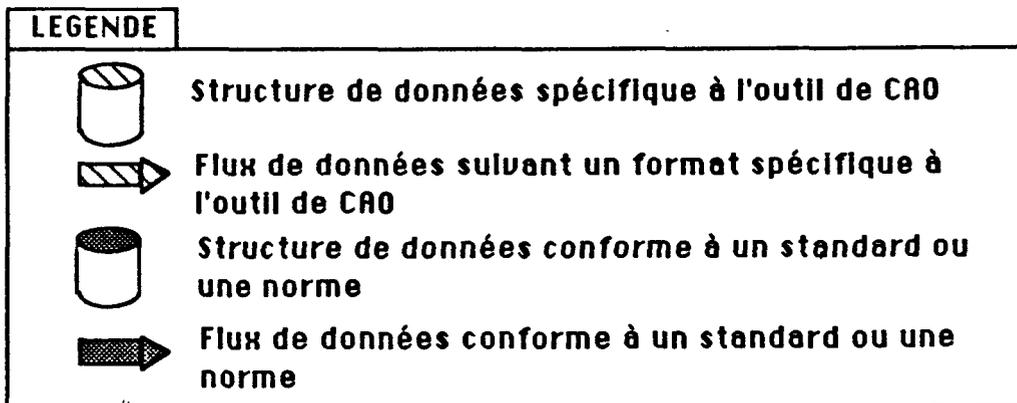
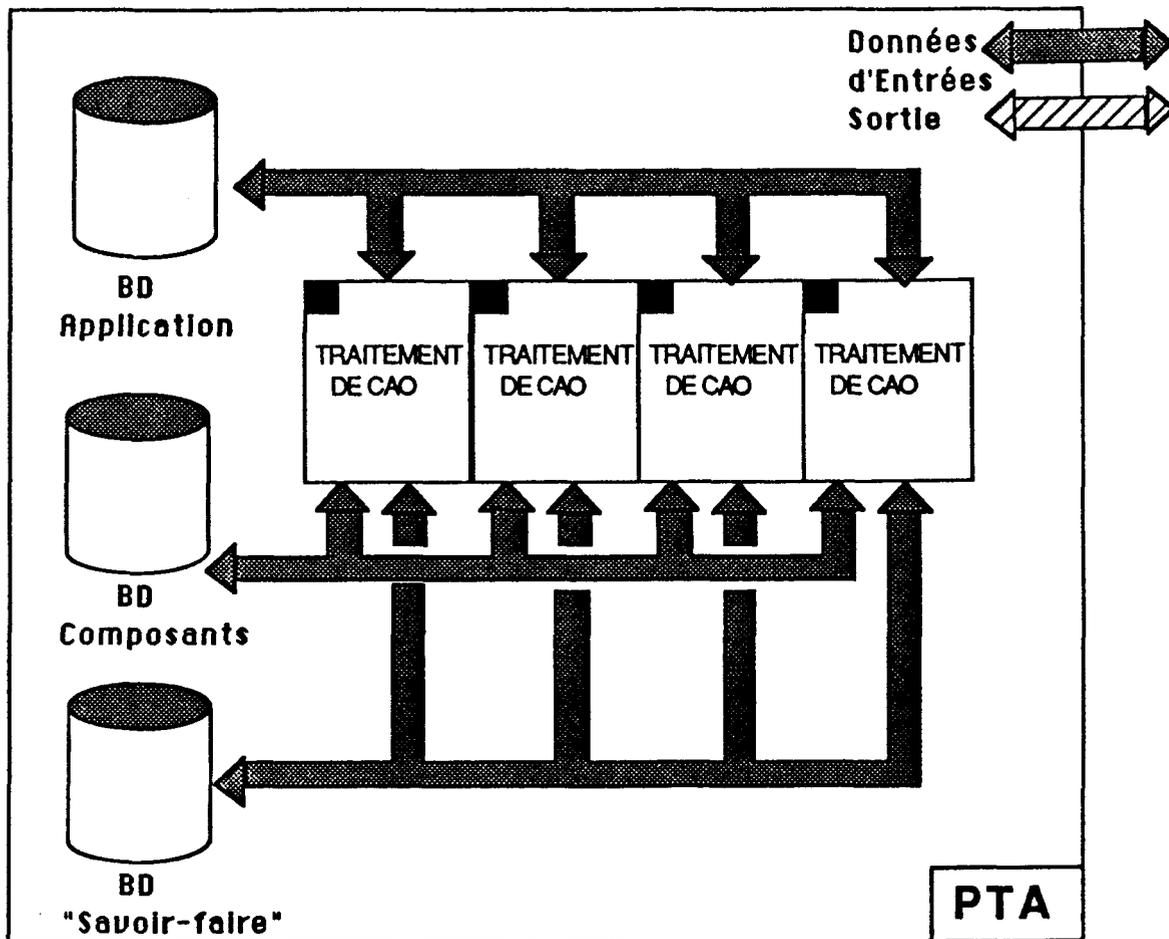


Fig 19: Architecture générale d'un Poste de Travail de l'Automaticien

PARTIE I - EXPOSE DU PROBLEME

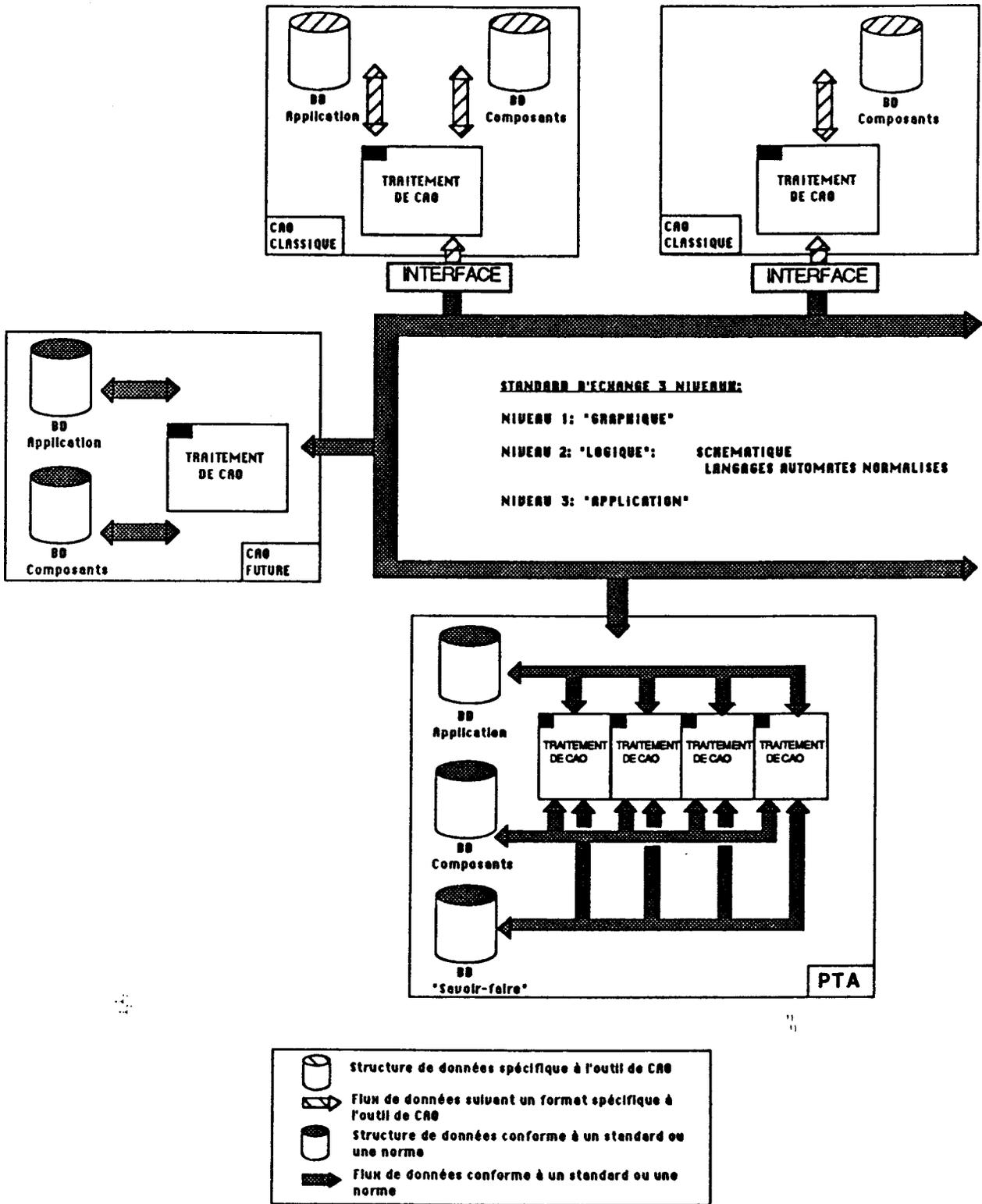


Fig 20: Schéma récapitulatif

7. ETAT DES TRAVAUX

71. Introduction

Nous avons mis en évidence les différents problèmes à résoudre pour créer une CAO performante et définir le Poste de Travail de l'Automaticien. Ainsi que nous l'avons vu, la tâche est importante et les axes possibles des différents travaux sont nombreux.

Avec l'appui des pouvoirs publics (Ministère de l'industrie et de l'aménagement du territoire, Ministère de la recherche et de la technologie) et sous l'impulsion des industriels du monde de l'automatisme, des études ont été lancées parmi lesquelles le projet BASEPTA au sein duquel nous avons pu contribuer à une standardisation des "données de l'automaticien".

72. Liste des projets

Nous allons ici très brièvement énumérer les différents projets les plus importants puis nous les situerons, sur un schéma, dans le PTA.

Projet BASEPTA

Définition d'un modèle conceptuel de la structure de données "application" d'une CAO d'automatismes.

Projet ALGA

Pré-étude de faisabilité d'un atelier logiciel pour le génie automatique.

Norme SET

Standard d'échange et de transfert des données techniques entre systèmes de CFAO (orientation mécanique).

Projet SET schématique

Extension de la norme SET aux données de schématique électrique et fluïdique.

Projet TECHNO-X

Etude d'un éditeur généralisé de schémas d'automatismes assurant la cohérence entre les schémas.

Normalisation langages automates

Projet de normalisation CEI des langages automates.

Normalisation DIN 4000

Normalisation DIN en cours sur les bases de constituants.

Projets réponses à l'appel d'offre du MIPTT (Octobre 87)

Une dizaine de projets sélectionnés développent des traitements et doivent s'appuyer sur la structure de données définie par BASEPTA.

PARTIE I - EXPOSE DU PROBLEME

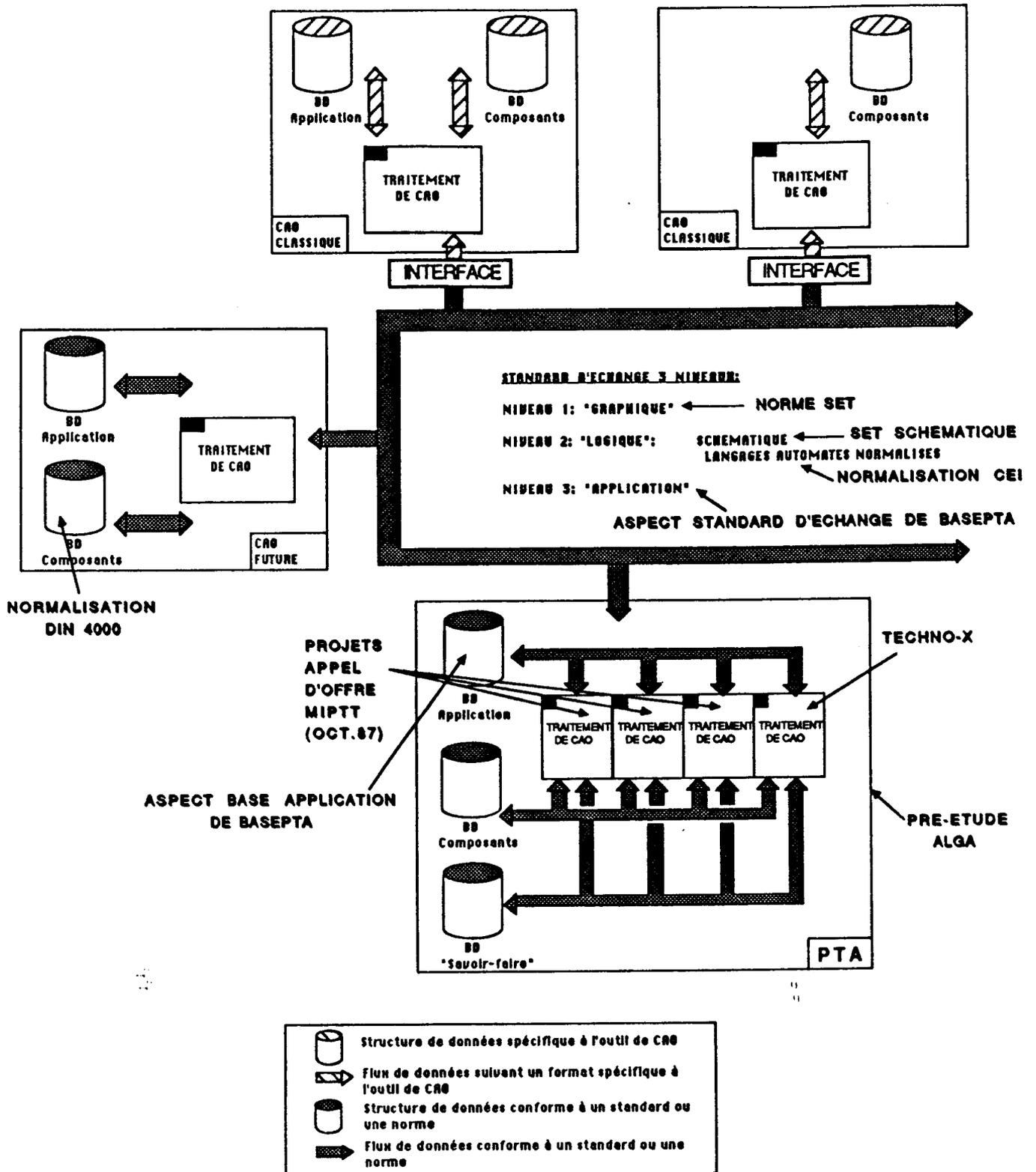


Fig 21: Les projets PTA - situation "technique"

CONCLUSION

Les outils logiciels à l'usage de l'automaticien étant ressentis comme notablement insuffisants par la communauté de l'automatisme, le développement d'un véritable PTA est une idée aujourd'hui acceptée et sur laquelle industriels, universitaires et pouvoirs publics ont montré leur volonté de s'impliquer en participant de façon multiple à divers projets.

La définition d'une structure de données renfermant les données caractéristiques de l'application et manipulées par tous les traitements au cours de l'ensemble du cycle de vie de l'automatisme constitue un élément majeur du futur Poste de Travail de l'Automaticien.

C'est un modèle de cette structure de données que nous allons présenter et illustrer dans la suite de ce mémoire.

PARTIE II

UN MODELE POUR LA STRUCTURE DE
DONNEES APPLICATION D'UNE CAO
D'AUTOMATISMES

INTRODUCTION

Nous allons présenter ici, d'une manière détaillée, un modèle décrivant une application automatisée. Ce modèle a été élaboré par l'équipe technique BASEPTA dont nous avons fait partie.

Nous commencerons tout d'abord par préciser ce que nous entendons par "modèle conceptuel de la Base Application" et nous rappellerons brièvement l'intérêt et l'objectif d'un tel modèle. Puis nous situerons l'exposé du modèle BASEPTA dans ce mémoire par rapport aux documents produits et diffusés par l'équipe du projet afin d'en souligner l'intérêt et la spécificité.

Après une description des principes et postulats retenus pour l'élaboration du modèle, nous présenterons de façon détaillée les éléments qui le constituent. Cette présentation sera précédée d'une courte introduction au formalisme choisi pour représenter le modèle de données. Nous nous efforcerons d'illustrer le plus souvent possible notre exposé par des exemples extraits de l'application type présentée en Partie I et dont le dossier est fourni en Annexe.

1. DEFINIR UN MODELE CONCEPTUEL POUR UNE STRUCTURE DE DONNEES PARTICULIERE: LA BASE DE DONNEES APPLICATION.

11. La Base Application: la solution au problème de la cohérence.

L'exposé que nous avons conduit en Partie I a permis de mieux définir les caractéristiques d'une CAO d'automatismes ou PTA. Comme nous l'avons vu, la conception d'une telle CAO est une tâche monumentale pour les prochaines années. Cependant, parmi les multiples travaux à réaliser, nous avons pu dégager un objectif précis et une solution pour le satisfaire:

Notre objectif est d'assurer la cohérence de l'application au travers du cycle de vie ainsi que la cohérence du dossier.

La solution envisagée est la réalisation d'une structure de données partagée (c'est à dire créée, modifiée, enrichie, exploitée...) par les différents traitements au cours du cycle de vie de l'automatisme. Cette structure de données constitue un des éléments fondamentaux du Poste de Travail de l'Automaticien. Elle est un préalable à la définition de traitements performants et communicants.

12. La définition d'un modèle conceptuel.

L'objectif final est de réaliser une structure de données "application" informatisée, c'est à dire implémentée sur une machine informatique afin d'être utilisée par des traitements logiciels.

La construction de cette Base Application informatisée se fait selon la méthode classique de construction d'une structure de données d'un système d'informations.

La première étape consiste à analyser la sémantique du réel, c'est à dire ses propriétés. Cette traduction est fonction de la manière dont est perçu le réel, en général sous forme d'objets, de relations entre objets et de propriétés de ces objets ou de ces relations. Cette perception est bien évidemment subjective. C'est ainsi qu'une propriété d'un objet peut être perçue comme une propriété intrinsèque ou comme une relation de cet objet

avec un autre objet. Dans le cas de la Base Application, le réel est l'application automatisée.

La traduction du réel se fait sous une forme plus rigoureuse qu'un simple ensemble de listes d'objets, de relations et de propriétés. Des classements, des regroupements ou à l'inverse des éclatements sont effectués afin d'obtenir des objets plus typés, plus précis. Le soucis principal est de parvenir à une description de la sémantique du réel précise, non ambiguë et non redondante. Cette description porte le nom de modèle conceptuel.

Il devient alors nécessaire d'introduire dans le modèle l'usage que l'on va faire des données, en particulier en fixant les accès et les paramètres d'accès aux données. Le modèle résultant est appelé modèle logique. Remarquons que certains outils logiciels (SGBD relationnels par exemple) n'imposent pas a priori de connaître et donc de décrire les différents accès aux données. La construction du modèle logique est donc, dans ce cas, inutile.

La dernière étape consiste en l'implémentation proprement dite de la structure de données. Il s'agit de décrire physiquement cette structure en prenant en compte des contraintes matérielles (choix de la machine d'accueil) et logicielles (nature et caractéristiques de l'outil logiciel SGBD qui gèrera la structure de données). Le modèle résultant est dit modèle physique.

Le modèle que nous présentons dans ce mémoire est un modèle de niveau conceptuel. Il ne préjuge donc en aucune manière du nombre et de la nature des accès aux données ni des choix de l'implémentation physique de la Base Application. Il s'agit simplement de la description d'une application automatisée, description précise et consistante des objets concrets (par exemple les constituants matériels) ou abstraits (par exemple les fonctions de commande) qui composent l'application, de leurs associations et de leurs propriétés. Cette description est réalisée grâce à un outil de modélisation que nous détaillerons plus loin.

Pour une meilleure compréhension, le modèle est décrit graphiquement. La forme graphique du modèle conceptuel est le schéma conceptuel.

13. Le travail du projet BASEPTA.

Projet coopératif réunissant les différents partenaires du monde de l'automatisme (utilisateurs, concepteurs, fabricants de CAO, constructeurs de constituants) et le Ministère de l'Industrie, BASEPTA avait comme objectif la définition du modèle conceptuel de la Base Application dit "modèle BASEPTA".

Ce modèle est le résultat du travail d'une équipe technique de cinq personnes issues des entreprises membres du projet. Cette équipe, véritable structure opérationnelle permanente, a fonctionné en étroite collaboration avec les différents correspondants techniques des sociétés. Le modèle résultant est donc le fruit d'un grand nombre d'apports techniques divers. Il traduit un consensus sur la perception de la Base Application et de son contenu. Il est du domaine public. [BAS 88A] [BAS 88B] [BAS 88C]

14. Objectifs de la présentation du modèle conceptuel dans ce mémoire.

Les objets constitutifs du modèle BASEPTA résultent d'un certain nombre de choix de l'équipe technique, choix qui ne sont plus apparents dans le modèle résultant. De plus, bien que l'équipe BASEPTA ait fait des efforts pédagogiques importants pour expliciter la sémantique de chaque objet, celle-ci est parfois en partie masquée. Il peut en résulter une mauvaise utilisation des objets du modèle, due à une "interprétation" erronée. En tant que membre de l'équipe technique, nous avons une meilleure compréhension du modèle et des motivations qui ont conduit à privilégier tel aspect ou à occulter tel autre. L'un des objectifs de ce mémoire est donc de lever les ambiguïtés qui demeurent dans les documents BASEPTA.

D'autre part, comme nous l'avons dit précédemment, le modèle BASEPTA résulte d'un consensus qui, à titre personnel, ne nous apparaît pas toujours satisfaisant. Dans certaines parties du modèle, il nous semble qu'il subsiste des insuffisances. Certains objets et certaines relations sont inutiles ou dangereux. Nous nous efforcerons de souligner ces faiblesses et de dégager les points forts du modèle BASEPTA.

Enfin compte-tenu des principes de base retenus pour

l'élaboration du modèle BASEPTA, principes que nous détaillerons plus loin, les objets sont relativement généraux. Toute implémentation de la structure de données devra être précédée d'un typage des objets, d'un affinement, en bref d'une personnalisation du modèle, par l'introduction de contraintes méthodologiques ou autres (par exemple liées au type de matériel que l'on utilise). Dans ce mémoire, nous donnerons des indications sur la manière de réaliser ce paramétrage pour donner de la consistance ou de "l'épaisseur" au modèle.

15. Contenu de la présentation du modèle conceptuel dans ce mémoire.

Nous allons donc présenter dans cette seconde partie notre perception du modèle BASEPTA (Version 3.0) [BAS 88C]. L'ensemble des entités du modèle sera exposé et illustré par des exemples choisis de préférence dans le dossier "Chaine d'emballage de rouleaux de papier". Les documents BASEPTA donnent des définitions précises des entités du modèle et de leurs attributs. Nous nous attacherons donc plutôt à souligner les idées fortes du modèle.

L'ordre dans lequel les objets du modèle (Entités ou Associations) seront introduits est tout à fait arbitraire. Il ne s'agit que d'un ordre de présentation choisi pour des raisons de clarté et de complexité croissante dans l'exposé du modèle. Cet ordre n'est en aucune façon significatif de la manière dont les traitements de CAO exploiteront la Base de données Application. Le modèle conceptuel n'a pas à prendre en compte l'utilisation des données.

2. LES PRINCIPES DU MODELE DE DONNEES

21. La cible du modèle : l'application en cours de traitement.

Le modèle que nous présentons permet de ne stocker que les données d'une seule application: l'application en cours de traitement. Il ne permet pas de décrire simultanément les données de plusieurs applications. Autrement dit: A plusieurs applications automatisées correspondent plusieurs Bases Application.

De plus le modèle décrit l'application dans son état actuel. Il ne permet pas a priori de conserver la trace de conception, c'est à dire par exemple l'ordre dans lequel les données ont été introduites dans la base ou l'ordre dans lequel les relations entre ces données ont été établies. Il se contente de représenter le résultat. La conservation des différentes versions successives de la structure de données est le domaine d'une autre base de données de type "Archivage". Nous verrons, lors de la présentation du modèle, que cette affirmation doit être nuancée et qu'il existe malgré tout dans le modèle des entités qui permettent, par exemple, de conserver les différentes versions d'un logiciel. Mais ici, dans le cadre de notre exposé des principes du modèle, il est essentiel de préciser qu'il s'agit là d'une dérogation à ces principes, d'une frontière du modèle.

D'autre part, les données décrites dans le modèle sont les données "instanciées" dans l'application et en aucune façon les données de bibliothèque. La description d'une structure de données pour les composants de bibliothèque (logiciels, matériels ou logiciels/matériels) est un tout autre problème. Nous verrons dans la présentation détaillée du modèle qu'il est possible de stocker par exemple les caractéristiques techniques des constituants matériels. Certes, ces caractéristiques techniques sont celles du constituant du catalogue, mais elles seront stockées dans la Base Application si elles sont jugées pertinentes du point de vue de l'application. Ainsi l'intensité nominale d'un contacteur le sera certainement, sa masse éventuellement, sa couleur probablement pas.

En résumé, le modèle de la structure de données application est limité. Il ne modélise pas les données des Bases Constituants, "Savoir-faire" (cf Partie I) ou Archivage. Mais, comme nous le verrons, il comporte des

entités permettant de faire le lien assez facilement avec ces structures de données.

22. Modéliser l'application et non sa représentation.

Les données du modèle sont des données décrivant l'application et non sa représentation par un dossier. Cela signifie que l'on ne trouvera dans le modèle:

- aucune entité graphique de type point, segment, cercle...
- aucune entité schématique de type symbole, point de connexion, noeud de connexion, liaison schématique...
- aucune entité de dossier de type sous-dossier, folio, cartouche...

Nous avons expliqué en Partie I que les documents et, plus généralement, le dossier pouvaient être considérés comme un ensemble de vues sur l'application. Notre modèle ne décrit pas ces vues mais la réalité de l'application qu'elles représentent.

Les liens entre les entités de la documentation et les entités du modèle application restent à définir. Mais, à titre d'exemple, nous donnerons dans notre présentation des indications sur la manière de relier les symboles des schémas de principe ou les vignettes des plans d'implantation en armoire aux entités convenables du modèle de la Base Application. Ces liens constituent un point important car la plupart des entités de l'application ne sont manipulées qu'au travers des schémas. Ainsi, on peut considérer qu'une manière courante d'introduire des constituants matériels dans la Base Application est de dessiner les symboles qui les représentent sur le schéma de principe.

Ce type de considération a conduit, lors des travaux sur le projet TECHNO-X [TEC 88A], projet de réflexion sur l'aspect documentation du PTA, à l'introduction du terme "couronne schématique" pour désigner l'interface entre les traitements et les données application. Ce terme implique que toute manipulation sur les objets de l'application se fait au travers des vues qu'en sont les schémas. Le terme schéma, baptisé aussi "technogramme" [TEC 88A], doit être pris ici au sens large pour désigner toute représentation graphique conventionnelle et symbolique sur un écran ou du

papier d'une réalité technologique. Cette idée nous semble un peu excessive, car elle contraint à donner une définition trop large et peu porteuse du schéma. Cependant, nous reconnaissons l'importance de la documentation et c'est pourquoi nous détaillerons chaque fois que cela sera possible les liens entre les données du modèle application et les données de la documentation.

23. Un modèle "methodology free".

Le modèle présenté se veut totalement indépendant de tout traitement. L'organisation des données est faite sans se préoccuper de la manière dont les traitements exploiteront ces données. Or ce sont les traitements qui supportent les méthodologies. L'indépendance vis à vis des traitements conduit donc à un modèle indépendant des méthodologies (modèle dit "methodology free") mais devant supporter n'importe quelle méthode. L'ambition affichée ici est que le modèle soit utilisable par n'importe quel traitement actuel ou futur, c'est à dire non encore défini. En introduisant dans la structure de données un certain nombre de liens images de la réalité mais non méthodologiques, le modèle facilitera le développement de traitements riches et performants.

Ce principe a conduit, comme nous le verrons plus loin, à des objets relativement généraux pour stocker les informations traduisant la structuration de l'application suivant différents critères. Sans considération méthodologique, il est impossible de typer ces entités plus précisément.

En comparaison, la description des constituants logiciels et matériels et de leur agencement donne lieu à des entités plus nombreuses et plus typées. Un affinement poussé du modèle, lorsque le principe de refus des méthodologies l'autorise, est une autre condition favorable pour le développement de traitements performants.

24. Les futures implémentations du modèle de données.

Les principes de base que nous venons de décrire ont conduit à un modèle très riche couvrant l'ensemble de l'application, mais que l'on pourrait considérer comme inutilisable tel quel. Il faudra en effet le personnaliser, le paramétrer par la prise en compte de

PARTIE II - MODELE DE STRUCTURE DE DONNEES APPLICATION

considérations méthodologiques afin de lui donner de la consistance. C'est cet aspect de configuration du modèle que devront traiter les premières implémentations. Il est raisonnable de penser que, compte-tenu de l'étendue du modèle, ces implémentations ne prendront en compte que des parties de ce modèle. Ce modèle se veut le modèle de la Base Application d'un PTA complet, PTA bien loin d'être réalisé aujourd'hui.

3. L'OUTIL DE MODELISATION

31. Introduction

Le modèle conceptuel de structure de données application que nous allons présenter est décrit grâce à un outil de modélisation basé sur le modèle Entité-Association [CHE 80A] [SAU 86A]. Ce modèle, très utilisé en informatique de gestion, est cependant l'objet de nombreuses critiques en ce qui concerne son utilisation en CAO. La sémantique des associations est assez pauvre, les contraintes que peut traduire le modèle sont faibles (elles se situent essentiellement au niveau des cardinalités).

Afin de prendre en compte, de la manière la plus fine possible, la richesse de la sémantique de l'application, des enrichissements du modèle Entité-Association ont été proposés. Il s'agit essentiellement de l'ajout d'une relation plus forte que la simple association, l'agrégation, et de la prise en compte des notions de classes et de sous-classes, notions usuelles dans les langages à objets [BAI 87A].

Nous allons dans cette section II-3 présenter l'outil de modélisation "Entité-Association enrichi" utilisé pour la description du modèle conceptuel de la structure de données application. Pour illustrer cette présentation, nous avons choisi des données de la schématique qui, comme nous l'avons vu précédemment (cf II-2.22), ne sont pas des objets de la Base Application.

32. Les concepts de base de l'outil de modélisation

321. L'entité

Comme nous l'avons vu plus haut, la perception du réel peut classiquement être exprimée sous forme de listes:

- liste d'objets,
- liste de relations entre ces objets,
- liste de propriétés de ces objets et de ces relations.

Une entité représente une classe d'objets concrets ou

abstrait du réel. Elle est déterminée par un certain nombre de caractéristiques appelées attributs dont le rôle est de rendre compte des propriétés des objets de la classe.

Les différents objets de la classe sont appelés instances de l'entité. Les attributs de l'entité prendront des valeurs pour les différentes instances.

Parmi les attributs de l'entité, certains permettent de différencier deux instances quelconques de l'entité. Ces attributs particuliers sont appelés identifiant de l'entité. L'identifiant d'une entité n'est pas unique. Ainsi, tout ensemble d'attributs contenant un sous-ensemble d'attributs qui est identifiant, est identifiant. Dans le modèle de la Base Application, un choix a été fait pour chaque entité. Nous parlerons par la suite de l'identifiant de l'entité pour désigner l'identifiant résultant de notre choix arbitraire. Dans le cas général, il ne sera formé que d'un seul attribut.

L'entité se représente sur le schéma conceptuel par un rectangle dans lequel figurent un mnémonique et la désignation de l'entité.

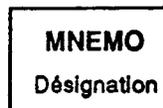


Fig 22

Exemple 1: Les folios d'un dossier donné appartiennent à une entité dont les attributs sont:

- numéro (qui est identifiant),
- état (en cours, terminé, ...)
- date de création,
- date de dernière modification...



Fig 23

exemples d'instances:

012	015
en cours	terminé
01.10.88	01.10.88
06.11.88	08.11.88

Exemple 2: les symboles d'un folio donné appartiennent à une entité symbole dont les attributs sont:

- l'identifiant,
- la nature (maître, esclave, passif...)
- la coordonnée en X du point d'implantation du symbole dans le folio,
- la coordonnée en Y du point d'implantation du symbole dans le folio,
- le nom du symbole de bibliothèque utilisé...

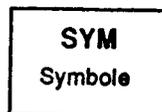


Fig 24

exemples d'instances:

S12	S23
maitre	esclave
2.5	6.0
3.0	8.5
BOB (bobine)	CF (contact à fermeture)

322. L'Association

Pour traduire les relations entre les objets du réel, nous utilisons la notion d'association entre deux entités. Une association entre deux entités 1 et 2 traduit que certaines instances de l'entité 1 peuvent être en relation avec certaines instances de l'entité 2. Elle constitue en quelque sorte une classe de relations de la même façon qu'une entité est une classe d'objets.

Comme une entité, une association peut posséder des attributs traduisant les propriétés de la relation dans le réel. Cependant, pour une association, la notion d'identifiant n'est pas nécessaire. En effet, toute instance de l'association est entièrement déterminée par les deux instances des entités qu'elle relie.

Une association entre deux entités se représente sur le schéma conceptuel par un polygone. Dans ce polygone figurent deux libellés (parfois appelés "rôles") correspondant aux deux sens de lecture de l'association (cf schéma de la figure 25).

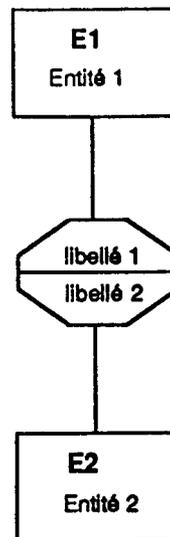


Fig 25

Exemple: Pour traduire le fait qu'un symbole se trouve sur un folio, nous pouvons introduire une association entre les deux entités SYM et FOL précédemment définies (cf figure 26).



Fig 26

Sur cet exemple, les deux libellés de l'association montrent clairement les deux sens de lecture de l'association:

- un symbole "se trouve dans" un folio,
- un folio "contient" des symboles.

Exemples d'instances de l'association:

Le symbole S12 se trouve dans le folio 015.
Le symbole S23 se trouve dans le folio 015.

Soit encore:

Le folio 015 contient le symbole S12.
Le folio 015 contient le symbole S23.

323. Cardinalités d'une association

Les cardinalités d'une association entre deux entités 1 et 2 permettent de définir précisément des contraintes sur le nombre d'instances de l'association auxquelles peuvent participer les instances de chaque entité.

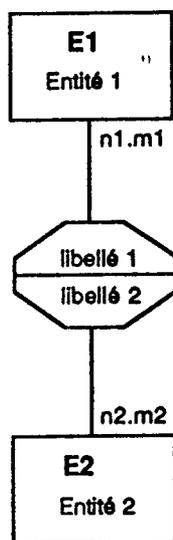


Fig 27

La cardinalité n_1, m_1 (respectivement n_2, m_2) règle les modalités de "participation" de l'entité 1 (respectivement 2) à l'association. Elle signifie que chaque instance de l'entité 1 (respectivement 2) est reliée à au moins n_1 (respectivement n_2) et au plus m_1 (respectivement m_2) instances de l'entité 2 (respectivement 1).

Lorsque les cardinalités sont connues précisément, elles figurent explicitement dans le schéma conceptuel. Lorsque le nombre d'instances maximal de l'association n'est pas défini, on indiquera conventionnellement N pour la valeur correspondant au maximum dans la cardinalité.

Exemple: Les cardinalités de l'association de la figure 28 ont la signification suivante:

- 1.1 - Tout symbole se trouve dans un folio
 - un symbole ne peut se trouver que dans un seul folio (rappelons que l'entité SYM est la classe des symboles instanciés dans un folio et non celle des symboles de bibliothèque).

- 0.N - un folio peut ne contenir aucun symbole (cas de folio de dessin et non de schéma)
 - un folio peut contenir un nombre quelconque de symboles.

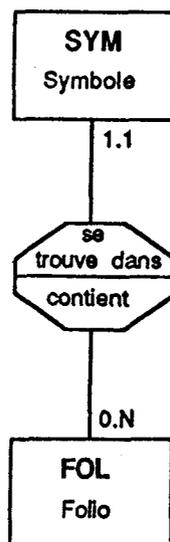


Fig 28

324. Attributs d'une association.

Comme nous l'avons vu précédemment, une association peut porter des attributs. Lorsque ce sera le cas, un astérisque l'indiquera au niveau de l'association sur le schéma.

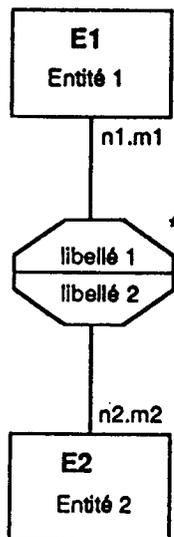


Fig 29

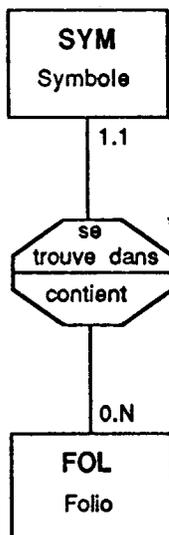
Exemple:

Fig 30

Il est possible, plutôt que de faire porter les coordonnées d'un symbole dans un folio par les attributs de l'entité SYM, de les considérer comme des attributs de l'association entre symboles et folios.

33. Un premier enrichissement du modèle Entité Association: la relation d'agrégation.

La relation d'agrégation est une association entre deux entités 1 et 2. L'entité 1 est l'agrégat, l'entité 2 est l'agrégé. L'agrégation traduit le fait que toute instance de l'entité 2 est une partie d'une instance de l'entité 1. Il en résulte que toute instance de l'entité 2 doit obligatoirement être reliée à une instance de l'entité 1 et ne peut donc exister isolément.

Il s'en déduit également que toute instance de l'entité 2 est identifiée par rapport à l'instance de l'entité 1 à laquelle elle est reliée. Ceci signifie que l'identifiant d'une instance de l'entité agrégée 2 dans la structure de données contient toujours l'identifiant de l'instance de l'entité 1 qui l'agrège.

L'agrégation est une association dont les cardinalités sont toujours 0.N pour l'agrégat et 1.N pour l'agrégé. Mais elle n'est pas équivalente à une simple association possédant ces cardinalités.

L'agrégation se représente comme une association mais une flèche et un petit rectangle indiquent l'entité agrégée. Les cardinalités étant toujours les mêmes, on indique simplement "AG" à côté de l'association sur le schéma conceptuel.

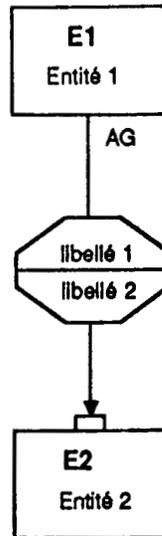


Fig 31

Exemple: Pour traduire le fait que tout symbole d'un schéma peut posséder des bornes, on peut utiliser deux entités SYM et B liées par une relation d'agrégation (cf figure 32).

L'entité SYM a été précédemment détaillée. L'entité B est la classe des bornes. Ses attributs pourraient être:

- un numéro d'ordre qui, composé avec l'identifiant du symbole auquel elle appartient, permet d'identifier dans la base toute instance de B,

- un type, utile par exemple à des traitements de schématique pour autoriser ou interdire la connexion de symboles.

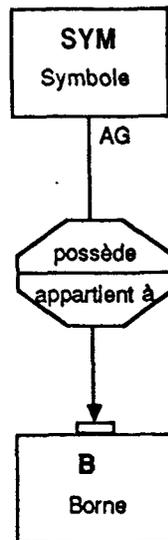


Fig 32

Exemples d'instances:

Le symbole S12 possède une borne de numéro d'ordre 11. Cette borne est identifiée dans la base comme "la borne de numéro d'ordre 11 du symbole S12".

34. Un deuxième enrichissement du modèle Entité Association : la relation de généralisation.

La relation de généralisation constitue un enrichissement du modèle Entité Association en direction des langages à objets. Ainsi que nous l'avons vu, les entités sont des classes d'objets. La relation de généralisation va permettre d'introduire la notion de sous-classe.

On dit que l'entité 1 est une généralisation de l'entité 2 lorsque l'entité 2 constitue une sous classe de l'entité 1. Tous les attributs de l'entité 1 sont des attributs de l'entité 2. Toutes les associations entre des entités du modèle et l'entité 1 existent également avec l'entité 2. Mais l'entité 2 peut posséder des attributs et des associations propres (spécificités de la sous-classe).

Graphiquement, la relation de généralisation se représente par une simple flèche orientée de la sous-classe vers la classe. Pour des commodités de lecture du schéma, la classe de généralisation sera grisée (cf figure 33).

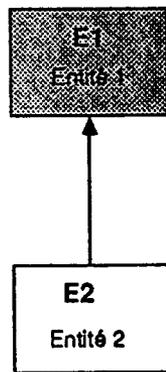


Fig 33

L'entité 1 est une généralisation de l'entité 2.

Exemple: L'entité SYM classe des symboles possède la sous-classe des symboles maîtres SYM/MAI (cf figure 34).

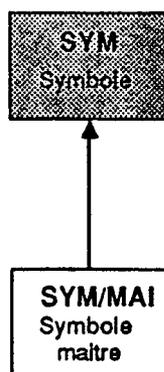


Fig 34

Plus généralement, on peut distinguer pour l'entité SYM les différentes sous-classes de la figure 35.

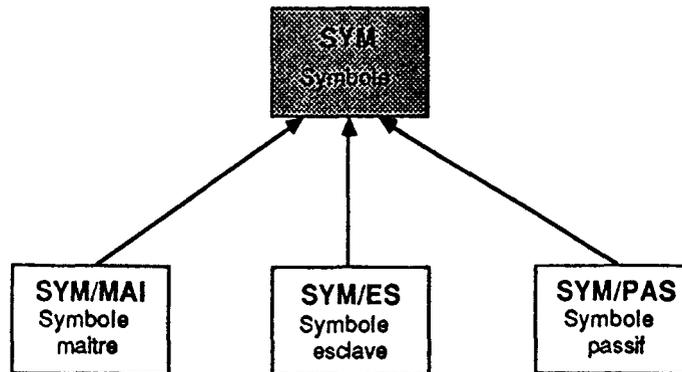


Fig 35

Il n'est alors plus nécessaire de faire porter à l'entité SYM un attribut nature (maître, esclave, passif).

Il faut remarquer que les différentes sous-classes constituent une partition, au sens mathématique du terme, de la classe générale.

35. Contraintes sur les associations

351. Contrainte d'exclusion

Une entité 1 peut être reliée à deux autres entités 2 et 3 par deux associations. La contrainte d'exclusion portant sur ces deux associations traduit le fait qu'une instance de l'entité 1 ne peut être simultanément liée à une instance de l'entité 2 et à une instance de l'entité 3. Elle se représente graphiquement par une croix cerclée reliée aux deux associations concernées.

Exemple: Sur le schéma de la figure 36, la contrainte d'exclusion entre les deux associations faisant intervenir l'entité TEX signifie qu'un texte ne peut être simultanément rattaché à un symbole et une liaison.

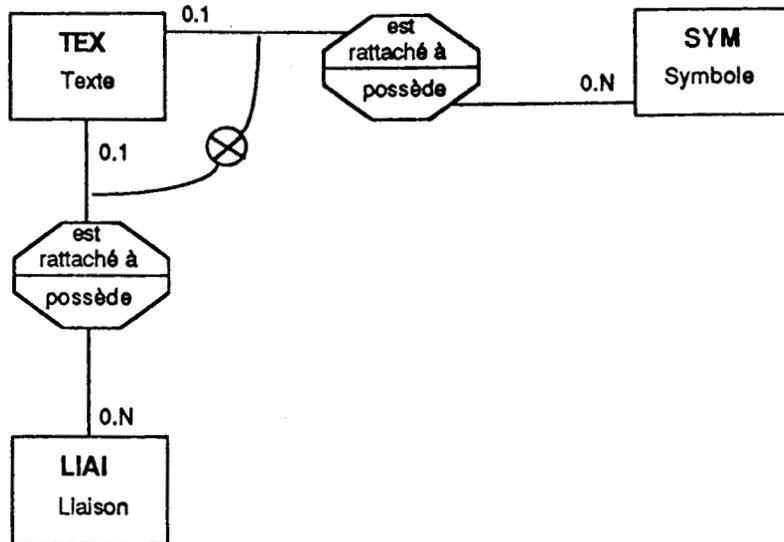


Fig 36

La contrainte d'exclusion peut toujours être remplacée par une spécialisation de l'entité participant aux associations portant cette contrainte. Les deux sous-classes que l'on peut faire apparaître, correspondent aux entités associées à chacune des associations.

Ainsi, dans l'exemple précédent l'entité TEX peut être spécialisée en deux sous-classes:

- la sous-classe TEX/LIAI des textes associés aux liaisons,
- la sous-classe TEX/SYM des textes associés aux symboles.

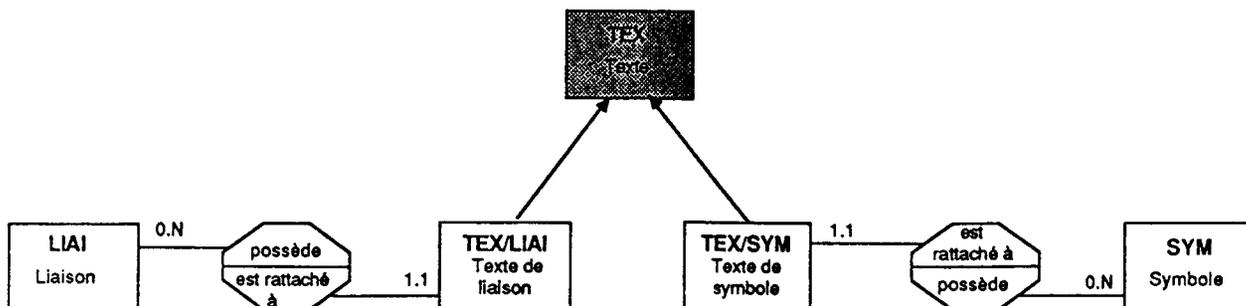


Fig 37

Comme cela apparaît sur l'exemple, une telle spécialisation en sous-classes permet de définir avec plus de précision les cardinalités. Ainsi le schéma de la figure 37 montre que tout texte est associé soit à un symbole, soit à une liaison (cardinalités 1.1 et 1.1), ce que ne traduisait pas le schéma de la figure 36 (cardinalités 0.1 et 0.1).

Cependant, les sous-classes résultant de la spécialisation n'ont parfois pas une signification simple par rapport à la perception du réel qui conduit à l'élaboration du modèle. C'est pourquoi on conserve parfois la contrainte d'exclusion dans sa forme primitive dans le but de ne pas introduire dans le modèle d'entités dont la sémantique n'apparaît pas nettement.

352. Autres contraintes.

Il peut arriver que certaines contraintes entre associations ne puissent pas être traduites par l'exclusion, les cardinalités ou l'agrégation. Elles seront alors indiquées et expliquées dans le modèle.

36. Conclusion

Comme la plupart des concepteurs de systèmes d'information, nous n'avons pu échapper à la redéfinition d'un outil de modélisation pour le schéma conceptuel. En effet, aucun outil disponible ne répondait efficacement à notre double objectif de lisibilité et de richesse dans l'expression de la sémantique des données.

Notre modèle Entité-Association enrichi présente l'avantage d'être facile à appréhender pour les familiers du modèle de CHEN. Il permet d'autre part de traduire finement beaucoup de contraintes.

Son principal inconvénient est sa spécificité. Les algorithmes de validation formelle pouvant détecter les incohérences du modèle conceptuel ne sont donc pas actuellement explicités. Dans l'état présent, le moyen le plus simple pour effectuer une validation formelle consisterait à transcrire le modèle conceptuel de la base application, avec les risques "d'appauvrissement" que cela comporte, dans un formalisme plus universel pour lequel les algorithmes sont décrits et supportés par des logiciels. Des travaux en ce sens sont en cours chez certains partenaires de BASEPTA. Ces travaux utilisent notamment le modèle NIAM de NIJSSSEN [ISO 82A] [NIJ 81A].

4. LES GRANDES FAMILLES DE DONNEES DU MODELE

41. Introduction

Le nombre important d'objets du modèle rend son exposé et son appréhension difficile. Il nous a donc semblé important de proposer une classification des données en grandes familles. Cette typologie n'est pas apparente au niveau du modèle proprement dit. Elle n'a pas de sens absolu mais constitue simplement un point de repère pour l'exposé.

42. La Fourniture

Toute application automatisée donne lieu à une fourniture, c'est à dire à un ensemble de matériels (constituants d'automatismes) et de logiciels (programmes de commande, de supervision...). La Base Application doit donc comporter un ensemble de données correspondant à cette fourniture. Il s'agit de données purement descriptives, traduisant l'état de la fourniture et non son comportement. L'approche est ici statique et non dynamique.

C'est ainsi qu'au niveau matériel nous décrirons les différents constituants physiques de l'application, la manière dont ils sont interconnectés, leur localisation physique (par exemple dans les armoires ou sur les pupitres), la façon dont ils sont approvisionnés et leurs caractéristiques techniques jugées "pertinentes du point de vue de l'application". Ce dernier point implique que la Base Application ne contiendra pas nécessairement l'intégralité des caractéristiques techniques du constituant catalogue réalisant l'approvisionnement.

Les données de modélisation du comportement de ces constituants ou de leur agencement ne sont pas stockées dans l'aspect "fourniture matérielle" de la Base Application.

Au niveau logiciel, l'aspect fourniture de la Base Application contient la description des différents programmes et des configurations logicielles des constituants programmables les supportant. Ici encore nous adoptons une démarche purement descriptive. C'est pourquoi l'aspect "fourniture logicielle" de la Base Application ne

prend pas en compte, par exemple, le contenu des organigrammes ou des arbres programmatiques des différents logiciels.

43. Les données de structuration

Si la nécessité de la partie fourniture de la Base s'impose d'elle même, il est clair que cette partie est insuffisante. Il existe d'autres types de données qui sont, de manière indiscutable, caractéristiques de l'application automatisée en cours de traitement. Ces types de données sont nécessaires pour que la Base Application puisse s'intégrer dans le cycle de vie.

C'est pourquoi le modèle de structure de données comporte un aspect "Données de structuration" de l'application. Dans cette partie du modèle nous stockerons par exemple la description des objectifs à satisfaire et des contraintes à respecter par le système, l'architecture fonctionnelle résultant des divers choix de conception, différents modèles de comportement de la commande, du logiciel, de l'association de constituants... Notons que cette liste est indicative et bien évidemment non exhaustive. Nous exposerons plus loin de manière plus détaillée les différentes données de cette partie du modèle. Nous pouvons cependant préciser ici qu'il s'agit de données abstraites représentant des concepts et non des objets concrets induisant, pour la plupart d'entre eux, diverses structurations de l'application. C'est pourquoi cet aspect du modèle porte le nom de "Données de structuration".

44. Les Données Annexes

Les deux aspects décrits précédemment, Fourniture matérielle et logicielle et Données de structuration, suffisent théoriquement à exprimer l'ensemble des diverses informations caractérisant le système automatisé. Mais il peut arriver que le concepteur d'un automatisme industriel ne sache pas "éclater" ces données, c'est à dire ne soit pas à même de les typer pour pouvoir les reporter dans les différentes classes ou entités du modèle. Ceci peut résulter d'une insuffisance dans l'identification et la reconnaissance de ces données ou simplement d'un choix délibéré. Le concepteur ou, plus précisément l'utilisateur sait seulement qu'il doit rattacher un ensemble de données à un objet de la base. Ainsi, il ne décrira pas nécessairement dans le modèle les données du compte-rendu de la réunion qui traite du choix de tel type de variateur

de vitesse. Il se contentera de considérer le contenu de ce compte-rendu comme un ensemble de données associé au variateur de vitesse.

Le modèle que nous exposons dans ce mémoire offre les possibilités pour réaliser cette association. C'est l'aspect "Données Annexes". Précisons cependant qu'il s'agit bien ici d'une commodité offerte aux traitements et que son utilisation excessive peut être contestable. La performance des traitements est en effet conditionnée par une répartition la plus fine possible des données dans la base. Manipuler des données en "blocs" va donc à l'encontre des performances des traitements.

45. Liens entre les différents aspects

Ainsi que nous l'avons expliqué, la découpe entre les différents aspects à pour seul objectif de faciliter la présentation du modèle. Les données des différents aspects sont bien évidemment reliées entre elles. Ainsi le modèle assure les liens entre le logiciel et le matériel, les liens entre la structuration et la fourniture permettant par exemple de conserver la trace des choix technologiques (A quelle fonction répond tel constituant) et bien sûr les liens entre les données annexes et la quasi-totalité des entités du modèle.

5. LA FOURNITURE : LES DONNEES DECRIVANT LE MATERIEL51. Introduction

Nous allons entamer la présentation du modèle de structure de données par l'aspect fourniture et plus précisément par la partie "matériel" de cet aspect fourniture. Dans cette section et dans toute la suite du mémoire, nous serons amené à représenter par des figures des parties du schéma conceptuel qui sera donc ici nécessairement incomplet. Pour une description complète de ce schéma conceptuel BASEPTA, nous renvoyons le lecteur aux travaux BASEPTA [BAS 89A].

D'autre part, le modèle sera illustré au travers d'exemples. Dans ces exemples, nous avons choisi de représenter les instances des entités comme indiqué sur le schéma de la figure 38. Les instances des associations seront représentées par des segments reliant les instances d'entités, segments fléchés pour les relations de spécialisation, segments fléchés avec la désignation "AG" pour les relations d'agrégation.

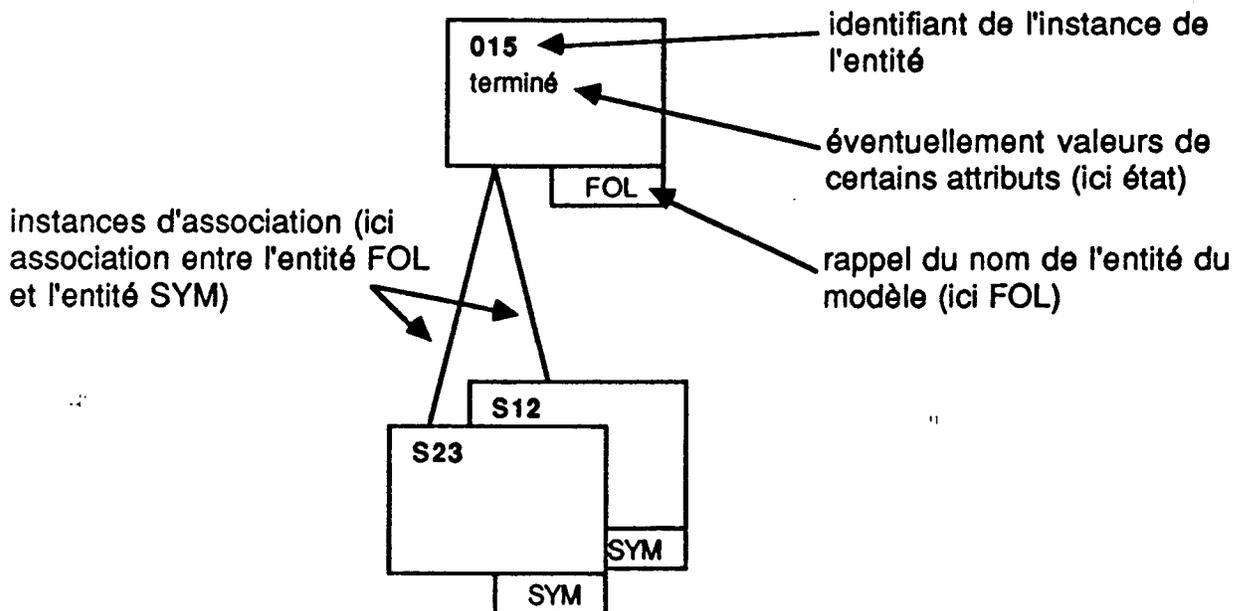


Fig 38

52. L'assemblage des constituants matériels de l'application

521. Le constituant : aspect logique

La fourniture matérielle de l'application consiste en un assemblage de constituants d'automatismes interconnectés par des liaisons (fils, câbles, tubes pneumatiques, hydrauliques...).

Il faut noter que la distinction entre constituant et liaison peut être relativement subjective. Nous reviendrons plus en détail sur ce point dans la suite de ce mémoire, mais nous pouvons dès à présent préciser qu'il n'existe pas de symétrie formelle entre constituant et liaison dans le modèle. Ainsi les constituants possèdent des bornes ou orifices dans le cas de l'hydraulique ou du pneumatique. Le modèle traite l'aspect logique de ces bornes mais il ne prend pas en compte les bornes des liaisons. De même, nous verrons que le modèle supporte une structuration hiérarchique des différents aspects logiques du constituant, mais pas des connexions.

En bref, le constituant est privilégié par rapport à la liaison dans notre modèle. Il y est décrit plus finement. Il pourra donc être préférable de considérer une liaison complexe telle qu'un câble avec des connecteurs spécifiques comme un constituant plutôt que comme une liaison. Un bon critère pour distinguer liaison et constituant serait l'approvisionnement: le constituant s'approvisionne de manière "discrète", par un nombre d'articles, la liaison de manière "continue", par ce que l'on appelle du "disponible" (rouleau de fil...). Dans notre exemple du câble, il est facile de voir que cette définition peut valablement être appliquée.

De manière générale, tous les constituants matériels de l'application sont identifiés de manière absolue dans l'application par une appellation unique. Cette appellation est, par exemple, le nom fonctionnel. Elle recouvre rarement un seul constituant physique. Ainsi l'appellation "KM1" regroupe les divers objets matériels réalisant le contacteur KM1: bobine, bloc comprenant les pôles et un contact additif, blocs de contacts additifs temporisés...Il serait plus exact de dire que cette appellation identifie non pas un constituant mais une "fonction constituant" tel la fonction contacteur KM1. Ceci justifie pleinement l'appellation "nom fonctionnel".

L'entité du modèle Constituant Matériel Application (CMA) permet de stocker cette fonction constituant. L'identifiant d'une instance de l'entité CMA sera l'appellation unique dans l'application de l'assemblage de constituants qu'elle traduit. Autrement dit, l'identifiant dans la Base Application d'une instance de CMA est significatif dans l'application automatisée. Ainsi, nous créerons une instance de CMA dont l'identifiant sera KM1.

D'un point de vue méthodologique, il est souhaitable que ces appellations soient significatives voire même normalisées. Ainsi, en électromécanique, le préfixe KM désigne une fonction contacteur de puissance, KA une fonction contacteur auxiliaire, F une fonction relais de protection, AU une fonction arrêt d'urgence... Nous pouvons remarquer que, dans notre exemple d'illustration, il est facile de déterminer les instances de l'entité CMA par une exploitation directe de la légende fonctionnelle.

La fonction constituant est généralement dissociée en plusieurs sous-fonctions correspondant à différents aspects du constituant. Le contacteur KM1 se décompose ainsi en sous-ensembles fonctionnels bobine, pôles, contact à ouverture, contact à fermeture... Ces différentes parties du constituant possèdent des bornes. Le modèle supporte cette découpe grâce à l'entité Constituant Matériel Élémentaire (CME). Cette entité est bien sûr associée à l'entité CMA par une relation d'agrégation, toute instance de CME étant une partie ou un "aspect" d'une instance de CMA. L'entité Élément De Raccordement (EDR) permet de stocker les bornes des instances de CME. Ici également, la relation d'agrégation contraint à identifier toute instance d'EDR par rapport à l'instance du CME à laquelle elle appartient. Remarquons que la terminologie "Élément De Raccordement" permet d'éviter le choix entre les mots borne et orifice ayant respectivement des connotations trop électriques et trop fluidiques.

Le schéma de la figure 39 récapitule la chaîne CMA-CME-EDR.

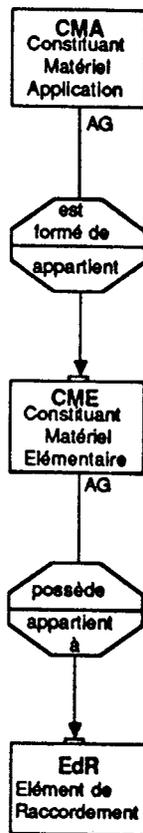


Fig 39

Envisageons, à titre d'illustration, le sectionneur porte-fusibles Q1 de notre exemple. Au travers des folios du schéma de principe, nous identifions divers aspects de Q1: la partie puissance proprement dite sur le schéma de puissance moteur G1 (Fig 40), un contact à fermeture et un contact à ouverture/fermeture sur le schéma de contrôle contacteurs (Fig 41).

PARTIE II - MODELE DE STRUCTURE DE DONNEES APPLICATION

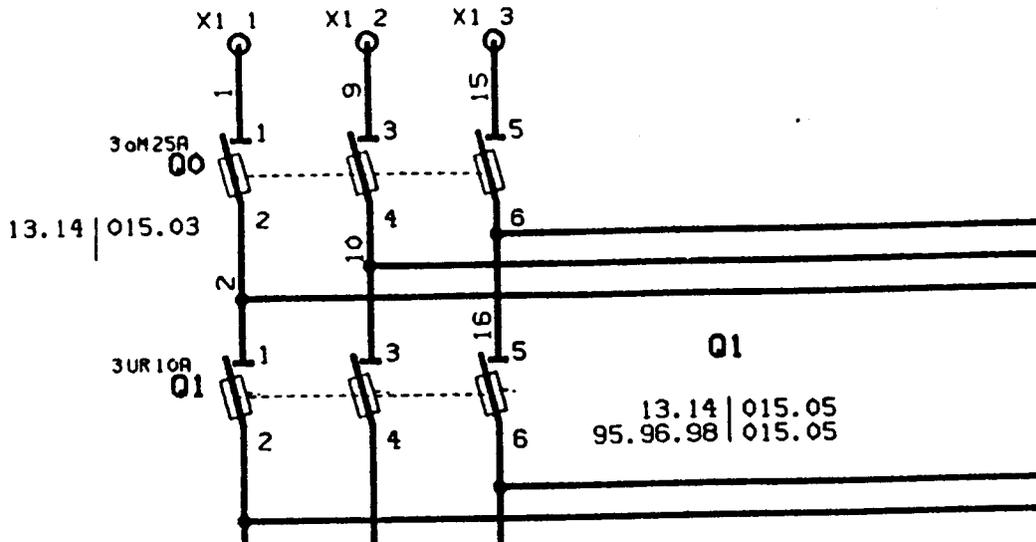


Fig 40

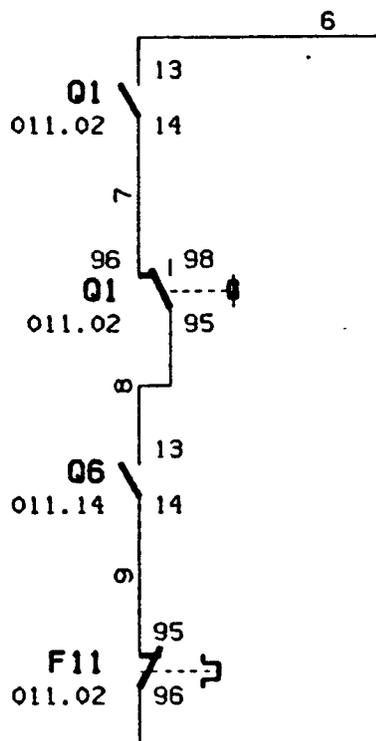


Fig 41

Ces informations sont traduites dans la base comme indiquées sur le schéma de la figure 42.

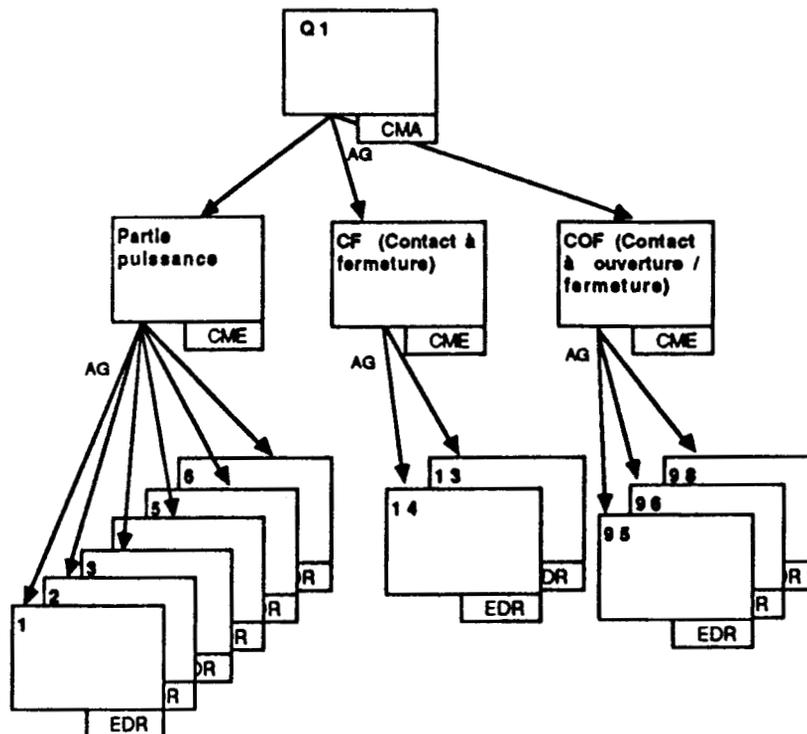


Fig 42

Sur cet exemple, les liens entre les données de schématique du schéma de principe et les données de la structure application sont simples à exprimer: à chaque symbole, on associe une instance de CME car, en fait, un symbole du schéma de principe est la représentation d'une fonction élémentaire du composant.

Il peut arriver aussi que le constituant matériel soit manipulé globalement, comme un tout. Il faut alors considérer une instance de CME associée directement au CMA et prenant en compte cet aspect global du constituant. Le modèle supporte donc une structure hiérarchique des

différents niveaux de CME. Ainsi, si le sectionneur Q1 précédent est manipulé globalement par un traitement, les données pourront être traduites dans la base comme indiqué sur le schéma de la figure 43.

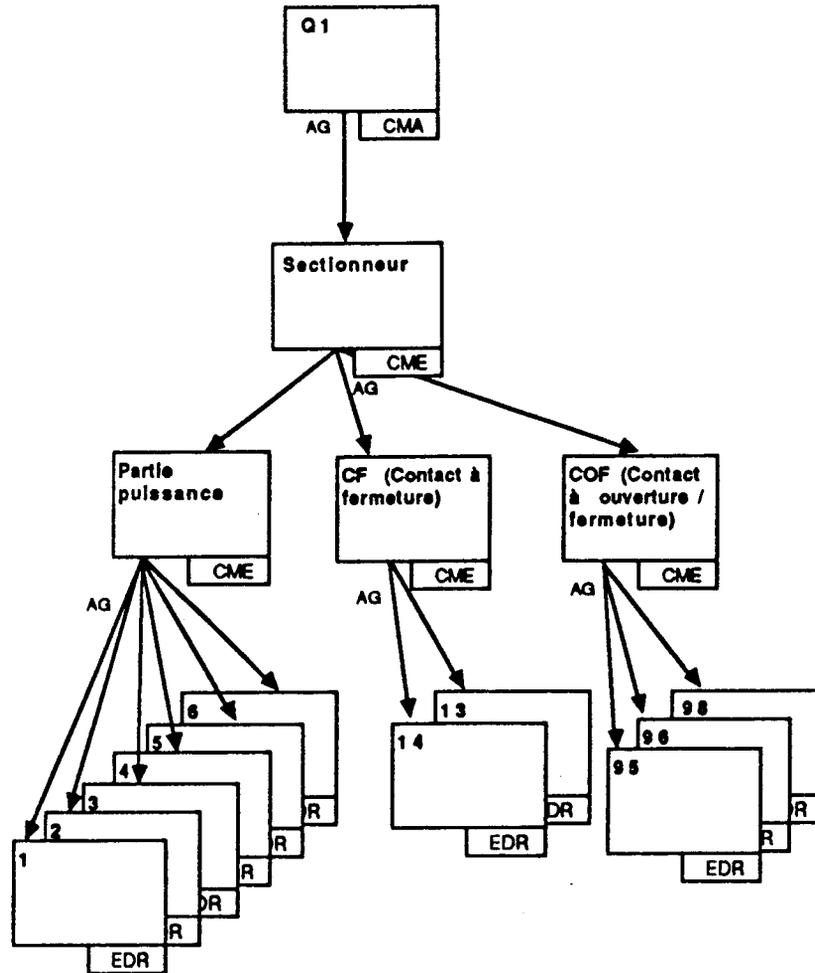


Fig 43

Il existe donc dans notre modèle une "association permettant de lier entre elles les instances de CME. Cette association est également une agrégation. Elle porte la contrainte suivante: une instance de CME appartient à une autre instance de CME de niveau plus élevé ou à une instance de CMA, mais pas au deux simultanément.

Nous obtenons en définitive le schéma de la figure 44 qui traduit l'aspect logique des constituants de l'application.

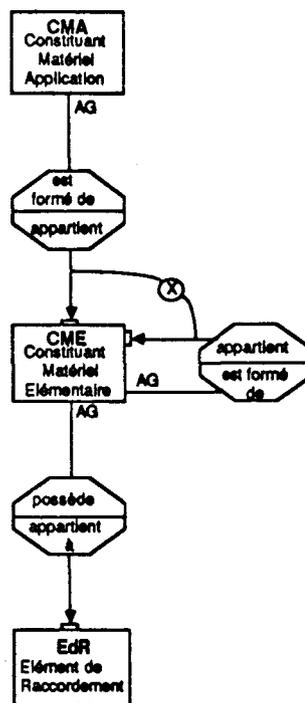


Fig 44

Remarque:

Il nous semble souhaitable d'imposer que toute instance de CMA soit associée à une instance de CME correspondant à l'aspect global du constituant matériel. Cette contrainte faciliterait le travail des traitements, le CMA étant alors considéré comme le nom du CME de plus haut niveau dans la hiérarchie. Précisons cependant que le modèle BASEPTA n'impose en rien une telle contrainte.

522. Assemblage logique de constituants

Après avoir décrit le constituant suivant son aspect logique, il convient à présent d'explicitier la manière dont la Base Application prend en compte l'interconnexion des constituants.

D'une manière générale, l'assemblage de constituants matériels d'automatismes se fait par la connexion de leurs

bornes. Cette relation de connexion entre les bornes est une relation d'équivalence dont les classes sont constituées d'éléments de raccordement ayant un même potentiel. Autrement dit, décrire l'assemblage des constituants matériels revient à regrouper leurs bornes dans des équipotentielles. Ici, nous ne traduisons qu'un aspect logique de l'interconnexion qui, physiquement, pourra être réalisée de multiples manières (fils, peignes...).

L'entité Equivaleur de Connexion (EVC) permet de représenter les équipotentielles. Le terme équipotentielle n'a pas été retenu car il est peu parlant pour des hydrauliciens ou des pneumaticiens, même si, mathématiquement, une isobare est bien une équipotentielle.

Le schéma de la figure 45 décrit donc l'assemblage logique des constituants.

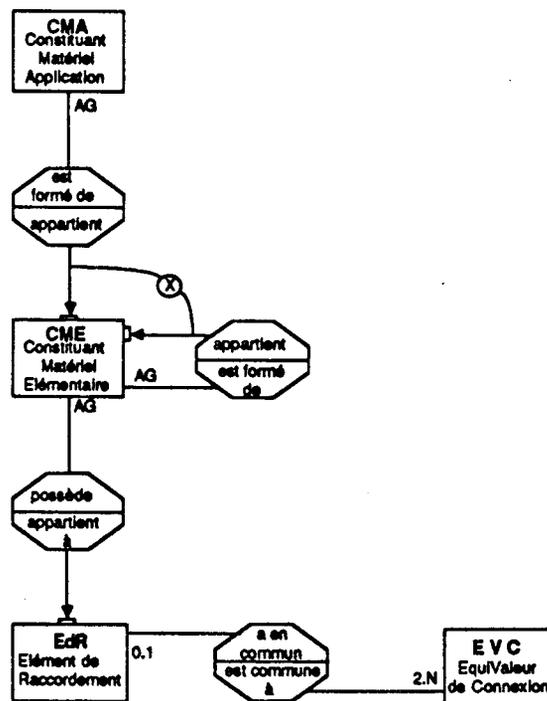


Fig 45

Reprenons le cas du schéma de principe de notre exemple d'illustration. L'extrait qui en est donné sur la figure 46 constitue une représentation (partielle) de la manière dont sont interconnectés logiquement les constituants Q1, Q6 et KM1. Il lui correspond dans la Base Application, les instances de la figure 47. A ce stade, nous ignorons tout des raccordements physiques effectifs des bornes de Q1, Q6 et KM1.

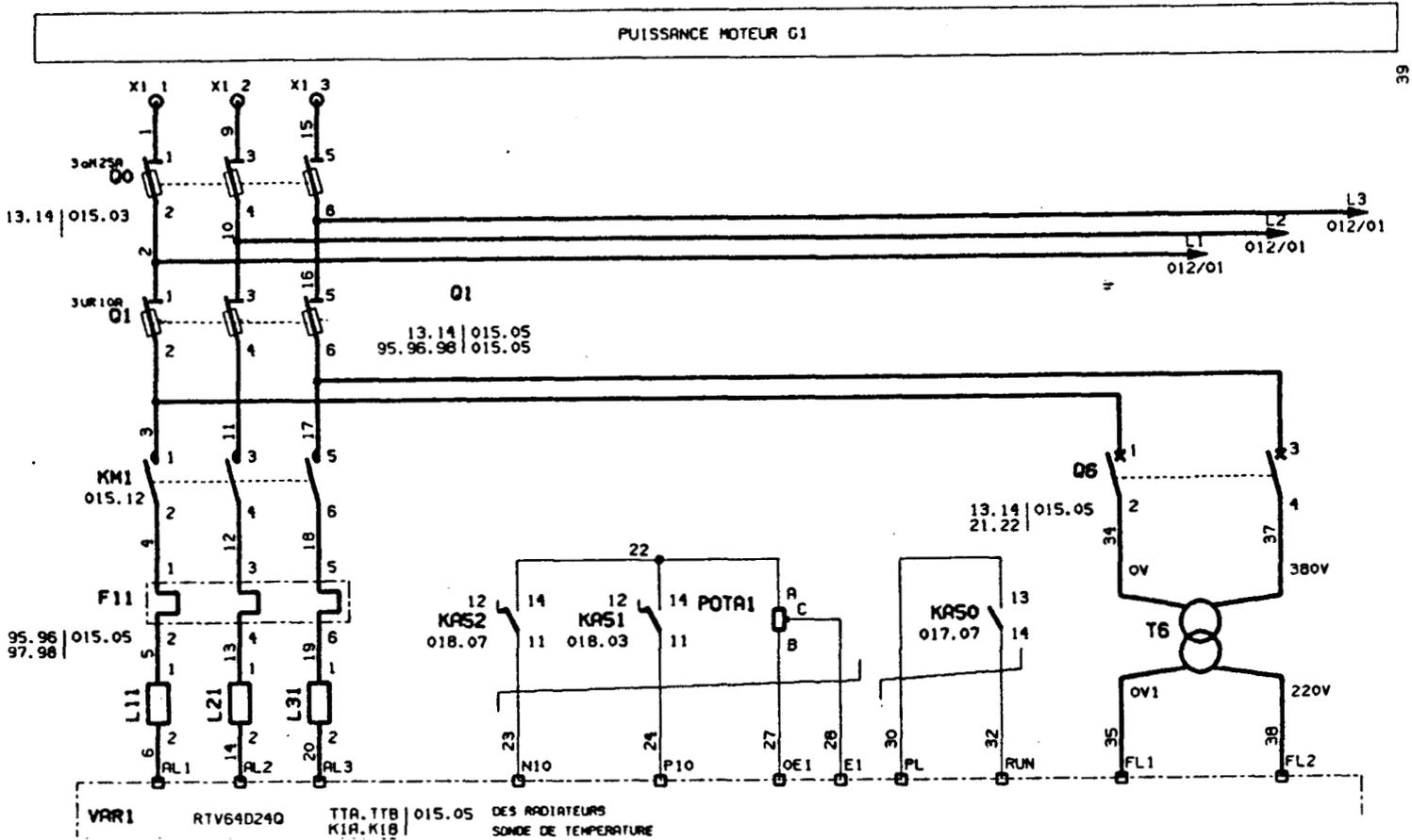


Fig 46

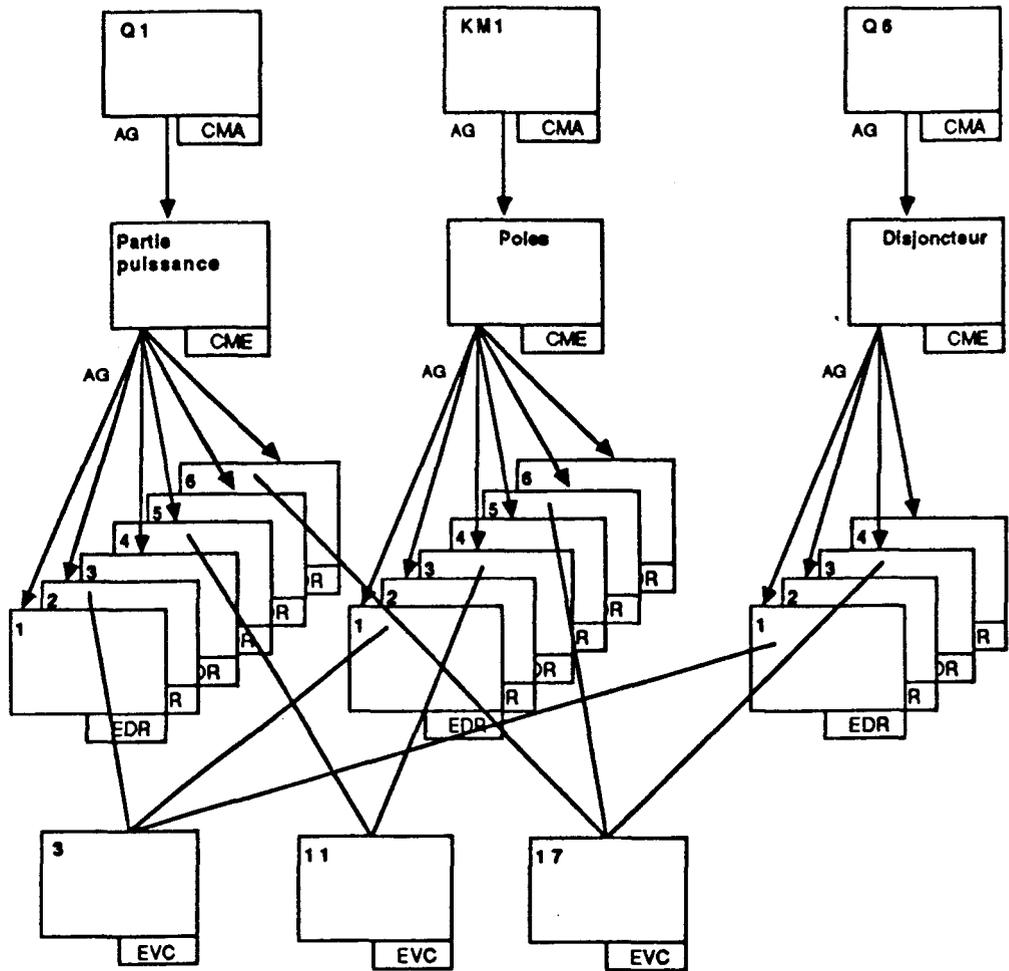


Fig 47

En poursuivant notre réflexion sur les liens entre données du schéma de principe et données de la Base Application, nous pouvons remarquer qu'une instance d'EVC se représente par une équipotentielle sur le schéma de principe. Une liaison schématique telle que celle qui traduit le fait que la borne 2 du symbole Q1 a été reliée à la borne 1 du symbole pôles de KM1 lors de la construction du schéma de principe, n'est pas associée à un objet de la Base Application. Cette donnée peut d'ailleurs à juste titre être considérée comme une donnée purement représentative. Seule la notion d'équipotentielle contient une information application.

En résumé, le schéma de la figure 45 permet de stocker la sémantique application contenue dans le schéma de principe.

523. Aspect logique d'une connexion réelle

Nous allons à présent préciser la manière dont sont connectées les bornes des constituants matériels. En effet, l'équivalent de connexion n'indique pas quelles sont les connexions physiquement réalisées.

L'entité Connexion (CONN) au mnémonique bien maladroit, permet de décrire d'un point de vue logique les connexions matérielles entre bornes de constituants (cf fig 48).

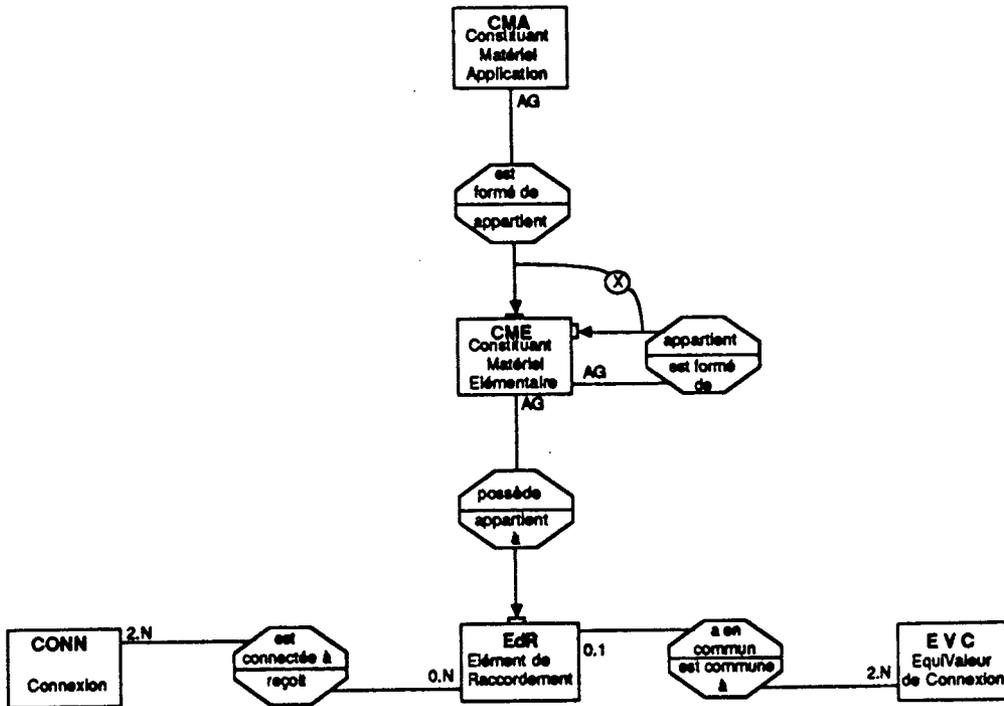


Fig 48

Notons la cardinalité 2.N qui interdit qu'une connexion existe dans la Base Application sans être reliée à au moins deux éléments de raccordement. La valeur "N" permet elle de traiter le cas de peignes associant plus de deux bornes.

L'association d'instances de CONN à des instances d'EDR n'est pas indépendante des associations entre ces instances d'EDR et des instances d'EVC. Plus précisément, il n'est possible d'établir des connexions qu'entre des

bornes de constituants ayant même valeur d'équipotentielle ou, inversement, des bornes d'EDR associées par des connexions directes (c'est à dire n'utilisant pas un constituant intermédiaire) ont en commun une même valeur d'équipotentielle. Nous rencontrons ici la première faiblesse importante du modèle BASEPTA : la contrainte entre les deux associations EDR-EVC et EDR-CONN n'est pas traduite. Les traitements sur la Base Application devront cependant la respecter systématiquement, cette contrainte étant bien une donnée application.

Ainsi, en reprenant l'exemple de la figure 46 pour l'équivalent de connexion 3, il existe plusieurs moyens d'établir des connexions en respectant l'équipotentielle. Les figures 49 et 50 en montrent deux.

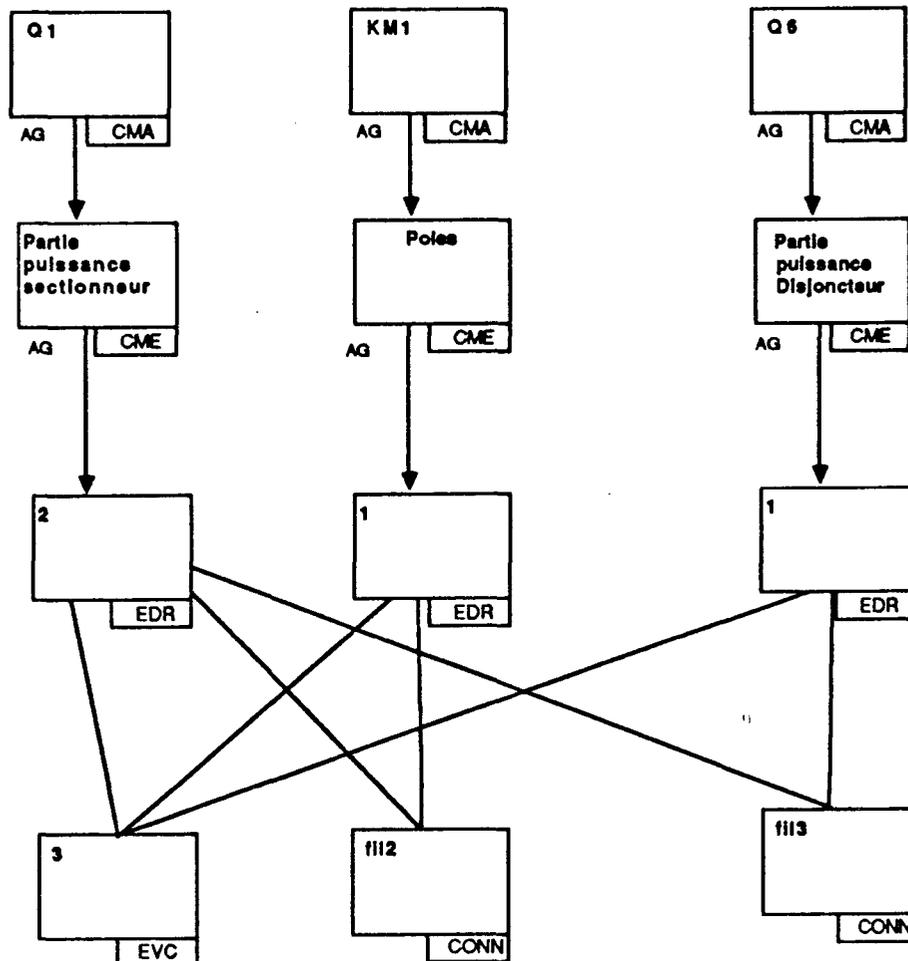


Fig 49

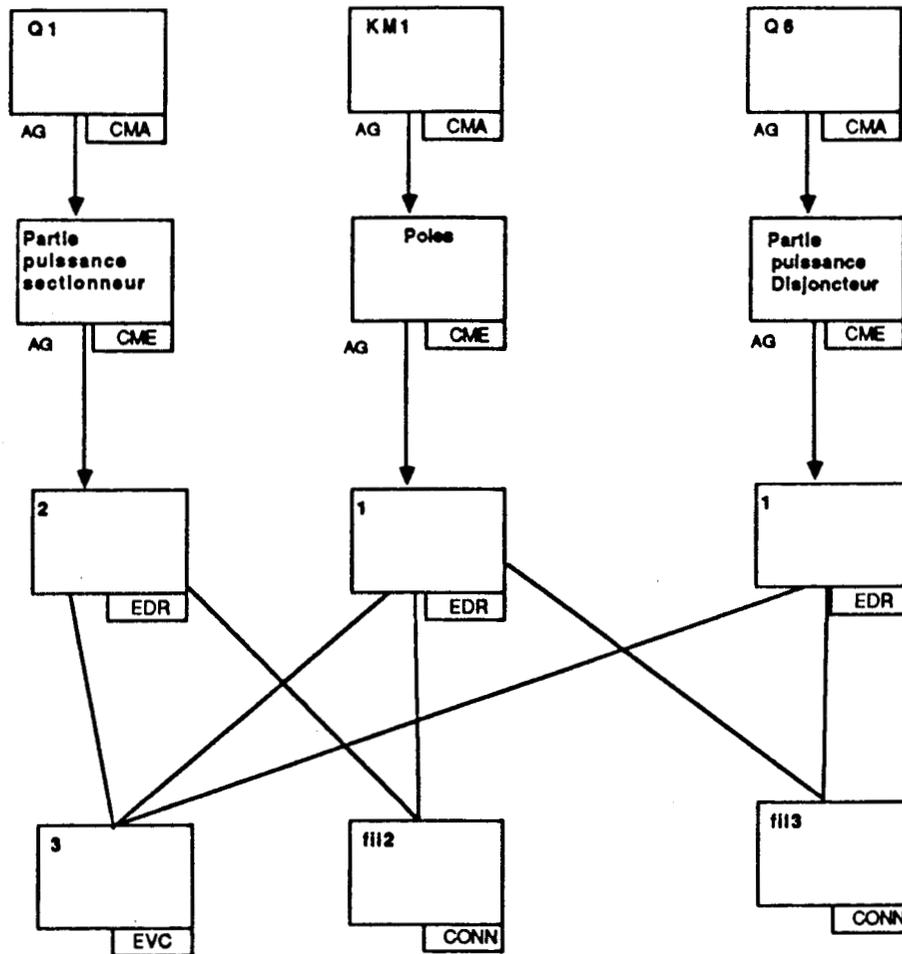


Fig 50

Nous remarquons que les instances de l'entité CONN ne peuvent en aucune façon être extraites du schéma de principe. Le schéma de câblage ou la liste de câblage peuvent en donner une partie incomplète car, comme nous le verrons, la connexion physique ne correspond pas nécessairement à l'existence d'un fil en électromécanique ou à un tube en fluide.

53. Les aspects réels et physiques des constituants et liaisons matériels de l'application

531. Aspect réel du constituant

Jusqu'ici, nous n'avons envisagé que les constituants et leur assemblage sur le plan logique. Mais ces constituants

sont des objets matériels réels.

L'entité Constituant Réel (CR) traduit cet aspect physique. Chaque instance de CR regroupe certaines sous-fonctions d'un constituant matériel, autrement dit certaines instances de CME rattachées à une même instance de CMA (cf fig 51). Ainsi, le contacteur KM1 donne lieu à plusieurs sous-fonctions: bobine, pôles. Un constituant réel correspondra à la bobine, un deuxième regroupera les pôles (cf schéma de la figure 52). La dissociation des CR en CME nécessite, comme l'a traduit de façon imagée Noël BOUTEILLE au cours des travaux BASEPTA, l'utilisation de la scie!!

Cette notion de constituant réel constitue une grande richesse potentielle pour les traitements sur la Base Application. Nous allons l'illustrer sur un exemple.

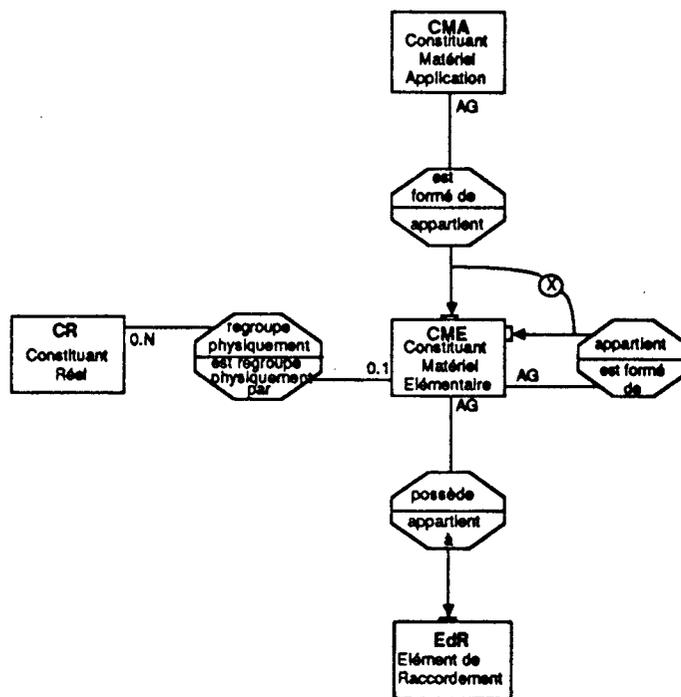


Fig 51

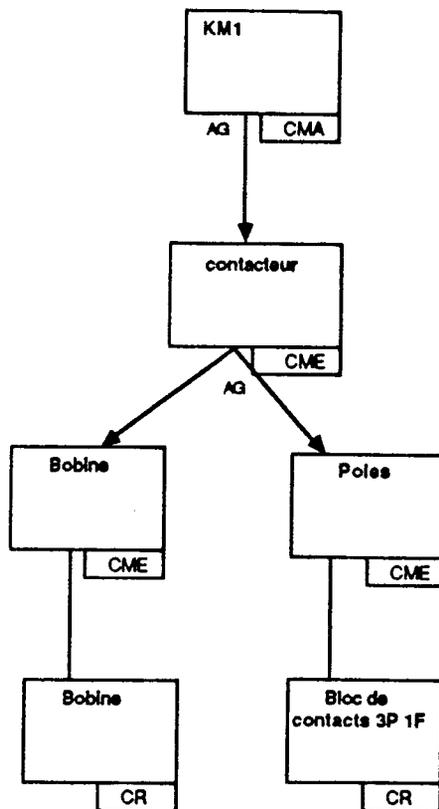


Fig 52

Supposons que l'étude nous ait conduit à introduire une autre instance de CME correspondant à un contact à fermeture supplémentaire pour KM1. Le bloc 3P1F peut l'accepter. Il n'est pas nécessaire de créer une autre instance de CR. Un deuxième contact à fermeture supplémentaire, par contre, pourra être choisi sur un bloc additif 1F1O (1 contact à fermeture, 1 contact à ouverture). Il y a donc ici obligation de créer une instance de CR supplémentaire. Le schéma résultant est donné par la figure 53. Le traitement utilise donc la structure de données et des connaissances sur la nature du CR (telle que sa composition en différents types de CME) pour ne pas remettre en question les instances de CR lorsque ce n'est pas nécessaire.

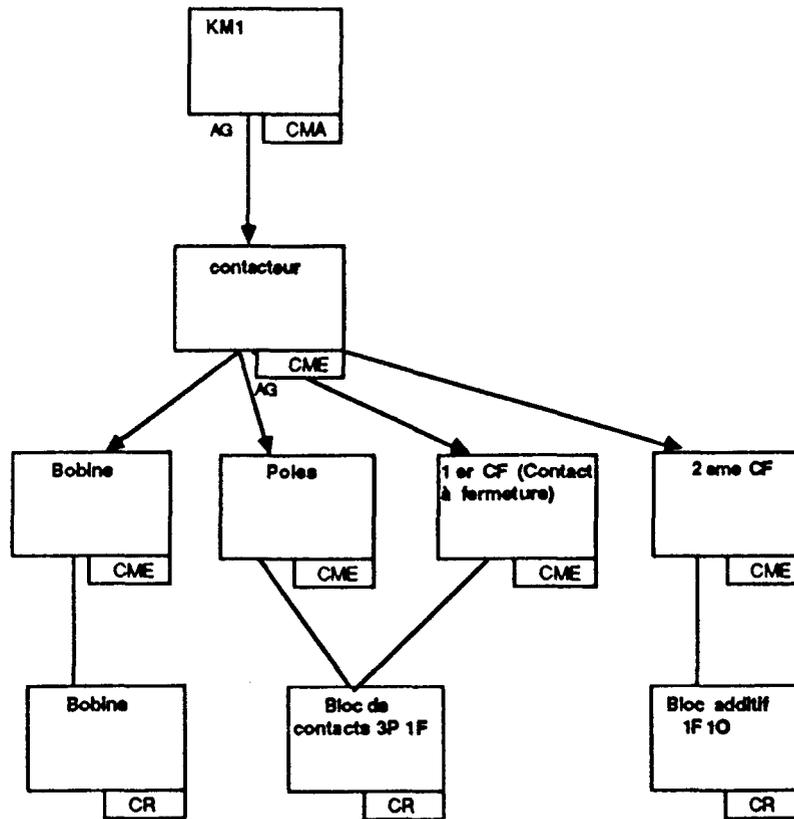


Fig 53

Nous pouvons remarquer que la nomenclature des appareils ne fournit pas une information aussi riche que celle induite par les associations CME-CR. En effet, la nomenclature donne une liste de matériels, donc implicitement une liste d'instances de CR, et les associe aux noms fonctionnels, c'est à dire aux instances de CMA. L'agrégation CMA-CME permet de dire que si une instance de CR est associée à une instance de CME, elle est aussi associée indirectement à l'instance du CMA à laquelle est rattachée l'instance du CME considérée. Notre association CME-CR porte donc l'information CMA-CR de la nomenclature. Mais elle est plus fine puisqu'elle précise quels aspects du CMA appartiennent aux différents CR. Parfois ce niveau de détail apparaît trop fastidieux et peu intéressant. Ainsi, pour le sectionneur Q1, il n'est pas toujours nécessaire de préciser que la partie puissance, le contact de pré coupure et le contact de fusion fusibles sont regroupés dans le constituant réel. Il est préférable d'associer cette instance de CR à l'instance de CME correspondant à l'aspect global du sectionneur Q1 (cf fig 54).

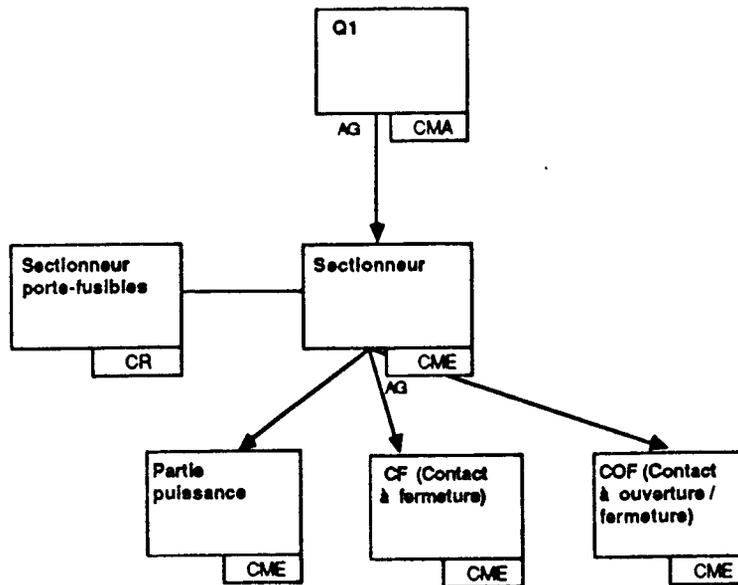


Fig 54

Pour une application donnée, l'ensemble des instances de CR constitue une partition de la fourniture matérielle.

532. Localisation physique d'un constituant réel.

Le constituant réel représentant l'aspect physique de la fourniture, il apparaît logique de faire porter les données de localisation des constituants sur l'entité CR plutôt que sur l'entité CME ou même CMA.

Le modèle permet donc de décrire une ou plusieurs localisations relatives et une ou plusieurs localisations absolues de tout constituant matériel (cf fig 55).

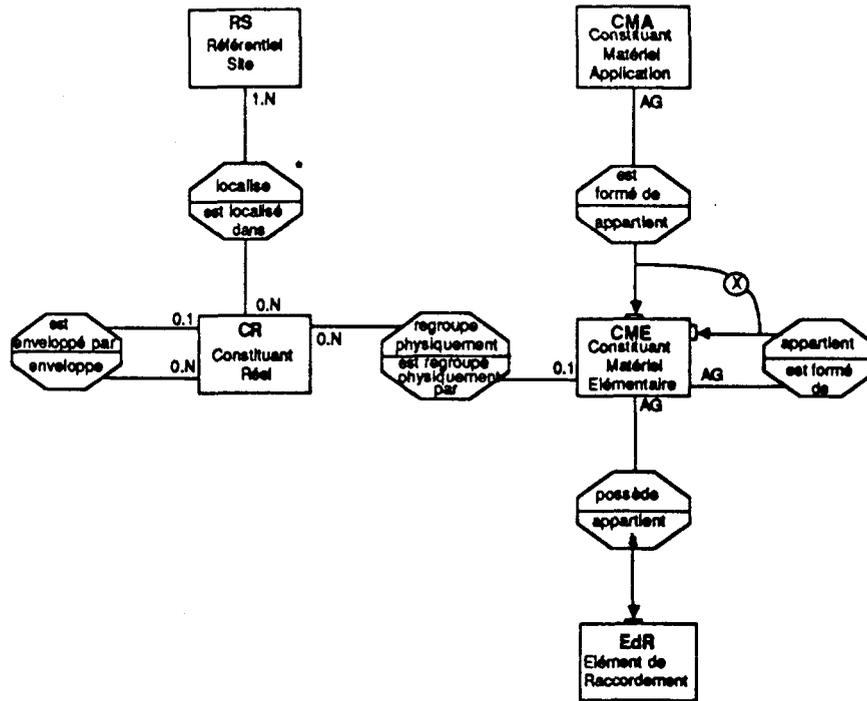


Fig 55

L'entité Référentiel Site (RS) permet de stocker différents systèmes de repérage ou référentiels de l'application. En associant une instance de CR à une instance de RS, on repère donc un constituant réel. L'association portera comme attribut les coordonnées de l'instance de CR dans le référentiel envisagé.

L'association de deux instances de CR permet de traiter le problème classique de la gestion d'enveloppes. Ainsi, le modèle permettra de traduire que le sectionneur Q1 se trouve dans l'armoire de commande ARM1. Il est dommage que cette association ne porte pas dans le modèle BASEPTA d'attribut. Les coordonnées de localisation du constituant dans son enveloppe auraient été les bienvenues.

En ce qui concerne les relations entre données de documentation et données application, il apparait clairement ici que les vignettes représentant les constituants matériels sur le plan d'implantation en armoire doivent être associées aux instances de l'entité CR et non à celle de l'entité CME. Le traitement d'implantation en armoire peut être valablement considéré comme une succession d'instanciations de l'association sur CR.

533. Les deux types de constituants réels.

Nous avons précédemment affirmé que, de manière générale, tout constituant matériel était identifié dans l'application par une appellation unique. Cette affirmation doit cependant être quelque peu nuancée. Il existe dans la fourniture matérielle un ensemble de constituants secondaires, faisant l'objet d'un approvisionnement effectif, mais pour lesquels il n'est pas nécessaire de fournir un nom fonctionnel. Ces constituants seront qualifiés d'accessoires. Ils ne correspondent pas à une fonction-constituant.

Compte-tenu des définitions des entités CMA, CME et CR, un constituant accessoire ne fait pas l'objet d'instances de CMA ou de CME. Par contre, il lui correspond une instance de CR.

Nous sommes alors conduits naturellement à utiliser la relation de généralisation en faisant apparaître deux sous-classes de l'entité CR. La première, CR/AP, correspond aux constituants réels de l'application, c'est à dire en liaison avec des fonctions-constituant. La seconde, CR/AC, correspond aux constituants accessoires (cf fig 56).

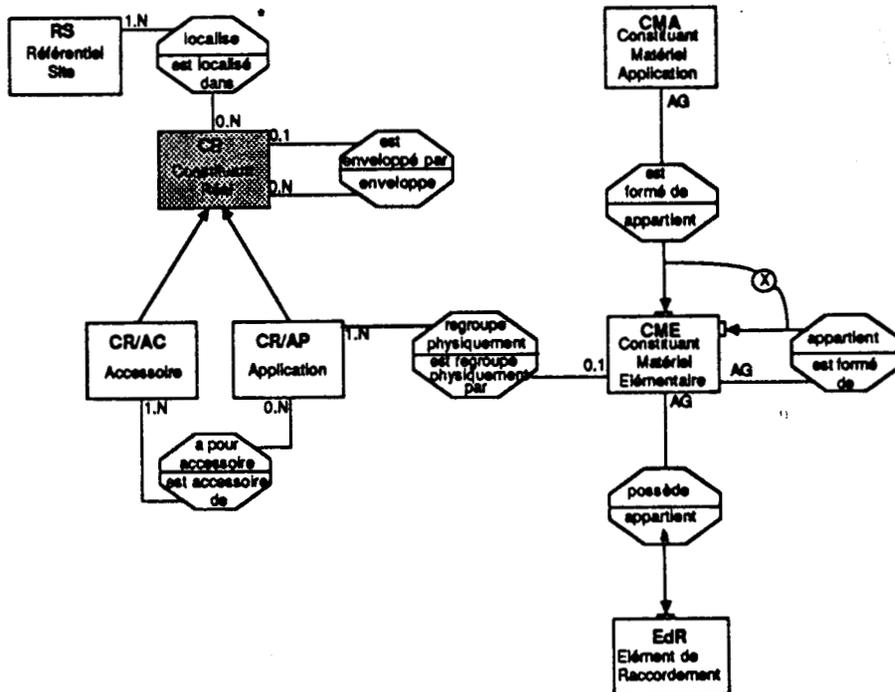


Fig 56

Si nous comparons le schéma de la figure 56 avec celui de la figure 54, nous constatons que la cardinalité 1.N de l'entité CR dans l'association CME-CR est plus restrictive sur la figure 56. En effet la cardinalité 0.N de la figure 54 était nécessaire pour prendre en compte les accessoires, instances de CR non reliées à des instances de CME. La généralisation permet donc de traduire au niveau du modèle une sémantique plus forte.

Une deuxième remarque concerne l'association entre les deux sous-classes CR/AC et CR/AP. Cette association implique qu'un accessoire ne peut être accessoire que d'un constituant application et pas d'un autre accessoire. Cette contrainte peut apparaître dans certains traitements gênante. Elle interdit des niveaux successifs d'accessoires.

Illustrons la notion de constituant accessoire sur un exemple. Supposons que le sectionneur Q1 doive se monter dans l'armoire ARM1. Cette armoire ne contient que des profilés DIN d'entraxe 120mm et compte-tenu de sa taille, le sectionneur Q1 ne peut se monter directement que sur des profilés DIN d'entraxe 60mm. Il faut donc, lors du traitement d'implantation en armoire, insérer une platine de montage se montant sur le profile DIN120, platine sur laquelle se montera Q1. Il n'est pas nécessaire de désigner cette platine par un nom fonctionnel. Elle constitue un accessoire du constituant réel réalisant physiquement le sectionneur Q1. Notons que, compte-tenu de la contrainte énoncée précédemment, les vis de montage de la platine doivent être considérées comme des accessoires du sectionneur et non de la platine. Le schéma de la figure 57 représente cet exemple. Les fusibles de Q1 y sont également considérés comme des accessoires.

Dans l'illustration précédente, nous venons de voir diverses natures d'accessoires de constituants réels. Le modèle BASEPTA laisse cet aspect dans l'ombre. Il nous apparaît indispensable de le compléter par un attribut sur la relation CR/AC-CR/AP indiquant la nature ou, plus exactement, la motivation de l'accessoire (pour montage, pour visualisation ...). De plus, l'association "est accessoire de " devrait supporter une hiérarchie sur les accessoires. Le schéma de la figure 58 propose une modification du modèle tenant compte de ces remarques.

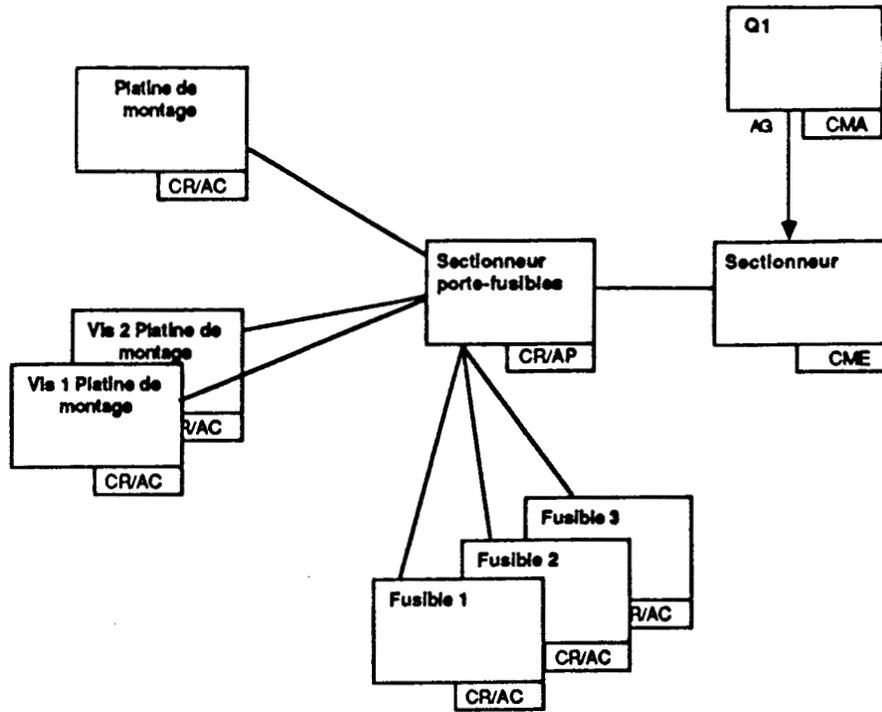


Fig 57

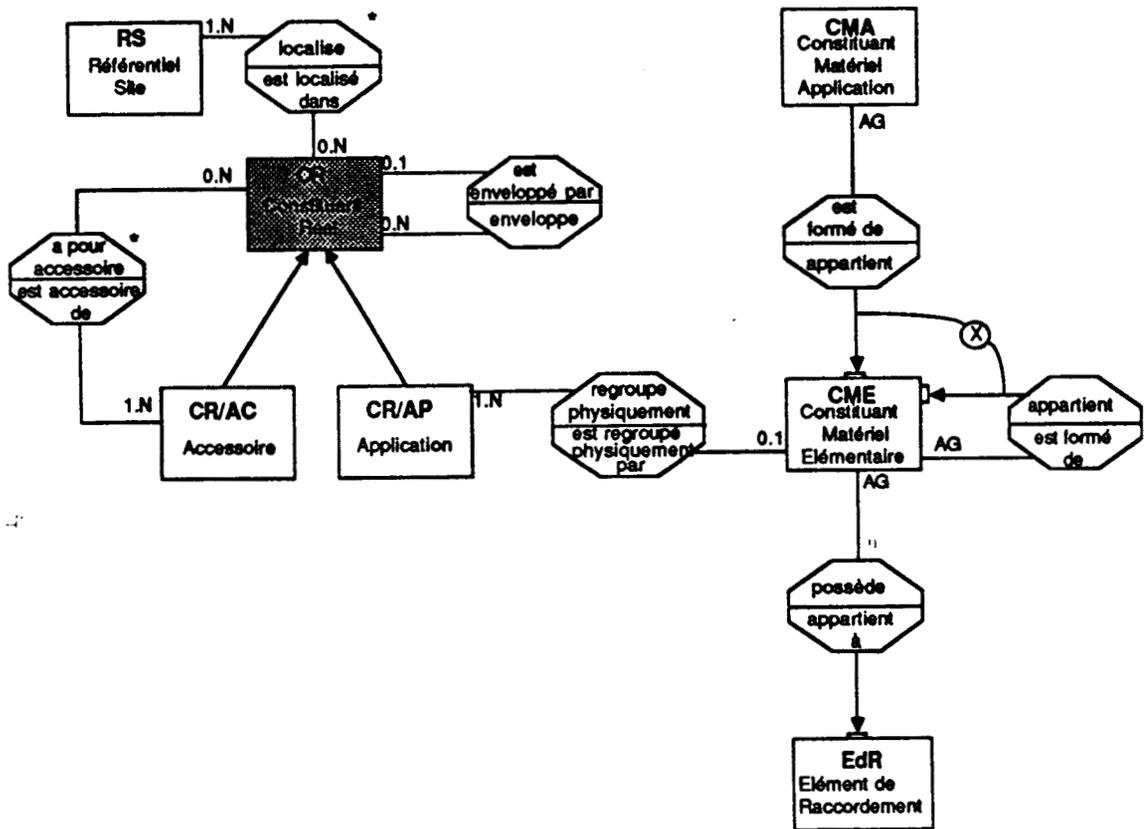


Fig 58

534. Aspect réel d'une connexion entre constituants.

Comme nous l'avons fait pour le constituant, il convient d'associer à l'aspect logique d'une liaison sa réalisation matérielle. Le modèle comprend donc une entité Liaison Physique (LP) traduisant la réalité de la connexion. Cette entité LP est associée à l'entité CONN (cf schéma de la figure 59).

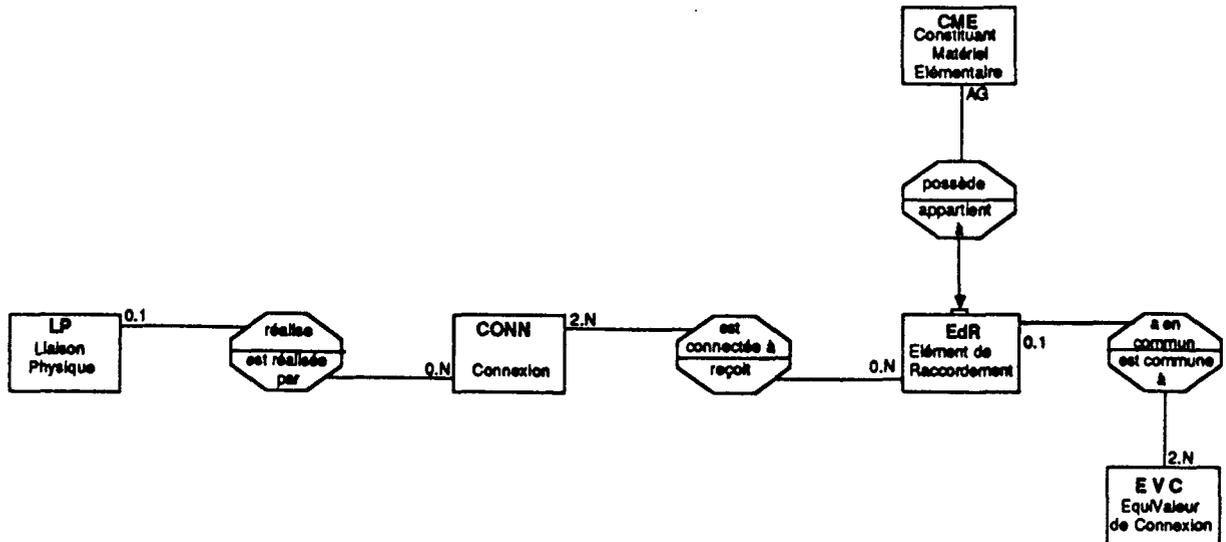


Fig 59

L'ambiguïté du terme "Liaison Physique" est grande. Nous allons donc préciser cette notion.

Comme nous l'avons vu, il existe plusieurs "manières de réaliser", de "matérialiser" une connexion logique. Par exemple, la connexion de deux bornes de constituants peut se faire par utilisation d'un conducteur (fil ou tube suivant la nature du signal entre les deux objets matériels), ou par simple rapprochement des constituants (mise en contact des bornes, montage direct des composants assurant automatiquement la connexion. Prenons l'exemple du vérin V1 et de son régulateur de vitesse (cf schéma de la figure 60).

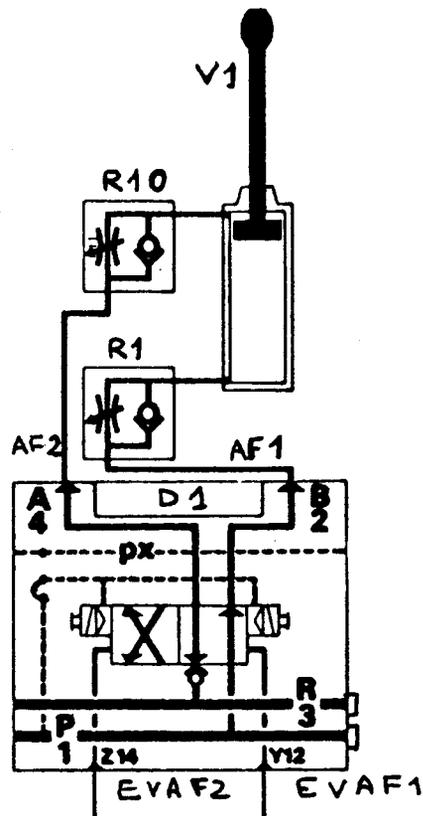


Fig 60

Le schéma montre clairement qu'entre les constituants vérin et régleur de vitesse, il y a connexion. Cependant, lors de la réalisation matérielle de cette connexion, nous n'utilisons pas de tube PVC. Le régleur est monté directement sur l'orifice du vérin réalisant par la même la connexion entre les constituants.

Nous pouvons maintenant expliciter l'entité LP. Nous dirons qu'il y a lieu de créer une instance de LP pour matérialiser une connexion entre constituants lorsqu'il y a ajout d'un élément matériel supplémentaire (c'est à dire autre que les deux constituants en présence) pour assurer cette connexion.

Il est alors immédiat que la connexion régleur de vitesse - vérin ne donne pas lieu à une instance de LP. Le schéma de la figure 60 sera donc traduit au niveau de la structure de données application comme indiqué sur la figure 61.

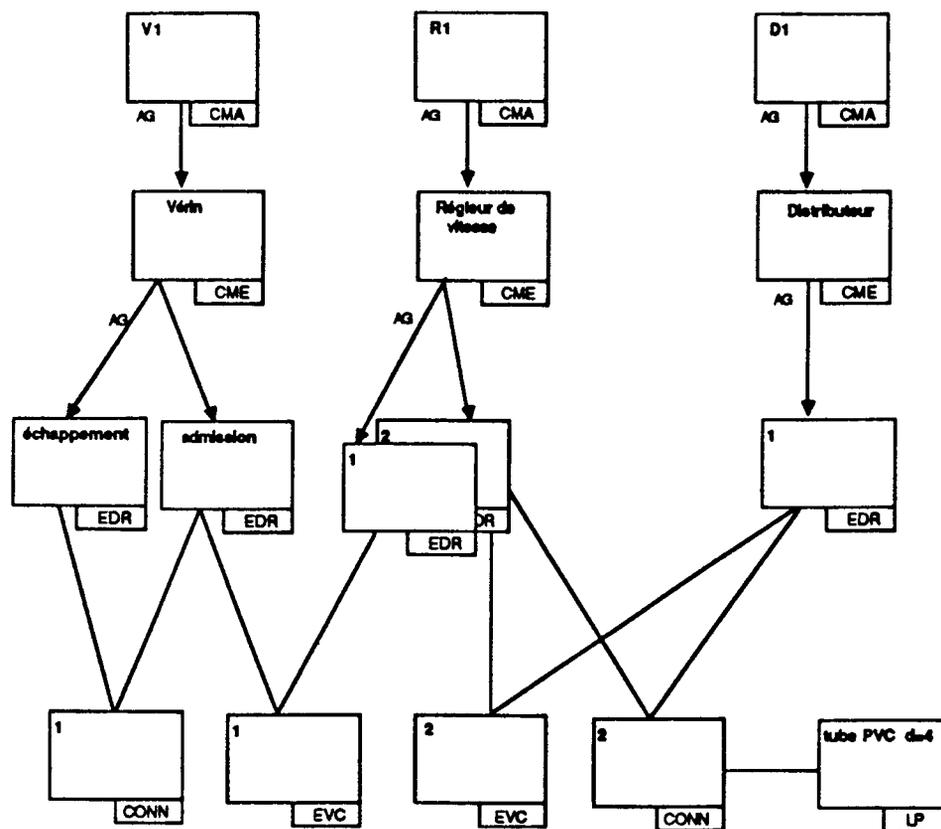


Fig 61

535. Localisation physique d'une liaison réelle

Comme pour les constituants matériels, il est souvent nécessaire de localiser les connexions. Pour le constituant matériel, cet aspect est pris en compte au niveau du constituant réel (CR). L'entité LP joue un rôle analogue pour la liaison.

Cependant, la localisation d'une connexion n'est pas totalement semblable à celle d'un constituant. Ainsi le repérage absolu des liaisons physiques dans un référentiel site n'est pas supporté par le modèle. Les liaisons physiques sont repérées de manière relative les unes par rapport aux autres grâce à une relation d'enveloppe (cf schéma de la figure 62). En première approche, cela peut sembler insuffisant mais en pratique la localisation d'une

connexion se limite à un repérage de fils dans des câbles ou des sous-câbles, c'est à dire précisément à des liaisons physiques dans leur enveloppe.

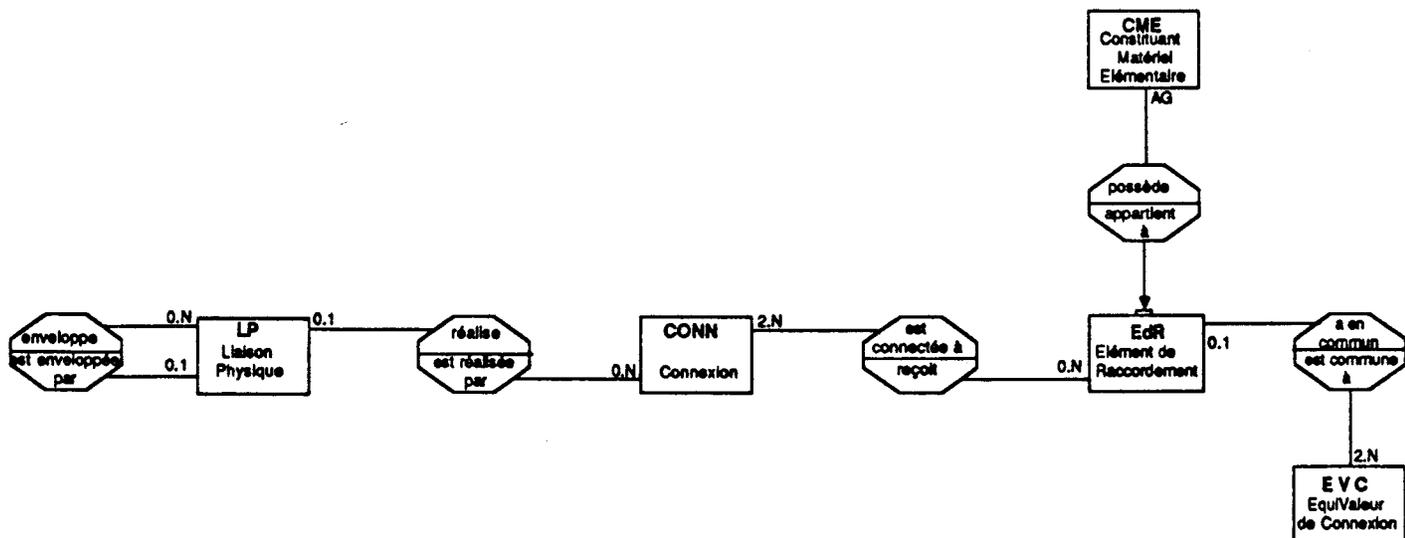


Fig 62

Le modèle paraît donc suffisant pour traduire la localisation usuelle des liaisons. Cependant, il serait souhaitable que la relation d'enveloppe sur l'entité Liaison Physique porte un attribut traduisant les coordonnées d'une instance de l'entité LP dans l'instance qui l'enveloppe (par exemple: numéro de fil dans le câble).

Pour conclure sur la liaison physique, nous allons préciser les liens entre le modèle et le schéma de câblage. Nous avons vu précédemment que les différentes instances de l'entité CONN et leurs associations avec les instances d'EDR étaient reliées à cette liste de câblage. Mais seules les instances de Connexion donnant lieu à une instance de Liaison Physique se trouveront dans cette liste. La relation d'enveloppe sur LP sera exploitée par les traitements de gestion de câbles.

PARTIE II - MODELE DE STRUCTURE DE DONNEES APPLICATION

Le schéma de la figure 63 résume les associations et entités du modèle que nous avons introduites jusqu'ici.

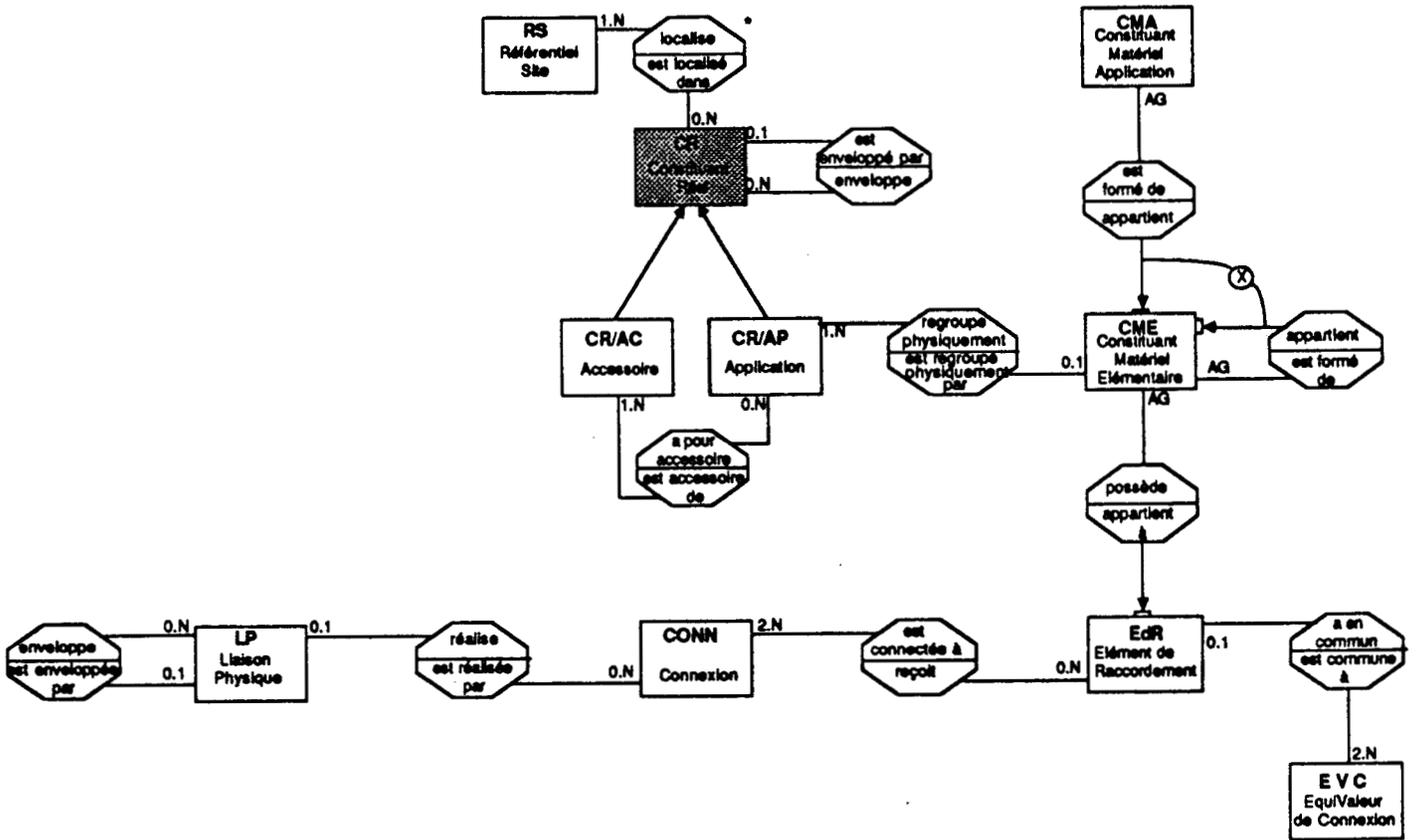


Fig 63

De façon simplifiée, nous avons envisagé la fourniture matérielle suivant trois axes: un aspect fonctionnel, un aspect logique et un aspect réel ou physique. Le schéma de la figure 64 re-situe ces trois aspects.

Aspect du constituant	Entités du modèle	Exemples de traitements s'appliquant typiquement aux différents aspects du modèle	Exemples de documents dont la sémantique peut être traduite par les différents aspects du modèle
Aspect fonctionnel (Fonction constituant)	CMA	Détermination des fonctions constituants	Légende fonctionnelle
Aspect logique (Sous-fonction)	CME EDR EVC	Description des principes de fonctionnement de l'assemblage matériel	Schéma de principe
Aspect réel ou physique	CONN CR-CR/AC-CR/AP LP RS	Regroupement des fonctions dans les constituants matériels Implantation en armoire Gestion des cables	Plan d'implantation Liste de montage Liste de cablage Schéma de cablage

Fig 64

54. L'aspect approvisionnement des constituants et liaisons de l'application.

541. Approvisionnement d'un constituant ou d'une connexion.

Constituants réels et liaisons physiques doivent être approvisionnés lors de la détermination précise et finale du matériel. Cette détermination consiste en réalité au choix d'un fournisseur et d'une référence catalogue (clé d'approvisionnement) chez ce fournisseur pour chaque objet matériel constituant ou liaison. Ce choix est fonction bien entendu d'un certain nombre de caractéristiques techniques souhaitées ou calculées pour le matériel. Ces caractéristiques sont des données application et nous verrons plus loin comment le modèle les supporte.

Pour traiter l'approvisionnement, la solution la plus naturelle pourrait être d'associer, sous forme d'attribut, des clés d'approvisionnement à chaque instance de CR et de LP. Mais cette solution n'a pas été retenue par l'équipe BASEPTA car elle entraîne une grande duplication d'informations.

Ainsi dans la chaîne d'emballage de rouleaux de papier, les deux sectionneurs porte-fusibles Q1 et Q10 sont identiques tant sur le plan de leurs caractéristiques techniques qu'en ce qui concerne leur approvisionnement. La solution évoquée précédemment contraindrait à stocker la même clé d'approvisionnement pour Q1 et Q10 et les mêmes caractéristiques techniques. Il est plus simple de considérer que les constituants réels Q1 et Q10 correspondent à un même article d'approvisionnement dont la clé d'approvisionnement est chez TELEMECANIQUE GK1-EV. C'est cette solution que nous avons retenue.

Toute instance de l'entité CR est reliée à une instance et une seule d'une entité nommée Article d'Approvisionnement (AA), instance qui constitue l'approvisionnement de ce constituant réel. Il va de soi que des constituants réels identiques, mais portant des noms différents dans l'application, sont reliés à la même instance de l'entité AA.

L'entité AA est associée à une entité CLÉ d'Approvisionnement (CLAP) permettant de stocker les différentes clés. Les deux associations entre AA et CLAP

permettent de traduire soit l'approvisionnement effectif de l'application (celui que l'on trouvera dans la nomenclature), soit les approvisionnements équivalents, éventuellement chez d'autres fournisseurs.

Le schéma de la figure 65 montre clairement qu'un Article d'Approvisionnement peut être l'approvisionnement d'un constituant réel ou d'une liaison physique.

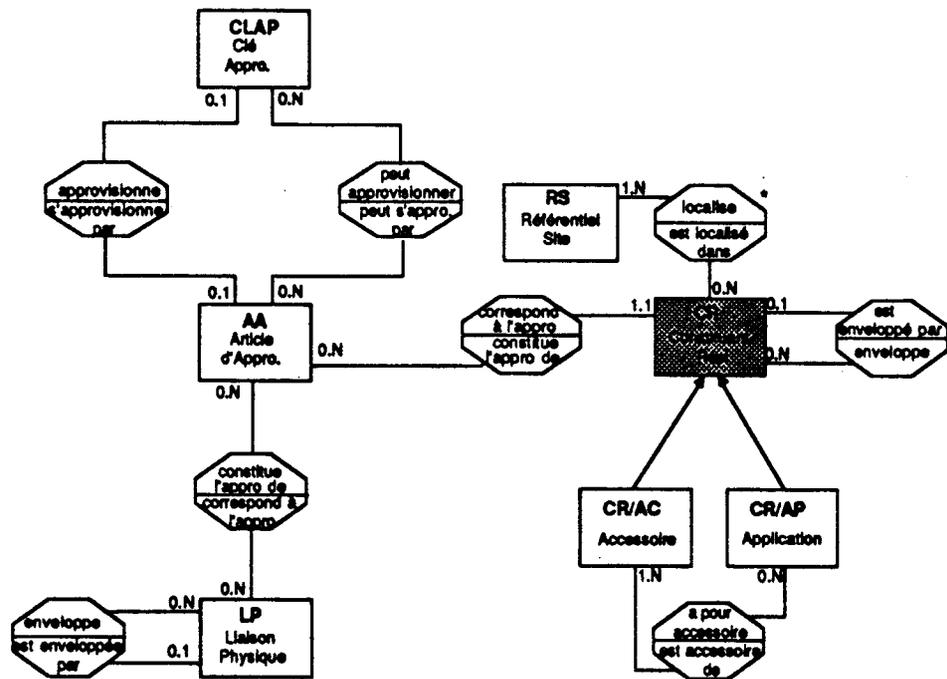


Fig 65

La liste des différentes instances de l'entité AA et le nombre de relations auxquelles elles participent constituent en fait une nomenclature interne de l'application, nomenclature qui deviendra externe lorsque l'on associera à chaque instance de l'entité AA une instance de l'entité CLAP. Le découpage en deux entités AA

et CLAP est bien adapté pour supporter ces deux phases successives.

Dans le cas des deux sectionneurs Q1 et Q10, le schéma de la figure 66 illustre les instances dans la Base Application.

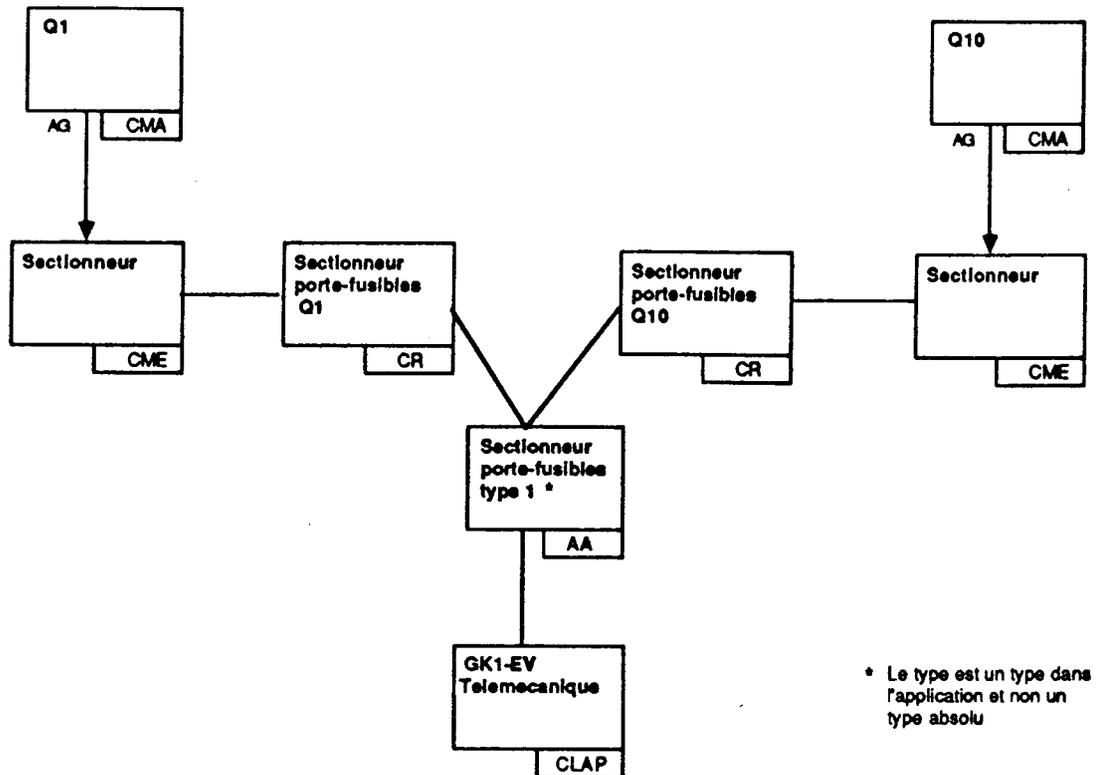


Fig 66

542. Notion de borne disponible

Nous avons vu précédemment que le constituant matériel possède des bornes. L'article d'approvisionnement possède également des éléments de raccordement que l'on peut qualifier de "génériques" par opposition aux bornes "instanciées" du CME. Lorsque l'on associe un article

d'approvisionnement à un constituant matériel, ses bornes sont en quelque sorte des bornes "disponibles" pour le constituant matériel élémentaire. Ces bornes disponibles constituent une entité Elément de Reccordement Disponible (EdRD), agrégée à l'entité Article d'Approvisionnement. Certaines de ces bornes disponibles peuvent ne pas être utilisées, c'est à dire effectivement raccordées dans l'application. D'autres le sont. Le modèle exprime cette utilisation en associant les entités EdR et EdRD (cf schéma de la figure 67).

La notion d'élément de raccordement disponible est intéressante pour le développement de traitements performants. En effet, il peut être souhaitable, lors de la prise en compte des évolutions du système automatisé, d'établir de nouvelles connexions entre constituants. Les instances d'Eléments de Raccordement Disponibles permettront de déterminer si l'approvisionnement doit être remis en cause ou s'il reste des "bornes disponibles" sur les constituants. C'est cette notion qui a conduit à préférer le terme d'Elément de Raccordement "Disponible" à celui d'Elément de Raccordement "utilisable", moins précis à notre avis.

543. Les différents types d'articles d'approvisionnement

Nous avons constaté qu'un article d'approvisionnement pouvait constituer soit l'approvisionnement d'un constituant réel, soit l'approvisionnement d'une liaison physique. Il apparaît assez nettement que ces deux cas sont exclusifs puisque constituant et liaison sont deux notions bien différentes. Il est donc naturel de considérer l'entité Article d'Approvisionnement comme une classe comportant au moins deux sous-classes: la sous-classe des articles d'approvisionnement de constituants réels et la sous-classe des articles d'approvisionnement de liaisons physiques. Comme cela se produit à chaque spécialisation d'entités en sous-classes, les cardinalités seront plus restrictives.

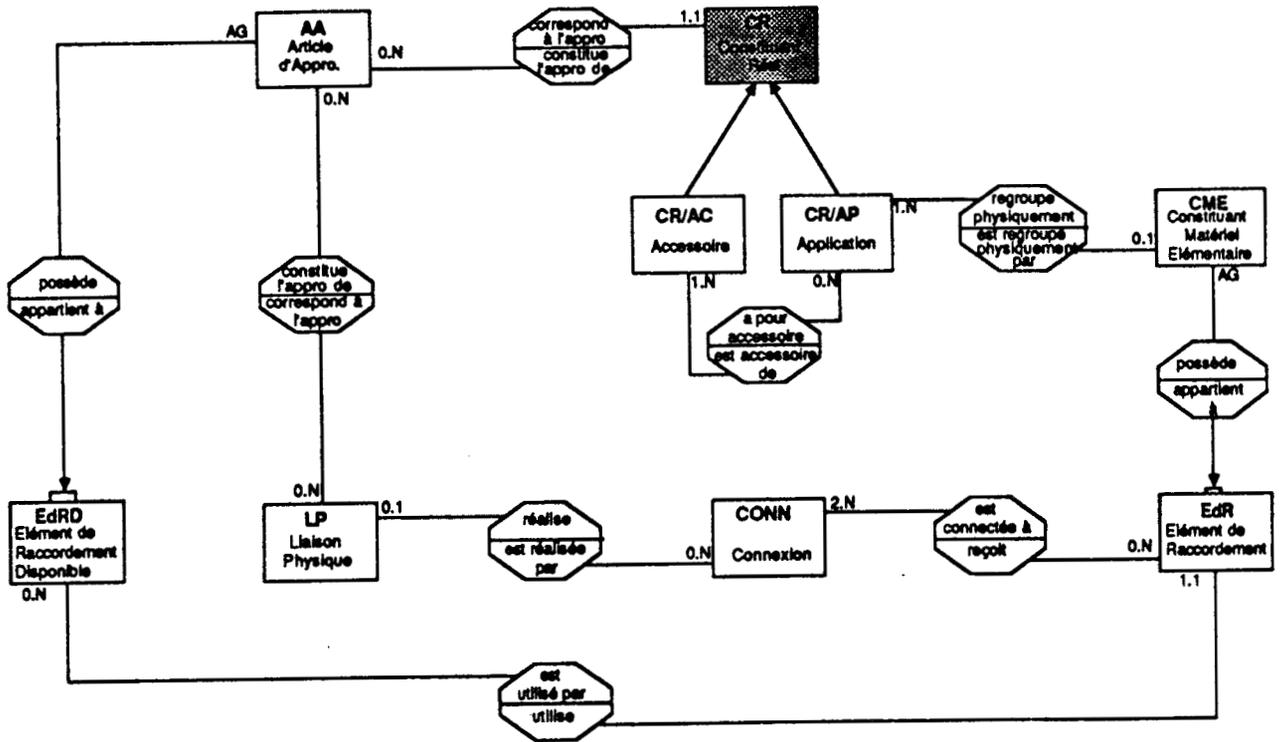


Fig 67

Il existe cependant un autre type d'article d'approvisionnement associé ni à des constituants, ni à des liaisons décrits dans l'application. Il s'agit des pièces de rechange. Ces pièces peuvent être effectivement approvisionnées afin, en cas de panne, de disposer immédiatement des rechanges. Mais il est aussi possible de ne pas les approvisionner effectivement et de simplement prévoir des approvisionnements possibles. Nous voyons ici une utilisation possible des deux associations entre AA et CLAP et nous justifions la cardinalité 0.1 au niveau de l'entité AA pour la relation d'approvisionnement effectif, une instance d'Article d'Approvisionnement n'étant pas obligatoirement associée à une instance de CLAP.

En définitive, le modèle comporte trois sous-classes de l'entité Article d'Approvisionnement: les deux premières précédemment citées et la sous-classe des articles d'approvisionnement de pièces de rechange (cf schéma de la figure 68).

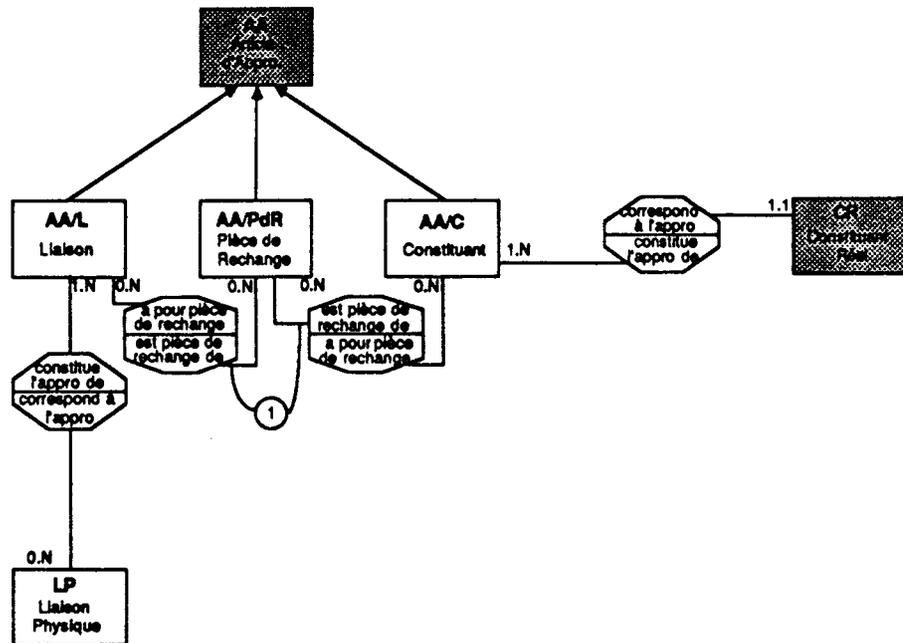


Fig 68

La contrainte de la figure 68 exprime qu'une pièce de rechange ne peut être simultanément pièce de rechange pour un article d'approvisionnement de type liaison et un article d'approvisionnement de type constituant, mais doit obligatoirement être associée à l'un ou à l'autre.

55. Caractéristiques technologiques des constituants de l'application

551. La classe fonctionnelle d'un constituant application

L'entité Article d'Approvisionnement que nous avons présenté précédemment permet de traduire la manière dont on réalise l'approvisionnement des constituants matériels.

L'établissement de nomenclatures consiste en l'association de clés d'approvisionnement aux articles d'approvisionnement. Cependant, en pratique, cette création de nomenclatures se fait dans une phase ultime de la conception. Les caractéristiques techniques des articles d'approvisionnement se déduisent immédiatement grâce aux informations données par le fournisseur lorsque les clés d'approvisionnement sont déterminées. Dans des phases préliminaires, il est essentiel de pouvoir stocker des informations technologiques sur les constituants matériels avant détermination de leur clé d'approvisionnement. Les entités que nous allons successivement introduire dans ce paragraphe 55 offrent cette possibilité.

Nous avons constaté que l'entité Constituant Matériel Application pouvait valablement être considérée comme la classe des différentes "fonctions constituant" de l'application. Il est naturel de regrouper les instances de CMA correspondant à la même nature de fonction constituant. Cette nature sera traduite grâce à un entité Classe Fonctionnelle (cf schéma de la figure 69). Chaque instance de CMA est rattachée à une et une seule instance de Classe Fonctionnelle (CF) qui représente son type de fonction constituant. Ainsi KM1, KM10 seront rattachés à la même instance de classe fonctionnelle "contacteur de puissance".



Fig 69

552. Une structure de "catalogue application" de constituants.

L'entité Classe Fonctionnelle n'est pas véritablement une entité application de par son caractère générique. Il s'agit en fait d'un objet intermédiaire entre l'application et le catalogue de constituants. En effet, si les différentes instances possibles de CF sont données

par le catalogue, leur présence dans la structure de données application ne se justifie que par la pertinence vis à vis de l'application d'une telle information. C'est pourquoi ne seront instanciées que les classes fonctionnelles identifiées dans l'application. Il ne s'agit pas de reproduire dans la base l'ensemble des classes fonctionnelles que pourrait fournir le catalogue.

Chaque classe fonctionnelle est formée d'un certain nombre de types d'article (entité Type d'Article TA). Ces types d'article sont l'image des différents types de constituant physique qui existent dans le domaine des automatismes, c'est à dire dans l'ensemble des catalogues de l'ensemble des fournisseurs de constituants d'automatismes. Ils sont le reflet d'un existant matériel et n'ont donc plus aucun aspect fonctionnel. Chaque type d'article se caractérise par des caractéristiques techniques (entité Caractéristique Technique C/T), c'est à dire par des types de propriétés technologiques significatifs pour le type d'article considéré (cf schéma de la figure 70).

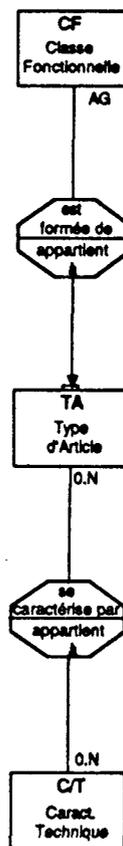


Fig 70

Comme pour la classe fonctionnelle, les entités TA et C/T sont des entités intermédiaires entre le catalogue et l'application. La chaîne CF-TA-C/T de la figure 70 peut en fait être considérée comme la structure générale d'un catalogue de constituants matériels classés par grandes fonctions. Soulignons qu'il s'agit de la structure du catalogue et non du catalogue lui-même, CF, TA et C/T étant des types (respectivement de fonction constituant, d'article matériel et de propriétés technologiques).

L'objectif de ces trois entités est de permettre des liens entre la structure de données application et le catalogue informatisé ou, plus généralement la Base Constituants. L'organisation présentée ici semble relativement conforme aux travaux en cours notamment dans le cadre des normes DIN série 4000. L'équipe technique de BASEPTA n'a donné aucune information quand aux instances possibles pour les trois entités considérées, laissant à la charge des traitements et des méthodologies qu'ils supportent le soin de les définir. Cependant une analyse fine du modèle montre deux contraintes fortes:

* la distinction est nette entre l'aspect fonctionnel du constituant catalogue (CF) et sa réalité matérielle (TA).

* le modèle n'autorise pas de structuration pour chacune des entités CF et TA, ce qui suppose de définir de manière absolue et sans aucun lien hiérarchique par exemple les instances possibles de CF.

Si la première contrainte nous semble poser aucun problème, il n'en va pas de même pour la seconde beaucoup trop restrictive. Pour certains traitements les instances possibles de l'entité CF pourraient être par exemple : Distribution, Actionneurs, Acquisition de données, Commande de puissance, Traitement des données et Dialogue Homme-machine. Pour d'autres, il sera intéressant de distinguer plus finement, au niveau par exemple de la commande de puissance, les classes fonctionnelles contacteurs, contacteurs-inverseurs, relais de protection, contacteurs auxiliaires, contacteurs disjoncteurs... Le modèle impose le choix d'un niveau de classe fonctionnelle. Or il nous semble tout à fait envisageable que, pour certains CMA, les traitements souhaitent une bonne précision sur la classe fonctionnelle et, pour d'autres, tolèrent une classe assez générale. Ceci serait facilement résolu pour une solution du type de celle de la figure 71.

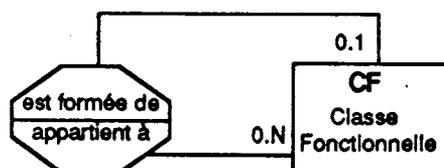


Fig 71

Ce type de structure devra être prévu pour la construction de la Base Constituants. On peut considérer qu'au niveau de la Base Application nous ne retrouvons que les instances de CF en ayant perdu le lien entre elles, ce lien figurant dans la Base Constituants. C'est la solution retenue par BASEPTA. Malgré ses inconvénients, elle est acceptable pour des entités à la "frontière" du modèle telles que CF, TA et C/T.

Illustrons les différentes entités de cet aspect structure de catalogue au travers de notre exemple. Q1 assure une fonction sectionneur. La classe fonctionnelle des sectionneurs comprend plusieurs types de sectionneurs: sectionneur porte-fusibles tripolaire, tétrapolaires... Le sectionneur porte-fusibles tripolaire sera par exemple caractérisé par une intensité thermique, une masse, la nature de la poignée de commande (latérale, frontale, extérieure). Le schéma de la figure 72 illustre les instanciations correspondantes.

Rappelons une fois encore que les instanciations doivent avoir un sens du point de vue de l'application et ne pas constituer une simple recopie du catalogue. Ainsi, la masse ne donnera lieu à une instance de C/T que si la masse est significative pour l'application considérée.

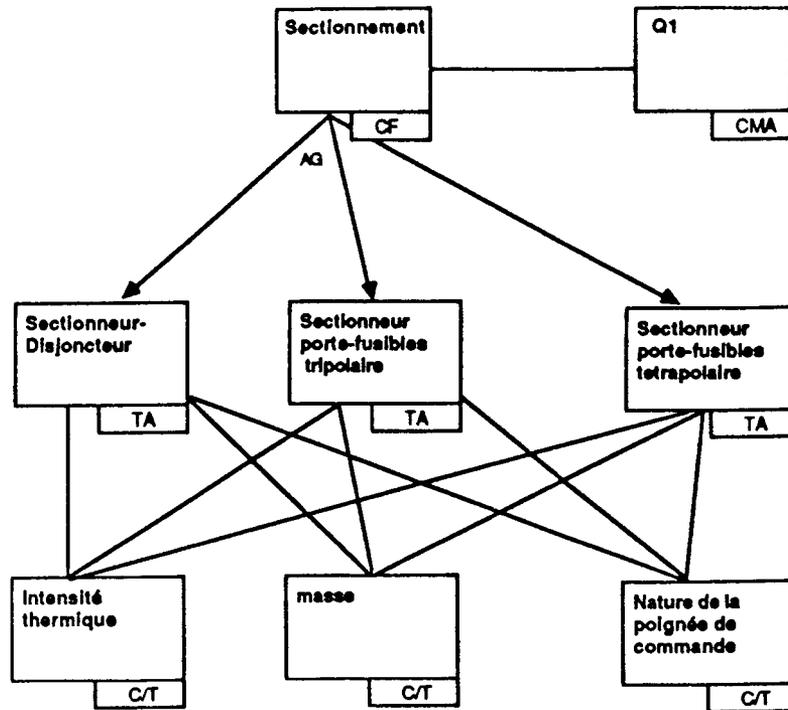


Fig 72

De même que le Constituant Matériel Application est associé à sa classe fonctionnelle, l'article d'approvisionnement est rattaché à son type d'article dans le but de pouvoir lui associer des données techniques. Une entité Donnée Technique (D/T) a été créée. Elle est rattachée nécessairement à une caractéristique technique dont elle constitue une valeur, mais également à l'article d'approvisionnement pour lequel la caractéristique technique prend cette valeur (cf schéma de la figure 73).

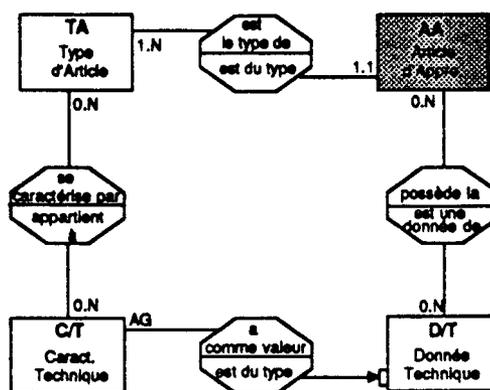


Fig 73

Associer des données techniques à un article d'approvisionnement aurait pu se faire grâce à des attributs de l'entité AA. Mais cela aurait supposé de connaître le nombre et la nature des caractéristiques techniques de l'article d'approvisionnement au moment de la définition du modèle. Or nombre et nature dépendent du type d'article d'approvisionnement. La solution retenue permet de manière très simple d'associer à un article d'approvisionnement autant de données techniques que l'on désire à condition que ces données soient significatives en regard du type d'article.

Dans le cas du sectionneur de notre exemple, la figure 74 montre la simplicité de mise en oeuvre de ces associations.

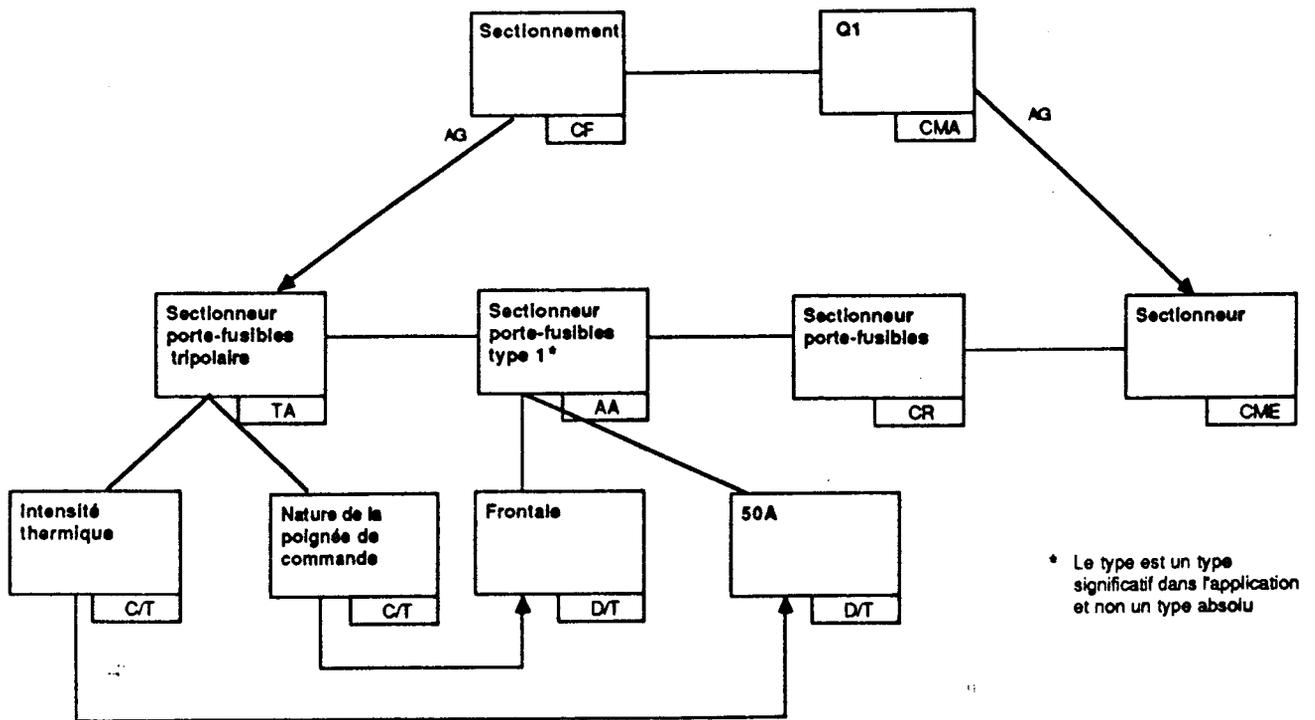


Fig 74

553. Caractéristiques techniques de bornes.

Il est courant d'associer à des bornes d'un constituant matériel approvisionné des données techniques. Les entités présentées ci-dessus permettent de le faire en considérant ces données techniques de bornes comme des données de connectivité de l'article d'approvisionnement qui les possède. Cette solution présente l'inconvénient de ne pas faire apparaître clairement la distinction entre les données se rapportant globalement au constituant et celles spécifiques à ses bornes.

Nous avons donc introduit une entité Type d'Élément de Raccordement TEdR caractérisant un élément de raccordement disponible d'un article d'approvisionnement et caractérisé par des caractéristiques techniques. Associer des données techniques à des bornes d'un article d'approvisionnement se fait d'une manière tout à fait analogue à l'association de données techniques globales à un article d'approvisionnement (cf schéma de la figure 75).

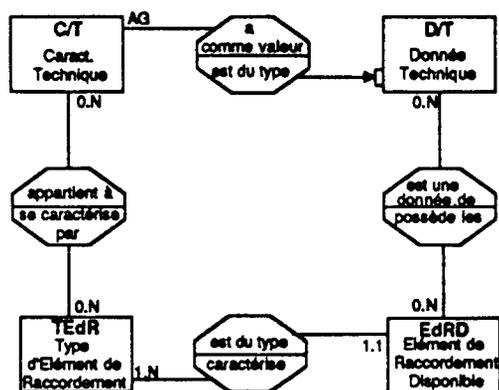


Fig 75

Comme précédemment, une donnée technique de borne doit être simultanément associée à une instance de C/T et une instance d'EdRD. La contrainte est explicitée sur le schéma récapitulatif de la figure 76. Elle s'interprète de

manière simple: toute instance de C/T (respectivement D/T) doit être associée, soit à une instance de TA (respectivement AA), soit à une instance de TEdR (respectivement EdRD), mais pas aux deux simultanément.

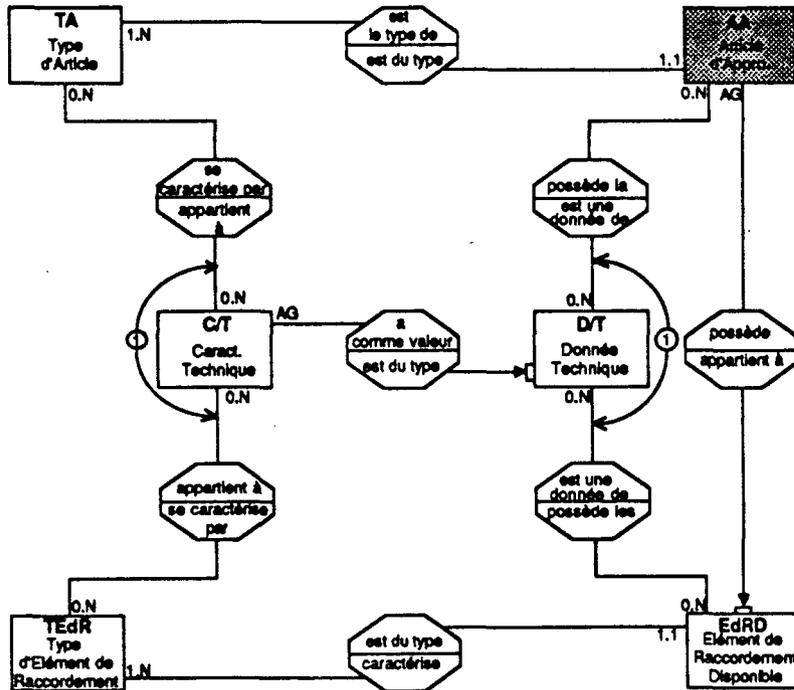


Fig 76

Remarquons pour terminer la notation des entités C/T et D/T qui indique que ces deux entités sont des sous-classes d'entités Caractéristique C et Donnée D que nous étudierons en section II-9.

56. Conclusion: vue générale de la fourniture matérielle

Le schéma de la figure 77 récapitule l'ensemble des entités et associations assurant la description de la fourniture matérielle.

PARTIE II - MODELE DE STRUCTURE DE DONNEES APPLICATION

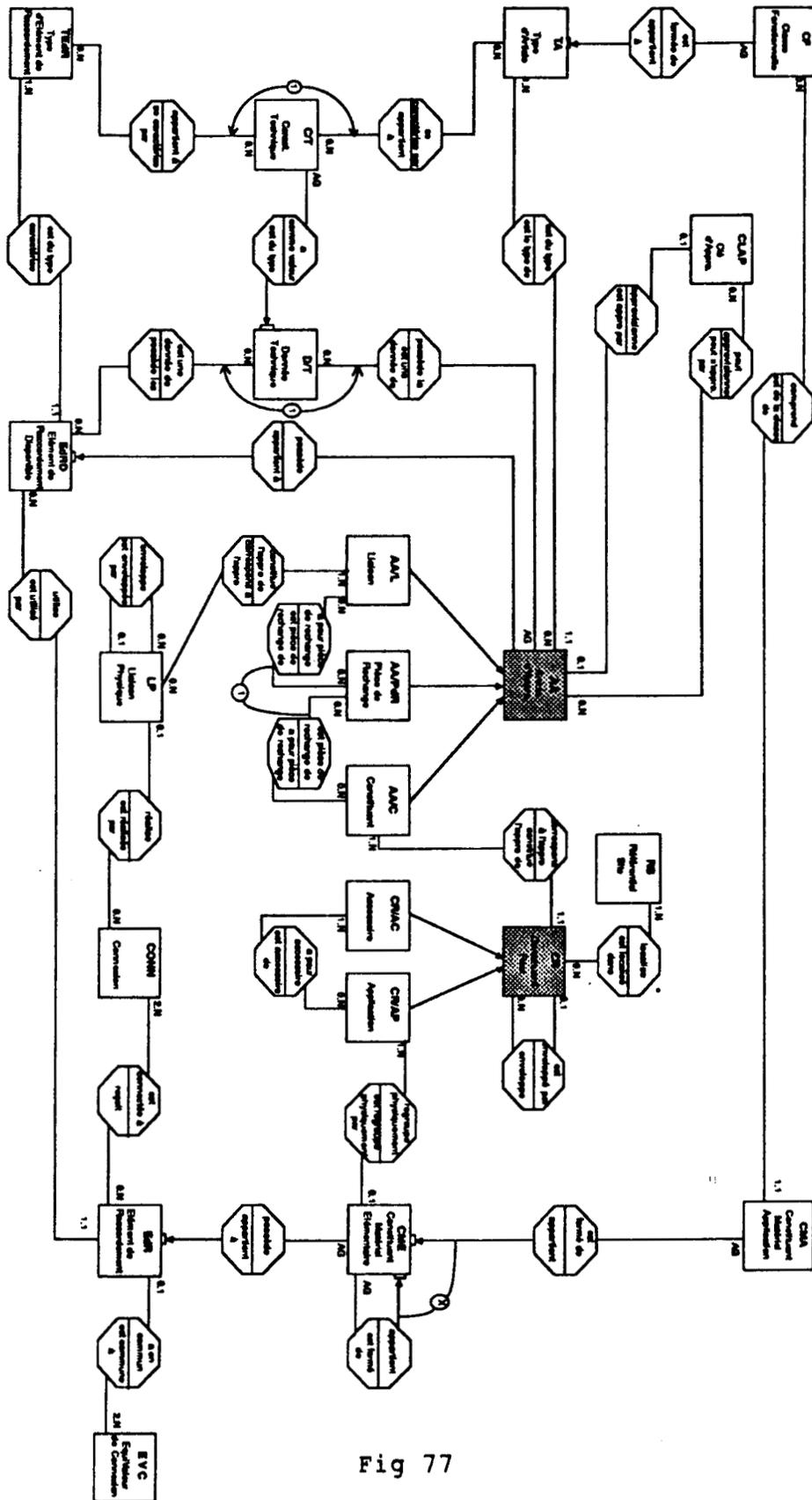


Fig 77

Ce schéma peut sembler complexe à utiliser pour les traitements identifiés aujourd'hui. Prenons un exemple. Les utilisateurs de CAO électromécanique ont coutume d'associer des informations techniques sur les symboles du schéma de principe qui, soit dit en passant, cesse alors d'être un véritable schéma de principe. Ainsi, sous le symbole du moteur triphasé, on indiquera P=1.5KW. Nous avons vu que le symbole du schéma de principe était lié au CME. Pour pouvoir saisir cette information sur la puissance du moteur, il faudra créer une instance de CR, une instance d'AA, rattacher cette instance d'AA à une instance de TA possédant une caractéristique technique "puissance nominale" pour laquelle on donnera une valeur 1.5 KW pour l'instance d'AA considérée. Ceci peut sembler fastidieux. Cependant, il ne faut pas oublier que le modèle permet de stocker l'état de l'application à un instant donné. Par la suite, les instances d'AA et de CR créées peuvent être remises en question, modifiées ou affinées. D'autre part, cet éclatement des données est la condition de performance des traitements de CAO travaillant sur la Base Application. En analysant finement l'exemple, nous constatons que la puissance associée au constituant élémentaire représenté par le symbole, est en réalité associée à l'article d'approvisionnement qui sera l'approvisionnement du constituant réel réalisant la fonction élémentaire moteur. Le modèle traduit donc la réalité au mieux, dans toute sa subtilité.

Pour conclure cette présentation de la fourniture matérielle, nous résumons dans le schéma de la figure 78 les divers aspects que nous avons envisagés pour les constituants.

Aspect du constituant	Entités du modèle	Exemples de traitements s'appliquant typiquement aux différents aspects du modèle	Exemples de documents dont la sémantique peut être traduite par les différents aspects du modèle
Aspect fonctionnel (Fonction constituant)	CMA	Détermination des fonctions constituants	Légende fonctionnelle
Aspect logique (Sous-fonction)	CME EDR EVC	Description des principes de fonctionnement de l'assemblage matériel	Schéma de principe
Aspect réel ou physique	CONN CR-CR/AC-CR/AP LP RS	Regroupement des fonctions dans les constituants matériels Implantation en armoire Gestion des câbles	Plan d'implantation Liste de montage Liste de câblage Schéma de câblage
Aspect approvisionnement	AA EdRD C/T-D/T CLAP	Choix de matériel en fonction de caractéristiques et données techniques	Nomenclature

Fig 78

6. LA FOURNITURE. LES DONNEES DECRIVANT LE LOGICIEL

61. Introduction

L'une des caractéristiques essentielles des applications automatisées est, comme nous l'avons vu, l'intégration logiciel - matériel. Dans cette section II-6, nous nous proposons donc de compléter la présentation des données décrivant la fourniture par l'aspect logiciel.

Aujourd'hui, il est relativement aisé d'obtenir un consensus pour départager ce qui est logiciel de ce qui est matériel. Cependant, il faut noter que l'intégration de ces deux aspects est de plus en plus étroite. Les constituants matériels d'automatismes intègrent très souvent une partie plus ou moins importante de logiciel. Le réglage des constituants consiste souvent en configuration ou paramétrage de ce dernier.

De plus, si l'accord se fait pour déterminer ce qui est "logiciel", cet accord disparaît lorsqu'il s'agit de définir ce qu'est "LE logiciel". La définition de chacun est conditionné par ses préoccupations, les outils dont il dispose et le recul méthodologique qu'il a acquis. Le programmeur des premiers automates, qui programmait directement en utilisant le code exécutable, ne risquait pas de confusion. Mais qu'en est-il de l'utilisateur d'une console moderne? Doit-il considérer qu'il dispose de deux niveaux de logiciel: le "littéral de l'automate" et le "code automate"? Cette classification relève d'une culture informatique dans le cadre de laquelle les choses sont claires et rassurantes: Le source c'est le logiciel qui sera "compilé" puis "lié" en exécutable. L'automaticien n'a pas à prendre en compte de telles considérations. En pratique, sur une console moderne, il ignore le plus souvent si le programme est compilé, interprété, semi-compilé ou semi-interprété. Ainsi, dans le cas d'un outil de type GRAL, le logiciel est-il constitué par: le code automate, le langage littéral de l'automate cible généré par l'outil GRAL, le code intermédiaire qu'il utilise ou la description du programme saisie par l'utilisateur? Remarquons d'ailleurs que cette description n'est pas une description fonctionnelle de la commande mais une description fonctionnelle du logiciel. Une description fonctionnelle de la commande ne doit pas prendre en compte le seul aspect logiciel mais l'ensemble de la commande.

En définitive, il apparait clairement qu'il n'est pas possible de donner une définition absolue du logiciel d'une application automatisée.

Pour éclairer cependant notre propos, nous proposons une classification des langages d'automatismes en trois catégories (cf schéma de la figure 79):

Le "langage de description de la commande" permet d'exprimer fonctionnellement la commande dans son intégralité, logicielle et matérielle. Il prend en compte des choix de conception mais pas de choix de réalisation programmée ou câblée.

Lorsque le choix d'une réalisation programmée a été fait, nous trouvons une expression utilisant le "langage de description du logiciel". Ce second langage peut être indépendant de la machine cible (cas du langage normalisé CEI ou de langages intermédiaires type GRAL) ou, au contraire, très lié à l'automate d'implantation (cas des langages de console constructeur). Il existe bien évidemment divers niveaux entre ces deux extrêmes, la tendance actuelle étant de s'éloigner le plus possible des contraintes de la machine afin que celles-ci soient maîtrisées non par le programmeur, mais par la CAO qui produit le logiciel. Faciliter la conception des programmes et améliorer leur portabilité et leur fiabilité sont bien sûr les objectifs d'une telle évolution.

Le "langage d'exécution" constitue le niveau le plus bas. Il est devenu aujourd'hui quasiment inaccessible à l'automaticien et reste à juste titre le domaine du constructeur d'automate.

Nous avons raisonné implicitement sur le cas de l'automate programmable, mais les trois niveaux exposés restent valables pour toute machine programmable tel un PC industriel.

Notre classification sommaire va nous permettre de préciser les données que nous allons décrire dans la partie fourniture logicielle de la Base Application. Il s'agit du programme exprimé dans le langage de description du logiciel. Lorsqu'il y a plusieurs expressions possibles correspondant à des langages de description de logiciel de niveaux différents, l'utilisateur de la Base Application (utilisateur de la Base doit être pris au sens large,

c'est à dire traitement s'appliquant sur la Base) choisira le niveau le plus pertinent. La partie du modèle "Données Annexes" (cf section II-9) permettra de conserver d'autres expressions.

Le logiciel exprimé en langage d'exécution ne sera pas dans le cas général stocké dans la Base Application. Quant à l'expression de la commande logicielle et matérielle, nous verrons en section II-8 comment les "Données de Structuration" permettent de la conserver.

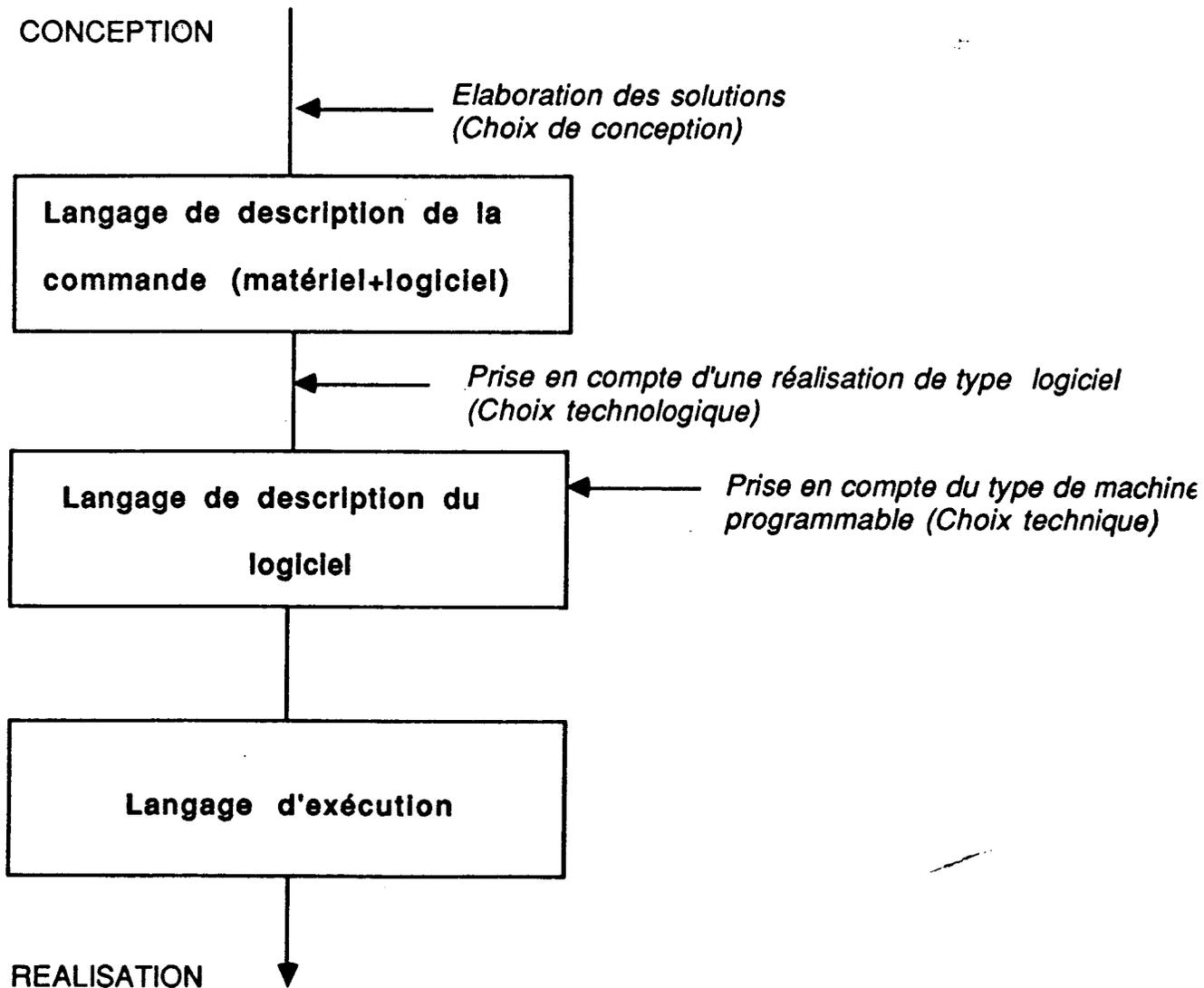


Fig 79

62. La configuration du logiciel

La configuration du logiciel application consiste la plupart du temps à déclarer "utilisable" un certain nombre de ressources que le logiciel d'exécution de la machine cible met à la disposition du logiciel application (variables internes, blocs fonctions, zone mémoire programme, zone mémoire données...). Afin de permettre une gestion optimale des ressources, la presque totalité des machines cibles du marché est configurable. C'est effectivement le cas des automates. Cependant, dans une optique future où le logiciel en tant que tel n'apparaîtrait pas à un concepteur qui se contenterait de travailler sur une description de la commande, la configuration optimale pourrait être choisie automatiquement et de façon transparente. Elle deviendrait alors "interne" et ne donnerait pas lieu à des données dans la Base Application.

Les possibilités de configuration logicielle dépendent de la machine cible. Pour un automate ou un PC industriel, seuls certains objets logiciels peuvent être configurés. L'entité Caractéristique de Configuration (C/C) permet de stocker les objets configurables (cf schéma de la figure 80).

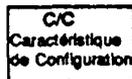


Fig 80

C'est ainsi que, par exemple, lorsque le nombre de blocs de temporisation est configurable dans l'automate, nous pouvons introduire une instance de C/C dont l'identifiant est "nombre de blocs de temporisation".

L'entité C/C est, comme nous le verrons en section II-9, une sous classe d'une classe générale Caractéristique (C).

La configuration logicielle d'une application consiste donc à donner des valeurs aux différentes caractéristiques de configuration, c'est à dire à associer des données aux diverses instances de C/C. Le modèle comporte donc une entité Donnée de Configuration (D/Conf), sous-classe de la classe générale des Données (D) et liée à l'entité C/C par une relation d'agrégation (cf schéma de la figure 81). Ce mécanisme est en tout point identique à celui présenté dans la section II-5 pour les caractéristiques et données techniques du matériel. Dans un souci d'homogénéité, il conviendrait d'ailleurs de remplacer le mnémonique C/C par un mnémonique C/Conf.

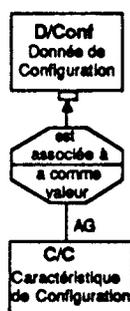


Fig 81

Comme nous l'avons expliqué, l'entité C/C est liée à la machine cible ou plus exactement au logiciel d'exploitation de cette machine cible. Il faudrait donc lier cette entité à l'aspect matériel, ce qui n'est pas fait dans le modèle BASEPTA. C'est à la charge des traitements de vérifier la cohérence des instanciations de C/C avec le constituant programmable.

63. La ressource système utilisée

La configuration logicielle réalisée, le concepteur du logiciel application dispose de ressources système logicielles "utilisables". L'entité Ressource Système Utilisée permet de stocker les ressources système

effectivement utilisées. Elle est donc associée à l'entité D/Conf, cette association traduisant le fait que toute ressource système utilisée doit avoir été configurée au préalable.

Du point de vue sémantique, une ressource système utilisée doit être cohérente avec le couple Caractéristique - Donnée de configuration. La relation d'agrégation entre C/C et D/Conf permet de traiter cette cohérence par une simple association RSU - D/Conf, toute instance de l'entité D/Conf étant, rappelons le, identifiée par rapport à une instance de C/C.

Nous avons identifié (cf schéma de la figure 82) trois types de ressource système utilisée induisant trois sous-classes de l'entité RSU.

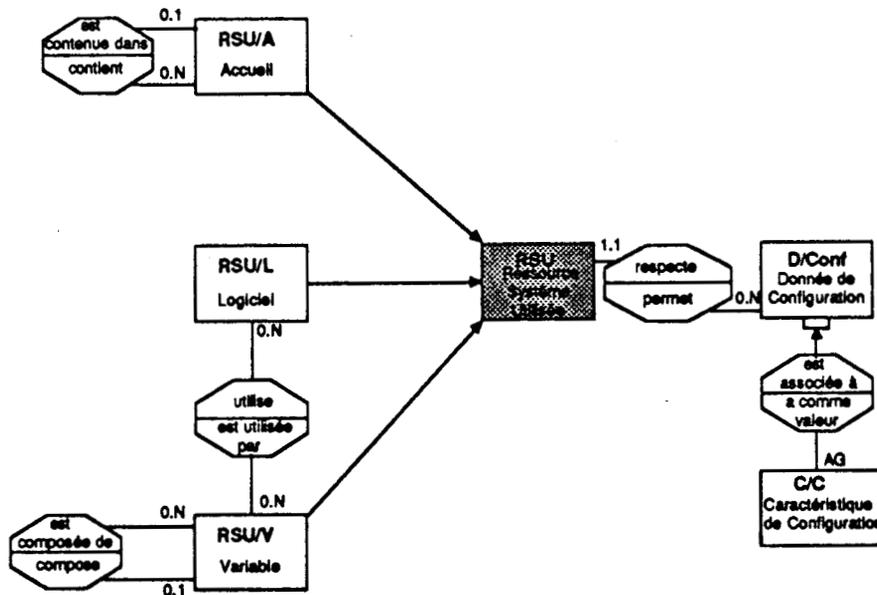


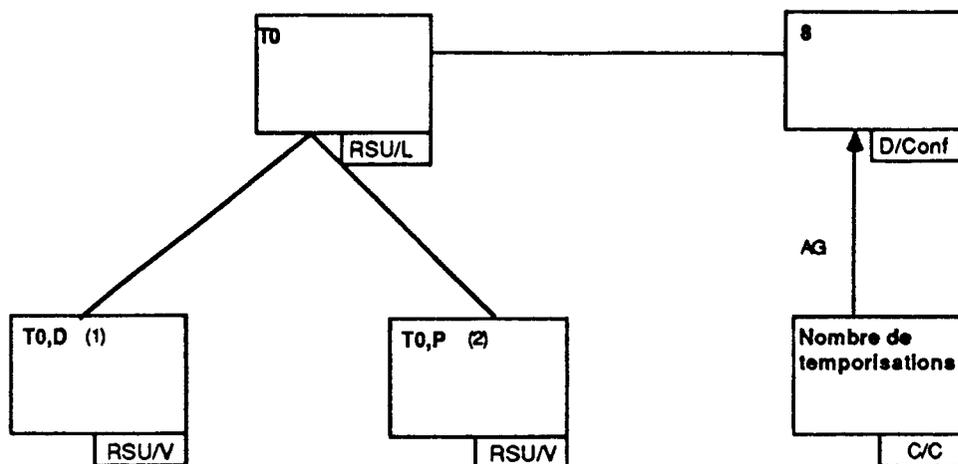
Fig 82

La sous-classe des ressources système utilisées de type variable (RSU/V) est la plus simple. Illustrons la sur un exemple: Sur la console T607 de TELEMECANIQUE, l'utilisation d'une information binaire, par exemple celle dont le nom système est I1,8, donne lieu à une instance de

RSU/V. Cette utilisation n'est possible que si les entrées automates ont été configurées c'est à dire, dans notre exemple, si en phase de configuration il a été indiqué qu'à l'emplacement 1 du rack 0 se trouvait une carte d'entrée tout ou rien. Nous voyons donc ici le lien entre RSU/V et configuration. De manière simplifiée, une instance de RSU/V correspond au "nom système" d'une variable utilisée dans le logiciel. Le modèle supporte une relation de "composition" entre les instances de RSU/V. Ainsi le "mot d'état système" peut être composé de "bits d'états systèmes" manipulés séparément.

La deuxième sous-classe de l'entité RSU décrit les ressources système utilisées de type logiciel (RSU/L). Il s'agit en fait d'éléments de logiciel disponibles sur la machine cible et utilisés dans le logiciel application. On peut généralement les qualifier de "blocs fonctions système". Temporisateurs, monostables, compteurs en sont des exemples simples. Mais, de façon générale, tout bloc fonction fourni par le système même évolué et paramétrable, tel un bloc fonction régulation, peut donner lieu à une instance de RSU/L s'il est effectivement utilisé dans le logiciel application. Une instance de RSU/L utilise généralement des instances de RSU/V. Ainsi la temporisation T0 pourra utiliser les informations temporisation écoulée, en cours (respectivement T0,D et T0,R dans PL7-3). Remarquons ici que la cardinalité 1.1 entre RSU et D/Conf est trop restrictive. 0.1 conviendrait mieux. En effet, comme nous le voyons sur la figure 83, si l'instance de RSU/L est mise en correspondance avec la configuration, les instances de RSU/V qu'elle utilise le sont de fait. Il n'est donc pas nécessaire que le modèle contraigne une association entre RSU et D/Conf. Cela peut même entraîner des incohérences.

La troisième et dernière sous-classe de l'entité RSU est difficile à comprendre dans une optique informatique classique, mais apparait assez nettement dans le cadre d'environnement de programmation proposé par des machines type automate. Elle correspond aux Ressources Système Utilisées de type Accueil (RSU/A), c'est à dire aux structures d'accueil du logiciel application. Nous pourrions dire de manière simpliste que les instances de RSU/A correspondent à l'identification des "endroits" où se trouve le logiciel application. En réalité, il ne s'agit pas uniquement d'une localisation. Chaque instance de RSU/A a un contexte d'activation et d'exécution et recouvre donc une notion plus riche.



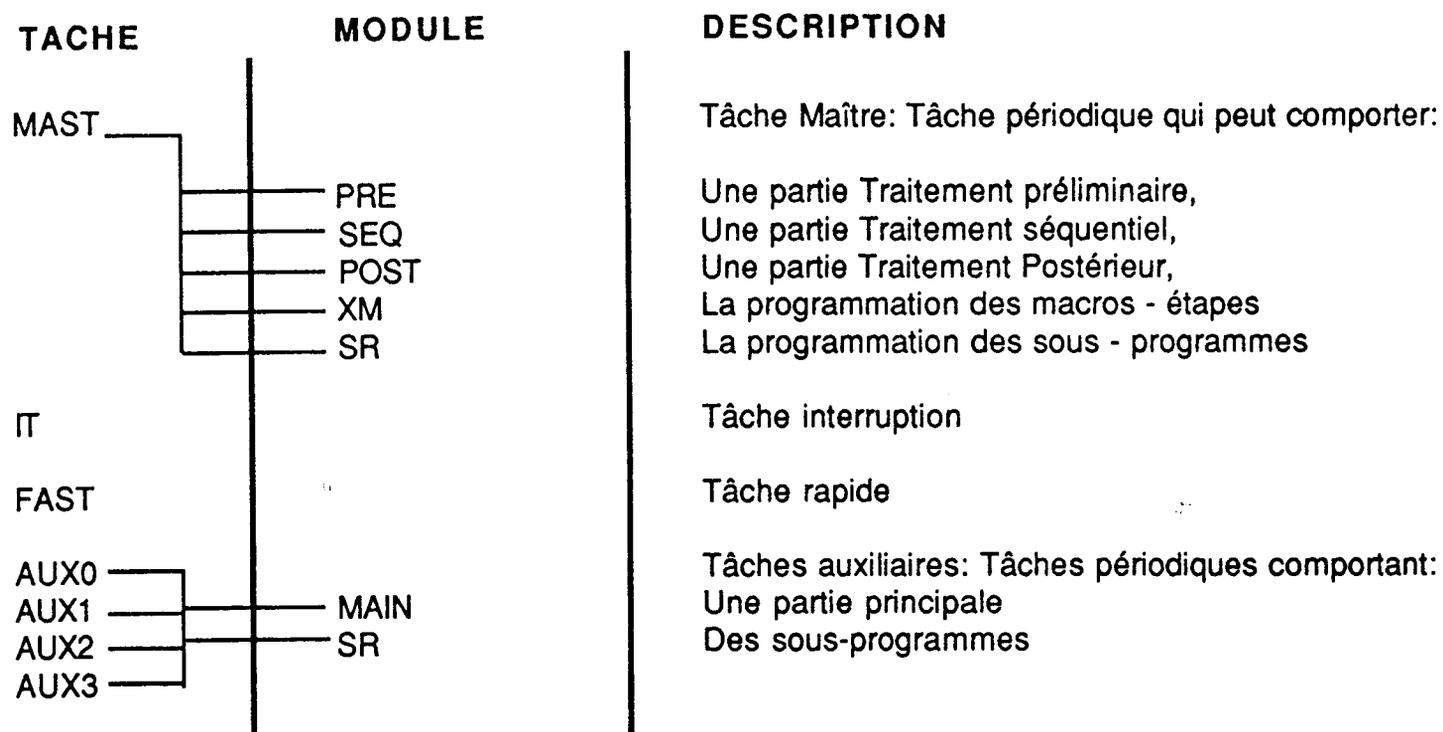
- (1) Indicateur de Temporisation écoulée
 (2) Valeur de pré-sélection

Nota: Si l'information Temporisation T0 en cours n'est pas manipulée dans le logiciel, il n'y a pas lieu de créer une instance de RSU/V pour cette information (T0,R)

Fig 83

Le schéma de la figure 84 montre les différentes ressources système de type accueil proposées par le logiciel PL7-3 (Grafset PL7-3). Elles donneront lieu à des instances de RSU/A si elles sont effectivement utilisées dans la réalisation du logiciel application. Elles sont de deux natures: les tâches et les modules. Un module est inclus dans une tâche. L'association sur l'entité RSU/A (cf figure 82) permettra de traduire cette information.

Comme nous venons de le voir, la ressource système utilisée de type accueil est bien adaptée à la problématique de l'automate. Elle s'utilise cependant aussi dans le cas d'une machine classique avec "un seul programme". Il n'existe alors qu'une seule instance de RSU/A correspondant à la structure d'accueil du programme.



64. L'aspect bibliothèque de logiciels

Nous avons envisagé, grâce à la ressource système utilisée de type logiciel, un aspect "bibliothèque système". Le concepteur du logiciel application dispose également la plupart du temps de bibliothèques dites "utilisateur". Il est donc nécessaire de stocker dans la base application le lien entre le logiciel application et les bibliothèques ayant contribué à sa réalisation et par voie de conséquence un minimum d'informations pertinentes sur ces bibliothèques. Notre modèle comporte donc des entités assez analogues à celles ayant trait au catalogue "application" des constituants matériels (cf II-55). Ces entités se rapportent à des catalogues des logiciels (cf schéma de la figure 85).

L'entité Bibliothèque (B) permet de stocker les identifiants de bibliothèques de logiciel dont le logiciel application utilise des éléments. Ces éléments donneront lieu à des instances d'une entité Élément de Logiciel de Bibliothèque (EDLB). Chaque élément de logiciel de bibliothèque manipule des variables logicielles. Ces

variables logicielles de bibliothèque sont qualifiées de "génériques" car elles donnent lieu à des instances dans le logiciel application. L'entité Variable Logicielle Générique (VLG) permet de les stocker. L'association entre les entités EDLB et VLG porte un attribut identifiant le mode d'accès de la variable logicielle générique dans l'élément de logiciel de bibliothèque. Ainsi, nous pourrions préciser s'il s'agit d'une variable de sortie ou d'entrée pour l'instance de l'EDLB considérée.

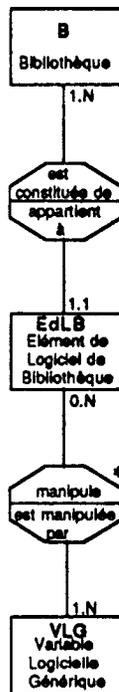


Fig 85

Précisons, une fois encore, que les instances des trois entités B, EDLB et VLG ne se justifient que par leur pertinence vis à vis de l'application. Ces trois entités, comme celle du catalogue application des constituants matériels, décrivent des objets intermédiaires entre l'application et les catalogues de logiciels. Elles constituent le squelette d'un modèle d'une Base de données de logiciels de bibliothèques. Il serait souhaitable de préciser et d'étendre l'analogie entre ces deux types de "catalogue application" dans le but d'obtenir une structure commune pour définir une Base Constituants,

constituants étant pris au sens large: matériels, logiciels, matériels+logiciels. Ceci ne concerne pas la Base Application mais constitue un point de départ pour une réflexion sur le modèle de la Base Constituants dont nous avons vu, en partie I, l'importance au sein du PTA.

Nous pouvons également remarquer l'analogie formelle entre la chaîne RSU/L - RSU/V d'une part et EDLB - VLG d'autre part, analogie qui souligne si besoin était que la distinction bibliothèque système - bibliothèque utilisateur est essentiellement une perception utilisateur et non une différence de fond absolue.

65. Le logiciel application

651. La structuration utilisateur du logiciel

Dans cette section II-6, nous avons jusqu'à présent introduit des données relativement secondaires: configuration, liens avec les bibliothèques... Nous allons à présent aborder la partie centrale de la fourniture logicielle et présenter les données concernant le ou les programmes application proprement dits.

Le logiciel application est structuré, même si cette structure se réduit à un seul élément: le programme dans sa totalité. Les éléments de cette structure sont décrits grâce à l'entité Élément de Logiciel (EDL) (cf schéma de la figure 86). L'organisation en éléments de logiciel est le résultat d'une découpe arbitraire issue de la conception logicielle. L'ensemble des instances de l'entité EDL constitue une partition au sens mathématique du logiciel application. Il n'y a donc pas de recouvrement possible entre éléments de logiciel, pas "d'inclusion" d'une instance d'EDL dans une autre. Cette limitation peut sembler contraignante mais nous verrons plus loin (cf II - 67) qu'il est possible de la surmonter.

Le logiciel application comprend également des variables logicielles manipulées par les différents éléments de logiciel. L'entité Variable Logicielle (VL) permet de les stocker. La vision du logiciel application exprimée dans la Base Application étant une vision utilisateur, il en

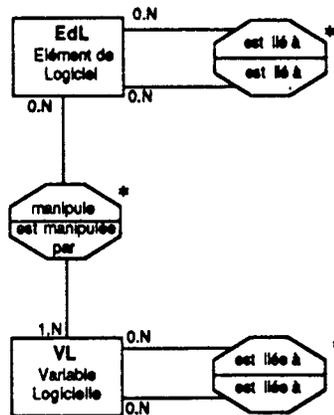


Fig 86

résulte que la notion de variable logicielle est une notion utilisateur. Il ne s'agit donc pas d'une variable au sens ressource du logiciel, mais d'une désignation utilisateur d'un objet du logiciel. Autrement dit, la même ressource variable peut être manipulée sous deux noms différents dans le même logiciel (variables équivalentes). Les deux noms donnent lieu à deux instances différentes de l'entité Variable Logicielle. Bien évidemment, la relation d'équivalence entre ces noms devra être traduite. C'est ce que permet, par exemple, l'association sur l'entité VL (cf schéma de la figure 86). Mais cette association est utilisable pour différentes natures de relations entre variables logicielles. Par exemple, elle permettra le lien entre le paramètre formel et le paramètre effectif. La diversité des relations entre variables logicielles a conduit à une association aux libellés très peu précis "est lié à". Cette association porte un attribut indiquant la nature du lien entre instances de l'entité VL.

L'association sur l'entité EDL est tout à fait analogue et possède également un attribut précisant le lien entre instances d'EDL (par exemple appel). L'utilisation de cette relation pour traduire des dépendances entre éléments de logiciel ne constitue pas le cas général. Il est en effet plus courant de considérer les éléments de logiciels comme liés par les variables logicielles qu'ils partagent. Pour un des éléments de logiciel, les variables

seront des variables de sortie, pour l'autre des variables d'entrée. Cette nature (entrée ou sortie) est traduite au niveau du modèle grâce à l'attribut sur l'association entre EDL et VL précisant le mode d'accès de la variable par l'élément de logiciel.

A titre d'exemple, nous reprenons sur la figure 87 un extrait du logiciel de notre exemple et nous montrons sur la figure 88 quelques instanciations d'EDL et VL auxquelles il donne lieu.

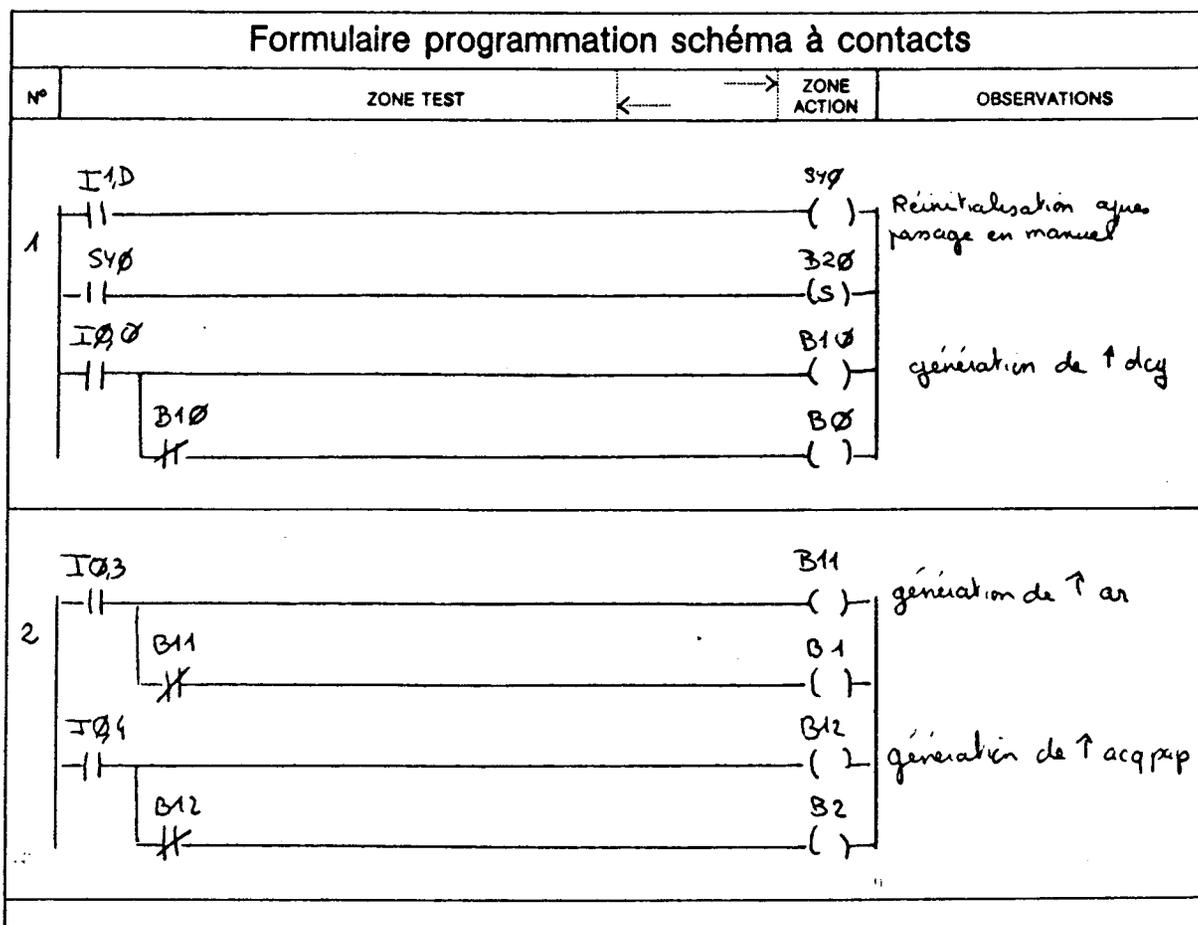


Fig 87

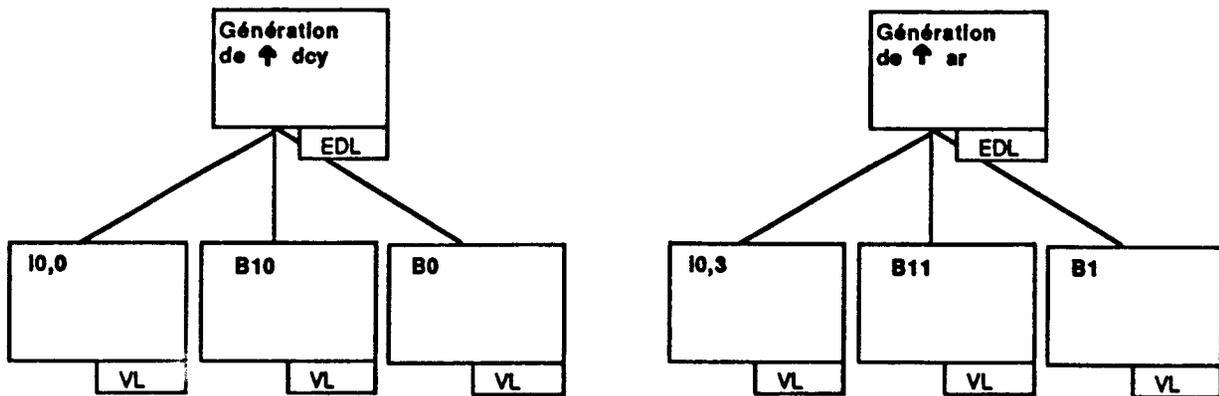


Fig 88

652. Réalisation d'un élément de logiciel

Les instances de l'entité EDL identifient des parties du logiciel mais en elles-même ne portent pas le contenu de ces parties de logiciel. L'existence d'un corps pour un élément de logiciel, c'est à dire en pratique de "lignes" de programme accessibles à l'utilisateur n'est pas obligatoire. Nous allons en effet montrer, dans ce paragraphe, qu'il existe plusieurs façons de réaliser un élément de logiciel. La terminologie "écrire" un élément de logiciel est proscrite à dessein.

Comme nous venons de le voir, la première et plus

naturelle façon de réaliser un élément de logiciel est de lui associer un corps, c'est à dire un ensemble d'instructions ou plus généralement de lignes de programme. L'entité Corps d'Elément de Logiciel (CORPS/EDL) sous classe générale de l'entité Corps que nous étudierons en section II-9 permet de stocker cet ensemble d'instructions (cf schéma de la figure 89).

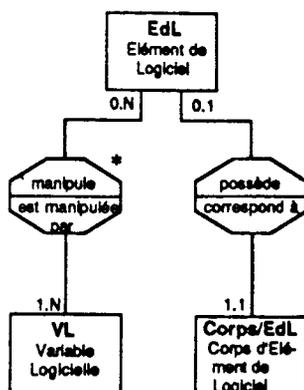


Fig 89

Remarquons le caractère "déporté" de l'entité CORPS/EDL qui n'est liée qu'à la seule entité EDL. En effet, si une instance de corps d'élément de logiciel possède un identifiant purement interne à la base de données, son identifiant application (ou utilisateur) est l'instance d'EDL dont elle est le corps. Il apparaît alors logique d'établir les liens entre les objets de la base et l'identifiant du contenu (l'EDL) plutôt qu'entre la base et le contenu lui-même (le CORPS/EDL). L'existence du corps d'élément de logiciel ne se justifie que par la seule nécessité de disposer dans la Base Application d'une description intégrale du logiciel application. Ainsi, si le modèle BASEPTA doit servir à la définition d'un standard d'échange, il est essentiel que les données échangées permettent, aux deux extrémités de la liaison, la reconstitution complète du logiciel.

Remarquons également la cardinalité 1.1 pour l'entité CORPS/EDL dans l'association EDL - CORPS/EDL. Il n'est possible d'associer qu'un seul corps à un élément de logiciel donné, c'est à dire pratiquement qu'une seule version de logiciel. Si l'utilisateur désire conserver d'autres versions dans la Base Application, il utilisera les facilités offertes par les données annexes (cf section II-9).

La deuxième façon de réaliser un élément de logiciel est donnée par le schéma de la figure 90.

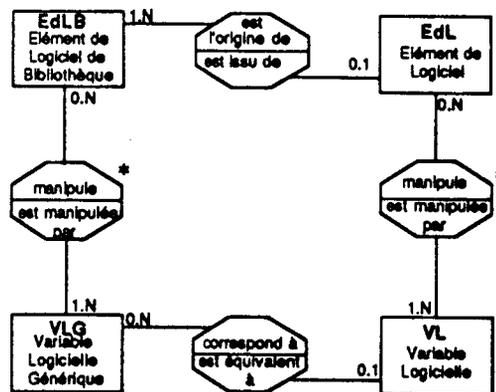


Fig 90

Elle consiste à utiliser un élément de logiciel d'une bibliothèque quelconque. Notons immédiatement que c'est le seul cas entraînant la création dans la base d'instances d'EDLB (cardinalités 1.N pour l'entité EDLB dans l'association EDLB - EDL). Il apparaît immédiatement que l'utilisation d'un élément de logiciel de bibliothèque peut être très variable. Il peut s'agir d'un appel avec passage de paramètres ou même d'une simple recopie d'un source dans le logiciel application (cf instruction INCLUDE de certains langages). Le modèle présenté ne prend pas en compte directement de telles finesses. Il se contente de traduire un lien de filiation entre le logiciel application et des éléments de bibliothèque. Il n'est d'ailleurs pas interdit d'avoir simultanément existence d'un corps et lien avec un élément de logiciel de bibliothèque.

Cependant, si l'élément de logiciel de bibliothèque manipule des variables logicielles génériques, il est nécessaire pour tout élément de logiciel application qui en est issu de conserver le lien entre ces variables génériques et les variables instanciées. Ceci permet de faire "hériter" des propriétés des variables de bibliothèque aux variables application. C'est l'intérêt de l'association VLG - VL.

La troisième possibilité pour réaliser un élément de logiciel application est l'utilisation d'un élément de logiciel fourni par le système de la machine d'accueil, c'est à dire une ressource système utilisée de type logiciel. Par exemple, le concepteur du logiciel peut avoir créé un élément de logiciel "Gestion de temporisation" et le réaliser en utilisant une ressource système de type temporisation. Il aurait pu tout aussi bien utiliser un élément de bibliothèque général assurant une gestion de temporisation ou même à l'extrême limite l'écrire lui-même.

Le modèle permet donc de relier les entités EDL et RSU/L. Mais cette liaison ne se fait pas par une association directe. Elle fait intervenir une entité intermédiaire Description d'un élément de logiciel (DESL/EDL). Le schéma résultant est celui de la figure 91.

L'entité DESL/EDL permet une plus grande précision sur le lien entre EDL et RSU/L. Un mécanisme analogue aurait pu être mis en oeuvre pour le lien entre EDL et EDLB. Le modèle BASEPTA ne l'a pas jugé nécessaire dans un premier temps, mais il nous semble fondamental de le développer dans des versions ultérieures.

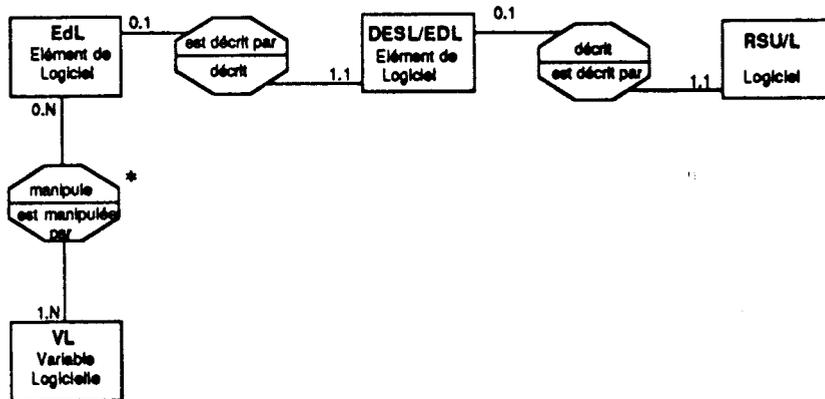


Fig 91

Toute ressource système utilisée de type logiciel possède un ensemble de caractéristiques pertinentes du point de vue application. Ainsi, pour un bloc temporisateur, la base de temps pourra être considérée. Les caractéristiques d'une instance de RSU/L seront regroupées dans un ensemble constituant une instance de l'entité DESL/EDL. C'est en fonction de cet ensemble de caractéristiques que l'on pourra attribuer à un élément de logiciel une ressource système ou inversement. Ce mécanisme est conçu pour améliorer la performance des traitements. Mais il présente un autre avantage. En effet, il est difficile de considérer que les caractéristiques sont plus attachées à l'élément de logiciel qu'à la ressource système et réciproquement. En les situant dans une entité intermédiaire le problème du choix est évité.

Le schéma de la figure 92 résume les entités et les associations que nous avons jusqu'ici présentées pour décrire un logiciel application et sa configuration.

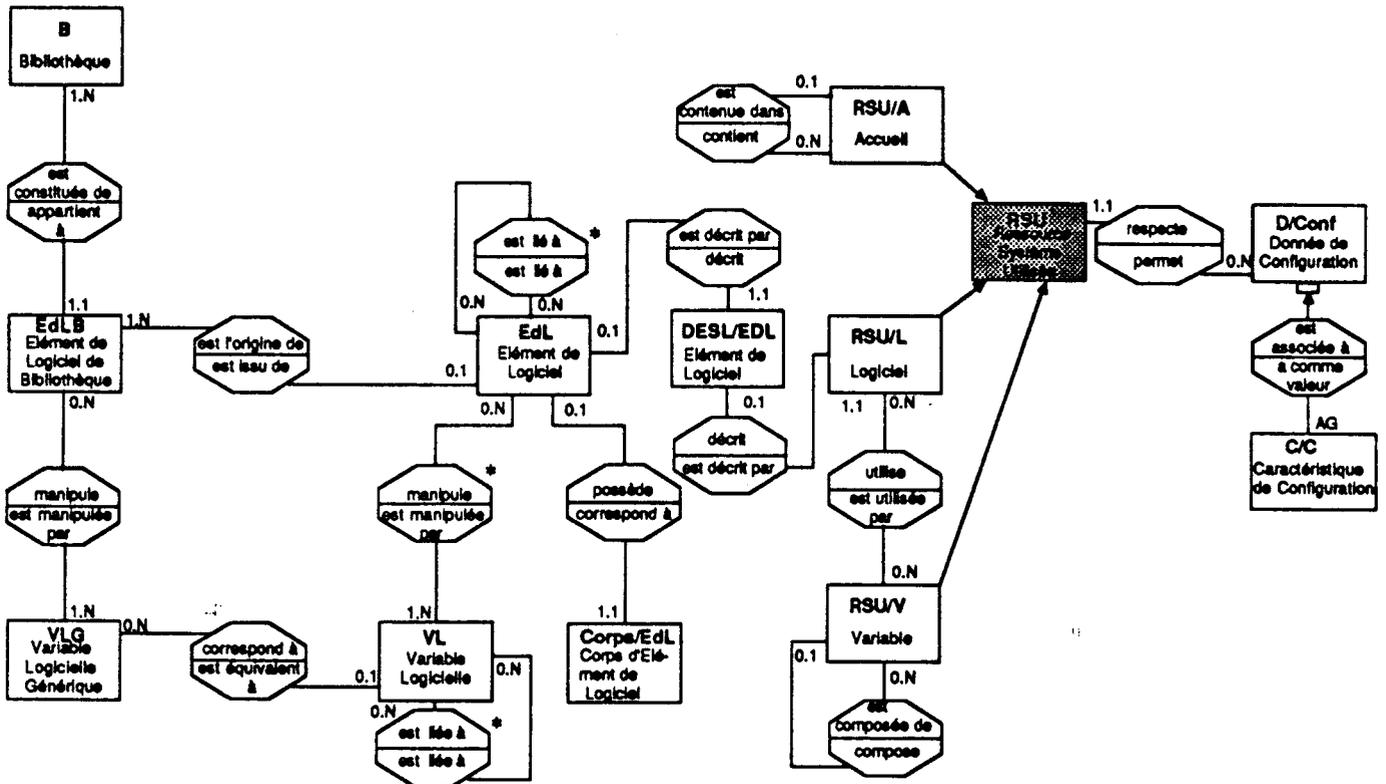


Fig 92

66. L'implantation du logiciel

661. Notion de logiciel implanté

Nous avons, au début de cette section, expliqué que les notions de source et d'objet n'étaient pas pertinentes dans notre approche. A la chaîne classique:

source
objet
exécutable,

nous avons préféré la démarche:

structuration du logiciel
écriture du logiciel
implantation.

Nous avons décrit les données du modèle concernant les deux premiers aspects et nous allons à présent aborder la notion d'implantation.

Un logiciel implanté est constitué d'un ensemble d'éléments de logiciel dans leurs structures d'accueil. Il est stocké grâce à l'entité Logiciel Implanté (LI). Cette entité est liée par une association à l'entité RSU/A (cf schéma de la figure 93)

Une ressource système de type accueil ne peut appartenir à plusieurs logiciels implantés.

662. Notion de système d'exploitation

Le logiciel implanté le sera sur un ou plusieurs constituants programmables. Ce ou ces constituants supportent un système d'exploitation caractérisé par une version. Ce système d'exploitation assure l'exécution du logiciel implanté.

L'entité Système d'exploitation (cf schéma de la figure 93) permet de stocker l'identifiant du système d'exploitation. Ce système d'exploitation est une fourniture du constructeur de constituants programmables. Mais il est essentiel de le nommer dans la Base Application, car le logiciel implanté n'a de sens que lié

à lui. Les cardinalités de l'association SE - LI sont d'ailleurs très restrictives: 1.1 et 1.1. Elles font apparaître l'aspect indissociable du couple formé par ces entités qui constitue en quelque sorte une "machine programmée".

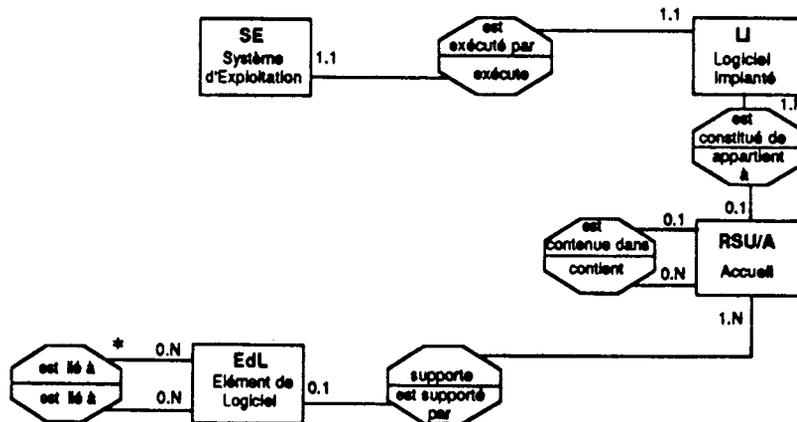


Fig 93

Remarquons que l'entité SE permet de stocker le nom du système d'exploitation et non son type. Si plusieurs constituants programmables supportent le même type de système, il y aura cependant plusieurs instances de l'entité SE, par exemple: moniteur PL7-3 de l'automate A1, moniteur PL7-3 de l'automate A2... Il apparaît donc ici la nécessité d'une évolution du modèle afin de traduire que dans l'exemple les deux instances de l'entité SE sont du même type (moniteur PL7-3). Il aurait fallu créer une entité Type de Système d'Exploitation (TSE) en liaison avec la partie matérielle de la fourniture, par exemple avec l'entité Article d'Approvisionnement.

La notion de système d'exploitation correspond à une réalité de la fourniture logicielle du fournisseur et non à une machine "virtuelle". Nous allons illustrer cette affirmation. Supposons un automate comportant une carte commande d'axes, carte dont une partie de la programmation est réalisée séparément du programme automate. L'optique

"machine virtuelle" conduirait à considérer l'ensemble de l'automate et de ses cartes comme une seule machine programmable avec un fonctionnement déterminé, soit un système d'exploitation virtuel prenant en compte la scrutation du programme automate et du programme carte. Le modèle que nous présentons conduit au contraire à envisager deux logiciels implantés: le programme automate et le programme carte d'axes exécuté par deux systèmes d'exploitation différents dont les versions bien que nécessairement compatibles n'ont aucune raison d'être identiques.

D'autre part, le logiciel implanté doit être en correspondance avec les données de configuration. Cet ensemble de données est regroupé dans une entité Configuration Logicielle (CL) associée à l'entité LI. Ainsi le modèle permettra de traduire la compatibilité entre la configuration et le programme proprement dit (cf schéma de la figure 94).

Une même configuration logicielle peut être utilisable pour plusieurs logiciels implantés et de même un logiciel implanté peut fonctionner avec plusieurs configurations différentes. Ce dernier cas doit être proscrit à notre avis, car pour un logiciel implanté il n'y a à un instant considéré qu'une seule configuration acceptable, celle qui existe dans la machine programmée. La cardinalité 0.N pour l'entité LI dans l'association LI - CL peut inciter à stocker plusieurs versions de configuration, ce qui est contraire aux principes du modèle.

67. Regroupements particuliers de logiciels

L'ensemble des instances de l'entité EDL constitue une partition du logiciel application, ce qui a pour conséquence d'interdire entre ces instances des relations d'inclusion ou plus généralement de hiérarchie.

Mais le concepteur peut souhaiter manipuler des groupes d'éléments de logiciel, avec d'éventuels recouvrements d'un groupe sur l'autre, pour des motivations très variées: constitution de fichiers source, regroupement de toutes les parties de logiciel traitant des sécurités... L'entité Ensemble Logiciel (EL) permet de stocker ces regroupements. Sur cette entité, figure une association permettant la traduction d'éventuelles hiérarchies entre divers ensembles logiciels (cf schéma de la figure 95).

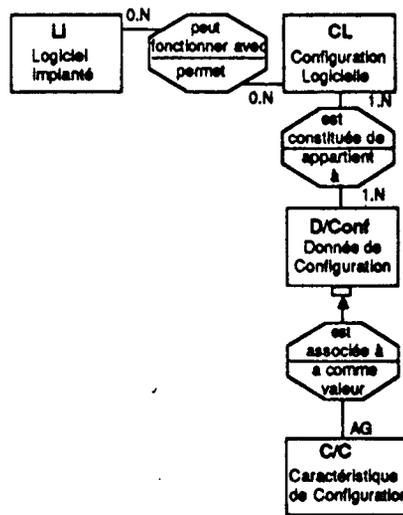


Fig 94

Il convient de noter que l'existence de l'entité EL fut et est toujours contesté au sein de l'équipe BASEPTA. Lorsque nous aborderons les données de structuration (cf section II-8), nous verrons en effet qu'il est possible d'effectuer sur la fourniture logicielle des regroupements d'éléments de logiciel selon des critères à définir. Or l'ensemble logiciel rentre dans cette catégorie de regroupements. L'entité peut être considéré en conséquence comme inutile puisque n'étant qu'une particularisation d'une des entités des données de structuration.

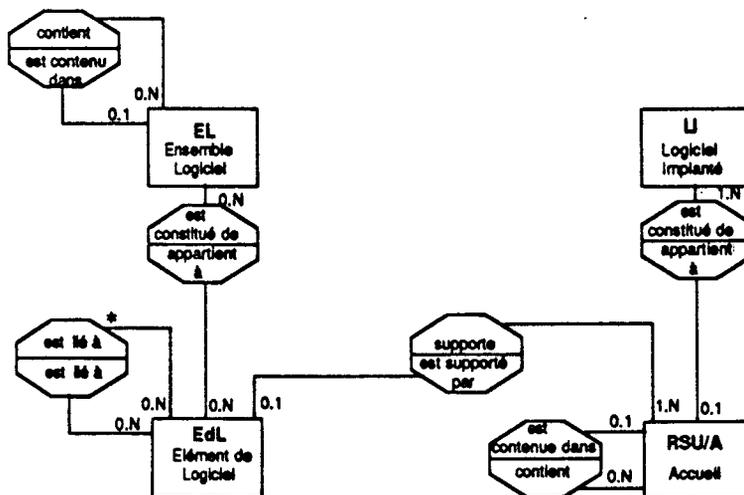


Fig 95

68. Relations logiciel application - système681. Relations variables logicielles - variables système

Lorsque nous avons décrit les différentes manières de réaliser un élément de logiciel (cf II-652), nous avons montré de quelle façon, il était possible d'utiliser une ressource système de type logiciel. L'association EDL-RSU/L se fait grâce à une entité intermédiaire regroupant les caractéristiques partagées par l'élément de logiciel et la ressource système utilisée. Pour associer une variable logicielle à une variable système, le mécanisme est en tout point identique. Il fait intervenir une entité Description Logicielle de type Variable (DESL/V) sous-classe, comme l'entité DESL/EDL, de la classe générale des descriptions logicielles (cf Fig 96).

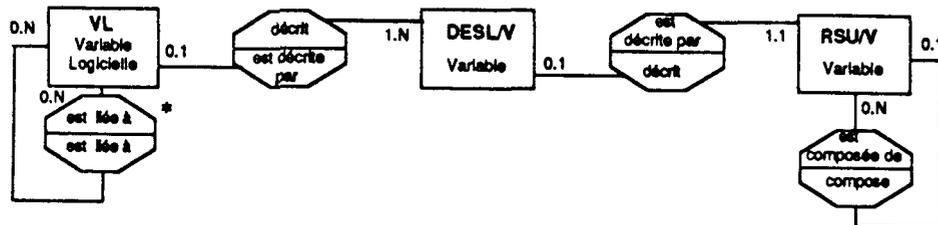


Fig 96

L'entité DESL/V permet de stocker l'identifiant d'un ensemble de caractéristiques de variables telles que sa nature physique (bit, mot...), son aspect logique (Entrée, Sortie, E/S...), son type logiciel (entier, réel, booléen...). Ces informations caractérisent aussi bien la variable logicielle que la ressource système. L'existence d'une entité intermédiaire facilite le travail des deux traitements envisageables. Le premier, le plus classique, consiste à configurer dans un premier temps les variables système puis à les utiliser en fonction de leurs caractéristiques. Le deuxième permettrait d'écrire du logiciel et de n'affecter qu'à la fin de cette écriture des ressources système aux variables logicielles. Le modèle ne privilégie aucun de ces deux traitements.

Remarquons également qu'il est possible d'associer le même ensemble de caractéristiques et par la suite la même ressource système à différentes variables logicielles. Cette possibilité provient de la définition retenue pour une variable logicielle. Deux mnémoniques différents pour la même ressource donnent lieu à deux instances de variables logicielles.

682. Relation élément de logiciel - ressource système de type logiciel

Nous avons déjà étudié (cf II- 652) l'association d'un élément de logiciel à une ressource système de type logiciel par l'intermédiaire de l'entité DESL/EDL.

683. Données de description logicielle

Les deux entités DESL/V et DESL/EDL sont des sous-classes de l'entité générale Description Logicielle. Chaque instance de cette entité DESL permet de désigner un ensemble d'informations caractéristiques d'un objet logiciel, variable ou programme, c'est à dire un ensemble de couples Caractéristiques de Description Logicielle (entité C/DESL), Donnée de Description Logicielle (entité D/DESL).



Fig 97

La figure 98 montre un exemple d'instanciation où sont décrites les caractéristiques logicielles de deux variables système.

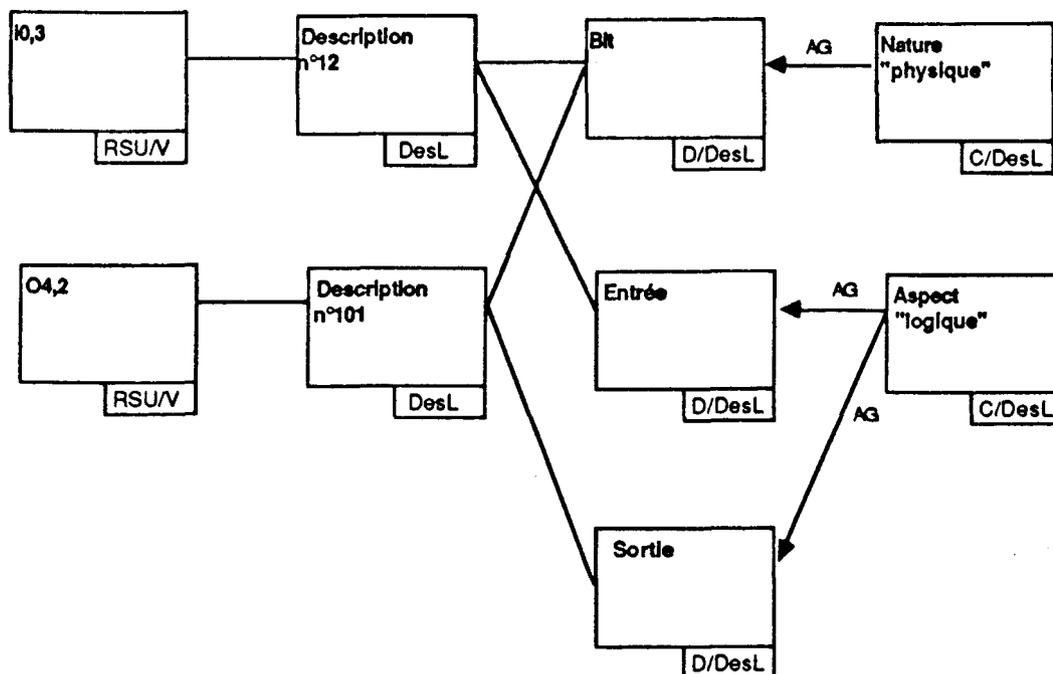


Fig 98

69. Conclusion: Vue générale de l'aspect fourniture logicielle

Sur la figure 99 est récapitulé l'ensemble des entités et des associations présentés dans cette section II-6 pour décrire de manière "anatomique" la fourniture logicielle. On y retrouve les différents aspects du logiciel, soit en partant de la droite: la configuration application, les ressources système utilisées, le logiciel application proprement dit et les liens avec les bibliothèques utilisateurs.

7. LA FOURNITURE. LES LIENS ENTRE LE LOGICIEL ET LE MATERIEL

Pour achever la présentation de la Base Fourniture, nous allons à présent détailler les liens entre les aspects matériel et logiciel. Ces liens sont concrétisés dans notre modèle par des associations entre entités de ces deux aspects.

Dans la partie I, nous avons vu qu'il n'existait pas actuellement d'outils de CAO prenant simultanément en compte les aspects logiciel et matériel. Or la demande des concepteurs de systèmes automatisés se cristallise de plus en plus sur ce point. Des solutions pragmatiques, en cours de développement, tentent de mettre en relation les données des deux types d'outils. Mais ces solutions se placent délibérément au niveau d'une communication physique de données élémentaires. La partie du modèle que nous allons détailler dans cette section II-7 décrit de façon plus précise la sémantique des liaisons entre logiciel et matériel. Elle permettra ainsi d'aborder le problème d'échange entre une CAO électrique et une CAO de programme avec une base conceptuelle plus riche.

71. Implantation d'un logiciel sur un constituant programmable.

Nous avons vu précédemment que le logiciel donnait lieu, en dernière étape de réalisation, à un système d'exploitation. Nous avons baptisé l'ensemble des deux "machine programmée". Cette machine programmée doit fonctionner sur un constituant programmable (automate, micro-ordinateur...). Mais elle peut éventuellement être répartie sur plusieurs constituants.

Le schéma de la figure 100 montre que le problème est traité par une association entre les entités CME et SE.

72. L'association variable système - voie automate.

Il est également essentiel de traduire le lien entre une variable d'entrée/sortie du programme et la voie d'entrée/sortie correspondante de l'automate ou plus généralement du constituant programmable. En fait, ce lien doit être fait non pas avec la variable logicielle (il faudrait alors plusieurs liens dans le cas des variables équivalentes) mais avec la ressource système de type

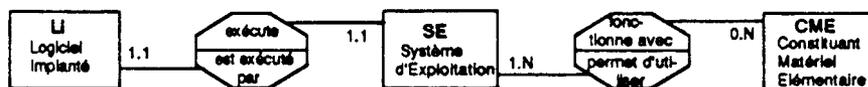


Fig 100

variable associée à la variable logicielle. De façon simple, nous pouvons dire que la ressource système de type variable est le nom dans le système de la voie automate. C'est ainsi que, très souvent les ressources système expriment dans leur appellation le type et la position de la voie dans la carte et dans le rack (cf la variable binaire d'entrée I1,8 de PL7-3 dont nous avons précédemment parlé).

Précisons également que nous avons choisi (cf schéma de la figure 101) d'associer la variable système à une voie automate (entité CME) et non à une borne automate (entité EDR). En effet, suivant le type de voie (avec ou sans commun) le rattachement de la variable à une des deux bornes de la voie serait arbitraire.

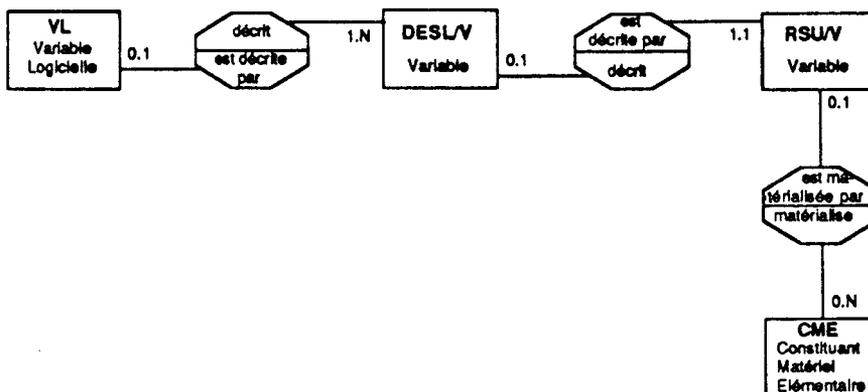


Fig 101

Les schémas des figures 100 et 101 traduisent les liens minimaux entre logiciel et matériel. En réalité les deux aspects sont plus étroitement liés. C'est ainsi que le système d'exploitation est d'un type compatible avec le constituant programmable. Nous avons déjà montré que ceci pouvait être traité grâce à une entité Type de Système d'Exploitation (TSE) liée à l'entité SE et à certaines entités de matériel. D'autre part les caractéristiques de configuration et les données de configuration sont compatibles avec le constituant programmable, ce qui suggère un lien entre les entités C/C, D/Conf et les entités du matériel.

Dans l'état actuel de la version 3.1, le modèle BASEPTA ne prend pas en compte ces relations, laissant à la charge des traitements opérant sur la base le soin de les établir. Dans les versions ultérieures, ce problème devra être traité. Pour la clarté du schéma, il sera nécessaire, nous semble-t-il, de créer des sous-classes de constituants correspondant aux constituants programmables.

8. LES DONNEES DE STRUCTURATION

81. Introduction

L'aspect "fourniture" de la structure de données application permet une description du système automatisé que l'on a souvent qualifiée d'anatomique. Mais cette vision purement descriptive est insuffisante.

En effet, le système automatisé peut également être caractérisé par un ensemble de données plus abstraites liées à des perceptions utilisateur de la fourniture matérielle et logicielle. Ces perceptions sont subjectives et donc non apparentes dans la réalité de l'application. Elles sont pourtant caractéristiques de cette application. Il en est ainsi pour les données de structuration de la commande qui décrivent le comportement attendu de la commande du système. Aux données "anatomiques", viennent donc s'ajouter des données "physiologiques": description ou spécification de comportement de la PC, de la PO, etc... De façon générale, ces différentes données structurent subjectivement la fourniture logicielle et matérielle. Nous avons donc choisi de baptiser la partie du modèle qui les concerne "Données de structuration".

Comme nous allons le voir, il est impossible de recenser de manière exhaustive les structurations possibles d'une application et les critères de structuration sans adopter d'a priori méthodologique. Cette impossibilité, résultant du principe modèle "methodology free" (cf section II-2), a contraint à n'avoir que des entités à caractère général. C'est donc en grande partie sur les objets de l'aspect données de structuration que vont s'appliquer les paramétrages méthodologiques.

82. Objectifs et contraintes du système automatisé: le lien avec le cahier des charges fonctionnel.

Les données du cahier des charges du système automatisé doivent être supportées par la structure de données application. Il faut cependant distinguer dans le cahier des charges plusieurs types de données différents. Comme l'a bien montré C. Sourisse [SOU 87A] [SOU 89A], le cahier des charges d'un automatisme industriel n'est jamais purement fonctionnel. Aux spécifications d'objectifs, s'ajoutent des descriptions de procédés ou de moyens à mettre en oeuvre. C'est la trilogie "But - Procédé - Moyen". Ainsi le cahier des charges ne se contente pas

d'exprimer un besoin, mais décrit également, en pratique, des éléments de solutions. Dans le cahier des charges de la chaîne d'emballage de rouleaux de papier, nous trouvons par exemple une expression d'objectif, gestion du nombre de rouleaux, de procédé, asservissement en vitesse des moteurs, de moyen, utilisation d'un automate TSX 47.

Les données du cahier des charges de type "moyen" sont décrites au niveau de l'aspect fourniture du modèle. Les données de type "procédé" sont décrites dans la partie données de structuration. Nous étudierions dans le paragraphe II-83 de quelle manière. Nous allons nous intéresser ici aux données de type "but".

Les données de type "but" sont donc les données du pur cahier des charges fonctionnel. Elles consistent essentiellement en des objectifs du système automatisé. L'entité Objectifs et Caractéristiques Globaux (OCG) permet de les stocker (cf schéma de la figure 102).

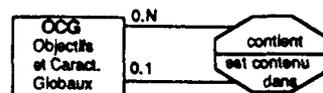


Fig 102

Le modèle autorise une hiérarchisation des objectifs.

Nous pouvons remarquer le caractère général de l'entité OCG. Un premier affinement consisterait à introduire la distinction fonction/critère/contrainte issue de l'analyse de la valeur [PET 88A]. Cette distinction s'applique bien à une description purement fonctionnelle, au sens fonction de service, de l'automatisme.

Un second affinement de l'entité OCG, de nature méthodologique, consisterait à faire apparaître des classes correspondant à la nature des objectifs: objectifs de qualité, objectifs de quantité, objectifs de sûreté (sécurité, disponibilité)... Une telle démarche, impossible dans une optique volontairement non méthodologique, est indispensable pour tout traitement

opérant sur les données de cahier des charges.

83. Les structurations de l'application

831. Description d'une structure

Dans les toutes premières versions du modèle BASEPTA figuraient des entités de structures de commande dites Eléments de Commande (EDC) et des entités de structure de Partie Opérative. Très rapidement, il est apparu deux écueils.

En premier lieu, les deux seules structures PC et PO ne sont pas suffisantes. Les partenaires du projet BASEPTA ont très vite désiré stocker d'autres structures telles que la structure process. Un accord sur le nombre et le type de structures devant être pris en compte dans le modèle de la Base Application est apparu impossible à cause des différentes méthodes mises en oeuvre.

D'autre part, les critères même de structuration diffèrent. Ainsi pour certains la Partie Opérative est organisée fonctionnellement, pour d'autres géographiquement, pour d'autres encore topo-fonctionnellement. Le sens des entités de structure PO ou PC devient alors rapidement discutable.

Le modèle BASEPTA a donc dû se limiter à une solution modeste mais ouverte. Dans la structure de données application, il est possible de stocker tout type de structure à condition d'en préciser obligatoirement le critère. Une structure se décrit par un ensemble d'éléments inter-reliés (entité Elément Résultant de Structuration ou ERS). Chaque instance d'ERS est agrégée à une instance d'une entité Critère de Structuration (CDS) indispensable pour établir sa signification (cf schéma de la figure 103).

De la même façon, la nature des structurations peut varier: structure hiérarchique, structure de liste... Plutôt que de créer sur l'entité ERS une association pour chaque type de structure, nous avons choisi de ne faire qu'une seule association portant un attribut qui traduit le type. Les schémas des figures 104 et 105 montrent deux exemples de structure de l'application chaîne automatique d'emballage de rouleaux de papier.

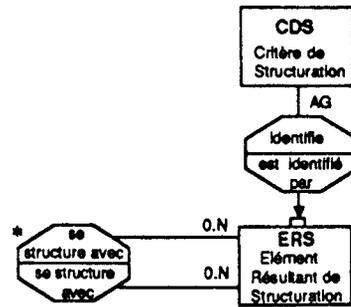
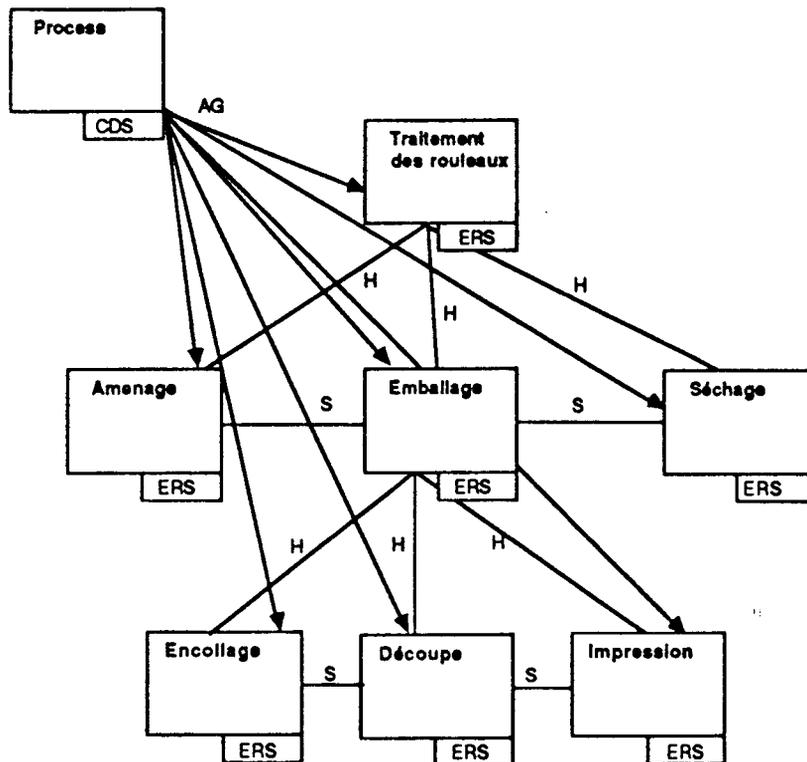


Fig 103



H lien de nature "hiérarchique"
S lien de nature "succession"

Fig 104

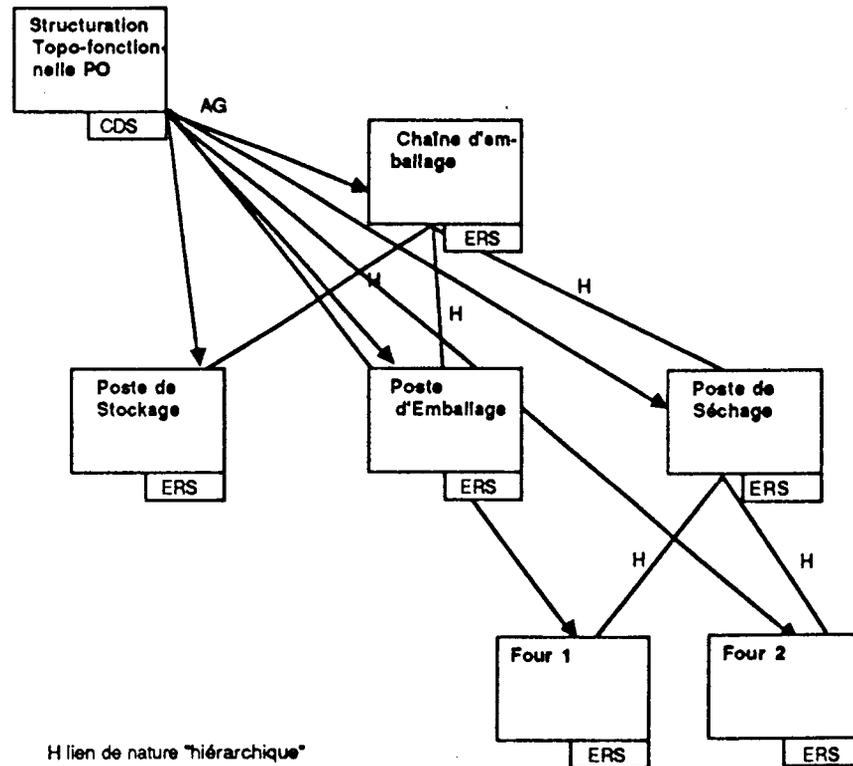


Fig 105

En préalable à toute utilisation de cette partie du modèle de structure de données application, il convient d'effectuer un paramétrage méthodologique. Ce paramétrage consiste en premier lieu à recenser et définir les critères de structuration admissibles, la nature des structures correspondant à ces critères et la typologie des éléments de ces structures.

Le cahier des charges général des automatismes des moyens de production de CITROEN [CIT 88A] définit deux types de structuration: la structuration de l'atelier dont il est facile de voir qu'elle est faite suivant un critère topo-fonctionnel et la structuration de la commande organisée suivant une découpe reflet de celle de l'atelier. Les entités résultant de structuration admissibles pour une structuration d'atelier sont de quatre types: lignes décomposables en modules comportant des stations

regroupant des postes. La configuration méthodologique du modèle pour CITROEN peut être issue directement de ces considérations.

832. Relations inter-structures

Comme nous l'avons laissé entendre, les méthodologues de CITROEN prescrivent une structuration de commande calquée sur la structuration d'atelier. Ainsi ils recommandent d'associer une entité de commande à chaque poste si la complexité et la taille et/ou l'autonomie le justifient. Le dialogue opérateur est structuré de même, le dialogue de commande proprement dit étant associé au niveau poste, le dialogue d'exploitation au niveau ligne ou module.

De façon plus générale, toutes les méthodes de conception induisent des liens inter-structures qu'il convient de traduire au niveau de notre modèle. Une association supplémentaire est donc portée sur l'entité ERS (cf schéma de la figure 106). Ses instances relient des instances d'ERS agrégées à des critères différents.

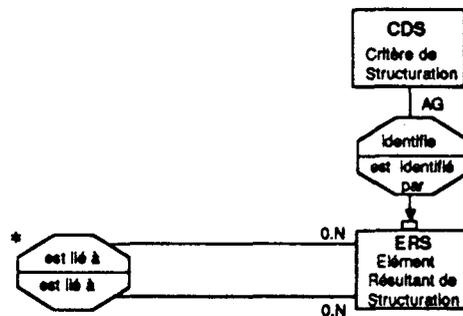


Fig 106

Pour des raisons déjà décrites l'association est très générale et porte un attribut indiquant la nature du lien. Par exemple, entre un élément de structure PC et un élément de structure de l'atelier, la signification du lien est "est la commande de".

Dans le cas de l'exemple Chaîne d'emballage de rouleaux de papier, on peut associer des instances de la structuration PO avec des instances de la structuration process par un lien du type "réalise" (cf schéma de la figure 107).

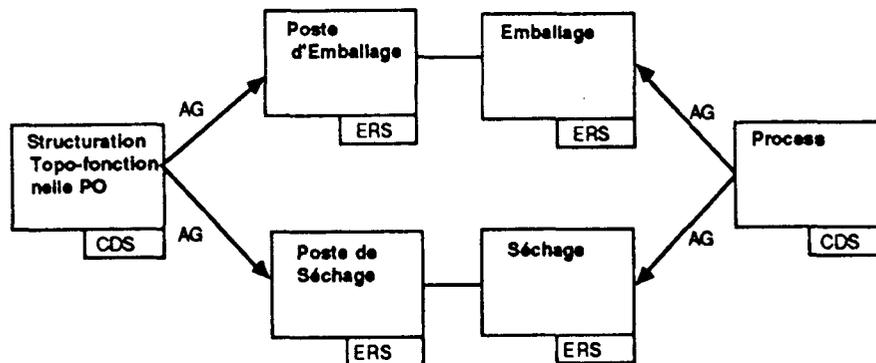


Fig 107

L'exemple de la figure 107 peut sembler trivial. Il n'en est rien. Il souligne au contraire l'ambiguïté de désigner par la même expression "Poste de Séchage" aussi bien le poste physique que la fonctionnalité process. Or il est essentiel de différencier les deux aspects pour ne pas compromettre la performance de traitements de conception.

833. Choix de conception: liens objectifs - structurations

Dans la trilogie But - Procédé - Moyen, les données de procédé peuvent être traduites par des instances des entités ERS et CDS. Mais il est alors souvent intéressant d'établir un lien entre ces instances et les objectifs décrits par des instances de l'entité OCG, auxquels elles répondent. Ce lien traduit un choix de conception mais non un choix technique puisque la réalisation matérielle ou logicielle n'est pas encore prise en compte. Le modèle

comporte donc entre les entités OCG et ERS une association portant un attribut nature de l'association dont la signification n'est pas absolue mais méthodologique (cf schéma de la figure 108). Le lien entre OCG et ERS sera d'ailleurs souvent difficile à établir.

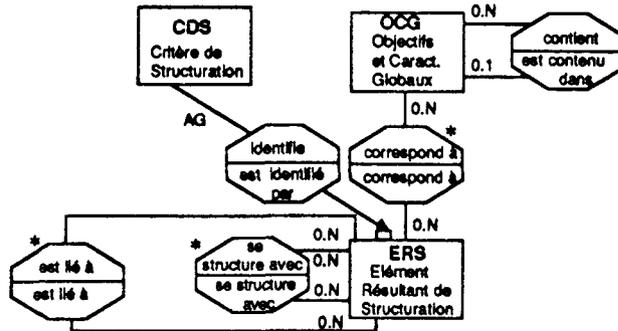


Fig 108

834. La modélisation comportementale

Les instances de l'entité ERS identifient des éléments de structures mais ne décrivent pas ces éléments. Il est souvent important de décrire leur comportement. L'entité Comportement (COMP) identifie le comportement d'un élément de structure (cf schéma de la figure 109). Elle comporte un attribut définissant le modèle avec lequel est exprimé le comportement tel que Grafcet, Réseau de Pétri... Avec les entités ERS et COMP, il est donc possible de stocker la description complète de la commande, soit sa structure et la modélisation de son comportement attendu.

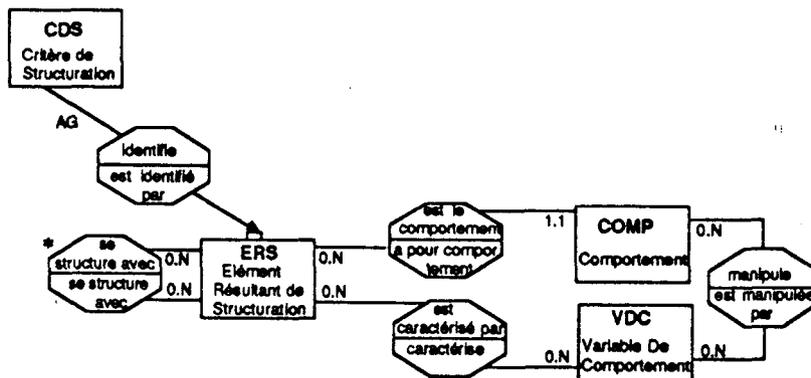


Fig 109

Les cardinalités sur l'association ERS - COMP montrent qu'il est possible d'associer à un même élément de structure plusieurs comportements différents, c'est à dire exprimés avec des modèles différents. Par contre un comportement n'est associé qu'à une seule instance d'ERS ce qui signifie en fait que, compte tenu de l'agrégation CDS - ERS, ce comportement n'a de sens que par rapport à un critère de structuration défini. Il est ainsi concevable d'avoir des critères tels que structuration de la commande attendue, structuration de la commande observée.

Un comportement manipule des variables dites variables de comportement (entité VDC). Ces variables assurent des liens entre les différents comportements, comme les variables logicielles permettent de lier les éléments de logiciel. Il pourrait à ce titre être intéressant d'ajouter un attribut sur l'association COMP - VDC indiquant la nature de la manipulation, c'est à dire par exemple si la variable de comportement est produite ou consommée.

Le schéma de la figure 109 comporte une association entre les entités ERS et VDC. Cette association peut sembler inutile, les variables de comportement caractérisant l'élément de structuration par l'intermédiaire du comportement. Cependant, il est fréquent de caractériser un élément de structure par un ensemble de variables de comportement sans souhaiter expliciter un comportement. A titre d'exemple, citons une instance d'ERS "Régulation de température" (agrégée au critère "structuration fonctionnelle de la commande") caractérisée par la variable de comportement Température.

Le schéma de la figure 110 illustre l'aspect modélisation comportementale sur une partie de l'exemple Chaîne automatique d'emballage de rouleaux de papier.

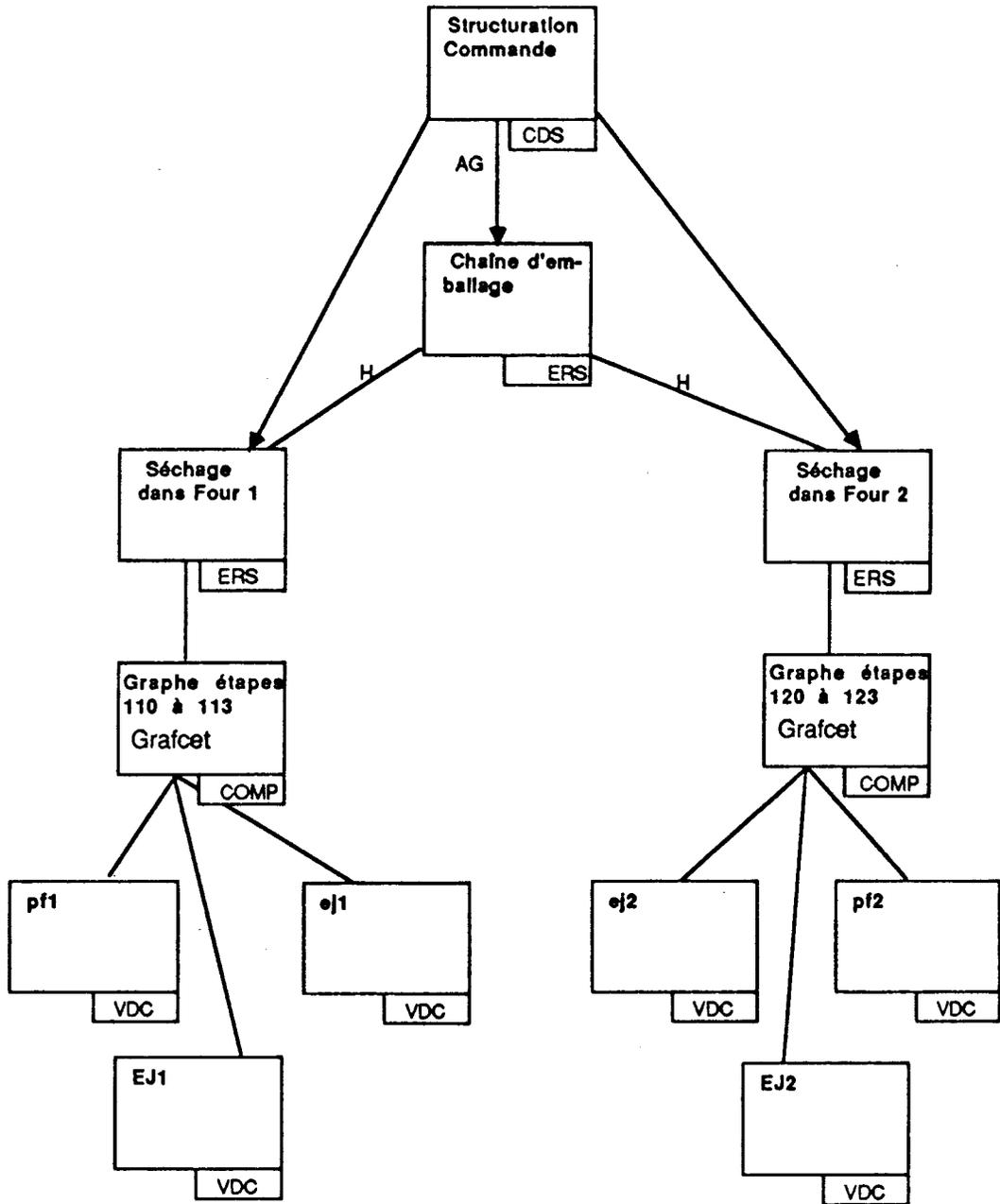


Fig 110

Notons enfin que l'entité COMP ne porte que l'identifiant du comportement et le modèle avec lequel il est exprimé. L'expression proprement dite du comportement doit cependant être stockée dans la Base Application, comme les lignes de programme d'un élément de logiciel devaient l'être. Une entité Corps de Comportement (CORPS/COMP) est donc associée à l'entité COMP (cf schéma de la figure 111). Elle porte un attribut traduisant le modèle

d'interprétation du corps et nécessaire à une exploitation par des traitements. Ce modèle d'interprétation est très lié à une syntaxe et ne doit pas être confondu avec l'attribut "type de modèle" de l'entité COMP lié plutôt aux règles sémantiques du modèle. Ainsi, pour un type de modèle "Grafcet", le modèle d'interprétation pourra être "expression littérale du grafcet", "grafcet CADEPA"...



Fig 111

84. Relations entre la structure de l'application et le logiciel.

841. Vision structurante du logiciel

Le schéma de la figure 112 résume l'ensemble des données que nous avons introduites pour décrire les objectifs et les structurations.

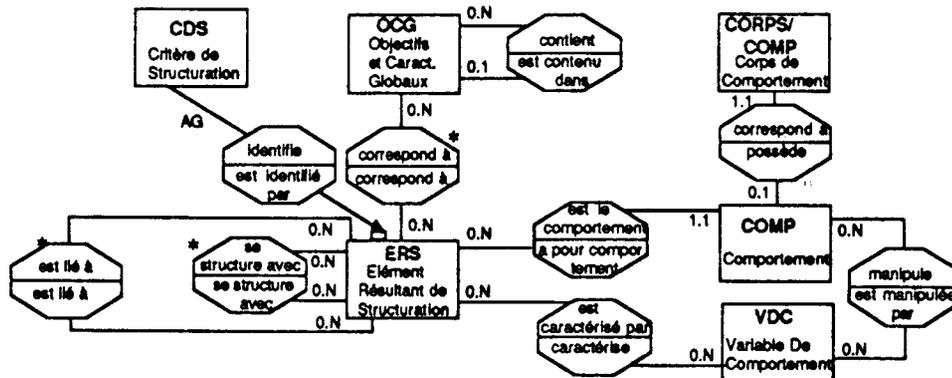


Fig 112

Ces données constituent des visions structurantes sur la fourniture, en particulier sur le logiciel. Il apparaît donc nécessaire de les relier avec les données décrivant la fourniture logicielle. Nous avons choisi d'associer l'entité ERS et l'entité EDL (cf schéma de la figure 113).

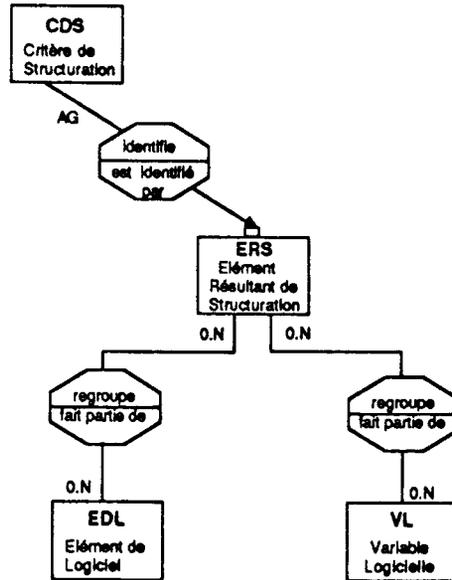


Fig 113

Ce mécanisme permet par exemple de regrouper toutes les instances des EDL traitant de la marche automatique (instance de l'entité ERS agrégée à un critère de structuration modes de marches) laquelle marche automatique peut être modélisée (association d'une instance de comportement). Il est également possible de regrouper les éléments de logiciel réalisant une fonction de commande (instance d'ERS agrégée à un critère "structuration de commande"). Cette fonction de commande est dans le cas général réalisée par du logiciel et du matériel. Le comportement qui peut lui être associée est alors un comportement global et non un comportement du seul logiciel. Mais il peut être également souhaitable de modéliser uniquement le comportement logiciel (données de l'analyse organique du logiciel). Les instances d'EDL seront alors associées à des instances d'ERS agrégées avec un critère "structuration du logiciel" et associées à des comportements souvent exprimés à l'aide d'organigrammes. Les possibilités d'utilisation du lien ERS - EDL sont donc très nombreuses.

Le lien entre éléments de structuration et éléments de logiciel est un lien très méthodologique et souvent difficile à exprimer. Parfois il est plus facile de se contenter de regrouper des variables logicielles dans des éléments de structuration. Le modèle comprend donc une association entre les entités ERS et VL (cf schéma de la figure 113).

Remarquons également qu'il est possible d'établir un ERS correspondant à l'entité EL avec un critère de structuration adéquat. Ceci explique le caractère non indispensable de l'entité EL que nous avons souligné en section II-6.

842. Association variable logicielle - variable de comportement.

L'association ERS - VL que nous venons de décrire reste très imprécise. Elle se contente d'exprimer qu'un "ensemble" de variables logicielles se rapporte à un élément de structure. Lorsque cet élément de structure est modélisé par des variables de comportement, il est souvent important de préciser le lien entre chaque variable de comportement et la variable logicielle qui la matérialise dans le programme. Les entités VL et VDC sont donc associées comme indiqué sur le schéma de la figure 114.



Fig 114

Ce type de lien est en général assez facile à établir. Il induit de fait une partition des variables logicielles en mettant en évidence les variables propres au logiciel et n'ayant pas d'autre signification dans l'application. Ces variables ne sont pas reliées à des variables de comportement, le cas typique est celui de variables de calcul intermédiaires du programme (B10 de la figure 88).

Sur le schéma de la figure 115, nous donnons un exemple des liens VDC - VL pour la chaîne d'emballage.

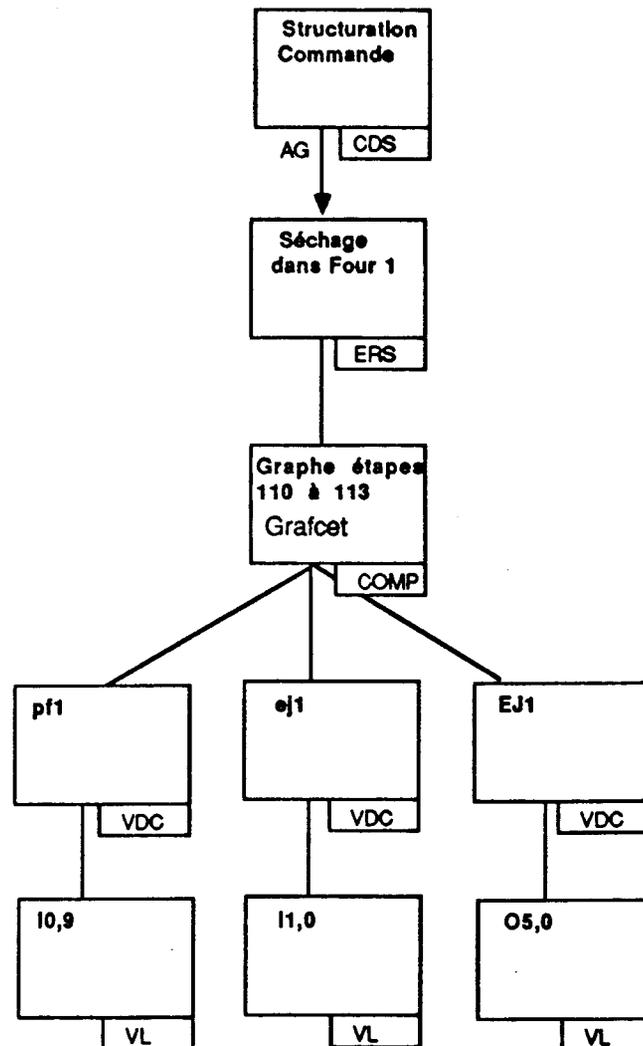


Fig 115

L'établissement des relations entre les instances des entités VDC et VL est une des tâches importantes d'un traitement permettant de passer automatiquement d'une spécification de la structure de commande à sa réalisation logicielle.

La figure 116 résume les liens entre les données de structuration et l'aspect logiciel de la fourniture.

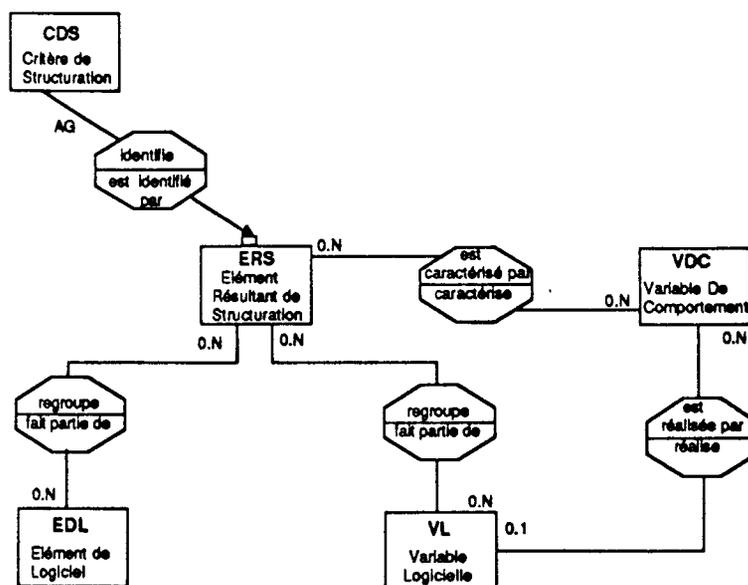


Fig 116

85. Relations entre la structuration de l'application et le matériel.

851. Vision structurante du matériel.

Comme nous l'avons fait pour la fourniture logicielle, il est nécessaire de relier les données de structuration aux données de la fourniture matérielle. Nous avons choisi de relier l'entité ERS à l'entité CME de façon assez naturelle puisque l'entité CME correspond à une vision logique des constituants (cf schéma de la figure 117).



Fig 117

Il est ainsi possible d'exprimer différentes versions structurantes de la fourniture matérielle.

Il convient ici de rappeler que dans le cas général, une instance d'ERS est simultanément reliée à des données du matériel et du logiciel. Ce sera notamment le cas pour toute description de la structure de commande indépendante de la réalisation technologique. La figure 118 en présente une illustration sur l'exemple chaîne d'emballage.

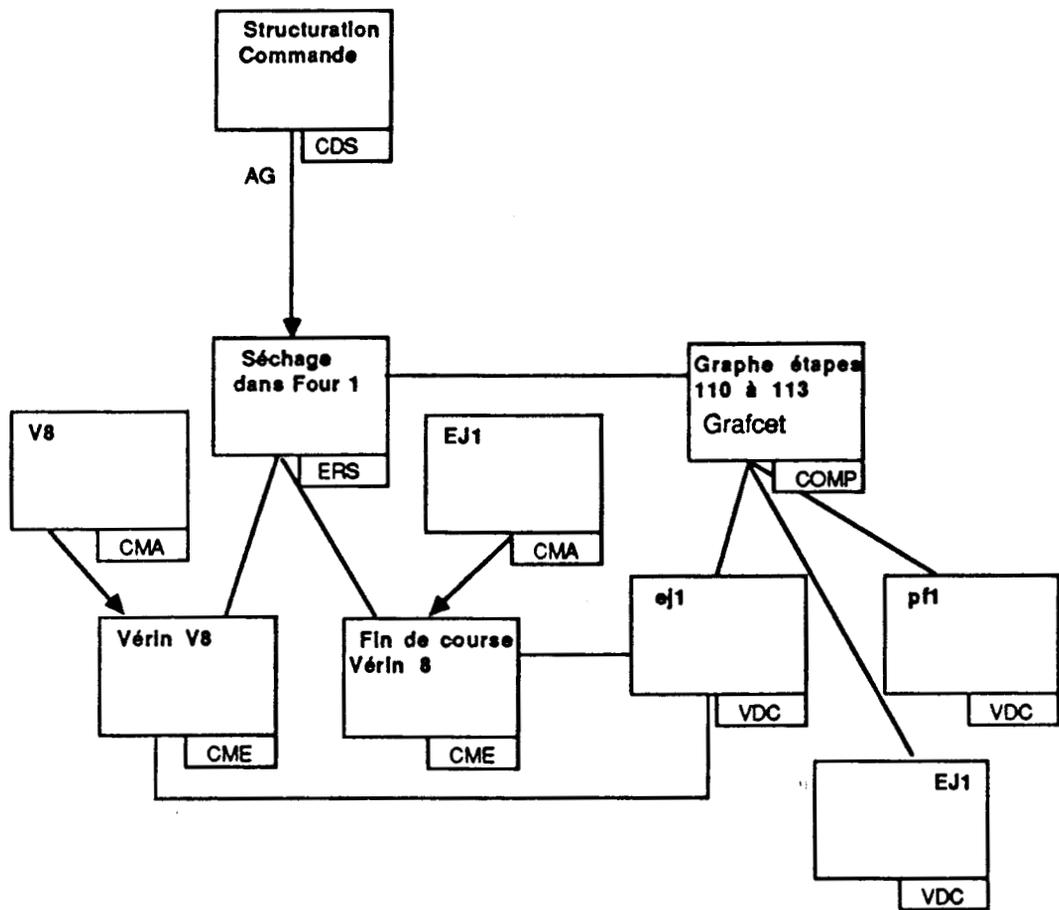


Fig 118

852. Variable de comportement d'un constituant matériel.

En observant le schéma de la figure 118, on constate qu'il serait intéressant de faire le lien entre une variable de comportement, eji et les constituants dont elle représente des états, soit ici aussi bien le vérin que le fin de course. Le modèle de structure de données application comprend donc une association entre les entités VDC et CME (cf schéma de la figure 119).

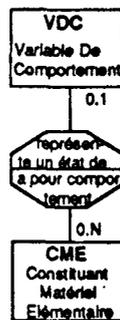


Fig 119

Appliqué à notre exemple, le schéma de la figure 118 devient donc le schéma de la figure 120.

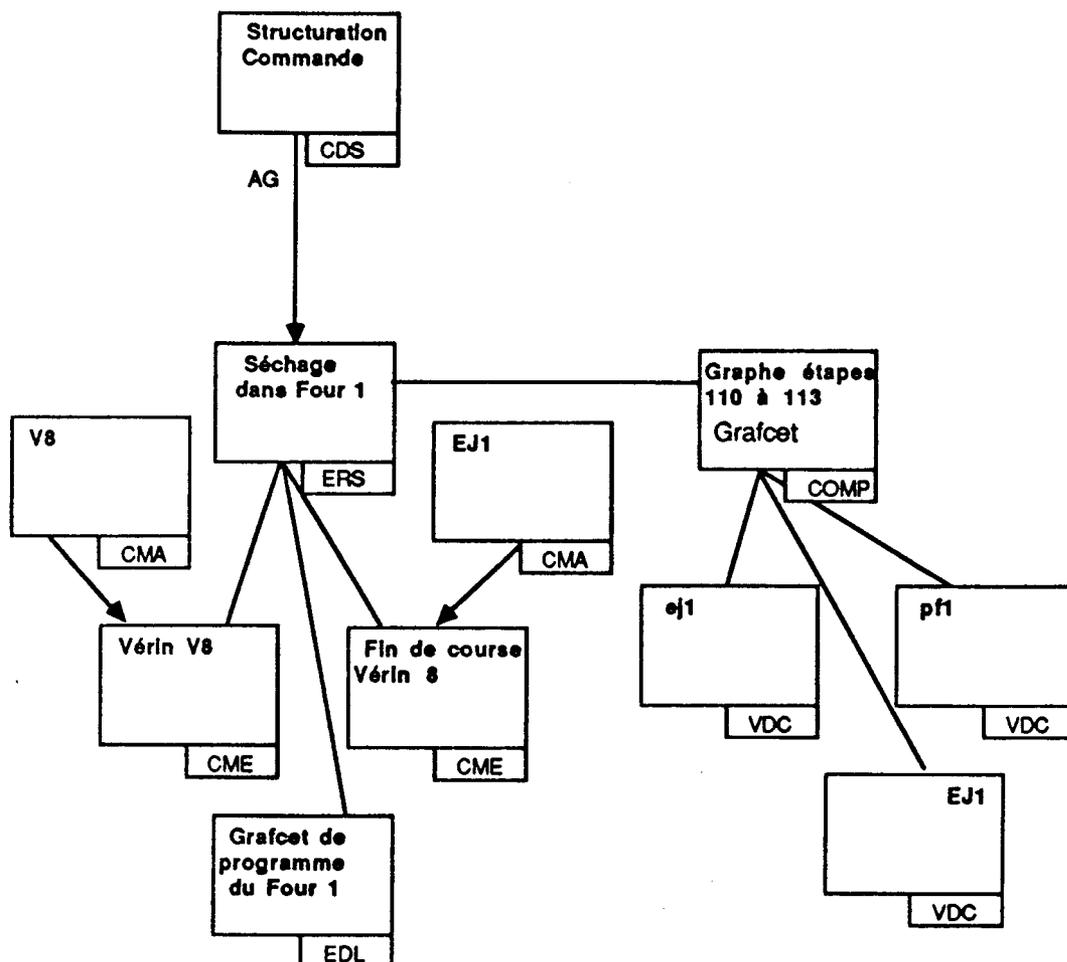


Fig 120

86. Conclusion: Vue générale des données de structuration et de leurs relations avec la fourniture.

Le schéma de la figure 121 récapitule l'ensemble des données de structuration.

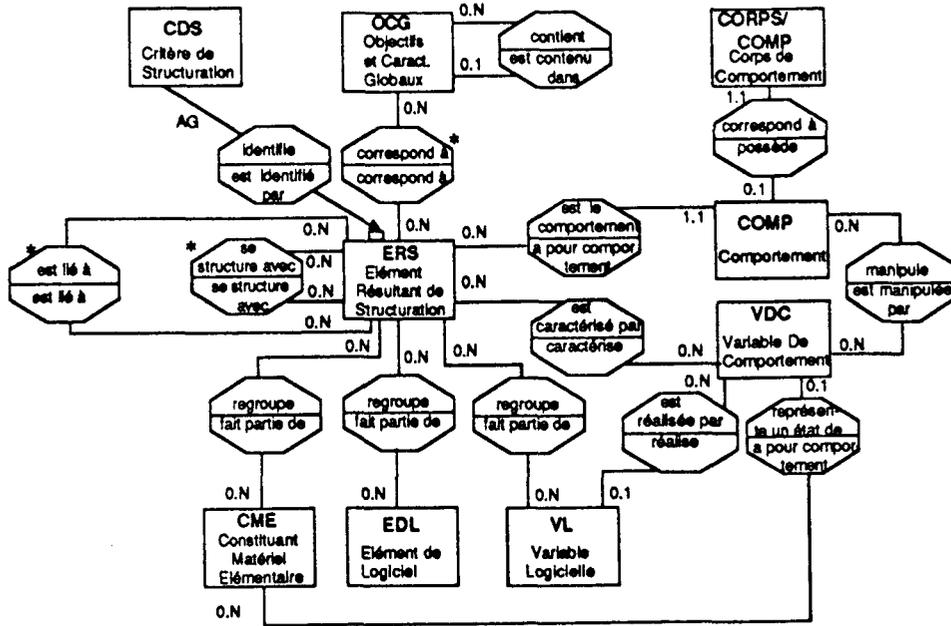


Fig 121

Nous avons vu le caractère général de ces objets. Il est par conséquent nécessaire pour exploiter cette partie du modèle de la paramétrer au préalable. Malgré tout, ces données sont relativement fines. Elles permettent de distinguer des notions différentes souvent dissimulées par l'adoption d'une même désignation. Ainsi I1,0 désigne parfois simultanément:

- 1 - la variable système ou instance de RSU/V
- 2 - la variable logicielle ou instance de VL
- 3 - la voie automate ou instance de CME
- 4 - la borne de voie automate ou instance d'EDR
- 5 - la variable de comportement ou instance de VDC

Le cas 1 est le seul cas "normal". Le cas 2 résulte de l'utilisation d'un automate ne disposant pas de programmation en symbolique. Le cas 3 est lié au fait que le nom système est souvent indicatif de l'adresse physique de la variable système. Le cas 4 résulte d'une assimilation excessive (problème de commun!) entre la voie automate et la borne automate. Le cas 5 est issu d'une erreur méthodologique due à l'utilisation d'une variable

PARTIE II - MODELE DE STRUCTURE DE DONNEES APPLICATION

de réalisation pour décrire une spécification de conception.

9. LES DONNEES ANNEXES

91. Introduction

Les deux aspects du modèle précédemment étudiés, fourniture matérielle et logicielle et données de structuration permettent d'exprimer la quasi-totalité des données caractéristiques du système automatisé. Mais ils supposent de savoir répartir les informations sur les différentes entités du modèle et établir des associations dont la nature est souvent méthodologique.

La partie Données Annexes que nous présentons dans cette section II-9 permet d'associer des ensembles de données, manipulés globalement, à tous les objets de l'application ainsi que des données générales de l'application automatisée. Son usage doit cependant être restreint. L'intérêt du modèle de données est effectivement d'offrir la possibilité d'atomiser les données pour faciliter le travail des traitements opérant sur la Base Application. La partie Données Annexes conduit au résultat contraire.

92. L'identifiant de l'application automatisée traitée

Il est souvent nécessaire de stocker des informations générales sur l'application que décrit la structure de données. Ces informations sont souvent des informations de type gestion d'affaire telles que numéro d'affaire (codification interne du concepteur), référence client, responsable d'affaire, etc... Notre modèle de données concerne plutôt les données techniques de l'application que ces données de gestion. Il n'offre donc qu'une seule entité baptisée Application (APPLI). Le principal objectif d'APPLI est de stocker l'identification du système automatisé objet de la structure de données (cf schéma de la figure 122). C'est en quelque sorte "l'étiquette" de la structure de données. Cette entité ne donne donc lieu qu'à une seule instance puisque la Base Application ne décrit qu'une seule application.



Fig 122

L'entité APPLI n'est associée à aucune autre entité car elle concerne l'application globalement. C'est par son intermédiaire que l'on pourra associer la structure de données application à la structure de données "affaire" dans le cas de l'intégration d'une CAO d'automatismes dans une CAO de réalisation de projets prenant en compte par exemple des aspects commerciaux.

93. Association d'un ensemble de données annexes à un élément de l'application

Toute entité du modèle, à l'exception des entités EDOnA, DATE et REFEX que nous allons introduire et de l'entité CORPS, est une sous classe de l'entité générale nommée Élément Référençable de l'Application (ERA). L'entité ERA est associée à une entité Ensemble de Données Annexes (EDOnA). Les instances d'EDOnA peuvent être inter-reliées grâce à une association (cf schéma de la figure 123). Par ce mécanisme, il est donc possible d'associer à toute entité un ensemble global, c'est à dire indissociable, d'informations.

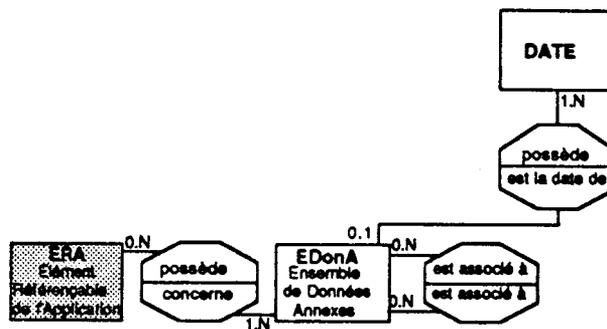


Fig 123

Une instance d'EDOnA peut être associée à une instance d'une entité DATE grâce à une association dont l'attribut précise la nature de la date telle que date de création, date de dernière modification de l'instance d' EDOnA...

D'autre part, l'entité EDOnA porte un attribut précisant la raison pour laquelle elle a été créée (archivage de la version, données pour maintenance, trace du processus de conception).

Ainsi nous avons vu en section II-6 que le logiciel n'était exprimé dans la Base que sous sa dernière version. Si l'utilisateur désire conserver ses anciennes versions, il associera à l'instance d'EDL concernée différentes instances d'EDonA avec une motivation archivage et pour lesquelles il fournira une date de version. Soulignons le caractère annexe de ces données. En effet, les versions successives de logiciel ne sont pas, à bien regarder, des données de l'application mais plutôt des données de conception.

De même les données techniques des constituants sont les données effectives de l'article d'approvisionnement. Or le souhait des futurs utilisateurs de traitements de CAO est souvent de conserver simultanément la donnée effective et la donnée calculée (et qui a permis le choix du constituant). A l'instance de l'entité Donnée Technique on associera une instance d'EDonA correspondant à la donnée calculée. Ici encore, nous sommes en présence d'une donnée de conception et non d'une donnée véritable de l'application.

Nous avons vu précédemment le caractère limité de l'entité APPLI. Pour conserver des informations plus riches ou plus nombreuses sur l'application, il suffit d'associer à l'instance d'APPLI une instance d'EDonA avec par exemple un attribut motivation "donnée de gestion".

94. Description de l'ensemble de données annexes

Une instance d'EDonA ne représente que la désignation de l'ensemble de données annexes. De la même manière que pour les entités EDL et COMP, il peut être nécessaire de conserver également le contenu de cet ensemble de données annexes dans la structure de données application. Ceci est possible grâce à l'entité Corps de Données Annexes (CORPS/DA) associée à l'entité EDonA comme indiqué sur le schéma de la figure 124.

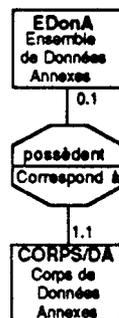


Fig 124

95. Référence externe d'un ensemble de données annexes.

Nous avons vu qu'il était possible d'associer à une instance d'EDonA un contenu sous la forme d'une instance de CORPS/DA lorsque ce contenu fait partie de l'application. Mais ce contenu peut aussi être extérieur. C'est le cas par exemple de consignes de maintenance que l'on souhaite associer à tel ou tel constituant. Il faut alors donner le moyen d'accéder à ce contenu externe. L'entité Référence Externe (REFEX) satisfait ce besoin (cf schéma de la figure 125).

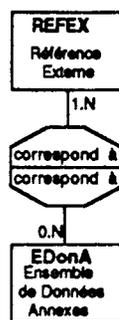


Fig 125

96. Les entités génériques.961. Le corps

Bien que ces entités ne fassent pas à proprement parler partie des Données Annexes, nous allons ici présenter les trois entités génériques Corps, Caractéristique et Donnée dont nous avons précédemment présenté les sous-classes.

L'entité CORPS comprend trois sous-classes (cf schéma de la figure 126): l'entité Corps d'Elément de Logiciel (CORPS/EDL), l'entité Corps de Comportement (CORPS/COMP) et l'entité Corps de Données Annexes (CORPS/DA). Elle porte un attribut appelé moyen d'interprétation permettant aux traitements opérant sur la Base d'interpréter et d'exploiter le corps.

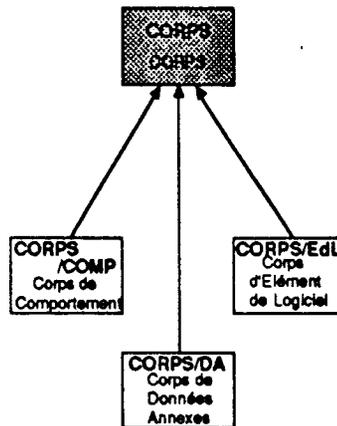


Fig 126

Les trois sous-classes de l'entité **CORPS** ne sont associées qu'à une seule autre entité, celle dont elles sont le corps, et par des cardinalités restrictives. En effet, les liens dans la structure de données doivent être faits avec le contenant (ensemble de données annexes, élément de logiciel, comportement) et non avec le contenu (données annexes, lignes de programme, expression du comportement). Cependant, pour des raisons de complétude de la Base Application, il est nécessaire de stocker les différents corps.

962. La caractéristique

L'entité **Caractéristique** (cf schéma de la figure 127) comporte trois sous-classes l'entité **Caractéristique de Configuration** (C/C), l'entité **Caractéristique de Description Logicielle** (C/DESL) et l'entité **Caractéristique Technique** (C/T).

L'identifiant d'une instance de **Caractéristique** est composé du nom de la caractéristique et de son unité. Ceci permet facilement de distinguer des caractéristiques telles que intensité (électrique) en Ampère, intensité (lumineuse) en Candéla, intensité (magnétique) en Tesla...

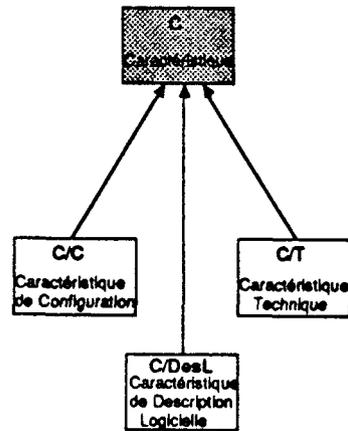


Fig 127

963. La Donnée

La classe générale des données (cf figure 128) comprend trois sous-classes: l'entité Donnée Technique (D/T), l'entité Donnée de Description Logicielle (D/DesL) et l'entité Donnée de Configuration.

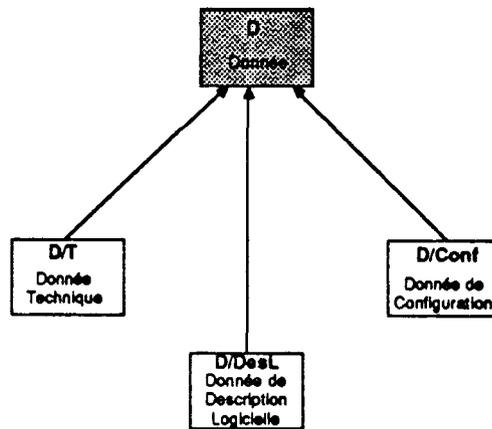


Fig 128

Une instance de Données est identifié par un identifiant interne à la structure de données, c'est à dire n'ayant pas de sens "application". Son attribut essentiel est la valeur. L'unité de la donnée sera l'unité de la caractéristique à laquelle elle sera agrégée. Il serait très utile de rajouter un attribut "unité pratique" sur l'entité Donnée. Lorsque cet attribut est instancié, la valeur de la donnée est exprimée dans cette unité. Dans le cas contraire, c'est l'unité de la caractéristique qui doit être utilisée. Ceci évite d'avoir des valeurs numériques peu commodes à manipuler.

97. Conclusion: Vue générale des Données Annexes.

Le schéma de la figure 129 récapitule les Données Annexes.

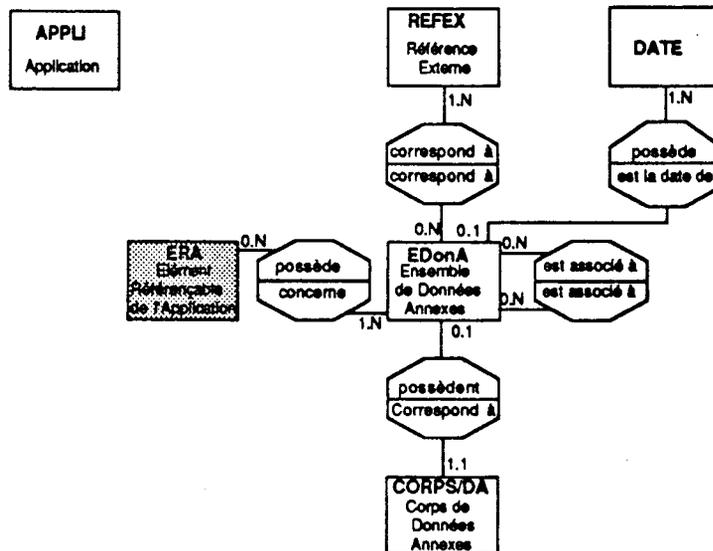


Fig 129

Afin de souligner à nouveau que cette partie du modèle est dangereuse d'utilisation, nous présentons ici un dernier exemple. Nous avons vu que les données de la documentation, symboles, vignettes d'implantation... devaient être reliées à diverses entités. Les Données Annexes offrent une possibilité pour associer globalement un schéma ou un folio par exemple à un constituant matériel, par l'intermédiaire d'une instance d'EDona. Il est aisé de voir les limitations sur les traitements qu'une telle solution induit.

CONCLUSION

Dans cette Partie II, nous avons présenté la totalité du modèle de structure de données Application. Sa richesse et sa complexité (57 entités, 75 associations) entraîneront des difficultés d'implémentation. De la même façon qu'un outil logiciel PTA couvrant l'ensemble du cycle de vie du système automatisé paraît irréaliste, il est en fort peu probable que la totalité du modèle de structure de données application soit intégrée dans un produit. Nous allons étudier en Partie III les problèmes liés à l'implémentation.

PARTIE III

EXPERIMENTATION DU MODELE DE

STRUCTURE DE DONNEES

INTRODUCTION

L'objectif de cette Partie III est de présenter une première expérimentation du modèle de données application exposé en Partie II. Le support de cette expérimentation est l'exemple chaîne automatique d'emballage de rouleaux de papier.

Dans une première section III-1, nous donnerons un extrait de la structure de données de cette application, et nous énoncerons quelques résultats concernant le volume et la finesse des données qu'elle contient.

Dans une deuxième section III-2, nous indiquerons comment implémenter cette structure de données à l'aide d'un système de gestion de base de données de type relationnel. Nous montrerons les difficultés inhérentes à une telle implémentation et nous relativiserons son intérêt.

1. TRAITEMENT D'UN EXEMPLE D'APPLICATION AUTOMATISEE

11. Structure de données de l'exemple chaîne automatique d'emballage de rouleaux de papier.

Nous présentons ici un extrait de la structure de données de l'exemple chaîne automatique d'emballage de rouleaux de papier. Il s'agit d'une partie relativement restreinte de l'ensemble des instances des entités et des associations du modèle de données. En effet, le nombre des données devient relativement important dès que l'on aborde de manière détaillée la description du système automatisé et une présentation exhaustive de la structure de données n'est pas compatible avec l'objectif de ce mémoire. C'est pourquoi nous avons choisi de ne présenter que certaines instances de certaines entités et associations.

La représentation graphique que nous avons utilisée pour introduire des exemples d'instances d'objets du modèle en Partie II a été ici abandonnée. Il est effectivement évident que cette représentation devient très rapidement inutilisable dès que le nombre d'instances sur la figure dépasse la dizaine. Nous avons donc préféré une représentation sous forme de tables très proche de la représentation physique dans un SGBD relationnel. Dans la section III-2, nous affinerons le passage du modèle Entité-Association au modèle relationnel. Considérons simplement pour l'instant que entités et associations se représentent par des tables. Les attributs de l'entité, en particulier ses attributs identifiants, constituent des colonnes de la table représentant l'entité. Les identifiants de deux entités reliées par une association ainsi que les attributs de cette relation constituent des colonnes de la table représentant l'association. Les lignes des tables représentent des instances. Le cas des entités liées par agrégation est un peu particulier. Nous utilisons une seule table pour décrire la chaîne d'agrégation.

CF-CMA

Identifiant CF	Identifiant CMA (nom application)
Automate programmable	A1
Commutateur	C1
Commutateur	CM2
Commutateur	CM3
Commutateur	CMCC
Commutateur	CMF1
Commutateur	CMF2
Disjoncteur	F13
Disjoncteur	F14
Disjoncteur controle	F12
Moteur	G1
Moteur	G2
Contacteur inverseur	KA4
Contacteur inverseur	KA5
Relais	KA1
Relais	KA50
Relais	KA51
Relais	KA52
Relais	KA500
Contacteur	KM1
Contacteur	KM2
Contacteur	KM10
Voyant	L1
Voyant	L2
Voyant	L3
Voyant	L4
Voyant	L5
Voyant	L6
Voyant	L7

CMA-CME-CME-EDR-EVC

Identifiant CMA (nom application)	type CME	repère CME	type CME	repère CME	repère EDR	nom EVC
KM2	contacteur	1	bobine	1	A1	014.14
KM2	contacteur	1	bobine	1	A2	013.13
KM2	contacteur	1	3poles	1	1	013.5
KM2	contacteur	1	3poles	1	2	013.6
KM2	contacteur	1	3poles	1	3	013.9
KM2	contacteur	1	3poles	1	4	013.10
KM2	contacteur	1	3poles	1	5	013.13
KM2	contacteur	1	3poles	1	6	013.14
KM2	contacteur	1	CF	1	13	013.18
KM2	contacteur	1	CF	1	14	013.14
KM2	contacteur	1	CF	2	53	013.18
KM2	contacteur	1	CF	2	54	014.15
KM2	contacteur	1	CF	3	73	014.11
KM2	contacteur	1	CF	3	74	014.12
KM2	contacteur	1	CF	4	83	014.06
KM2	contacteur	1	CF	4	84	014.07
KA5	contacteur	1	bobine	1	A1	014.13
KA5	contacteur	1	bobine	1	A2	013.13
KA5	contacteur	1	pole	1	1	013.27
KA5	contacteur	1	pole	1	2	013.28
KA5	contacteur	1	CO	1	21	014.8
KA5	contacteur	1	CO	1	22	014.9
KA5	contacteur	1	CF	1	53	013.18
KA5	contacteur	1	CF	1	54	013.16
KA5	contacteur	1	CF	2	73	013.18
KA5	contacteur	1	CF	2	74	014.17
KA5	contacteur	1	CF	3	83	013.10
KA5	contacteur	1	CF	3	84	013.11
KA5	contacteur	1	bobine	1	A1	014.13
KA5	contacteur	1	bobine	1	A2	013.13
KA5	contacteur	1	pole	1	1	013.27
KA5	contacteur	1	pole	1	2	013.28
KA5	contacteur	1	CO	1	21	014.8
KA5	contacteur	1	CO	1	22	014.9
KA5	contacteur	1	CF	1	53	013.18
KA5	contacteur	1	CF	1	54	013.16
KA5	contacteur	1	CF	2	73	013.18
KA5	contacteur	1	CF	2	74	014.17
KA5	contacteur	1	CF	3	83	013.10
KA5	contacteur	1	CF	3	84	013.11

CMA-CME-CR

Identifiant CMA (nom application)	type CME	repère CME	type CME	repère CME	nom CR
KM2	contacteur	1	bobine	1	contacteur1
KM2	contacteur	1	3poles	1	contacteur1
KM2	contacteur	1	CF	1	contacteur1
KM2	contacteur	1	CF	2	additif1
KM2	contacteur	1	CF	3	additif1
KM2	contacteur	1	CF	4	additif1
KA5	contacteur	1	bobine	1	contacteur2
KA5	contacteur	1	pole	1	contacteur2
KA5	contacteur	1	CO	1	contacteur2
KA5	contacteur	1	CF	1	additif2
KA5	contacteur	1	CF	2	additif2
KA5	contacteur	1	CF	3	additif2
KA4	contacteur	1	bobine	1	contacteur3
KA4	contacteur	1	pole	1	contacteur3
KA4	contacteur	1	pole	2	contacteur3
KA4	contacteur	1	CO	1	contacteur3
KA4	contacteur	1	CF	1	additif3
KA4	contacteur	1	CF	2	additif3
KA4	contacteur	1	CF	3	additif3

CMA-CME-CR

Identifiant CMA (nom application)	type CME	repère CME	nom CR
D1	distributeur B	1	distributeur1
D2	distributeur M	1	distributeur2
D3	distributeur M	1	distributeur3
D4	distributeur M	1	distributeur4
R1	régleur A	1	régleur1
R10	régleur A	1	régleur2
R2	régleur B	1	régleur3
R3	régleur B	1	régleur4
BPF1	bouton	1	bouton1
BPF2	bouton	1	bouton2
BPF3	bouton	1	bouton3
BPAA1	bouton	1	bouton4
BPF9	bouton	1	bouton5
BPF10	bouton	1	bouton6

CR-AA

AA-CLAP

nom CR	clé (appro application) AA	clé (appro application) AA	Valeur CLAP	Fournisseur
contacteur1	CT1	CT1	LC1-D123MA65	TELEMECANIQUE
contacteur2	CT2	CT2	LC2-D129MA65	TELEMECANIQUE
contacteur3	CT2	ADD1	LA1-D31A65	TELEMECANIQUE
additif1	ADD1	DST1	PVD-C142229	TELEMECANIQUE
additif2	ADD1	DST2	PVD-C141229	TELEMECANIQUE
additif3	ADD1	RGV1	PWR-A1445	TELEMECANIQUE
distributeur1	DST1	RGV2	PWR-B1148	TELEMECANIQUE
distributeur2	DST2	BP1	XB2-BA21	TELEMECANIQUE
distributeur3	DST2	BP2	XB2-BA42	TELEMECANIQUE
distributeur4	DST2			
régleur1	RGV1			
régleur2	RGV1			
régleur3	RGV2			
régleur4	RGV2			
bouton1	BP1			
bouton2	BP1			
bouton3	BP1			
bouton4	BP2			
bouton5	BP1			
bouton6	BP1			

AA-TA

clé (appro application) AA	nom TA
CT1	Contact B+C
CT2	Contact-inv. B+C
ADD1	Bloc additif
DST1	Distributeur
DST2	Distributeur
RGV1	Régleur
RGV2	Régleur
BP1	Bouton C+T
BP2	Bouton C+T

TA-C/T

Nom TA	Unité C/T	Nom C/T
Contact B+C	V	Tension bobine
Contact B+C	A	Courant maximal
Contact B+C	A	Pouvoir de coupure
Contact inv B+C	V	Tension bobine
Contact inv B+C	A	Courant maximal
Contact inv B+C	A	Pouvoir de coupure
Bouton C+T	SO (Sans Objet)	Type bouton
Bouton C+T	SO	Désignation bouton
Bouton C+T	SO	Couleur
Distributeur	SO	Type de connexion
Distributeur	SO	Diamètre tubes de connexion
Régleur	SO	Type de connexion
Régleur	mm ou °	Diamètre tubes de connexion

C/T

Unité C/T	Nom C/T	Champs de valeur
V	Tension bobine	—
A	Courant maximal	—
A	Pouvoir de coupure	—
SO (Sans Objet)	Type bouton	à impulsion, temporisé, tournant...
SO	Désignation bouton	dépassant, affleurant, affleurant avec marquage...
SO	Couleur	noir, vert, jaune, rouge, bleu...
SO	Type de connexion	instantanée orientable, fileté...
mm ou °	Diamètre tubes de connexion	4, 6, 8 (mm) 1/4, 1/2, 3/8 (")...

AA - C/T - D/T

Clé (app. application)	nom C/T	Unité C/T	Identifiant D/T	valeur D/T
CT1	Tension bobine	V	D1	220
CT1	Courant maximal	A	D2	25
CT1	Pouvoir de coupure	A	D3	250
CT2	Tension de bobine	V	D4	220
CT2	Courant maximal	A	D5	25
CT2	Pouvoir de coupure	A	D6	250
DST1	type de connexion	SO	D7	instantanée
DST1	d. tubes de connexion	mm ou °	D8	4 (mm)
DST2	type de connexion	SO	D9	instantanée
DST2	d. tubes de connexion	mm ou °	D10	4 (mm)
RGV1	type de connexion	SO	D11	instantanée
RGV1	d. tubes de connexion	mm ou °	D12	4 (mm)
RGV2	type de connexion	SO	D13	instantanée
RGV2	d. tubes de connexion	mm ou °	D14	4 (mm)
BP1	type bouton	SO	D15	à impulsion
BP1	désignation bouton	SO	D16	affleurant
BP1	couleur	SO	D17	noir
BP2	type bouton	SO	D18	à impulsion
BP2	désignation bouton	SO	D19	affleurant
BP2	couleur	SO	D20	rouge

CDS-ERS

nom CDS	nom ERS
Process	Traitement des rouleaux
Process	Amenage
Process	Emballage
Process	Séchage
Process	Encollage
Process	Découpe
Process	Impression
Structuration topo-fonctionnelle PO	Chaîne d'Emballage
Structuration topo-fonctionnelle PO	Poste de stockage
Structuration topo-fonctionnelle PO	Poste d'emballage
Structuration topo-fonctionnelle PO	Poste de séchage
Structuration topo-fonctionnelle PO	Four 1
Structuration topo-fonctionnelle PO	Four 2
Structuration fonctionnelle commande	Chaîne d'emballage
Structuration fonctionnelle commande	Emballage
Structuration fonctionnelle commande	Amenage
Structuration fonctionnelle commande	Encollage
Structuration fonctionnelle commande	Découpe
Structuration fonctionnelle commande	Impression
Structuration fonctionnelle commande	Evacuation
Structuration fonctionnelle commande	Aiguillage. Fours
Structuration fonctionnelle commande	Séchage dans four 1
Structuration fonctionnelle commande	Séchage dans four 2
Structuration modes de marche	Traitement automatique des rouleaux
Structuration modes de marche	Traitement avec un seul four
Structuration modes de marche	Arrêt en position de sécurité
Structuration modes de marche	Arrêt dans l'état initial
Structuration modes de marche	Arrêt pour recharge papier ou rouleaux
Structuration modes de marche	Mise PO dans état initial et marche manuelle

CDS-ERS-VDC

nom CDS	nom ERS	nom VDC
Structuration fonctionnelle commande	Amenage	pe
Structuration fonctionnelle commande	Amenage	dg1
Structuration fonctionnelle commande	Amenage	dg2
Structuration fonctionnelle commande	Encollage	pe
Structuration fonctionnelle commande	Encollage	G1+
Structuration fonctionnelle commande	Encollage	G2
Structuration fonctionnelle commande	Encollage	ENC
Structuration fonctionnelle commande	Découpe	pe
Structuration fonctionnelle commande	Découpe	G1+
Structuration fonctionnelle commande	Découpe	dec
Structuration fonctionnelle commande	Découpe	COUT
Structuration fonctionnelle commande	Impression	G1-
Structuration fonctionnelle commande	Evacuation	EVA
Structuration fonctionnelle commande	Aiguillage Fours	af1
Structuration fonctionnelle commande	Séchage dans four 1	pf1
Structuration fonctionnelle commande	Séchage dans four 1	ej1
Structuration fonctionnelle commande	Séchage dans four 1	EJ1
Structuration fonctionnelle commande	Séchage dans four 2	pf2

VDC-VL

nom VDC	nom VL
dcy	<u>10,0</u>
pe	<u>10,6</u>
ps	<u>10,7</u>
pf1	10,9
pf2	<u>10,A</u>
pp1	<u>10,B</u>
pp2	<u>10,C</u>
ej1	11,0
ej2	11,1
RDG1	O4,0
RDG2	O4,1
MARQ	O4,2
ENC	O4,3
COUT	O4,4
EVA	O4,5
AF1	O4,6
AF2	O4,7
EJ1	O5,0

DESL/V-D/DESL-C/DESL

nom DESL/V	nom C/DESL	unité C/DESL	Identifiant D/DESL	valeur D/DESL
D001	Nature physique	SO	D30	bit
D001	Aspect logique	SO	D31	entrée
D003	Nature physique	SO	D32	bit
D003	Aspect logique	SO	D33	entrée
D005	Nature physique	SO	D34	bit
D005	Aspect logique	SO	D35	entrée
D008	Nature physique	SO	D36	bit
D008	Aspect logique	SO	D37	sortie
D009	Nature physique	SO	D38	bit
D009	Aspect logique	SO	D39	sortie
D011	Nature physique	SO	D40	bit
D011	Aspect logique	SO	D41	sortie
D012	Nature physique	SO	D42	bit
D012	Aspect logique	SO	D43	interne
D015	Nature physique	SO	D44	bit
D015	Aspect logique	SO	D45	interne
D018	Nature physique	SO	D46	bit
D018	Aspect logique	SO	D47	interne
D019	Nature physique	SO	D48	mot

VL-DESL/V-RSU/V

nom VL	nom DESL/V	nom RSU/V
I0,0	D001	I0,0
I0,1	D002	I0,1
I0,2	D003	I0,2
I0,3	D004	I0,3
I0,4	D005	I0,4
O4,0	D006	O4,0
O4,1	D007	O4,1
O4,2	D008	O4,2
O4,3	D009	O4,3
O4,4	D010	O4,4
O4,5	D011	O4,5
B0	D012	B0
B1	D013	B1
B2	D014	B2
B3	D015	B3
B10	D016	B10
B20	D017	B20
T0,D	D018	T0,D
T0,P	D019	T0,P
T1,P	D020	T1,P

12. Analyse des résultats

L'application du modèle de dossier à l'exemple chaîne d'emballage automatique de rouleaux de papier permet de tirer un certain nombre de conclusions. Cet exemple, bien que créé de toutes pièces est cependant significatif d'une certaine réalité industrielle, comme nous l'avons vu en Partie I. L'analyse des résultats nous autorise donc à émettre des considérations générales.

La première remarque que l'on peut faire, concerne la taille de la structure de données. Malgré la simplicité de l'exemple, le nombre des données est très important. Ainsi, la prise en compte des différents aspects des constituants matériels multiplie les instances d'entités. Ces instances ne sont d'ailleurs pas toujours faciles à appréhender en dehors des traitements. Ceci explique la difficulté de trouver des identifiants différents (mais signifiants pour un utilisateur) pour les instances de CME, CR, AA d'un même constituant. Il n'est pas habituel de désigner par des vocables distincts l'aspect approvisionnement, l'aspect physique et les aspects logiques. En réalité ces désignations sont souvent internes aux traitements qui les gèrent.

Il résulte de tout ceci que la taille de la structure de données s'accroît très vite dans des proportions qu'on ne peut justifier sans parler des traitements exploitant cette structure. Nous pouvons aussi en déduire qu'il n'est pas raisonnable d'envisager de stocker la structure de données application en un seul lieu. Pour des impératifs de taille mémoire, mais aussi de rapidité de traitement, il apparaît que la structure de données application devra, tout au moins dans les premières implantations être physiquement répartie. Cette répartition devra être fonction des traitements.

La deuxième constatation qui s'impose est la performance du modèle tant dans la richesse des données que dans la complexité des relations qu'il permet de traduire. La tentation a été grande dans le traitement de l'exemple chaîne d'emballage d'abuser de cette performance, notamment en explicitant des informations sur le système automatisé que la documentation généralement sous-entend. C'est par exemple le cas du câblage, traité simplement par l'ajout de bornes de bornier sur le schéma développé ou de la structuration du logiciel qui n'apparaît pas clairement dans le listing du programme. Ce sont autant d'informations omises au niveau de la documentation pour

des raisons de simplification, que le modèle incite à expliciter. Il s'agit d'ailleurs, à notre avis, d'un des points forts du modèle. Rappelons ici que le but du travail de BASEPTA était de faciliter les développements ultérieurs d'outils de CAO, notamment en gérant au niveau de la structure de données application tous les liens de nature application, liens actuellement à la charge des traitements. Le modèle que nous avons présenté répond à cet objectif. Le revers de la médaille est ici encore une tendance à l'explosion combinatoire du nombre de données. Le niveau de détail permis par notre modèle peut donc être dangereux. Il devra être fonction de la nature et de la performance des traitements opérant sur la Base Application (Pourquoi atomiser des données si elles sont manipulées globalement ?).

La troisième remarque est une confirmation. Elle concerne le niveau d'abstraction des données du modèle. Les données de la structure de données application sont des données application non représentées. Elles sont, par essence même, difficiles à concevoir puisque la seule manipulation qui nous en est familière est effectuée au travers de la documentation. L'hypothèse de départ de notre travail était, dans la construction de la Base Application, de bien dissocier données application et données de représentation. La conclusion est que pour une bonne utilisation de la Base Application, il faudra soigneusement étudier les données de documentation et leurs liens avec les données application.

La quatrième remarque concerne les quantités respectives de données des différents aspects du modèle, fourniture et données de structuration notamment. La prédominance des données de la fourniture, en particulier matérielle est très nette. Les données de structuration, plus abstraites, sont plus difficiles à exprimer. Le manque de méthodologie est particulièrement sensible.

Dans le dossier, grâce aux différents schémas, l'intégralité de l'assemblage matériel est exprimé. Les données de la fourniture matérielle sont alors assez simples à identifier. Mais ce n'est pas le cas des données de structuration. L'expression de ces données dans la documentation est rarement ou confusément faite. Dans l'exemple, il a fallu opérer une analyse détaillée des schémas PO, des grafjets de commande et souvent extrapoler. La définition d'un langage de description de la totalité de la commande contribuerait à améliorer la situation. Mais plus généralement, il faudra conduire une réflexion méthodologique. Celle ci devra identifier et

standardiser les différentes structurations et parvenir à un affinement des données de structuration comparable à celui des données de la fourniture. De même, il faudra étudier les liens entre données de structuration et données de la fourniture, liens qui se sont avérés, sur l'exemple, difficiles à établir et à exploiter. Il est d'ailleurs à noter que notre modèle de structure de données application peut fort bien servir de base à cette réflexion méthodologique. Le traitement de l'exemple a ainsi permis de déterminer assez naturellement quelques critères de structuration et d'identifier des liens inter-structures dont l'utilisation nous semble généralisable.

2. IMPLANTATION DE LA STRUCTURE DE DONNEES APPLICATION GRACE A UN SGBD RELATIONNEL.

21. Objectifs

Pour compléter l'expérimentation "papier" décrite en section III-1, nous allons maintenant présenter les mécanismes d'une réalisation informatique de la structure de données application. Nous avons choisi d'exposer une implémentation basée sur l'utilisation d'un système de gestion de base de données. Le SGBD envisagé sera du type relationnel à cause de sa relative simplicité de mise en oeuvre, mais aussi parce que les systèmes relationnels sont aujourd'hui les plus communs.

Les fonctions dont nous disposerons dans un premier temps seront élémentaires, du type consultation, création, modification. Ces fonctions sont accessibles par le langage de manipulation de données (LMD) du SGBD. Il convient ici de relativiser l'intérêt d'une telle implémentation. En effet, compte-tenu du caractère abstrait des données, la création d'une instance d'entité a peu de sens. Ce ne serait pas le cas d'une base de données Constituants pour laquelle créer un article, créer une référence catalogue sont des fonctions utilisateurs en tant que telles. Les fonctions de bas niveau du type création, consultation, modification dans le cas de la Base Application n'en sont pas. Ce sont des primitives d'accès devant être intégrées dans des traitements de plus haut niveau qui, eux, fournissent des fonctions utilisateur. Elles sont donc en général transparentes pour un utilisateur de CAO.

Cependant, une implémentation avec un SGBD relationnel permet d'appréhender plus facilement la performance et l'intérêt de notre modèle qu'au travers de traitements manuels. Elle présente un intérêt "pédagogique".

22. Mécanismes de transcription du schéma conceptuel.

221. Introduction

Le passage d'un schéma décrit dans le modèle Entité-Association en un schéma relationnel est un travail relativement classique et bien modélisé [PLA 88A]. Notre outil de modélisation Entité-Association enrichi étant une

extension du modèle Entité-Association, les mécanismes de transcription, dits aussi de dérivation, sont donc analogues. Seul se pose le problème de la traduction des extensions.

222. Présentation sommaire d'un système relationnel.

Rappelons qu'un schéma relationnel est constitué de plusieurs relations. Une relation R se définit comme un sous ensemble d'un produit cartésien d'ensembles E_i ($0 < i < n+1$), les composantes X_i de chaque n-uplet appartenant à la relation prenant leur valeur dans les ensembles E_i . On définit également en général dans un schéma relationnel un ensemble de contraintes ou propositions logiques que doivent vérifier les composantes des n-uplets des relations.

D'un point de vue implémentation grâce à un SGBD relationnel, les relations sont traduites par des tables dont les colonnes constituent des sous-ensembles des ensembles E_i et les lignes des n-uplets de la relation. La création ou déclaration des structures de tables ainsi que les contraintes entre les colonnes ou sur les valeurs des n-uplets est effectuée lors de la création de la structure de données grâce au langage de description de données (LDD) que fournit le SGBD.

223. Dérivation d'une entité "non contrainte"

Une entité non contrainte, c'est à dire n'intervenant pas dans une relation d'agrégation est traduite par une table. Son identifiant est la clé primaire de la table. Ses attributs sont les colonnes de la table. Si l'entité ne possède qu'un seul attribut, son identifiant, elle n'est pas convertie.

Le schéma de la figure 130 montre un exemple de dérivation d'entité non contrainte.

Dans le cas d'entités liées par des relations de généralisation, nous ne traduisons pas l'entité classe générale. Les sous classes devront donc obligatoirement prendre en compte les attributs de la classe générale en plus de leurs attributs spécifiques. Il en sera de même pour les associations. Un système relationnel ne sait pas exploiter la relation de généralisation.

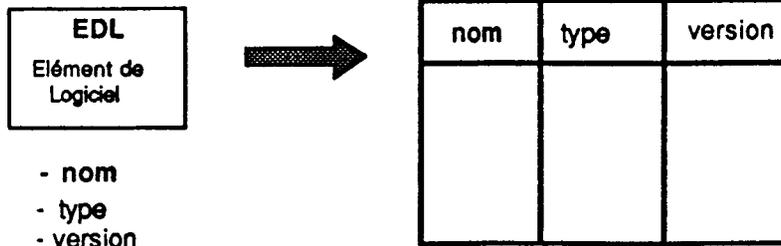


Fig 130

224. Dérivation d'une association classique.

Une association "classique", c'est à dire n'étant ni une généralisation, ni une agrégation, sera dérivée en fonction de ses cardinalités. Si les cardinalités sont de type 0.N ou 1.N, l'association devient une table, les identifiants des entités qu'elle relie et ses propriétés spécifiques, lorsqu'elle en a, deviennent des colonnes de la table. La clé primaire de cette table est constituée par les identifiants des deux entités associées.

Le schéma de la figure 131 en donne un exemple.

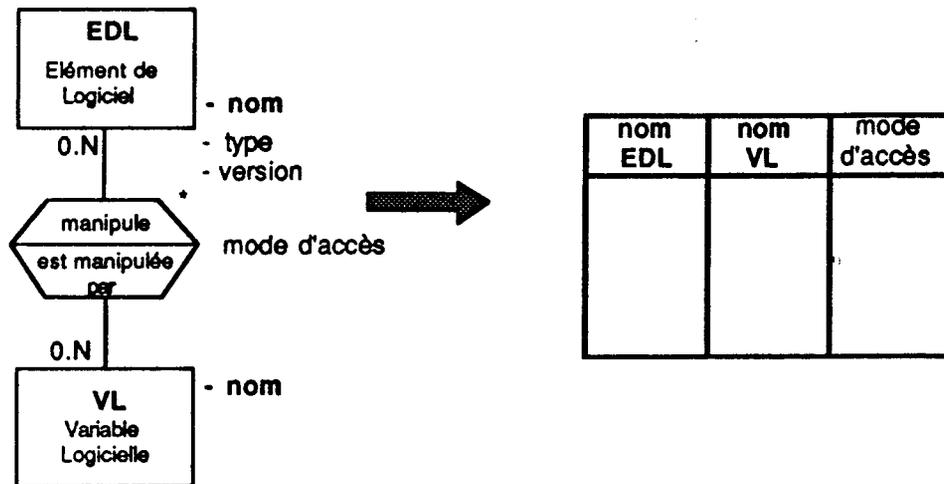


Fig 131

Si l'une des deux cardinalités de l'association est du type 0.1 ou 1.1, l'association se traduit par l'ajout de colonnes à la table qui traduit celle des entités qui participe à l'association avec la cardinalité 0.1 ou 1.1. Le nombre de colonnes supplémentaires est égal au nombre d'attributs identifiants de la deuxième entité plus d'éventuels attributs de l'association. La clé primaire de la table est l'identifiant de la première entité.

Le schéma de la figure 132 illustre ce type de dérivation.

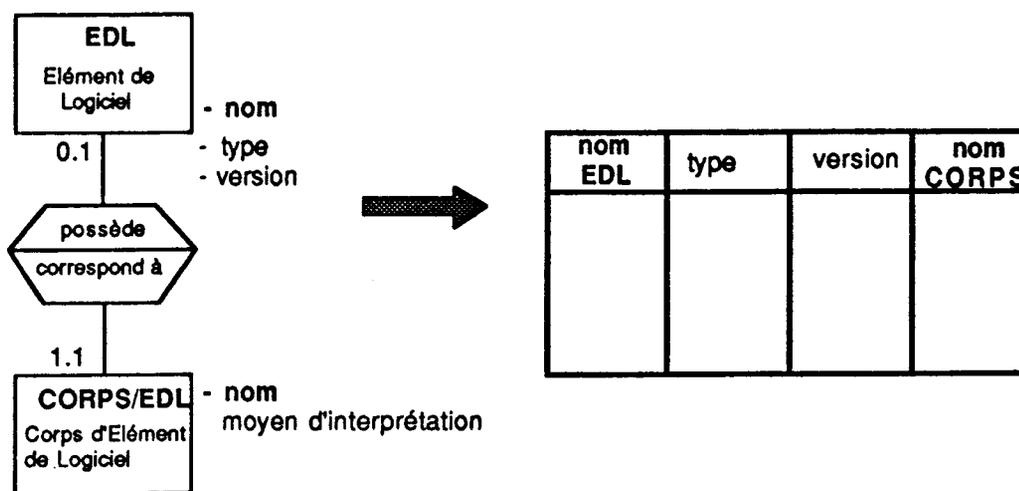


Fig 132

Dans le cas de relation de type "boucle" il faudra orienter la relation en précisant origine et destination.

225. Dérivation d'une agrégation

Dans le cas d'une relation de type agrégation, les deux entités et l'association sont traduites par une seule table dont les colonnes sont constituées des attributs des trois objets. La clé primaire est formée par les identifiants des deux entités.

Le schéma de la figure 133 en donne une illustration.

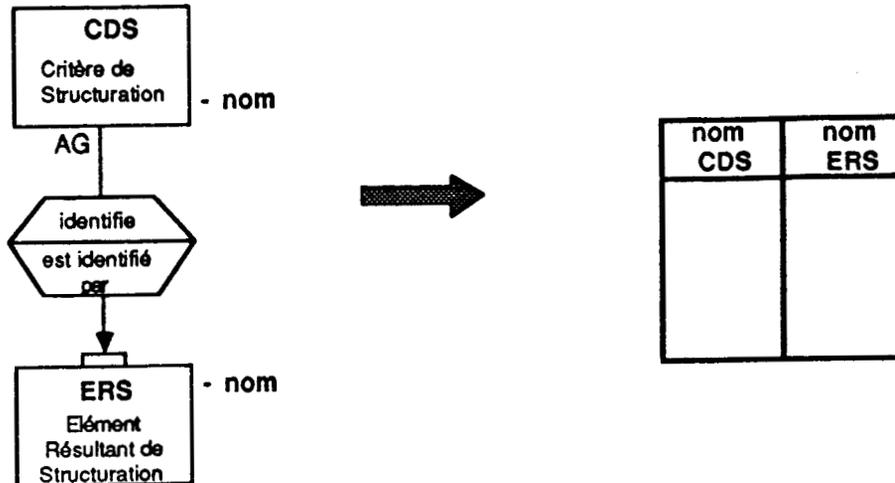


Fig 133

Dans le cas particulier d'entités associées par des successions d'agrégations, type CMA - CME - EDR, la table sera de taille variable, ce qui pose problème. En effet, il est possible de traduire des structures de taille variable avec la majorité des SGBD relationnels, mais la taille maximale de ces structures doit être fixée, ce qui n'est pas le cas dans le modèle. La profondeur de l'arborescence de description d'un CME, par exemple, n'est effectivement pas limitée.

226. Traduction des contraintes.

Les mécanismes précédents sont insuffisants pour décrire toutes les contraintes exprimables dans le modèle Entité-Association enrichi. En particulier, il est facile de constater que seules les cardinalités maximales sont prises en compte. Les cardinalités minimales, lorsqu'elles sont non nulles doivent cependant être exprimées. De même, les contraintes sur les instances d'associations doivent être prises en compte. Dans ce cas, la situation dépend de l'existence ou non d'un mécanisme de traduction de contraintes dans le SGBD.

227. Conclusion

Ainsi que nous l'avons vu, le modèle relationnel permet mal de traduire le schéma exprimé en modèle Entité-Association enrichi: non prise en compte de la relation de généralisation, non systématisation du traitement des

PARTIE III - EXPERIMENTATION DU MODELE DE STRUCTURE

cardinalités minimales, problème de description des agrégations successives, difficulté de traduction des "boucles", traitement des contraintes dépendant de l'outil SGBD. L'implémentation du modèle de structure de données application dans un SGBD relationnel, implémentation dont nous avons nuancé l'intérêt précédemment, apparaît donc très imparfaite.

PARTIE IV

PERSPECTIVES D'UTILISATION

DU MODELE DE STRUCTURE DE

DONNEES

INTRODUCTION

L'implantation de la totalité de la structure de données application avec un simple système de gestion de base de données est, comme nous l'avons vu en Partie III, très imparfaite et d'un intérêt tout relatif. La seule véritable validation de notre modèle ne pourra être faite que par des intégrations partielles de ce modèle de données dans des traitements de CAO qui restent à développer. En tout état de cause, la validité de notre modèle ne pourra jamais être établie de façon absolue. En effet, si la cohérence du schéma conceptuel peut être vérifiée, sa complétude ne sera toujours que relative aux traitements existants. De nouveaux outils de CAO plus performants pourront donc entraîner une modification du modèle (enrichissement ou affinement).

Pour valider le modèle de structure de données application, il faut donc développer des outils logiciels pour automaticiens plus riches que ceux existant actuellement et s'appuyant explicitement sur le modèle. Cependant, s'il est vrai que la définition du modèle BASEPTA constitue une première étape, le chemin à parcourir pour la création de ces outils PTA est encore long. Il passe par la résolution préalable d'un certain nombre de points théoriques. L'objectif de cette Partie IV est d'énoncer brièvement les plus importantes de ces nouvelles voies de recherche.

1. REALISATION D'UN SGBD SUPPORTANT LE FORMALISME ENTITE-ASSOCIATION ENRICHI.

11. Quel type de SGBD pour implémenter le schéma conceptuel de structure de données ?

Nous avons souligné précédemment la perte sémantique résultant de la dérivation de notre modèle dans un formalisme relationnel. L'essentiel de ce qui fait la richesse de l'outil de modélisation est peu ou mal pris en compte. Des tentatives sont actuellement en cours dans diverses entreprises partenaires des projets PTA pour dériver le modèle dans d'autres formalismes.

Le modèle NIAM [NIJ 81A] a la faveur de certains. De notre point de vue, le caractère peu structuré ("à plat") des schémas conceptuels NIAM est redhibitoire. Le schéma devient très éclaté et très difficile d'appréhension. Par ailleurs le modèle NIAM nous paraît très intéressant s'il est utilisé pour bâtir le modèle conceptuel dès le début de l'étude. Par contre, il est très difficile d'obtenir une transcription fidèle d'un schéma exprimé avec l'outil Entité-Association enrichi en un schéma NIAM.

Une modélisation inspirée des langages à objets [BAI 87A] peut aussi apparaître intéressante. Mais, si elle permet une bonne exploitation des relations de généralisation, elle traduit par contre très difficilement la relation d'agrégation.

La solution que nous allons évoquer ici est la réalisation d'un SGBD supportant spécifiquement le formalisme Entité-Association enrichi.

12. Spécifications générales du LMD d'un SGBD Entité-Association enrichi.

En théorie, le formalisme Entité-Association enrichi étant non ambigu, la description fonctionnelle des primitives de manipulation d'un schéma conceptuel exprimé dans un tel formalisme est directement explicitable. Cependant, afin de faciliter le travail de futurs réalisateurs de SGBD, des travaux sont en cours pour définir le langage de manipulation de données (LMD) d'un SGBD Entité-Association enrichi [BOU 89A]. Nous donnons ci-après les grandes lignes de ce langage.

PARTIE IV - PERSPECTIVES D'UTILISATION DU MODELE DE DONNEES

Les primitives générales de manipulation des données sont les suivantes :

niveau 0	RECHERCHE D'UNE INSTANCE D'ENTITE			
niveau 1	CREATION D'UNE INSTANCE D'ENTITE	SUPPRESSION D'UNE INSTANCE D'ENTITE	LECTURE D'UNE INSTANCE D'ENTITE	RECHERCHE D'UNE INSTANCE D'ASSOCIATION
niveau 2	MODIFICATION D'UNE INSTANCE D'ENTITE	CREATION D'UNE INSTANCE D'ASSOCIATION	SUPPRESSION D'UNE INSTANCE D'ASSOCIATION	LECTURE D'UNE INSTANCE D'ASSOCIATION
niveau 3	MODIFICATION D'UNE INSTANCE D'ASSOCIATION			

Ces primitives traiteront le cas des:

- entités non agrégées et non généralisées
- entités généralisées
- associations non agrégation et non généralisation
- entités agrégées avec leurs relations d'agrégation.

Elles sont donc différentes suivant ces quatre types d'objets manipulés, mais bien sur restent indépendantes de la sémantique des objets du modèle BASEPTA. Ainsi la primitive de création d'une instance de l'entité VL est fonctionnellement la même que celle de création d'une instance d'EDL puisqu'il s'agit de deux données de même nature.

Remarquons que la relation de généralisation ne donne pas lieu à des primitives de manipulation autre que celles concernant une entité généralisée ou sous-classe. La relation d'agrégation et les entités qu'elle relie sont traitées globalement.

A ces primitives, viendront s'ajouter des primitives de vérification de cohérence s'assurant que les contraintes d'exclusion, par exemple, sont bien vérifiées ainsi que les cardinalités. En effet, compte-tenu de la complexité du schéma, il semble souhaitable de ne pas contraindre au respect des cardinalités lors de la création des instances.

La manipulation de la structure de données serait alors très complexe. Ainsi, pour créer une instance de CR/AP, il faudrait simultanément créer une instance de CME, de CMA, de AA/C de TA et de CF et relier convenablement ces différentes instances.

2. LA BASE DOCUMENTATION

Nous avons déjà amplement montré le caractère abstrait des données de notre modèle et indiqué qu'il était nécessaire de les relier aux données de représentation.

Le schéma de la figure 78 page II-74 a permis d'exprimer une partie des liens entre les données de la fourniture matérielle et leur représentation dans les schémas. La représentation des constituants matériels est de fait la plus simple à relier à notre modèle. Mais il faut aussi établir les liens entre le logiciel et sa documentation (listings de programme) et bien sûr entre les données de structuration et l'ensemble des documents d'analyse et de conception (architecture, étude opérationnelle, étude fonctionnelle). Ce dernier point est incontestablement le plus délicat. C'est ce que l'on peut comprendre sur l'exemple suivant: Un lien entre ERS et un objet de documentation type "étape de grafcet" n'a pas de sens de notre point de vue. La solution la meilleure serait de relier des graphes connexes globalement à des instances d'ERS. Mais ces liens sont très difficiles à établir et souvent peu exploitables.

Il faut donc conduire une analyse approfondie de la documentation et de ses données. De tels travaux ont déjà été entamés dans le cadre du projet Techno-X [TEC 88A]. Mais le modèle BASEPTA n'était alors pas entièrement achevé. Il est donc nécessaire maintenant de poursuivre ces travaux dans le sens d'une étude des liens entre données de documentation et données application après avoir identifié puis écarté les données purement de documentation (références croisées, renvois inter-folios, numérotation de folios...).

3. LA BASE CONSTITUANTS

Le développement d'outils CAO pour automaticien ne saurait être possible sans une étude approfondie sur la modélisation de catalogues de constituants.

Dans la partie II, nous avons décrit des données en liaison avec le catalogue. Le schéma de la figure 134 fournit un point de départ pour l'établissement du schéma conceptuel d'une base de constituants matériels.

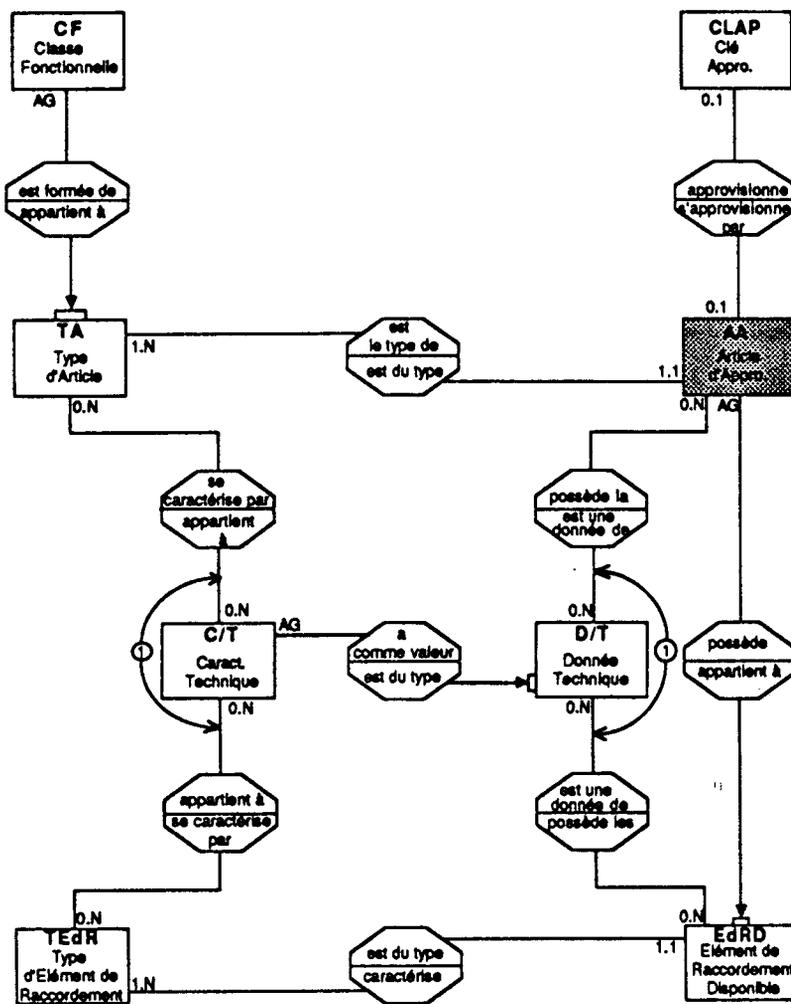


Fig 134

Dans le cas de la Base Constituants les entités du schéma de la figure 134 prennent une signification légèrement différente de celle exposée en Partie II puisqu'il ne s'agit plus d'objets application mais d'objets de catalogues. Tel est le cas par exemple de l'entité Article d'Approvisionnement. Cependant afin d'établir plus facilement les liens entre Base Application et Base Constituants, il est souhaitable d'avoir pour ces deux structures de données des schémas conceptuels aussi proches que possibles.

Le schéma de la figure 134 ne constitue qu'un point de départ dans le domaine des constituants matériels. La réflexion sur la Base Constituants doit être plus large et respecter les remarques suivantes:

1 - Les constituants logiciels (EDLB) doivent être pris en compte au même titre que les constituants matériels.

2 - La réflexion sur la Base Constituants doit être menée simultanément avec la réflexion sur la Base Savoir-Faire (cf I - 63). En effet, la différence entre ces deux structures de données est circonstancielle et relative à un état de l'offre à un moment donné. Remarquons d'ailleurs que l'évolution actuelle consiste en l'intégration croissante de savoir-faire dans les constituants. Ceci revient à dire que les données vont évoluer de la Base Savoir-faire vers la Base Constituants. Tel est par exemple le cas du contacteur intégral, du variateur de vitesse et de bien d'autres constituants. Conceptuellement les données de ces deux bases sont donc très voisines. Cependant la Base Constituants est le domaine du fournisseur, la Base Savoir-Faire celui de l'utilisateur.

3 - La réflexion sur la Base Constituants peut valablement être conduite avec simultanément la spécification puis la réalisation de traitements CAO du type "Assistance au choix de constituants". C'est le problème du catalogue à choix assisté par ordinateur. Ceci conforte les réflexions précédentes sur les Bases Constituants et Savoir-Faire.

4. PRODUITS, OUTILS ET ATELIER PTA

Après avoir dégagé les principaux axes de travail futurs, nous allons, pour conclure, émettre quelques hypothèses et conjectures sur l'évolution de l'offre en produits CAO. Cependant, il semble auparavant nécessaire de revenir au problème de la définition du standard d'échange dont nous avons précédemment dit qu'il pourrait être une suite immédiate de la définition du modèle de structure de données application.

Aujourd'hui, la définition d'un standard d'échange pour les données application nous semble sans intérêt immédiat. Pour le montrer, reprenons le cas du standard SET [SET 85A] et de son extension à la schématique SES. Tant que les données manipulées par les CAO n'étaient que graphiques, le standard SET classique était amplement suffisant. Lorsque les outils de schématique ont commencé à prendre en compte la notion de logique de connexion, la nécessité d'un standard d'échange de données de connectique et de schématique s'est faite jour. Or les CAO actuelles ne manipulent pratiquement aucune donnée application. Le standard d'échange de données application n'est donc pas une nécessité. Seul un cas semble présenter un intérêt. Il s'agit de l'échange variable automate / voie automate entre une CAO d'automatisme programmé et un logiciel de schématique. Remarquons d'ailleurs qu'il s'agit là quasiment des seules données application clairement identifiées dans les CAO actuelles. Dans ce cas, il est possible de définir une syntaxe ou format neutre pour ces données, c'est à dire établir une représentation physique des données organisées logiquement suivant la partie de notre modèle adéquate.

Les perspectives d'avenir concernant le problème PTA sont les suivantes:

A terme, les automaticiens disposeront d'ateliers PTA ou ateliers de Génie Automatique, c'est à dire d'ensembles intégrés d'outils constitués ou non à partir de produits disponibles sur le marché permettant la production de systèmes automatisés. Les outils intégrés seront des outils PTA, c'est à dire respectant les concepts PTA que nous avons déjà présentés tout au long de ce mémoire. Ces concepts peuvent se résumer de la manière suivante:

- Prise en compte de la totalité du cycle de vie.
- Existence de méthodes intégrées et évolutives.

- Existence d'une Base Application permanente et enrichie tout au long du cycle de vie.
- Existence d'une Base Constituants et d'une Base Savoir-Faire.
- Respect d'organisations standardisées pour les bases de données afin de permettre les échanges de données inter-outils.

L'atelier PTA est donc une notion utilisateur (cf schéma de la figure 135). Le fournisseur de CAO réalise des produits PTA que l'utilisateur intègre dans son atelier en tant qu'outils PTA (cf schéma de la figure 136).

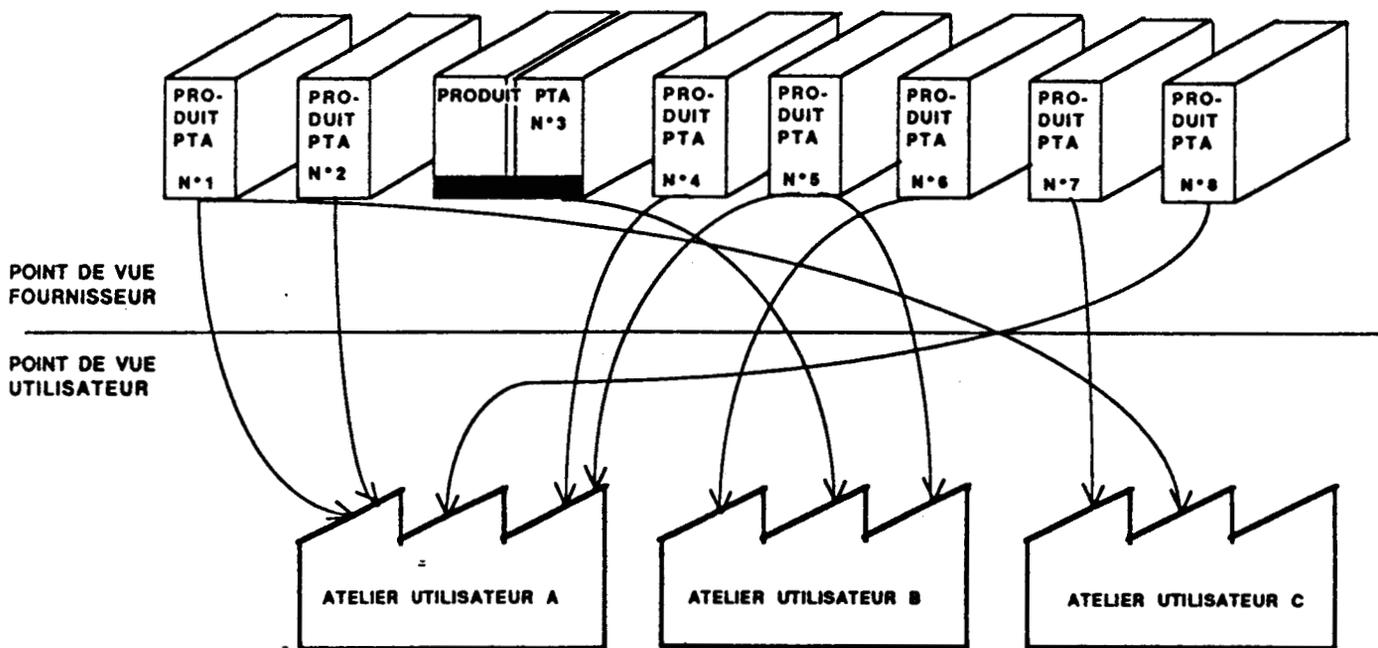


Fig 135

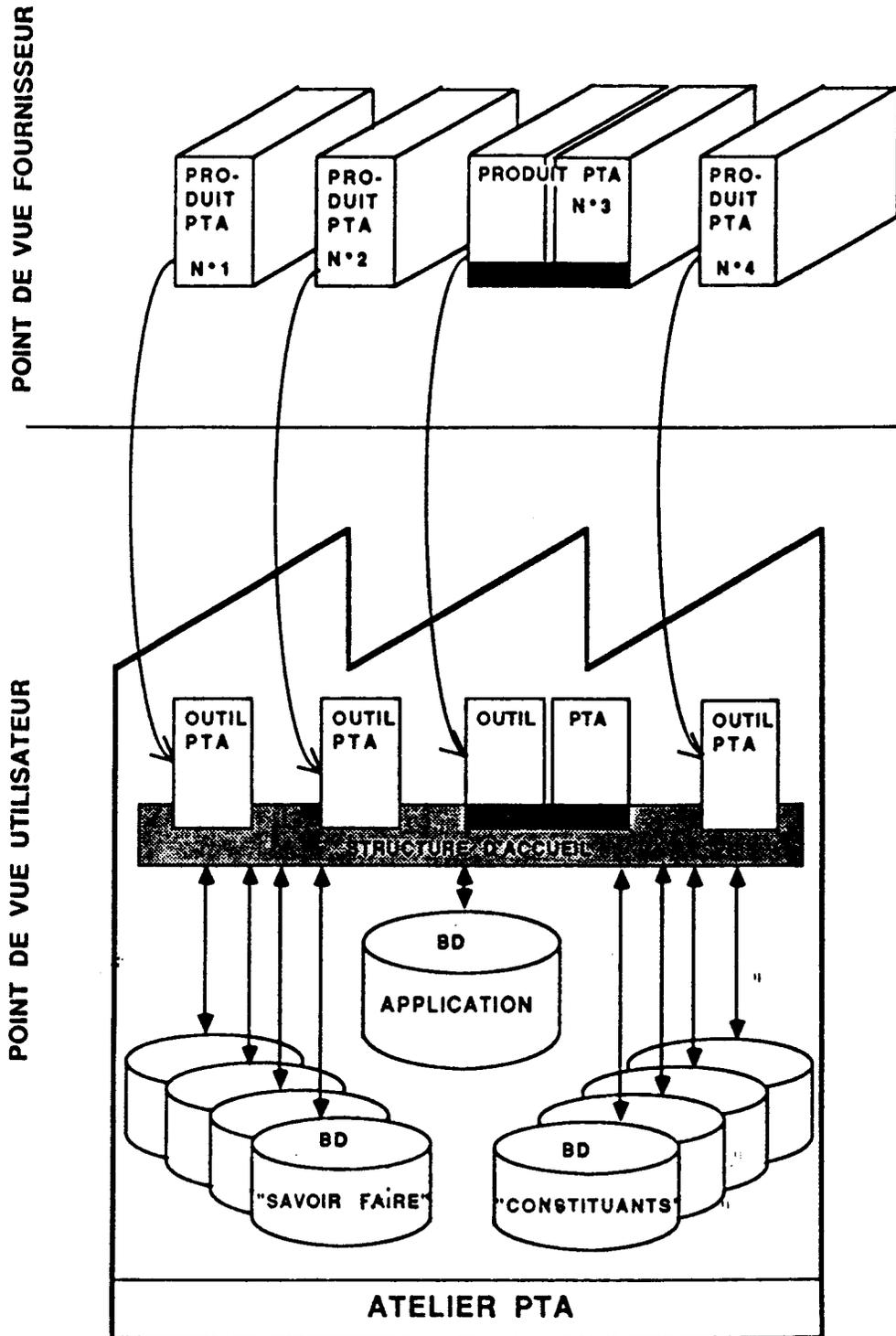


Fig 136

Le marché de la CAO offrira donc des éléments de constitution de l'atelier PTA, cette constitution restant à la charge des utilisateurs de l'atelier. L'analogie avec le Génie Logiciel est donc ici flagrante.

L'essentiel des futures recherches consiste donc à se concentrer sur les conditions nécessaires d'une bonne intégration, c'est à dire les bases de données Constituants et Savoir-Faire déjà évoquées, mais aussi la structure d'accueil dans laquelle les primitives de manipulation de données (cf section IV - 1) jouent un rôle important.

C O N C L U S I O N

CONCLUSION

Le modèle de structure de données application que nous avons présenté dans ce mémoire, offre un niveau de détail assez important. Cette finesse nous autorise légitimement à penser qu'il favorisera le développement de traitements de CAO performants et communicants. Mais cette complexité a aussi pour conséquence la difficulté de communication du modèle. Or, pour sa bonne utilisation, il est essentiel que le modèle soit compris et "partagé" par les développeurs de CAO et les fabricants de consoles automate. Ce mémoire a tenté de contribuer à cette compréhension.

Les développements à venir d'outils logiciels PTA permettront de valider le concept de Base Application et d'apprécier l'exactitude de notre modèle conceptuel de cette Base Application. Sur ces deux points, compte-tenu de l'existant en terme d'outils de CAO pour automaticien, il est aujourd'hui impossible de se prononcer de façon catégorique.

Notre travail a permis également de souligner, si besoin était, le manque cruel de méthodes de conception de systèmes automatisés. Le développement des outils de Génie Automatique doit commencer obligatoirement par un travail de définition, de mise au point et d'harmonisation des méthodes de conception. Ce constat doit inciter à la plus grande modestie face au problème du CIM (Computer Integrated Manufacturing). La démarche intégrée produit, procédé, processus, partie opérative, partie commande est effectivement d'une grande complexité. Les problèmes posés par la conception d'automatismes de bonne qualité n'abordant que les deux derniers niveaux de la démarche sont encore bien loin d'être maîtrisés. De ce fait, il ne faut pas espérer à moyen terme de solutions aux problèmes plus généraux de la productique.

En tout état de cause, il reste aux chercheurs beaucoup de travail dans le domaine des méthodes d'automatismes avant de disposer de bases solides pour la réalisation de produits logiciels intégrant la démarche productique complète.

B I B L I O G R A P H I E

(C L A S S E M E N T P A R O R D R E A L P H A B E T I Q U E)

BIBLIOGRAPHIE (CLASSEMENT PAR ORDRE ALPHABETIQUE)

- [AFC 87A] AFCET - GROUPE SYSTEMES LOGIQUES - Formalisation d'extension du grafcet macro-étape et forçage - Electronique industrielle n°128 - 15.06.87
- [AUT 87A] CLUB AUTOMATION - Compte-rendu de la journée d'information et de ses débats. Les outils de l'automaticien. 2 ème journée "Le cahier des charges" 9 décembre 1987
- [BAI 87A] BAILLY C. - CHALLINE J.F. - FERRI J.C. - GLOESS P. - MARCHESIN B. - Les langages orientés objets - CEPADUES EDITIONS - 1987
- [BAS 88A] PROJET BASEPTA - Plaquette de présentation du projet AUTOMATION 88 - Stand du MIPTT/SERICS
- [BAS 88B] Equipe BASEPTA: FRACHET J.P.- ALANCHE P.- DOLLE F.- BOUTEILLE N.- DELARBRE T. - DIVERCHY F.- PERRIEN H. SAIDI A.- Modèle conceptuel des données en automatisation: résultats des travaux du groupe BASEPTA Congrès AFCET Automatique - Grenoble 10 Octobre 1988
- [BAS 89A] EQUIPE TECHNIQUE BASEPTA - Documents de fin de projet - Fasc. 0 : Objectifs et principes généraux
Fasc. 1 : Modèle conceptuel version 3.1
Fasc. 2 : Commentaires sur le modèle conceptuel version 3.1
Fasc. 3 : Illustration des évolutions de la base sur un exemple
Fasc. 4 : Outil de modélisation
5/01/1989
- [BER 85A] BERTRAND D. - BOUTEILLE D. - COLLOT R. - GARNIER J.C. - HENAU J.F. - Les automatismes électropneumatiques et pneumatiques - Editions de l'Usine Nouvelle - 1985
- [BLA 79A] BLANCHARD M. - Comprendre, maîtriser et appliquer le Grafcet - CEPADUES EDITION - 1979
- [BOU 87A] BOUTEILLE D. - BOUTEILLE N. - CHANTREUIL S. - COLLOT - FRACHET J.P. - LE GRAS H. - MERLAUD C. - SELLOSSE J. - SFAR A. - Les automatismes programmables - CEPADUES EDITION - Septembre 1987
- [BOU 89A] BOUTEILLE N. Primitives d'accès à BASEPTA - Document interne TELEMECANIQUE - 1989

BIBLIOGRAPHIE (CLASSEMENT PAR ORDRE ALPHABETIQUE)

- [BOS 79A] BOSSY J.C. - BRARD P. - FAUGERE P. - MERLAUD C. -
Le Grafcet, sa pratique et ses applications
EDUCALIVRE - 1979
- [BRA 87A] BRARD P. - Structure d'un dossier d'affaire TELEMECANIQUE - Document interne TELEMECANIQUE.
- [BRA 88A] BRARD P. - Outil de description des automatismes séquentiels - le GRAFCET - TECHNIQUES DE L'INGENIEUR - 1988
- [CAD 87A] CADEPA Automatisation assistée par ordinateur - Documentation produit - SGN VALORIS - 1987
- [CAL 87A] CALIN Progiciel de Conception assistée des Automatismes Logiques Industriels par analyse Grafcet - Documentation produit - CYBERNETIX - 1987
- [CHE 80A] CHEN P. Entity-Relationships approach to systems analysis and design - NORTH HOLLAND Publishing company 1980
- [CIT 88A] AUTOMOBILES CITROEN - Cahier des charges général. Automatisation des moyens de production. Document interne Citroen - Version 2.2 - juin 88
- [CJG 87A] CJGRAF Le Grafcet à la carte - Documentation produit - Version 1.0 - CJB AUTOMATION - 1987
- [FRA 86A] FRACHET J.P.- ARRIOLA I. - PAILLER C.
Une modélisation de la prestation d'automatisation pour les processus discontinus. Application à la structuration du dossier - 1986
- [FRA 87A] FRACHET J.P. Une introduction au génie automatique. Application à la faisabilité d'une chaîne intégrée d'outils CAO pour la conception et l'exploitation des machines automatiques industrielles
Thèse de doctorat d'état - Avril 1987
- [FRA 87B] FRACHET J.P. - ALANCHE P. - Schéma conceptuel d'une Base de Données pour une CAO d'équipements de production automatisée - janvier 87

BIBLIOGRAPHIE (CLASSEMENT PAR ORDRE ALPHABETIQUE)

- [GRA 87A] GRAL L'atelier logiciel des automates programmables -
Documentation produit - SYGRAL - 1987
- [GRE 85A] GREPA - Le Grafcet. De nouveaux concepts -
Cepadues Editions - 1985
- [IEC 848] NORME CEI - Publication 848 - Première édition -
Etablissement des diagrammes fonctionnels pour système
de commande - 1987
- [ISO 82A] ISO TC97/SC5/WG3 - Concepts and terminology for the
conceptual schema and the information base - 1982
- [LAM 87A] LAMBERT - SALVI - CAO Biens d'équipements. Eléments du
cahier des charges Automatismess
Document interne RENAULT - 18/02/87
- [NIJ 81A] NIJSSEN - An architecture for knowledge base software -
Australian computer society
- [OME 87A] OMEGA Un atelier logiciel pour automaticiens
Documentation produit - 3IP - 1987
- [PER 89A] PERRIEN H. - Application Database of a CAD system for
Automation Engineering. Promises and weaknesses of a
solution : the BASEPTA model - Congrès AIPAC'89 - Nancy
3 - 5 Juillet 1989
- [PLA 88A] PLANCHE R. - Maîtriser la modélisation conceptuelle -
Collection MIPS - MASSON 1988
- [SAU 86A] SAUGE P. - SGBD: Quelle méthode pour le modèle concep-
tuel - Le Monde Informatique - Décembre 1986
- [SET 85A] AFNOR Norme expérimentale Z68-300 Automatisation indus-
trielle - représentation externe des données de défini-
tion des produits - Spécification du standard d'échange
et de transfert (SET) - Version 85-08 - Aout 85
- [SOU 87A] SOURISSE C.- Pourquoi et comment établir un cahier des
charges- Forum de la Production automatisée - ARLES
26 et 27 Octobre 1987

BIBLIOGRAPHIE (CLASSEMENT PAR ORDRE ALPHABETIQUE)

- [SOU 89A] SOURISSE C. - Principes et méthodes d'élaboration du Cahier des Charges des automatismes industriels - AUTOMATION 89 - 17-19 Mai 1989
- [PTA 87A] CAO pour automatismes - PTA - Poste de Travail pour l'Automatisation - Etude réalisée par le Club utilisateur RNUR MICHELIN ADEPA PSA SGN avec le concours du MRES - 8 janvier 1987
- [TEL 82A] TELEMECANIQUE PNEUMATIQUE - Manuel des automatismes à commande pneumatique - janvier 1982
- [TEC 88A] PROJET TECHNO-X - PERRIEN H. - BRUEL .P - MAYEUR L. - Document de synthèse - 7 septembre 1988

B I B L I O G R A P H I E

(C L A S S E M E N T P A R T H E M E)

OUVRAGES GENERAUX

- [BOU 87A] BOUTEILLE D. - BOUTEILLE N. - CHANTREUIL S. - COLLOT -
FRACHET J.P. - LE GRAS H. - MERLAUD C. - SELLOSSE J. -
SFAR A. - Les automatismes programmables -
CEPADUES EDITION - Septembre 1987
- [BER 85A] BERTRAND D. - BOUTEILLE D. - COLLOT R. - GARNIER J.C. -
HENAU J.F. - Les automatismes électropneumatiques et
pneumatiques - Editions de l'Usine Nouvelle - 1985
- [CIT 88A] AUTOMOBILES CITROEN - Cahier des charges général.
Automatisme des moyens de production.
Document interne Citroen - Version 2.2 - juin 88
- [FRA 86A] FRACHET J.P.- ARRIOLA I. - PAILLER C.
Une modélisation de la prestation d'automatisation pour
les processus discontinus. Application à la structura-
tion du dossier - 1986
- [FRA 87A] FRACHET J.P. Une introduction au génie automatique.
Application à la faisabilité d'une chaîne intégrée
d'outils CAO pour la conception et l'exploitation des
machines automatiques industrielles
Thèse de doctorat d'état - Avril 1987
- [TEL 82A] TELEMECANIQUE PNEUMATIQUE - Manuel des automatismes à
commande pneumatique - janvier 1982

GRAF CET

- [GRE 85A] GREPA - Le Grafcet. De nouveaux concepts -
Cepadues Editions - 1985
- [AFC 87A] AFCET - GROUPE SYSTEMES LOGIQUES - Formalisation d'ex-
tension du grafcet macro-étape et forçage -
Electronique industrielle n°128 - 15.06.87
- [BLA 79A] BLANCHARD M. - Comprendre, maitriser et appliquer le
Grafcet - CEPADUES EDITION - 1979
- [BOS 79A] BOSSY J.C. - BRARD P. - FAUGERE P. - MERLAUD C. -
Le Grafcet, sa pratique et ses applications
EDUCALIVRE - 1979

BIBLIOGRAPHIE (CLASSEMENT PAR THEMES)

- [BRA 88A] BRARD P. - Outil de description des automatismes séquentiels - le GRAFCET - TECHNIQUES DE L'INGENIEUR - 1988

STRUCTURE DE DONNEES APPLICATION

- [PER 89A] PERRIEN H. - Application Database of a CAD system for Automation Engineering. Promises and weaknesses of a solution: The BASEPTA model - Congrès AIPAC'89 - Nancy 3 - 5 juillet 1989
- [FRA 87B] FRACHET J.P. - ALANCHE P. - Schéma conceptuel d'une Base de Données pour une CAO d'équipements de production automatisée - janvier 87
- [BAS 88A] PROJET BASEPTA - Plaquette de présentation du projet AUTOMATION 88 - Stand du MIPTT/SERICS
- [BAS 88B] Equipe BASEPTA: FRACHET J.P.- ALANCHE P.- DOLLE F.- BOUTEILLE N.- DELARBRE T. - DIVERCHY F.- PERRIEN H. SAIDI A.- Modèle conceptuel des données en automatisation: résultats des travaux du groupe BASEPTA Congrès AFCET Automatique - Grenoble 10 Octobre 1988
- [BAS 89A] EQUIPE TECHNIQUE BASEPTA - Documents de fin de projet
Fasc. 0: Objectifs et principes généraux
Fasc. 1: Modèle conceptuel version 3.1
Fasc. 2: Commentaires sur le modèle conceptuel version 3.1
Fasc. 3: Illustration des évolutions de la base sur un exemple
Fasc. 4: Outil de modélisation
5/01/1989

PTA

- [PTA 87A] CAO pour automatismes - PTA - Poste de Travail pour l'Automatisation -
Etude réalisée par le Club utilisateur RNUR MICHELIN ADEPA PSA SGN avec le concours du MRES - 8 janvier 1987

BIBLIOGRAPHIE (CLASSEMENT PAR THEMES)

NORMES ET STANDARDS

- [IEC 848] NORME CEI - Publication 848 - Première édition -
Etablissement des diagrammes fonctionnels pour système
de commande - 1987
- [ISO 82A] ISO TC97/SC5/WG3 - Concepts and terminology for the
conceptual schema and the information base - 1982
- [SET 85A] AFNOR Norme expérimentale Z68-300 Automatisation indus-
trielle - représentation externe des données de défini-
tion des produits - Spécification du standard d'échange
et de transfert (SET) - Version 85-08 - Aout 85
- [BRA 87A] BRARD P. - Structure d'un dossier d'affaire TELEMECA-
NIQUE - Document interne TELEMECANIQUE.

CAHIER DES CHARGES

- [SOU 87A] SOURISSE C.- Pourquoi et comment établir un cahier des
charges- Forum de la Production automatisée - ARLES
26 et 27 Octobre 1987
- [SOU 89A] SOURISSE C. - Principes et méthodes d'élaboration du
Cahier des Charges des automatismes industriels -
AUTOMATION 89 - 17-19 Mai 1989
- [AUT 87A] CLUB AUTOMATION - Compte-rendu de la journée d'informa-
tion et de ses débats. Les outils de l'automaticien.
2 ème journée "Le cahier des charges"
9 décembre 1987

PRODUITS COMMERCIALISES

- [CAD 87A] CADEPA Automatisation assistée par ordinateur -
Documentation produit - SGN VALORIS - 1987
- [CAL 87A] CALIN Progiciel de Conception assistée des Automatismes
Logiques Industriels par analyse Grafcet - Documenta-
tion produit - CYBERNETIX - 1987
- [CJG 87A] CJGRAF Le Grafcet à la carte - Documentation produit -
Version 1.0 - CJB AUTOMATION - 1987

BIBLIOGRAPHIE (CLASSEMENT PAR THEMES)

[GRA 87A] GRAL L'atelier logiciel des automates programmables -
Documentation produit - SYGRAL - 1987

[OME 87A] OMEGA Un atelier logiciel pour automaticiens
Documentation produit - 3IP - 1987

CAO ELECTRIQUES

[LAM 87A] LAMBERT - SALVI - CAO Biens d'équipements. Eléments du
cahier des charges Automatismess
Document interne RENAULT - 18/02/87

[TEC 88A] PROJET TECHNO-X - PERRIEN H. - BRUEL .P - MAYEUR L. -
Document de synthèse - 7 septembre 1988

BASE DE DONNEES

[BOU 89A] BOUTEILLE N. Primitives d'accès à BASEPTA -
Document interne TELEMECANIQUE - 1989

[CHE 80A] CHEN P. Entity-Relationships approach to systems
analysis and design - NORTH HOLLAND Publishing company
1980

[NIJ 81A] NIJSSSEN - An architecture for knowledge base software -
Australian computer society

[PLA 88A] PLANCHE R. - Maîtriser la modélisation conceptuelle -
Collection MIPS - MASSON 1989

[SAU 86A] SAUGE P. - SGBD: Quelle méthode pour le modèle concep-
tuel - Le Monde Informatique - Décembre 1986

DIVERS

[BAI 87A] BAILLY C. - CHALLINE J.F. - FERRI J.C - GLOESS P. -
MARCHESIN B. - Les langages orientés objets -
CEPADUES EDITIONS - 1987

L I S T E D E S

A B R E V I A T I O N S U S U E L L E S

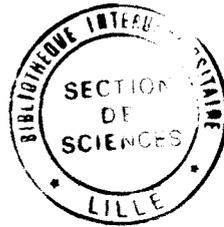
LISTE DES ABREVIATIONS USUELLES

NOTA: Les désignations d'entités du modèle conceptuel sont précédées d'un astérisque

AA	*Article d'Approvisionnement
ALGA	Atelier Logiciel de Génie Automatique
APPLI	*Application
B	*Bibliothèque
BASEPTA	Base Application et Standard d'Echange pour le Poste de Travail de l'Automaticien
BD	Base de Données
BEA	Bureau d'Etudes en Automatismes
C	*Caractéristique
CAO	Conception Assistée par Ordinateur
CDAO	Conception et Dessin Assistés par Ordinateur
CDC	Cahier Des Charges
CEI	Comission Electrotechnique Internationale
CF	*Classe Fonctionnelle
CL	*Configuration Logicielle
CLAP	*Clé d'Approvisionnement
CMA	*Constituant Matériel Application
CME	*Constituant Matériel Elémentaire
CNOMO	Comité Normatif Outillages Machines-Outils
COMP	*Comportement
CONN	*Connexion
CORPS	*Corps
CR	*Constituant Réel
CSAO	Conception de Schémas Assistée par Ordinateur
D	*Donnée
DAO	Dessin Assisté par Ordinateur
DATE	*Date
DES	*Description
EDL	*Elément De Logiciel
EDLB	*Elément De Logiciel de Bibliothèque
EDONA	*Ensemble de DONnées Annexes
EDR	*Elément De Raccordement
EDRD	*Elément De Raccordement Disponible
EL	*Ensemble Logiciel
ERA	*Elément Référençable de l'Application
ERS	*Elément Résultant de Structuration
EVC	*Equivalleur de Connexion
GEMMA	Guide d'Etude des Modes de Marches et d'Arrêt
LDD	Langage de Description de Données
LI	*Logiciel Implanté
LMD	Langage de Manipulation de Données
LP	*Liaison Physique
MIPTT	Ministère de l'Industrie et des PTT
MRES	Ministère de la Recherche et de l'Enseignement Supérieur
OCG	*Objectifs et Caractéristiques Globaux
PC	Partie Commande
PO	Partie Opérative
PTA	Poste de Travail pour l'Automaticien ou Poste de Travail pour l'Automatisation

LISTE DES ABREVIATIONS USUELLES

PTA-E	Poste de Travail pour l'Automaticien - Europe ou Processes and Tools for Automation Engineering
REFEX	*REFérence EXterne
RS	*Référentiel Site
RSU	*Ressource Système Utilisée
SE	*Système d'Exploitation
SET	Standard d'Echange et de Transfert
SGBD	Système de Gestion de Base de Données
TA	*Type d'Article
TE	TELEMECANIQUE ELECTRIQUE
TEDR	*Type d'Elément De Raccordement
VDC	*Variable De Comportement
VL	*Variable Logicielle
VLG	*Variable Logicielle Générique



RESUME

Le développement de la productique et la complexité croissante des systèmes automatisés industriels rendent nécessaire la réalisation d'outils logiciels de CAO performants à l'usage de l'automaticien. L'intégration de ces outils dans un atelier logiciel dit "Atelier PTA" (Poste de Travail pour l'Automatisation) suppose la prise en compte de la totalité du cycle de vie de l'application automatisée. La cohérence des données décrivant le système automatisé doit être impérativement assurée depuis les spécifications jusqu'aux phases d'exploitation et de maintenance en passant par les différentes étapes de conception. La solution est la constitution d'une structure de données dite "application" enrichie, modifiée et exploitée par les différents traitements de CAO.

Ce mémoire présente un modèle de niveau conceptuel de la structure de données application d'une CAO d'automatismes. Exprimé à l'aide d'un formalisme enrichi à partir de l'outil de modélisation Entité - Association, ce modèle est illustré et appliqué à un exemple d'équipement de production industriel.

MOTS-CLEFS

SYSTEMES AUTOMATISES

CAO D'AUTOMATISMES

CYCLE DE VIE

STRUCTURE DE DONNEES

MODELE CONCEPTUEL

ENTITES ASSOCIATIONS

METHODE DE CONCEPTION

ATELIER DE GENIE AUTOMATIQUE