



HAL
open science

**Méthodologie pour la construction de systèmes CAO :
SIGMA-CAO : une méthode globale pour la
spécification, la mise en œuvre et l'utilisation de
systèmes intégrés pour la conception assistée par
ordinateur**

Bertrand David

► **To cite this version:**

Bertrand David. Méthodologie pour la construction de systèmes CAO : SIGMA-CAO : une méthode globale pour la spécification, la mise en œuvre et l'utilisation de systèmes intégrés pour la conception assistée par ordinateur. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1981. tel-00295241

HAL Id: tel-00295241

<https://theses.hal.science/tel-00295241>

Submitted on 11 Jul 2008

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

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

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

et à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR ES SCIENCES

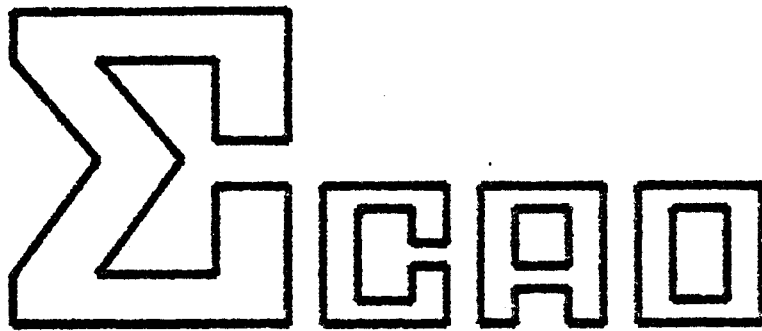
"Informatique"

par

Bertrand DAVID

**METHODOLOGIE
POUR LA CONSTRUCTION DE SYSTEMES CAO :
SIGMA - CAO**

une méthode globale pour la spécification,
la mise en œuvre et l'utilisation de systèmes intégrés
pour la Conception Assistée par Ordinateur.



Thèse soutenue le 30 septembre 1981 devant la Commission d'Examen :

Monsieur L. BOLLIET : Président

Messieurs J.M. BRUN
C. DELOBEL
M. GRIFFITHS
M. LUCAS
J. MERMET
P. ROMAND

} Examineurs

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président

Monsieur Joseph KLEIN : Vice-Président

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

| | | |
|-----|------------------------|---|
| MM. | AMBLARD Pierre | Clinique de dermatologie |
| | ARNAUD Paul | Chimie |
| | ARVIEU Robert | I.S.N. |
| | AUBERT Guy | Physique |
| | AYANT Yves | Physique approfondie |
| Mme | BARBIER Marie-Jeanne | Electrochimie |
| MM. | BARBIER Jean-Claude | Physique expérimentale |
| | BARBIER Reynold | Géologie appliquée |
| | BARJON Robert | Physique nucléaire |
| | BARNOUD Fernand | Biosynthèse de la cellulose |
| | BARRA Jean-René | Statistiques |
| | BARRIE Joseph | Clinique chirurgicale A |
| | BEAUDOING André | Clinique de pédiatrie et puériculture |
| | BELORIZKY Elie | Physique |
| | BARNARD Alain | Mathématiques pures |
| Mme | BERTRANDIAS Françoise | Mathématiques pures |
| MM. | BERTRANDIAS Jean-Paul | Mathématiques pures |
| | BEZES Henri | Clinique chirurgicale et traumatologie |
| | BLAMBERT Maurice | Mathématiques pures |
| | BOLLIET Louis | Informatique (I.U.T. B) |
| | BONNET Jean-Louis | Clinique ophtalmologie |
| | BONNET-EYMARD Joseph | Clinique hépato-gastro-entérologie |
| Mme | BONNIER Marie-Jeanne | Chimie générale |
| MM. | BOUCHERLE André | Chimie et toxicologie |
| | BOUCHEZ Robert | Physique nucléaire |
| | BOUSSARD Jean-Claude | Mathématiques appliquées |
| | BOUTET DE MONVEL Louis | Mathématiques pures |
| | BRAVARD Yves | Géographie |
| | CABANEL Guy | Clinique rhumatologique et hydrologique |
| | CALAS François | Anatomie |
| | CARLIER Georges | Biologie végétale |
| | CARRAZ Gilbert | Biologie animale et pharmacodynamie |

.../...

| | | |
|-----|---------------------|---|
| MM. | CAU Gabriel | Médecine légale et toxicologie |
| | CAUQUIS Georges | Chimie organique |
| | CHABAUTY Claude | Mathématiques pures |
| | CHARACHON Robert | Clinique ot-rhino-laryngologique |
| | CHATEAU Robert | Clinique de neurologie |
| | CHIBON Pierre | Biologie animale |
| | COEUR André | Pharmacie chimique et chimie analytique |
| | COUDERC Pierre | Anatomie pathologique |
| | DEBELMAS Jacques | Géologie générale |
| | DEGRANGE Charles | Zoologie |
| | DELORMAS Pierre | Pneumophtisiologie |
| | DEPORTES Charles | Chimie minérale |
| | DESRE Pierre | Métallurgie |
| | DODU Jacques | Mécanique appliquée (I.U.T. I) |
| | DOLIQUE Jean-Michel | Physique des plasmas |
| | DREYFUS Bernard | Thermodynamique |
| | DUCROS Pierre | Cristallographie |
| | FONTAINE Jean-Marc | Mathématiques pures |
| | GAGNAIRE Didier | Chimie physique |
| | GALVANI Octave | Mathématiques pures |
| | GASTINEL Noël | Analyse numérique |
| | GAVEND Michel | Pharmacologie |
| | GEINDRE Michel | Electroradiologie |
| | GERBER Robert | Mathématiques pures |
| | GERMAIN Jean-Pierre | Mécanique |
| | GIRAUD Pierre | Géologie |
| | JANIN Bernard | Géographie |
| | KAHANE André | Physique générale |
| | KLEIN Joseph | Mathématiques pures |
| | KOSZUL Jean-Louis | Mathématiques pures |
| | KRAVTCHENKO Julien | Mécanique |
| | LACAZE Albert | Thermodynamique |
| | LACHARME Jean | Biologie végétale |
| Mme | LAJZEROWICZ Janine | Physique |
| MM. | LAJZEROWICZ Joseph | Physique |
| | LATREILLE René | Chirurgie générale |
| | LATURAZE Jean | Biochimie pharmaceutique |
| | LAURENT Pierre | Mathématiques appliquées |
| | LEDRU Jean | Clinique médicale B |
| | LE ROY Philippe | Mécanique (I.U.T. I) |

| | | |
|------|----------------------------|-----------------------------------|
| MM. | LLIBOUTRY Louis | Géophysique |
| | LOISEAUX Jean-Marie | Sciences nucléaires |
| | LONGEQUEUE Jean-Pierre | Physique nucléaire |
| | LOUP Jean | Géographie |
| Mlle | LUTZ Elisabeth | Mathématiques pures |
| MM. | MALINAS Yves | Clinique obstétricale |
| | MARTIN-NOEL Pierre | Clinique cardiologique |
| | MAYNARD Roger | Physique du solide |
| | MAZARE Yves | Clinique Médicale A |
| | MICHEL Robert | Minéralogie et pétrographie |
| | MICOUD Max | Clinique maladies infectieuses |
| | MOURIQUAND Claude | Histologie |
| | MOUSSA André | Chimie nucléaire |
| | NEGRE Robert | Mécanique |
| | NOZIERES Philippe | Spectrométrie physique |
| | OZENDA Paul | Botanique |
| | PAYAN Jean-Jacques | Mathématiques pures |
| | PEBAY-PEYROULA Jean-Claude | Physique |
| | PERRET Jean | Séméiologie médicale (neurologie) |
| | RASSAT André | Chimie systématique |
| | RENARD Michel | Thermodynamique |
| | REVOL Michel | Urologie |
| | RINALDI Renaud | Physique |
| | DE ROUGEMONT Jacques | Neuro-Chirurgie |
| | SARRAZIN Roger | Clinique chirurgicale B |
| | SEIGNEURIN Raymond | Microbiologie et hygiène |
| | SENGEL Philippe | Zoologie |
| | SIBILLE Robert | Construction mécanique (I.U.T. I) |
| | SOUTIF Michel | Physique générale |
| | TANCHE Maurice | Physiologie |
| | VAILLANT François | Zoologie |
| | VALENTIN Jacques | Physique nucléaire |
| Mme | VERAIN Alice | Pharmacie galénique |
| MM. | VERAIN André | Physique biophysique |
| | VEYRET Paul | Géographie |
| | VIGNAIS Pierre | Biochimie médicale |

PROFESSEURS ASSOCIES

MM. CRABBE Pierre
SUNIER Jules

CERMO
Physique

PROFESSEURS SANS CHAIRE

| | | |
|------|-------------------------|--------------------------------|
| Mlle | AGNIUS-DELORS Claudine | Physique pharmaceutique |
| | ALARY Josette | Chimie analytique |
| MM. | AMBROISE-THOMAS Pierre | Parasitologie |
| | ARMAND Gilbert | Géographie |
| | BENZAKEN Claude | Mathématiques appliquées |
| | BIAREZ Jean-Pierre | Mécanique |
| | BILLET Jean | Géographie |
| | BOUCHET Yves | Anatomie |
| | BRUGEL Lucien | Energétique (I.U.T. I) |
| | BUISSON René | Physique (I.U.T. I) |
| | BUTEL Jean | Orthopédie |
| | COHEN-ADDAD Jean-Pierre | Spectrométrie physique |
| | COLOMB Maurice | Biochimie médicale |
| | CONTE René | Physique (I.U.T. I) |
| | DELOBEL Claude | M.I.A.G. |
| | DEPASSEL Roger | Mécanique des fluides |
| | GAUTRON René | Chimie |
| | GIDON Paul | Géologie et minéralogie |
| | GLENAT René | Chimie organique |
| | GROULADE Joseph | Biochimie médicale |
| | HACQUES Gérard | Calcul numérique |
| | HOLLARD Daniel | Hématologie |
| | HUGONOT Robert | Hygiène et médecine préventive |
| | IDELMAN Simon | Physiologie animale |
| | JOLY Jean-René | Mathématiques pures |
| | JULLIEN Pierre | Mathématiques appliquées |
| Mme | KAHANE Josette | Physique |
| MM. | KRAKOWIACK Sacha | Mathématiques appliquées |
| | KUHN Gérard | Physique (I.U.T. I) |
| | LUU DUC Cuong | Chimie organique - pharmacie |
| | MICHOULIER Jean | Physique (I.U.T. I) |
| Mme | MINIER Colette | Physique (I.U.T. I) |

| | | |
|------|-----------------------|--------------------------|
| MM. | PELMONT Jean | Biochimie |
| | PERRIAUX Jean-Jacques | Géologie et minéralogie |
| | PFISTER Jean-Claude | Physique du solide |
| Mlle | PIERY Yvette | Physiologie animale |
| MM. | RAYNAUD Hervé | M.I.A.G. |
| | REBECQ Jacques | Biologie (CUS) |
| | REYMOND Jean-Charles | Chirurgie générale |
| | RICHARD Lucien | Biologie végétale |
| Mme | RINAUDO Marguerite | Chimie macromoléculaire |
| MM. | SARROT-REYNAULD Jean | Géologie |
| | SIROT Louis | Chirurgie générale |
| Mme | SOUTIF Jeanne | Physique générale |
| MM. | STIEGLITZ Paul | Anesthésiologie |
| | VIALON Pierre | Géologie |
| | VAN CUTSEM Bernard | Mathématiques appliquées |

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

| | | |
|-----|------------------------|--|
| MM. | ARMAND Yves | Chimie (I.U.T. I) |
| | BACHELOT Yvan | Endocrinologie |
| | BARGE Michel | Neuro-chirurgie |
| | BEGUIN Claude | Chimie organique |
| Mme | BERIEL Hélène | Pharmacodynamie |
| MM. | BOST Michel | Pédiatrie |
| | BOUCHARLAT Jacques | Psychiatrie adultes |
| Mme | BOUCHE Liane | Mathématiques (CUS) |
| MM. | BRODEAU François | Mathématiques (I.U.T. B) (Personne étrangère habilitée à être directeur de thèse) |
| | BERNARD Pierre | Gynécologie |
| | CHAMBAZ Edmond | Biochimie médicale |
| | CHAMPETIER Jean | Anatomie et organogénèse |
| | CHARDON Michel | Géographie |
| | CHERADAME Hervé | Chimie papetière |
| | CHIAVERINA Jean | Biologie appliquée (EFP) |
| | COLIN DE VERDIERE Yves | Mathématiques pures |
| | CONTAMIN Charles | Chirurgie thoracique et cardio-vasculaire |
| | CORDONNER Daniel | Néphrologie |
| | COULOMB Max | Radiologie |
| | CROUZET Guy | Radiologie |

| | | |
|-----|---------------------------|--|
| MM. | CYROT Michel | Physique du solide |
| | DENIS Bernard | Cardiologie |
| | DOUCE Roland | Physiologie végétale |
| | DUSSAUD René | Mathématiques (CUS) |
| Mme | ETERRADOSSI Jacqueline | Physiologie |
| MM. | FAURE Jacques | Médecine légale |
| | FAURE Gilbert | Urologie |
| | GAUTIER Robert | Chirurgie générale |
| | GIDON Maurice | Géologie |
| | GROS Yves | Physique (I.U.T. I) |
| | GUIGNIER Michel | Thérapeutique |
| | GUITTON Jacques | Chimie |
| | HICTER Pierre | Chimie |
| | JALBERT Pierre | Histologie |
| | JUNIEN-LAVILLAVROY Claude | O.R.L. |
| | KOLODIE Lucien | Hématologie |
| | LE NOC Pierre | Bactériologie-virologie |
| | MACHE Régis | Physiologie végétale |
| | MAGNIN Robert | Hygiène et médecine préventive |
| | MALLION Jean-Michel | Médecine du travail |
| | MARECHAL Jean | Mécanique (I.U.T. I) |
| | MARTIN-BOUYER Michel | Chimie (CUS) |
| | MASSOT Christian | Médecine interne |
| | NEMOZ Alain | Thermodynamique |
| | NOUGARET Marcel | Automatique (I.U.T. I) |
| | PARAMELLE Bernard | Pneumologie |
| | PECCOUD François | Analyse (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse) |
| | PEFFEN René | Métallurgie (I.U.T. I) |
| | PERRIER Guy | Géophysique-glaciologie |
| | PHELIP Xavier | Rhumatologie |
| | RACHALL Michel | Médecine interne |
| | RACINET Claude | Gynécologie et obstétrique |
| | RAMBAUD Pierre | Pédiatrie |
| | RAPHAEL Bernard | Stomatologie |
| Mme | RENAUDET Jacqueline | Bactériologie (pharmacie) |
| MM. | ROBERT Jean-Bernard | Chimie-physique |
| | ROMIER Guy | Mathématiques (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse) |
| | SAKAROVITCH Michel | Mathématiques appliquées |

| | |
|------------------------------|---------------------|
| MM. SCHAEGER René | Cancérologie |
| Mme SEIGLE-MURANDI Françoise | Crytogamie |
| MM. STOEIBNER Pierre | Anatomie pathologie |
| STUTZ Pierre | Mécanique |
| VROUSOS Constantin | Radiologie |

MAITRES DE CONFERENCES ASSOCIES

| | |
|---------------------|--------------------------|
| MM. DEVINE Roderick | Spectro Physique |
| KANEKO Akira | Mathématiques pures |
| JOHNSON Thomas | Mathématiques appliquées |
| RAY Tuhina | Physique |

MAITRE DE CONFERENCES DELEGUE

| | |
|-------------------|-----------------------------------|
| M. ROCHAT Jacques | Hygiène et hydrologie (pharmacie) |
|-------------------|-----------------------------------|

Fait à Saint Martin d'Hères, novembre 1977

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD
Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

| | | |
|-----|------------------------|--|
| MM. | ANCEAU François | Informatique fondamentale et appliquée |
| | BENOIT Jean | Radioélectricité |
| | BESSON Jean | Chimie Minérale |
| | BLIMAN Samuel | Electronique |
| | BLOCH Daniel | Physique du Solide - Cristallographie |
| | BOIS Philippe | Mécanique |
| | BONNETAIN Lucien | Génie Chimique |
| | BONNIER Etienne | Métallurgie |
| | BOUVARD Maurice | Génie Mécanique |
| | BRISSONNEAU Pierre | Physique des Matériaux |
| | BUYLE-BODIN Maurice | Electronique |
| | CHARTIER Germain | Electronique |
| | CHERADAME Hervé | Chimie Physique Macromoléculaires |
| Mme | CHERUY Arlette | Automatique |
| MM. | CHIAVERINA Jean | Biologie, Biochimie, Agronomie |
| | COHEN Joseph | Electronique |
| | COUMES André | Electronique |
| | DURAND Francis | Métallurgie |
| | DURAND Jean-Louis | Physique Nucléaire et Corpusculaire |
| | FELICI Noël | Electrotechnique |
| | FOULARD Claude | Automatique |
| | GUYOT Pierre | Métallurgie Physique |
| | IVANES Marcel | Electrotechnique |
| | JOUBERT Jean-Claude | Physique du Solide - Cristallographie |
| | LACOUME Jean-Louis | Géographie - Traitement du Signal |
| | LANCIA Roland | Electronique - Automatique |
| | LESIEUR Marcel | Mécanique |
| | LESPINARD Georges | Mécanique |
| | LONGEQUEUE Jean-Pierre | Physique Nucléaire Corpusculaire |
| | MOREAU René | Mécanique |
| | MORET Roger | Physique Nucléaire Corpusculaire |
| | PARIAUD Jean-Charles | Chimie - Physique |
| | PAUTHENET René | Physique du Solide - Cristallographie |
| | PERRET René | Automatique |

.../...

| | | |
|-----|--------------------------|--|
| MM. | PERRET Robert | Electrotechnique |
| | PIAU Jean-Michel | Mécanique |
| | PIERRARD Jean-Marie | Mécanique |
| | POLOUJADOFF Michel | Electrotechnique |
| | POUPOT Christian | Electronique - Automatique |
| | RAMEAU Jean-Jacques | Chimie |
| | ROBERT André | Chimie Appliquée et des matériaux |
| | ROBERT François | Analyse numérique |
| | SABONNADIÈRE Jean-Claude | Electrotechnique |
| Mme | SAUCIER Gabrielle | Informatique fondamentale et appliquée |
| M. | SOHM Jean-Claude | Chimie - Physique |
| Mme | SCHLENKER Claire | Physique du Solide - Cristallographie |
| MM. | TRAYNARD Philippe | Chimie - Physique |
| | VEILLON Gérard | Informatique fondamentale et appliquée |
| | ZADWORNY François | Electronique |

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

| | | |
|-----|-------------------|------------------------|
| M. | FRUCHART Robert | Directeur de Recherche |
| MM. | ANSARA Ibrahim | Maître de Recherche |
| | BRONOEL Guy | Maître de Recherche |
| | CARRE René | Maître de Recherche |
| | DAVID René | Maître de Recherche |
| | DRIOLE Jean | Maître de Recherche |
| | KAMARINOS Georges | Maître de Recherche |
| | KLEITZ Michel | Maître de Recherche |
| | LANDAU Ioan-Doré | Maître de Recherche |
| | MERMET Jean | Maître de Recherche |
| | MUNIER Jacques | Maître de Recherche |

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

| | |
|-----|---------------------|
| MM. | ALLIBERT Michel |
| | BERNARD Claude |
| | CAILLET Marcel |
| Mme | CHATILLON Catherine |
| MM. | COULON Michel |
| | HAMMOU Abdelkader |
| | JOUD Jean-Charles |
| | RAVAINE Denis |
| | SAINFORT |

C.E.N.G.

MM. SARRAZIN Pierre
SOUQUET Jean-Louis
TOUZAIN Philippe
URBAIN Georges

Laboratoire des Ultra-Réfractaires ODEILLO

E.N.S.M.E.E.

MM. BISCONDI Michel
BOOS Jean-Yves
GUILHOT Bernard
KOBILANSKI André
LALAUZE René
LANCELOT François
LE COZE Jean
LESBATS Pierre
SOUSTELLE Michel
THEVENOT François
THOMAS Gérard
TRAN MINH Canh
DRIVER Julian
RIEU Jean

E.N.S.E.R.G.

MM. BOREL Joseph
CHEHIKIAN Alain
VIKTOROVITCH Pierre

E.N.S.I.E.G.

MM. BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre
Mme JOURDAIN Geneviève
MM. LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G.

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G.

MM. COURTIN Jacques
LATOMBE Jean-Claude
LUCAS Michel
VERDILLON André

Je tiens à remercier

Monsieur le Professeur Louis BOLLIET, qui a bien voulu me faire l'honneur de présider le jury de cette thèse,

Monsieur Jean-Marc BRUN, Maître de Recherche au CNRS, qui m'a suivi tout au long de l'élaboration de ce travail et qui a accepté d'en être le premier juge,

Monsieur le Professeur Michael GRIFFITHS, qui a été présent lors de mes débuts en informatique et qui m'a judicieusement conseillé et m'encouragé chaque fois que cela était nécessaire,

Monsieur le Professeur Michel LUCAS, avec qui j'ai passé de longues heures de discussion et de travail notamment lorsqu'il travaillait dans le même laboratoire,

Monsieur Jean MERMET, Maître de Recherche au CNRS et directeur de cette thèse, qui a eu le courage (ou l'inconscience) de m'accueillir dans son équipe, qui en découvrant mon goût pour la liberté et la nouveauté a su me donner une autonomie nécessaire tout en m'orientant toujours vers des applications nouvelles de la CAO et me guidant en douceur vers le but que je m'étais fixé. Il m'a permis de créer des équipes autonomes tout en assurant une partie importante de la charge administrative qui en découle. Je l'en remercie vivement.

Monsieur Paul ROMAND, ancien président de l'A.F.MICADO, ancien attaché scientifique à la direction de la société CROUZET, actuellement directeur d'une PME, qui m'a permis de confronter mes idées universitaires avec la réalité industrielle.

Ce travail a nécessité des contacts fréquents avec de nombreuses personnes notamment les membres de l'association MICADO. Que tous ceux qui ont eu à me supporter soient remerciés.

Je tiens à rappeler que ce travail n'aurait pas pu être entrepris sans:

- l'équipe SIGMA-ARCHI composée principalement de Victor RIVERO, Rita VILLAMAYOR, François Z'GRAGGEN et Frédéric ZANELLI,
- l'équipe SIGMA-CAO, composée principalement de Gérard VITRY, Patrick VENIER et Ginette BUISSON. Ce sont sans doute eux qui m'ont le plus aidé dans la conception et la réalisation de la méthode, des outils et enfin de cet ouvrage,
- l'équipe travaillant sur la CAO chez CROUZET, composée de MM. ALEONARD, BENEVISE, GIRON, LE FRANC et ROMAND.

Je remercie également tous les membres du groupe "Méthodologie de la Conception Assistée par Ordinateur et la Communication Graphique" et plus particulièrement Dominique BORRIONE, Philippe BOULLE, Yvon BRESSY, Claude LE FAOU, Jean-François GRABOWIECKI, Geneviève GRAS, Patrice UVIETTA et notre secrétaire Josiane CARRY avec lesquels j'ai passé plusieurs années dans une ambiance studieuse et amicale, sans oublier Michel DELAUNAY, qui ne fait pas partie du groupe, mais dont la présence n'est pas passée inaperçue.

Enfin je remercie l'ordinateur HB68, son système MULTICS, le programme COMPOSE et le terminal DIABLO 1641, qui m'ont rendus de grands services dans la réalisation matérielle de ce document, ainsi que le Service de Reprographie pour son tirage rapide.

TIM

SOMMAIRE

SOMMAIRE

RESUME

INTRODUCTION

P A R T I E I : PRESENTATION DU PROBLEME

CHAPITRE 1: La C.A.O.

- 1.1 Définition de la C.A.O.
- 1.2 Considérations historiques
 - 1.2.1 Le point de vue du concepteur
 - 1.2.2 Le point de vue du constructeur de systèmes CAO
- 1.3 Etude de systèmes généraux pour la C.A.O.
- 1.4 Critique de la situation actuelle

CHAPITRE 2: Une approche globale pour la construction de systèmes CAO

- 2.1 Présentation succincte de l'approche
- 2.2 Les raisons de cette approche
- 2.3 Et les modalités

P A R T I E I I : ASPECT METHODOLOGIQUE

CHAPITRE 3: Modélisation du cadre de l'action

- 3.1 Contexte général
- 3.2 Formalisation
- 3.3 Recherche d'invariants
- 3.4 Méthode pour introduire la CAO dans une entreprise
 - 3.4.0 Présentation de la méthode
 - 3.4.1 Etude du processus actuel
 - 3.4.2 Les possibilités offertes par la CAO
 - 3.4.3 Définition du processus de conception adapté à la CAO
 - 3.4.4 Etablissement du cahier des charges fonctionnel du système CAO
 - 3.4.5 Etude de faisabilité - maquette
 - 3.4.6 Rédaction du cahier des charges détaillé
 - 3.4.7 Etablissement du cahier des charges organique
 - 3.4.8 Plan d'investissement et plan de mise en oeuvre
 - 3.4.9 Mise en oeuvre
 - 3.4.10 Calendrier des opérations
 - 3.4.11 Structure de l'équipe chargée de l'étude

B.DAVID : METHODOLOGIE POUR LA CONSTRUCTION DE SYSTEMES CAO

CHAPITRE 4: Processus de conception

- 4.1 Qu'est ce que la conception ?
- 4.2 Modélisation de l'activité de conception
- 4.3 Notre modèle de conception
- 4.4 Langage de spécification du modèle de conception

CHAPITRE 5: Données dans le processus

- 5.1 Pourquoi une Base de Données en CAO?
- 5.2 Spécificité des Bases de Données pour la CAO
- 5.3 Recherche d'un modèle pour la Base de Données du
Projet
 - 5.3.1 Structure
 - 5.3.2 Sémantique
 - 5.3.3 Cohérence
 - 5.3.4 Opérations
 - 5.3.5 Modèle conceptuel
- 5.4 Langage conceptuel
- 5.5 Essais de mise en oeuvre - expérimentation
 - 5.5.1 Organisation du rangement des données
 - 5.5.2 Expérimentation à l'aide des fichiers séquentiels
 - 5.5.3 Expérimentation d'un système de gestion de
bases de données
- 5.6 Modes d'accès
 - 5.6.1 Contexte
 - 5.6.2 Solution proposée
- 5.7 Base de Données Techniques

CHAPITRE 6: Dialogues homme-machine

- 6.1 Définition et classification
- 6.2 LDC - Une méthode unifiée pour la mise en oeuvre de
langages de commandes
 - 6.2.1 Spécification des dialogues d'une application
 - 6.2.2 Structure-support de gestion des dialogues
 - 6.2.3 Les extensions
 - 6.2.4 Gestion des modes et des moyens de dialogues

CHAPITRE 7: Tâches dans le processus

- 7.1 Définition et classifications
- 7.2 Description d'une tâche
 - 7.2.1 Description externe
 - 7.2.2 Description interne
- 7.3 Spécification d'une tâche
- 7.4 Activation et enchaînement de tâches

B.DAVID : METHODOLOGIE POUR LA CONSTRUCTION DE SYSTEMES CAO

CHAPITRE 8: Spécification d'une application

- 8.1 Méthodes de spécification
 - 8.1.1 Spécification de systèmes d'information
 - 8.1.2 Spécification en génie logiciel
 - 8.1.3 Conclusions
- 8.2 Spécification d'un système CAO
 - 8.2.1 Intérêt et principes
 - 8.2.2 Place du langage de spécifications dans la modélisation CAO
- 8.3 COLA - un langage pour la spécification fonctionnelle d'applications CAO
 - 8.3.1 Modèle conceptuel des applications CAO
 - 8.3.2 Langage de spécifications COLA
 - 8.3.3 Traitement du langage de spécifications
- 8.4 Différentes approches de l'utilisation du langage de spécifications
 - 8.4.1 Par le processus
 - 8.4.2 Par les tâches
 - 8.4.3 Par les données
 - 8.4.4 Par les utilisateurs
- 8.5 Exemple de spécification

CHAPITRE 9: Support informatique - Machine abstraite

- 9.1 Machine abstraite: définition et justification
- 9.2 Passage à une réalisation concrète
- 9.3 Contenu de la machine abstraite pour la CAO (MACAO)
- 9.4 Moniteur de communication
 - 9.4.1 Mécanisme des Unités Logiques
 - 9.4.2 Analyse lexicographique
 - 9.4.3 Sorties sous forme canonique
 - 9.4.4 Structure du moniteur de communication
- 9.5 Moniteur du processus de conception
- 9.6 Base de Données
- 9.7 Base d'algorithmes

CHAPITRE 10: Prototype - méthode de construction

- 10.1 Définition et justification d'un prototype
- 10.2 Passage des spécifications au prototype
- 10.3 Système pour la construction de prototypes
- 10.4 Les autres opérations

P A R T I E I I I : ASPECT TECHNIQUE

CHAPITRE 11: Outils pour la construction

- 11.1 Outils de construction
 - 11.1.1 Traitement de COLA
 - 11.1.2 Analyse des spécifications
 - 11.1.3 Traitement des dialogues

B.DAVID : METHODOLOGIE POUR LA CONSTRUCTION DE SYSTEMES CAO

- 11.1.4 Traitement des tâches
- 11.1.5 Traitement des données
- 11.2 Outils logiques
 - 11.2.1 Cohérence
- 11.3 Outils généraux
 - 11.3.1 Graphiques
 - 11.3.2 Graphiques de modélisation
 - 11.3.3 Non graphiques
- 11.4 Outils technologiques

CHAPITRE 12: Différentes réalisations de la MACAO

- 12.1 MINI-SIGMA
 - 12.1.1 Objectifs
 - 12.1.2 Contexte
 - 12.1.3 Choix
 - 12.1.4 Realisation
- 12.2 APPLI-SIGMA
 - 12.2.1 Objectifs
 - 12.2.2 Contexte
 - 12.2.3 Choix
 - 12.2.4 Realisation
- 12.3 VRAI-SIGMA
 - 12.3.1 Objectifs
 - 12.3.2 Contexte
 - 12.3.3 Choix
 - 12.3.4 Réalisation

CHAPITRE 13 Implantation de MACAO sur différents sites

- 13.1 Objectifs
- 13.2 Eléments de choix de la configuration
- 13.3 Le système CAO comme système distribué
 - 13.3.1 Répartition fonctionnelle
 - 13.3.2 Répartition physique
 - 13.3.3 Mise en oeuvre de la répartition
- 13.4 Typologie des matériels et des logiciels
- 13.5 Langage de spécifications organiques
- 13.6 Méthode de mise en oeuvre

CONCLUSIONS

A N N E X E S : EXEMPLES D'UTILISATION DE L'APPROCHE

ANNEXE 1: L'architecture

- A1.1 Cadre et objectifs de l'étude
- A1.2 Justification de l'approche intégrée
- A1.3 Objectifs du système SIGMA - ARCHI
- A1.4 Différents niveaux d'intégration
- A1.5 Processus de conception architecturale
- A1.6 Informations gérées par le système

B.DAVID : METHODOLOGIE POUR LA CONSTRUCTION DE SYSTEMES CAO

- A1.7 Traitements disponibles
 - A1.7.1 Définition du projet
 - A1.7.2 Traitements des propriétés et des relations
 - A1.7.3 Traitement des formes brutes
 - A1.7.4 Traitement des formes finies
 - A1.7.5 Traitement d'évaluation
 - A1.7.6 Traitements documentaires
- A1.8 Utilisation du système SIGMA-ARCHI
- A1.9 Mise en oeuvre du système SIGMA-ARCHI dans le contexte de SIGMA-CAO
- A1.10 Conclusion

ANNEXE 2: Les équipements électroniques

- A2.1 Cadre et but de l'étude
 - A2.1.1 Rappel de la démarche méthodologique
 - A2.1.2 Calendrier des opérations
 - A2.1.3 Structure de l'équipe chargée de l'étude
- A2.2 Déroulement de l'étude
 - A2.2.1 Etude du processus actuel de conception
 - A2.2.2 Les possibilités offertes par la CAO
 - A2.2.3 Processus Conception-Etude-Développement adapté à la CFAO
 - A2.2.4 Etablissement du cahier des charges fonctionnel du système CAO
 - A2.2.5 Etude de faisabilité
 - A2.2.5.1 Maquette du système support
 - A2.2.5.2 Maquette de l'application
 - A2.2.6 Rédaction du cahier des charges détaillé
 - A2.2.7 Etablissement du cahier des charges organique
- A2.3 Rappel des résultats

BIBLIOGRAPHIE

INTRODUCTION

INTRODUCTION

Tout semble actuellement favorable à la Conception Assistée par Ordinateur (C.A.O.): l'intérêt général, la baisse du coût des matériels informatiques, leur miniaturisation, la crise du pétrole ayant pour conséquence l'augmentation des prix des énergies et influençant plus ou moins directement ceux des matières premières en général. Il est donc important de concevoir mieux et souvent plus vite. La C.A.O. peut y contribuer de façon significative. C'est pourquoi elle n'est plus l'apanage de spécialistes de l'informatique et de quelques domaines privilégiés. Toutefois le contexte n'est pas le même dans les grosses entreprises qui peuvent créer des gros centres de C.A.O. et constituer des équipes importantes avec un matériel conséquent, et dans les petites et moyennes industries (PMI) qui ne peuvent pas en faire autant. Les raisons en sont multiples et parmi celles-ci, on peut citer principalement: des marges d'autofinancement souvent réduites, une réduction des coûts d'études insuffisamment rentables, difficilement finançables et la nécessité de compétences nouvelles souvent inexistantes.

C'est pourquoi il existe de plus en plus de demandes d'information sur la Conception Assistée par Ordinateur, émanant aussi bien des PMI que des grosses entreprises non encore utilisatrices de la CAO.

Cet intérêt croissant pour la CAO peut être plus ou moins profond et il peut être classé en principalement quatre niveaux, selon qu'il concerne:

- une information générale sur la CAO: principes, objectifs, principaux domaines d'application et résultats obtenus,
- une étude de l'aspect organisationnel de la CAO, et sa transposition au domaine concerné,
- une étude de faisabilité de l'application choisie, s'accompagnant généralement de la réalisation d'un prototype. L'utilisateur se rend alors mieux compte des implications du futur système,
- une étude et une mise en oeuvre d'un système CAO dans l'entreprise: c'est à dire l'implantation effective d'un système CAO adapté au problème traité et aux conditions spécifiques de l'entreprise (taille, organisation, style de travail, matériel informatique, possibilités d'investissement, etc...).

Nous avons constaté que l'on peut se contenter d'une démarche très empirique (sous forme de présentations et documents) uniquement au premier niveau et que pour les autres, l'efficacité dépend directement de la façon dont on organise le travail.

Nous avons alors cherché à apporter une réponse globale à ce problème d'actualité. Nous nous sommes astreints à proposer une méthode et des outils directement utilisables dans le contexte actuel (habitudes, matériel disponible).

Nous avons essayé de dégager les informations nécessaires à un système CAO. Il nous fallait alors trouver un moyen commode pour un concepteur (non familiarisé avec la CAO) d'exprimer celles-ci. Cela nous a conduits à définir un langage de spécifications.

Ensuite nous avons cherché à assurer la complétude et la cohérence de ces informations et à construire une version prototype servant de support pour l'étude de faisabilité.

Le problème du choix de matériel (gros, mini, micro, la répartition, l'utilisation de processeurs spécialisés), a aussi été abordé, ainsi que la méthode de mise en oeuvre sur le matériel choisi.

Ce travail, a été mené depuis six ans dans le contexte grenoblois, au sein d'un laboratoire d'informatique disposant d'un matériel très complet.

Dans l'équipe CAO, nous avons commencé par traiter le problème de la CAO des systèmes logiques et électroniques, puis nous avons étudié de nouvelles applications CAO comme par exemple l'architecture de bâtiments. Enfin nous participons actuellement à la mise en oeuvre d'un système intégré pour des ensembles logiques et électroniques.

Par l'intermédiaire de l'association MICADO (Association Française ayant pour Mission la Conception Assistée et le Dessin par Ordinateur), nous avons été en contact avec des industriels, et nous avons suscité des réflexions lors de groupes de travail.

Pour des problèmes informatiques aussi bien techniques que méthodologiques, nous avons particulièrement bénéficié de la proximité des recherches entreprises au sein du Laboratoire IMAG, notamment en ce qui concerne les aspects graphique, langage, système et base de données mais nous nous sommes également intéressés aux différentes études menées en France et à l'étranger.

Les résultats de cette étude sont multiples, aussi bien pratiques que théoriques. Au niveau méthodologique, nous avons

réussi à dégager une méthode globale de spécification et de mise en oeuvre originale et complète d'applications CAO. Le système sous-jacent permet une réalisation effective de cette méthode. Les outils définis et développés complètent efficacement le système. Des applications concrètes ont été traitées nous permettant de livrer nos premières impressions sur l'utilisation de la méthode.

Dans ce qui suit, nous essayons de rendre le mieux possible compte des résultats obtenus, tout en montrant les tenants et les aboutissants de la méthode, et en mettant en évidence les raisons qui nous ont amenés à effectuer les choix retenus.

Après une présentation succincte de l'approche, nous discutons les raisons et les modalités qui nous y ont conduits. Ensuite, nous étudions de façon détaillée les trois points importants: l'aspect méthodologique, l'aspect technique et les exemples d'applications que nous avons traités.

Dans la partie méthodologique, nous étudions tout d'abord l'organisation générale des systèmes CAO et nous en cherchons une formalisation. Puis nous nous posons la question: par quoi un système CAO est-il caractérisé? Pour y répondre, nous examinons successivement le processus de conception, les tâches dans ce processus, les données utilisées et les dialogues homme-machine. Pour chacun de ces points, nous présentons quelques exemples, nous montrons le but recherché, et les solutions envisagées. Nous faisons alors un choix que nous discutons et justifions.

Ensuite nous présentons différentes approches de spécification d'un système CAO. Nous discutons l'intérêt d'un langage de spécification, et nous présentons le langage COLA.

Puis nous cherchons un support informatique pour accueillir les informations spécifiées. Pour cela nous nous appuyons sur la notion de machine abstraite que nous définissons.

Les possibilités de passage de la machine abstraite vers des réalisations concrètes sont ensuite examinées et des solutions types proposées.

Pour terminer cette partie, nous introduisons la notion de prototype et après l'avoir justifiée, nous en montrons la réalisation et l'utilisation ainsi que les adaptations à apporter à la machine abstraite pour qu'elle puisse servir de prototype.

Dans la partie technique, nous présentons les réalisations concrètes des différents composants du système:

- les outils de spécification et de construction s'appuyant sur le langage de spécifications,
- un exemple de réalisation complète de la machine abstraite,

- la prise en compte de la répartition,
- différents cas de transport de la machine sur les sites variés.

La dernière partie (située en annexe), se compose d'exemples que nous avons traités et qui utilisent de façon plus ou moins poussée notre approche. En effet, ce travail n'est pas purement théorique mais découle d'applications industrielles réelles: celles-ci ont permis de tester la capacité d'adaptation de notre approche, au fur et à mesure de son élaboration. Citons les principaux domaines abordés: l'architecture, les équipements électro-mécaniques, les équipements électroniques, la fabrication de chaussures et de meubles en bois massif, la cartographie écologique et linguistique.

Toutes ces applications ne sont pas présentées en détail, les deux seules que nous détaillons sont la CAo en architecture et la CAO des équipements électro-mécaniques (se trouvant respectivement dans l'annexe 1 et l'annexe 2), les autres sont simplement esquissées.

En conclusion, nous essayons de faire une synthèse des résultats directs et indirects de notre étude, et d'en dégager les perspectives.

PARTIE I

PRESENTATION DU PROBLEME

RESUME: Cette partie situe la problématique de la construction de systèmes de Conception Assistée par Ordinateur. Elle contient une étude historique de la CAO du point de vue des utilisateurs ou des concepteurs et une présentation d'une nouvelle approche de construction, ainsi que les raisons et les modalités de celle-ci.

CHAPITRE 1

LA CONCEPTION ASSISTEE PAR ORDINATEUR

RESUME: Ce chapitre est consacré à l'analyse de la Conception Assistée par Ordinateur. Après avoir donné notre vision de la CAO, on examine l'évolution historique aussi bien du point de vue du concepteur-utilisateur que de celui du constructeur de systèmes CAO. L'accent est mis sur la notion d'intégration et l'analyse des systèmes CAO est orientée vers la construction de systèmes intégrés, ce qui permet d'introduire la notion de système général pour la CAO et de présenter une analyse des systèmes généraux pour la CAO. Ce chapitre se termine par la critique de la situation actuelle, en insistant sur l'absence de méthodologie de construction de systèmes CAO.

1. LA CONCEPTION ASSISTEE PAR ORDINATEUR

1.1 DEFINITION DE LA C.A.O.

L'utilisation de l'informatique dans des domaines sans cesse plus variés devient de plus en plus courante. Les bureaux d'études n'ont pas échappé à cette évolution, car la Conception Assistée par Ordinateur (C.A.O.) fait pénétrer l'informatique dans le processus Conception - Etude - Développement.

L'idée de base sur laquelle repose la CAO est très simple; il suffit en effet de comparer les capacités fondamentales de l'homme (réflexion, adaptation, intuition, globalisation, acquisition et utilisation de l'expérience) avec celles de la machine (grande capacité de stockage, puissance de traitement, fonctionnement systématique et répétitif, grande fiabilité) pour constater qu'elles sont complémentaires. Il est donc naturel de supposer qu'en associant l'homme et la machine dans la résolution, d'un problème de conception, l'on puisse obtenir de meilleurs résultats.

1.2 CONSIDERATIONS HISTORIQUES

1.2.1 Le point de vue du concepteur

Les premiers utilisateurs de la CAO ont été les informaticiens eux-mêmes qui s'en sont servi pour effectuer des calculs simples lors de la conception des calculateurs. Puis a suivi l'époque de la simulation, c'est à dire de la vérification de fonctionnement sur une maquette virtuelle (programmée dans l'ordinateur).

Le premier grand pas de la CAO a été dû au progrès technologique, qui a permis de remplacer le traitement par lots par un travail interactif, permettant à l'utilisateur de converser avec son programme et donc d'établir un véritable dialogue entre l'homme et la machine.

Il est incontestable que c'est à partir du moment où l'on est arrivé d'abord à produire, puis à introduire et modifier les informations sous forme de dessins, que la CAO s'est ouvert véritablement les portes des bureaux d'études. Le concepteur pouvait donc "dialoguer" naturellement avec l'ordinateur au lieu de produire et ensuite d'analyser des informations numériques, il

a utilisé un poste de travail reconnaissant des informations graphiques. Il a donc retrouvé la planche à dessin, les plans sous la présentation habituelle; il a également travaillé devant un écran lui permettant de modifier instantanément différentes parties du dessin. Cette façon de concevoir était très intéressante car elle a permis de faire mieux et plus vite qu'à la main. Le champ des applications CAO s'est élargi et l'intérêt pour ces nouveaux outils dans la conception, n'a pas cessé de croître.

A partir de ce moment, l'évolution de la CAO n'a pas été seulement tributaire des progrès de l'informatique mais a suivi l'évolution déterminée par les concepteurs eux-mêmes. Ceux-ci ont constaté que l'utilisation de programmes spécifiques correspondant aux tâches ponctuelles dans le processus de conception, nécessitait une préparation importante des données et une période assez longue d'interprétation des résultats et de préparation de nouvelles données, pour le programme suivant. Ils ont donc demandé de supprimer ce travail fastidieux afin de permettre l'enchaînement de différents programmes.

On a donc construit des séries de programmes correspondant à des démarches méthodologiques bien précises.

Leur intérêt n'était pas négligeable, mais leur inconvénient majeur était dû à leur rigidité. On s'est aperçu que le concepteur souhaitait pouvoir enchaîner les programmes tout à fait à sa guise et éventuellement introduire de nouveaux programmes effectuant de nouveaux traitements.

Les séries de traitements étant trop rigides, on a préféré raisonner et organiser la démarche autour des données en les considérant comme le noyau du système: des programmes accèdent à ces données, les testent, les modifient, etc...

De cette façon, on a abouti à la notion de SYSTEME INTEGRE pour la CAO, caractérisé par l'intégration, au sein d'un même système, des différentes phases qui vont de la conception jusqu'à la production. Ceci permet d'assurer une conception plus globale: la création des données qui décrivent l'objet à concevoir, la manipulation de ces données pour aboutir à une forme achevée de conception, et la génération des informations nécessaires à la fabrication de cet objet à partir des données de description issues de la conception, et stockées dans l'ordinateur.

L'intérêt d'un système intégré pour la CAO semble se situer notamment: dans une meilleure organisation du temps nécessaire au projet, car l'ordinateur se charge d'effectuer le travail répétitif et permet au concepteur de se consacrer aux tâches véritablement de conception. La suppression de la plupart des maquettes physiques, remplacées par des maquettes virtuelles

(programmées), constitue aussi un gain de temps important. L'utilisation de la main-d'oeuvre devient plus rationnelle, sa compétence et sa qualification changent.

L'approche globale de la conception permet une réduction du nombre d'erreurs de conception, facilite le passage conception - fabrication, et fournit un cadre intéressant pour la conduite du projet.

Dans certains domaines d'application comme l'électronique par exemple, le passage par un système intégré pour la CAO semble une obligation, car on ne trouve pas d'autre solution compte-tenu de la précision et/ou de la complexité nécessaires.

En résumé, on peut estimer que les bénéfices des systèmes intégrés pour la CAO se situent dans les gains:

- en temps : la durée de la conception est réduite,
- en coût : le coût global de la conception est abaissé, ce qui peut être négligeable, mais surtout le coût de fabrication qui est abaissé,
- en qualité : la qualité est accrue.

Mais il est surtout important de remarquer l'impact de la CAO sur la fabrication: la meilleure conception, de qualité accrue entraîne l'augmentation de la qualité de la fabrication et abaisse son coût.

Néanmoins, pour concrétiser une telle démarche, les obstacles sont nombreux, les plus importants étant:

- le coût élevé de la mise en place d'un tel système,
- le risque encouru que l'approche choisie ne soit pas satisfaisante,
- les problèmes psychologiques que le nouveau cadre peut poser aux concepteurs, habitués aux anciennes méthodes,
- la souplesse du système, primordiale pour pouvoir tenir compte de l'évolution aussi bien des méthodes de conception, que des produits de cette conception.

On peut constater que, à partir d'un problème de mécanisation (automatisation) de l'enchaînement de différents outils technologiques, on est arrivé à des notions beaucoup plus fondamentales et complexes liées au processus de conception, aux méthodes de conception, etc... Mais là, on aborde le présent et même le futur de la CAO.

Si nous regardons de plus près les diverses applications de la CAO, nous pouvons assez facilement constater qu'il existe deux grandes classes:

- celle des grosses entreprises, notamment en électronique, en automobile et en aviation, qui disposent de bureaux d'études importants et d'un volant financier permettant de mettre sur pieds, de façon interne, des équipes et des systèmes CAO intégrés importants. Leur but est de résoudre les problèmes particuliers au domaine concerné. Elles bâtissent donc des systèmes intégrés, mais très spécifiques aux domaines d'application concernés et aux matériels utilisés.
- les petites et moyennes entreprises, qui n'ont pas la possibilité économique d'effectuer des recherches en CAO chez elles. Elles sont donc à la recherche de solutions existantes proposées soit par des sociétés de service (clé en main) ou des constructeurs de matériel, soit par des associations comme MICADO (Association française ayant pour Mission la Conception Assistée et le Dessin par Ordinateur), pouvant prendre en compte leur problème spécifique et le replacer dans un contexte plus général. Nos travaux leur sont donc essentiellement destinés.

1.2.2 Le point de vue du constructeur de systèmes CAO

Du point de vue informatique, le domaine de la CAO est à la charnière de l'informatique et des domaines d'application; quiconque y travaille doit aborder d'une façon occasionnelle de nombreux domaines (langages, systèmes, algorithmique, gestion de données, graphique, ...). Le but est de constituer un système valable, équilibré, facile d'emploi et peu coûteux, tout en étant le plus proche de l'état d'avancement des recherches dans les différents domaines, et à la limite, en influençant certains axes de recherche.

La mise en place des applications CAO a suivi initialement l'évolution classique dans la programmation d'un domaine d'application. On a commencé par écrire des programmes spécifiques, puis, au vu des parties communes à plusieurs programmes, on a constitué des sous-programmes, puis des bibliothèques de sous-programmes pouvant être partagés (pour le graphique par exemple). On a alors commencé à dégager des programmes plus généraux, traitant un aspect du problème, quelle que soit l'application (visualisation en deux dimensions par exemple). Pour faire face à la demande de plus en plus grande des utilisateurs, et pour organiser le travail de mise en oeuvre de façon économique, notamment à partir du moment où la construction de systèmes intégrés a été formulée, on a étudié plus en détail leur nature. On est donc allé beaucoup plus loin que dans d'autres domaines, car on a cherché une structure et des concepts de base qui régissent un système intégré.

Cette étude a fourni les bases d'une approche bâtie sur la notion de SYSTEME GENERAL pour la CAO. Il s'agit d'un ensemble

d'outils de base réalisant une fois pour toutes les fonctions assurées par n'importe quel système CAO, en particulier la gestion des données, l'enchaînement des programmes et les dialogues homme-machine. Nous discuterons cette approche de façon plus approfondie dans le paragraphe suivant.

La production d'un système intégré pour la CAO dans un domaine particulier, se traduit par le développement d'un sous-système, à partir du système général (spécialisation du système général). Cette construction peut être assistée par ordinateur puisque l'on peut avoir recours aux outils assurant de façon automatique la construction des diverses parties du système.

Il est tout à fait admis que ce travail de construction nécessite la présence d'un informaticien de haut niveau, et ne peut pratiquement jamais être réalisé par le concepteur - utilisateur lui-même.

Ceci introduit un maillon entre l'utilisateur et l'ordinateur, et crée le problème de la rapidité de l'évolution du système.

Il semble donc intéressant de chercher à supprimer cet intermédiaire, c'est à dire essayer de constituer des outils à la portée des non-spécialistes de l'informatique.

Une solution possible pourrait résider dans une généralisation encore plus grande (suppression de sous-systèmes et de programmes spécifiques). C'est en tout cas ce qu'a tenté J.C.LATOMBE dans le système TROPIC (Lat.77), en utilisant les techniques de représentation et de résolution de problèmes issues de l'intelligence artificielle.

Ce stade nécessite encore aujourd'hui de grosses configurations et pose des problèmes psychologiques aux concepteurs, car la formulation de très haut niveau (par but et non par moyens) ne leur est pas encore habituelle. Par contre, il est possible de diminuer la programmation, non pas en réduisant les programmes spécifiques mais réalisant la production de ceux-ci d'une manière automatique.

En tout cas, les problèmes de capacité d'évolution et de facilité d'emploi des systèmes CAO par les concepteurs - utilisateurs, restent encore ouverts.

1.3 ETUDE DES SYSTEMES GENERAUX POUR LA CAO

Le rôle des systèmes généraux pour la CAO est de fournir un cadre propice à la construction d'un système CAO spécialisé.

Nous présentons quelques systèmes - types. Le premier système est sans doute le système ICES (Roo.67), développé au MIT dans les années 1965-67. Il s'agit d'un système général, car il ne s'adresse pas directement à une branche spécifique de l'ingénierie. Il permet, grâce à la notion de sous-système, de construire des systèmes spécialisés pour différents domaines d'application. C'est un système intégré, car il permet des échanges de données au sein d'un et entre plusieurs sous-systèmes. ICES fournit des outils pour la prise en compte des dialogues et pour la définition des structures de données partagées entre différents sous-systèmes.

Les traitements sont programmés en ICETRAN (extension de FORTRAN). Ce langage, qui est traduit par un pré-processeur en Fortran, augmente les possibilités de Fortran car il comporte quatre groupes d'instructions, respectivement destinés à: l'édition de liens, la gestion de tableaux dynamiques, la structuration de données et l'accès à la mémoire secondaire.

Ce système permet la construction de systèmes intégrés et il est lui-même intégré, car les outils de définition de dialogues et de structuration de données, constituent des sous-systèmes à part entière.

Un certain nombre de systèmes généraux se sont inspirés d'ICES notamment le système REGENT (LeS.77), développé en Allemagne Fédérale, qui reprend exactement la définition de ICES, avec PL1 comme langage support.

D'autres projets ont vu jour, et ont apporté certaines visions nouvelles:

- le projet GERMINAL (JRV.74), développé au CERT de TOULOUSE, dont la démarche est fondée sur des moyens de structuration de données plus récents (issus d'ALGOL 68), d'une base de données intégrée et d'un langage noyau spécifique et toujours interprété.
- le système BAL (BLR.74), essaie d'apporter une réponse à la façon d'utiliser et surtout de réutiliser des algorithmes dans le contexte d'une grosse entreprise.
- le projet RAINBOW (WCE.76), étudié à l'Université de Cambridge (Angleterre), traite le problème du partage des données et des traitements entre deux ordinateurs. Par ailleurs, un problème de structure de données générale est également abordé.
- le projet CASCADE (Mas.76), dont le champ d'intérêt est la construction du système ECAO (Enseignement par la CAO), montre qu'un système général pour la CAO peut être mis en oeuvre sur un mini-ordinateur.

D'autres aspects de généralité dans les systèmes CAO ont été étudiés notamment l'aspect graphique (description, manipulation et représentation d'informations graphiques). Plusieurs systèmes doivent être mentionnés: EUCLID (Bru.78), SYSTRID (MMS.79), BDS (ELS.76), et en particulier GLIDE (EaH.77), qui réunit dans la même recherche, les aspects graphiques, langage et base de données.

Il serait injuste d'oublier les essais de recherche de méthodes générales de résolution de problèmes, en particulier les tentatives de CADSYS (Whi.74) et de TROPIC (Lat.77), et les essais de plus en plus nombreux de réalisation de systèmes experts pour la CAO. Ces systèmes utilisent des techniques d'intelligence artificielle en CAO afin d'accroître la contribution de l'ordinateur dans la résolution des problèmes de conception. Ces systèmes associent les techniques de résolution de problèmes à l'utilisation d'une base de connaissances expertes. C'est un domaine en pleine expansion mais ne débouchant pas encore sur des réalisations industrielles.

1.4 CRITIQUE DE LA SITUATION ACTUELLE

Le lecteur pourrait se demander pourquoi les systèmes CAO véritablement industriels n'ont pas été évoqués plus longuement. La raison en est simple. Il existe actuellement deux grandes classes de systèmes CAO: ceux spécialisés dans un domaine d'application ou de une tâche très précise de ce domaine (le système REDAC pour les circuits par exemple), et ceux capables de réaliser une fonction générale quel que soit le domaine d'application, par exemple le dessin assisté par ordinateur (les systèmes APPLICON, COMPUTERVISION, GERBER, ...).

On constate généralement qu'il s'agit de systèmes fermés réalisant le mieux possible ce pour quoi ils ont été conçus. L'adaptation à un problème particulier ou l'intégration au sein d'un système plus vaste ne semblent pas encore être des arguments commerciaux suffisants. C'est pourquoi cette capacité, si elle existe, n'est pas encore dévoilée.

En ce qui concerne les grandes applications intégrées elles comportent souvent des fonctions réalisées par des systèmes spécialisés et l'intégration est assurée par un niveau supplémentaire de système. C'est ce niveau qui nous intéresse particulièrement.

L'intérêt d'une généralisation de systèmes de CAO est indéniable, notamment dans le contexte actuel de l'utilisation de plus en plus large de la CAO. Néanmoins, il est possible de formuler quelques critiques quant à la façon dont cette généralisation a été menée jusqu'à présent.

Si l'on étudie l'origine de cette généralisation, on s'aperçoit que l'on a considéré comme point de départ des systèmes CAO existants et spécifiques chacun à un domaine particulier. La généralisation a consisté à étudier le comportement de ces systèmes et à en dégager des similitudes, puis des fonctions à réaliser. Cette démarche entreprise uniquement par des informaticiens a donné naissance aux systèmes généraux pour la CAO. Il semble important de remarquer que cette façon de procéder n'est pas spécifique à la CAO mais correspond à un comportement habituel chez les informaticiens. L'on constate cependant quelques lacunes quant à l'organisation et au fonctionnement informatiques de ces systèmes: les systèmes généraux cités dans le paragraphe précédent en sont des exemples significatifs.

Nous soulignons notamment les points suivants:

- CONFIGURATION DE MATERIEL: pour l'ensemble des systèmes la configuration de matériel est figée. Elle est en général constituée d'une grosse machine (sauf pour CASCADE) unique (sauf pour RAINBOW, qui se place dans un contexte particulier de couplage de deux machines, une petite et une grosse). Pourtant il semble fondamental en CAO de suivre les utilisateurs sur leur lieu de travail, en respectant les configurations existantes.
- ORGANISATION DU SYSTEME: face à une configuration mono-machine on trouve un système monolithique intégrant aussi bien des outils de mise en place des applications que des applications elles-mêmes.
- LANGAGE SUPPORT ET PORTABILITE: les langages supports sont souvent spécifiques à la machine (certains systèmes s'appuient sur l'assembleur), la portabilité en est donc réduite et le transport lourd et coûteux.
- FONCTIONNEMENT: les lacunes lors de l'utilisation sont aussi nombreuses: nous avons constaté que certains systèmes comportent des limitations suivantes:
 - + mode d'utilisation: seulement traitement par lot, en différé,
 - + poste de travail: réduit souvent à un téléimprimeur,
 - + graphique: l'utilisation des techniques graphiques n'est pas généralisée,
 - + gestion des données: souvent rudimentaire,
 - + gestion et enchaînement des algorithmes: souvent manuels.
- TYPES D'UTILISATION: l'utilisation de tels systèmes est typiquement informatique, on construit puis on utilise, ce qui pose le problème fondamental de l'évolutivité du système

afin de résoudre celle du processus de conception. Ce problème est lié à celui de la (des) personne(s) intermédiaire(s). En effet les outils de mise en place sont des outils de complexité certaine, il n'est donc pas possible de les faire manipuler par des personnes non initiées. L'informaticien est donc nécessaire pour la mise en oeuvre des évolutions, ce qui peut poser des problèmes quant à la rapidité de leur prise en compte.

Les problèmes informatiques que nous avons évoqués ne sont pas exhaustifs mais correspondent aux plus importants.

Finalement, le problème essentiel vient du fait que dans cette approche de généralisation, l'on soit parti des systèmes informatiques et non de la conception proprement dite. On a procédé à la normalisation de la mécanisation sans se soucier de la sémantique de la conception elle-même. De cette façon on a laissé de côté l'étude et la modélisation du processus de conception et donc, la recherche de la logique sous-jacente au fonctionnement du système CAO.

C'est une lacune fondamentale qui empêche toute évolution future. En essayant de la combler il sera possible d'apporter des éléments de réponse quant à l'organisation fonctionnelle d'un système CAO. Cela permettra notamment de dégager des méthodes pour la réalisation de nouvelles applications.

Dans nos travaux nous avons essayé d'apporter notre contribution à ce problème délicat mais d'actualité. Nous avons étudié les moyens pour bâtir une méthode de construction qui tienne mieux compte de la réalité, tout en intégrant les possibilités offertes par les systèmes généraux actuels.

Nous avons donc voulu déplacer l'approche que constituent les systèmes généraux vers le haut, c'est à dire vers la sémantique de la conception et vers le bas, c'est à dire vers une meilleure utilisation des possibilités qu'offre l'informatique actuellement.

CHAPITRE 2

UNE APPROCHE GLOBALE POUR LA CONSTRUCTION DE SYSTEMES CAO

RESUME: Ce chapitre retrace l'approche qui nous a conduit à proposer une méthode globale de spécification et de mise en oeuvre de systèmes CAO, ainsi que les raisons principales de cette approche: le flou dû au manque d'informations sur le processus de conception et les conséquences de l'introduction de la CAO dans une entreprise (modification des méthodes de travail, risque, coût élevé, impact considérable). Il se termine par l'étude des modalités de cette méthode grâce à une étude bibliographique de la construction de systèmes et du génie logiciel.

2. UNE APPROCHE GLOBALE POUR LA CONSTRUCTION DE SYSTEMES CAO

2.1 PRESENTATION SUCCINCTE DE L'APPROCHE

Après avoir, dans le chapitre 1, analysé le contexte actuel de la CAO, nous présentons maintenant une façon d'améliorer les systèmes CAO et notamment les méthodes de construction de ces systèmes.

Une expérience significative sur le terrain a montré que les lacunes à combler sont nombreuses, et que seule une méthode globale de résolution permettra d'y parvenir.

Dans le cadre du projet SIGMA-CAO, nous avons essayé d'utiliser l'ensemble des connaissances en production de logiciels, pour définir une méthode globale de spécification et de mise en oeuvre des systèmes CAO.

Dans cette méthode, nous proposons de réaliser une application CAO en trois phases (Fig. 2.1): spécification fonctionnelle, construction de prototype pour vérifier et raffiner les spécifications, construction d'une version définitive du système.

La première phase de spécifications permet de définir de façon fonctionnelle le comportement souhaité du système CAO: de quoi veut-on partir, à quoi veut-on aboutir, par quel processus, dans quel contexte, etc...

La deuxième phase réalise la construction d'une version prototype du système, qui permet une expérimentation grandeur nature du fonctionnement du futur système.

La troisième phase se caractérise par la mise en oeuvre d'une version finale du système, après avoir déterminé une bonne organisation du système, c'est à dire après avoir choisi différents constituants, soit par rapport à une structure (matériel et logiciel) donnée, soit en en proposant une.

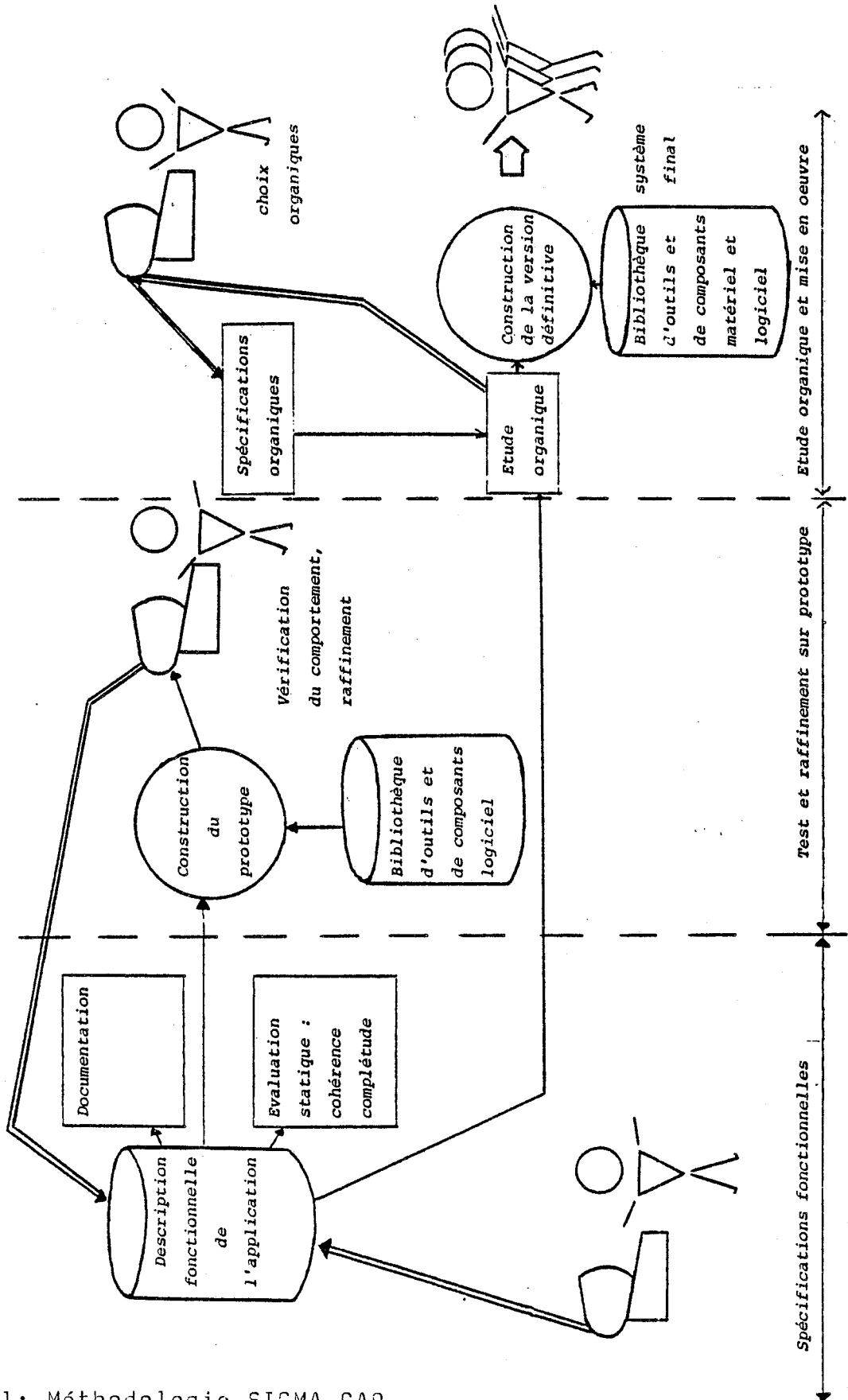


FIGURE 2.1: Méthodologie SIGMA-CAO

Pour chacune de ces phases, il existe des outils facilitant les tâches à entreprendre.

En intégrant ces trois phases à l'intérieur d'une méthode globale de construction, on peut intégrer ces outils, en particulier lors de la première phase: les spécifications sont définies à l'aide d'un langage qui est analysé par un processeur. Ce processeur produit alors une représentation formalisée, sur laquelle on peut entreprendre des traitements pour:

- effectuer des vérifications statiques de cohérence et d'intégrité des spécifications,
- produire automatiquement une documentation liée à ces spécifications. La construction du prototype s'effectue à partir des spécifications et peut être assistée par ordinateur.
- expérimenter le fonctionnement afin de vérifier des spécifications et éventuellement les modifier et/ou les raffiner.
- faciliter la mise en oeuvre définitive.

2.2 LES RAISONS DE CETTE APPROCHE

Après les débuts dans la CAO des systèmes logiques nous avons mené des applications très variées aussi bien du point de vue du secteur concerné que des problèmes rencontrés et traités.

Un domaine dans lequel nous avons beaucoup investi, mais dont les résultats sont restés au stade de résultats de laboratoire, est celui de la CAO en Architecture. Après une analyse détaillée du domaine, nous avons conçu et réalisé différentes versions d'un système CAO. Nous avons mis en pratique les différents niveaux d'intégration et différents niveaux de spécialisation de ces systèmes (systèmes généraux, systèmes spécialisés - sous-systèmes).

Les autres applications que nous avons menées ont un caractère opérationnel plus important. Elles ont été entreprises en collaboration directe avec des industriels demandeurs et avec le concours de l'A.F.MICADO.

Pour la société MERLIN-GERIN nous avons entrepris une étude et la réalisation d'un système de CAO de mécanismes électro-mécaniques. L'intérêt de cette étude se situe surtout dans la mise en oeuvre d'un système CAO intégré, ouvert et basé principalement sur un outil de dessin. Cette application montre qu'un outil de dessin peut dans certains cas constituer le coeur d'un système CAO.

Pour la société CROUZET nous avons conduit une étude très complète sur l'introduction de la CAO dans une entreprise. Cette étude nous a surtout permis de tester et d'affiner notre méthode d'introduction de la CAO en milieu industriel. Le domaine de conception très complexe, disposant déjà de nombreux outils autonomes, nous a beaucoup aidé dans la justification d'un système intégré.

Les autres applications sont actuellement en cours et nous ne disposons que des résultats des études préalables concernant notamment un diagnostic de la situation actuelle et l'opportunité de faire appel à la CAO.

C'est ainsi que nous avons étudié l'intérêt d'introduire la CFAO dans l'industrie de la chaussure. Les conclusions obtenues démontrent que seule une intégration conception - production assistées par ordinateur semble être rentable dans ce domaine où le coût de conception est très réduit par rapport au coût des matières premières.

Cette même conclusion nous l'avons tiré d'une étude d'opportunité effectuée pour un industriel de meubles en bois massif.

L'expérience acquise grâce au développement de ces applications de CAO nous enseigne que nous devons distinguer deux étapes importantes dans la vie d'une application: d'une part la définition et la mise en oeuvre de cette application, et d'autre part son utilisation.

Selon la nature de l'application, ces deux étapes peuvent être plus ou moins nettement séparées. La plupart du temps, elles sont bien marquées, notamment pour des applications vraiment nouvelles et on constate que le potentiel nécessaire pour la bonne exécution de chacune de ces phases est tout à fait différent.

Dans la première étape (la définition et la mise en oeuvre), il s'agit d'établir le dialogue entre le constructeur du système et le futur utilisateur du système produit. Il est primordial de disposer d'un contexte permettant une collaboration étroite entre les deux interlocuteurs, en vue d'échanger des informations sur les tenants et les aboutissants de l'application, et sur les possibilités de la CAO, afin de saisir l'application dans son ensemble et d'en proposer une solution adéquate.

Dans la deuxième étape (l'utilisation), il s'agit d'exploiter le système produit dans un contexte (matériel, humain et social) très différent du précédent. En effet l'exploitation se fera par un utilisateur-concepteur opérationnel dans un cadre

réel (matériel existant ou à définir), pour respecter les contraintes de fiabilité et d'efficacité très strictes.

Nous nous proposons de prendre en compte successivement ces deux étapes dans la vie de l'application, et de bâtir une méthode globale permettant de guider et de traiter les divers aspects de l'application de façon cohérente et efficace.

Les trois points clés de cette méthode sont: la spécification, la construction du prototype et la construction définitive. Pour spécifier un système CAO, nous fournissons un support permettant d'obtenir des spécifications complètes et cohérentes. Nous construisons alors un prototype de l'application, celui-ci étant défini comme un système fonctionnellement équivalent à l'application définitive, mais fonctionnellement seulement, c'est à dire faisant abstraction des choix de réalisation à tous les niveaux (matériel et logiciel). L'intérêt de ce prototype semble être fondamental car il assure, avant la mise en oeuvre définitive, la cohérence effective des spécifications. Ceci est particulièrement important si le domaine d'application et/ou le personnel sont peu familiarisés avec la CAO.

Après avoir obtenu des résultats satisfaisants, on peut entreprendre la construction d'une version finale du système, à partir des spécifications et du fonctionnement du prototype. Une étude de comportement permet de dégager les caractéristiques d'implantation du système, compte-tenu de la nature et du volume d'informations stockées et traitées (place mémoire, espace disque, puissance de calcul, nature du poste de travail, logiciel de base nécessaire), de vérifier que la structure existante est capable de recevoir le système conçu, et de définir une structure d'implantation souhaitable. Le résultat de cette étude conduit à la stratégie d'implantation sur la configuration cible.

Les avantages de cette démarche sont multiples: le fait de traiter successivement les différents aspects (spécifications, vérifications fonctionnelles, choix de réalisation), permet de concentrer l'attention sur les points fondamentaux. Par exemple, lors de la construction du prototype, on dissocie les contraintes fonctionnelles des contraintes de réalisation, ce qui a pour effet d'augmenter la rapidité de la réalisation. En effet, la disparition des contraintes de réalisation accroît les possibilités d'utilisation de composants fonctionnels préexistants, et permet d'avoir recours aux techniques de la construction assistée par ordinateur. Pour que cette démarche soit effectivement utile, la réalisation du prototype doit être vraiment rapide par rapport à la réalisation directe de l'application. Il faut également pouvoir facilement modifier le prototype pour réduire le temps entre la décision de modifier et le test de cette modification.

De cette façon, on permet aux critiques, aux modifications et aux nouvelles orientations de s'exprimer très rapidement et dès les premières phases du processus de construction.

Le prototype peut donc être utilisé comme un stimulateur et servir par exemple' au raffinement du cahier des charges du système à produire.

On associe alors véritablement le constructeur du système et le futur utilisateur, ce qui est très important, car on diminue ainsi considérablement un risque d'échec dû à une définition trop floue ou au manque de compréhension entre les deux interlocuteurs. Ceci est vrai pour toute application informatique, mais est absolument fondamental en CAO, où il faut tenir compte des trois caractéristiques suivantes:

- le flou dû au manque d'informations sur le processus de conception,
- les modifications des méthodes de travail induites par l'utilisation d'un système CAO,
- le risque de l'introduction de la CAO dans une entreprise, son coût élevé, et son impact considérable.

La séparation des aspects fonctionnels et de réalisation rend possible aussi la prise en compte, le moment venu, de différentes solutions d'implantation pour le système fonctionnellement défini, comme par exemple:

- la distribution du système CAO sur un réseau.
- l'utilisation du même système sur des machines différentes, ne disposant pas des mêmes composants logiciels (système d'exploitation, gestion des données, ...), ou matériels (terminaux graphiques, ...).
- la possibilité de faire appel à des processeurs spécialisés (processeur fichier, processeur base de données, ...).

2.3 ET LES MODALITES

Pour mener à bien la mise en place de la méthode que nous avons esquissée dans la paragraphe précédent, nous nous sommes appuyés surtout:

- sur la connaissance de la CAO acquise soit directement par les nombreuses applications menées au sein de l'équipe "Méthodologie de la CAO", soit par l'intermédiaire de divers organismes (dont l'A.F.MICADO) soit encore par l'étude des travaux, pratiques ou théoriques, publiés sur la CAO,
- sur les connaissances de la modélisation à travers des recherches sur l'approche systémique, l'analyse de système, la théorie du système général, etc..
- sur les progrès de l'informatique et notamment du génie logiciel.

Sur ce dernier point, un aperçu bibliographique nous apparaît utile même si l'exhaustivité semble impossible. Nous avons choisi de présenter les travaux en les classant de la façon suivante:

1/ LES CONCEPTS: l'article (RGI.75) présente assez complètement la tendance actuelle,

2/ LES ETUDES PONCTUELLES: les études et les réalisations qui essaient d'apporter des solutions pour diverses tâches du processus d'élaboration de logiciels.

Citons:

- la programmation structurée (DDH.74, ACM.74),
- les constructions modulaires de systèmes (Par.72a, Par.72b, Der.74, KLM.75, Dek.76, Wir.77, Mos.77)
- les nouveaux langages de programmation (LiZ.75, Lis.76, WLS.76, Jor.77, GaH.75, CAC.77),
- les langages de spécification (SEC.76, RoS.77, Ros.77, SAD.76, HaZ.76, Abr.78, BBD.77, Alf.77, NCC.78, SMC.74, Cap.78),
- les techniques de programmation automatique (Bal.73, Rut.76, Rut.77, Lon.77, Ter.78, BGW.78)

3/ LES METHODES GLOBALES: il s'agit de réunir les solutions proposées dans le paragraphe précédent, pour bâtir une méthode globale de spécification, de mise en oeuvre et de maintenance de logiciels.

Dans ce domaine, les tentatives sont beaucoup moins nombreuses et on peut considérer deux types d'approches:

- la première consiste en une démarche manuelle pouvant être assistée par ordinateur (spécifications, découpage en modules, programmation structurée,...) (SMC.74, RaH.78, Chu.75)

- la seconde est fondée sur les techniques de l'intelligence artificielle; avec production automatique à partir des spécifications de programmes (les techniques de la programmation automatique)(BGW.78),
- 4/ LES CAS SPECIFIQUES: il semble intéressant de remarquer que les problèmes de génie logiciel ont aussi été abordés dans divers domaines (IGe.77) de l'informatique, comme par exemple les Bases de Données et la Conception Assistée par Ordinateur, et ont élaboré des solutions intéressantes.
- en ce qui concerne les Bases de données, le projet le plus important semble être celui de l'Université de Michigan (TRH.73, TeW.76, TeH.78, ISD.75, Bah.75), en France, nous constatons une activité importante dans ce domaine. Nous ne citons ici que deux projets significatifs: le projet MACSI (Pec.75, ACh.75, Gir.77) et le projet REMORA (Per.76, Thi.76, Lam.77), qui jouit actuellement de développements très importants (RFR.79).
 - en Conception Assistée par Ordinateur, on a vu apparaître le terme Système Général pour la CAO, il y a déjà dix ans (Roo.67). Ce terme exprime la volonté de dégager les concepts et les fonctions d'un système CAO, et de fournir une méthode et des outils pour bâtir sur ce système général, un système orienté vers une application donnée (V1W.73, CER.74, All.77).

Pour définir les méthodes globales (3ème type dans notre classification), on peut évidemment procéder de deux façons: soit en rassemblant les outils ponctuels (SADT + ISDOS par exemple), soit en essayant de généraliser les méthodes utilisées dans les contextes spécifiques (4ème type) (généralisation du projet ISDOS par exemple). Nous nous intéressons essentiellement à cette deuxième approche.

PARTIE II

ASPECT METHODOLOGIQUE

RESUME: Cette partie expose tous les aspects méthodologiques qui ont contribué à l'élaboration de la méthode. On présente les quatre composants fondamentaux de l'invariant conceptuel: processus de conception, données utilisées, dialogues homme-machine et tâches du processus. Puis on décrit un langage de spécifications (COLA) à l'aide duquel une application CAO peut être exprimée. L'invariant informatique est ensuite introduit sous forme de machine abstraite pour la CAO (MACAO). On termine par la justification de la notion de prototype et on montre la façon de projeter les spécifications sur la machine abstraite.

CHAPITRE 3

MODELISATION DU CADRE D'ACTION

RESUME: Ce chapitre étudie la modélisation globale du cadre de la CAO. Le processus Conception-Etude-Développement est introduit comme une généralisation de la CAO et comme une définition plus précise de la CFAO (Conception et Fabrication Assistées par Ordinateur). Un schéma global d'un système CAO intégré, support du processus CÉD, est alors présenté: cette généralisation a pour but de fixer le cadre conceptuel d'un système CAO. La partie fondamentale concerne la recherche et la formalisation d'invariants des systèmes CAO. Ces invariants sont au nombre de trois (méthodologique, conceptuel et informatique).

3. MODELISATION DU CADRE D'ACTION

La méthode que nous avons tenté de présenter et justifier dans le chapitre précédent est le résultat d'un ensemble d'études que nous voulons rappeler maintenant.

A partir d'exemples nous avons cherché à découvrir tout ce qui n'est pas spécifique à une application, tout ce qui peut avoir un caractère plus général, direct ou indirect. Nous avons dégagé des éléments, des notions et des aspects que nous avons essayé de modéliser et de formaliser.

Puis nous les avons organisés de façon à obtenir différents niveaux d'invariants.

3.1 CONTEXTE GENERAL

Un système intégré pour la CAO est le support du processus Conception - Etude - Développement. Il doit donc prendre en compte l'environnement de ce processus et ce processus lui-même. Naturellement dans différents domaines d'application les processus Conception - Etude - Développement (C.E.D.) ne sont pas les mêmes. Néanmoins il est possible de dégager un cadre commun à ces différents processus. Evidemment ce processus n'a pas la même signification, le même contenu et la même rigueur pour un architecte ou pour une entreprise d'aviation, mais les éléments utilisés sont semblables.

Nous avons choisi d'utiliser le terme processus C.E.D qui nous semble plus approprié que le terme processus de conception: il est en effet difficile de déterminer où s'arrête la conception. Si le point de départ est assez bien défini, le point d'arrivée l'est beaucoup moins car entre la conception pure (de l'objet) et la conception de la méthode de fabrication et des outils associés, il s'agit encore de conception. C'est pourquoi il nous semble que le terme C.E.D., que nous allons utiliser sans vouloir l'imposer, correspond plus à la réalité et montre bien la recherche de continuité dans ce processus. Il nous semble plus précis que le terme CFAO (Conception et Fabrication Assistées par Ordinateur)(CAD/CAM) qui est plus macroscopique.

Ce découpage en Conception, Etude, Développement est plus ou moins net selon la complexité de l'objet, le nombre de participants, la durée du processus, le nombre d'objets à produire, etc..

Nous pouvons donc dans un premier temps, schématiser le processus C.E.D. comme un processus de transformation de l'objet à un certain niveau de définition vers un autre plus précis et plus concret (Fig.II - 3.1).

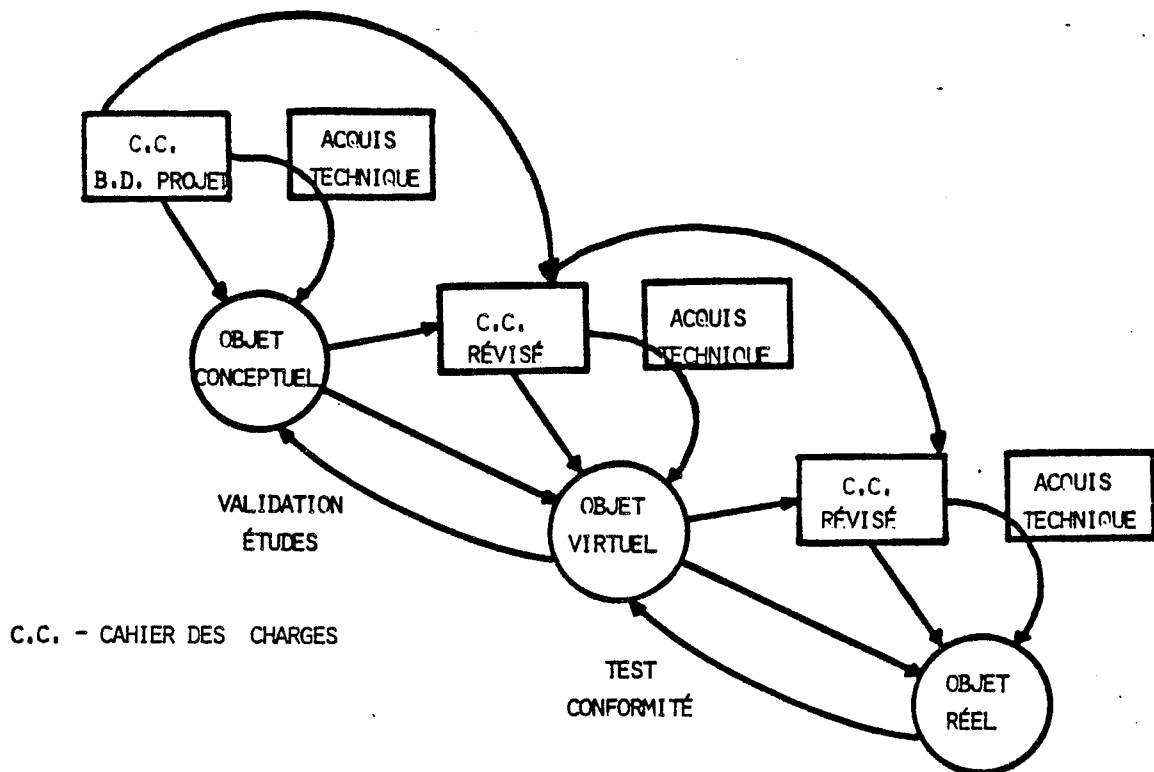


FIGURE II - 3.1: Processus CONCEPTION - ETUDE - DEVELOPPEMENT : processus de transformation

Au sein de chacune des étapes de transformation on entreprend un ensemble de tâches. Ces tâches ont des caractéristiques diverses (complexité, généralité, spécificité, répétitivité, etc..). Parmi celles-ci, deux au moins sont toujours présentes: la tâche de conduite de projet et celle de constitution du dossier (document sur l'état d'avancement du processus).

Ce processus se développe dans un environnement qui exerce des influences sur son déroulement. Deux composantes fondamentales de cet environnement sont:

- le concepteur ou l'équipe de concepteurs, qui ont un rôle actif car l'homme possède l'autorité décisionnelle, même en Conception Assistée par Ordinateur.
- l'environnement informationnel, c'est à dire l'ensemble des informations dont on dispose et dont on doit tenir compte pendant le déroulement du processus.

Nous avons surtout retenu trois types d'information, chacun constituant une base de données. (le terme "base de données" est ici utilisé sans aucune arrière pensée informatique ou de réalisation mais tout simplement au sens de base d'information.)

Le premier type contient les données les plus liées à un projet précis, il constitue ce que nous appelons BASE DE DONNEES PROJET (B.D.P.). Cette base contient toutes les informations relatives au projet traité (cahier des charges, déroulement du projet, résultats obtenus). C'est donc sur cette base que l'équipe de projet travaille et fait constamment évoluer les informations qui s'y trouvent.

Le second type constitue l'acquis technique de l'entreprise: il regroupe des données techniques indépendantes d'un projet précis et fournit des informations sur la conception, sur des méthodes de production, etc.. Nous l'appelons BASE DE DONNEES TECHNIQUES (B.D.T.).

Le troisième type contient des informations de caractère plus général: aspects financiers, gestion de personnel, moyens (coût des produits, délais d'approvisionnement, etc...). Nous l'appelons donc BASE DE DONNEES GENERALE (B.D.G.). Son existence est due au fait que le processus CED se déroule au sein d'un organisme vivant (entreprise par exemple), dont le contexte économique et organisationnel doit être pris en compte.

En effet, jusqu'à présent, les systèmes CAO se sont contentés essentiellement de la mécanisation de différentes tâches du processus de conception. Il semble maintenant possible et utile de permettre à l'utilisateur - concepteur de bénéficier d'aides offertes par l'ordinateur dans les domaines suivants: l'adéquation des spécifications, l'intégration de ces spécifications, l'exploitation de l'acquis technique lors d'une conception. Ceci afin de permettre une maîtrise plus complète de l'innovation, tout en conservant les avantages déjà acquis concernant la mécanisation des tâches sans créativité, la promotion des tâches créatives et l'intégration de tâches constitutives du processus de conception.

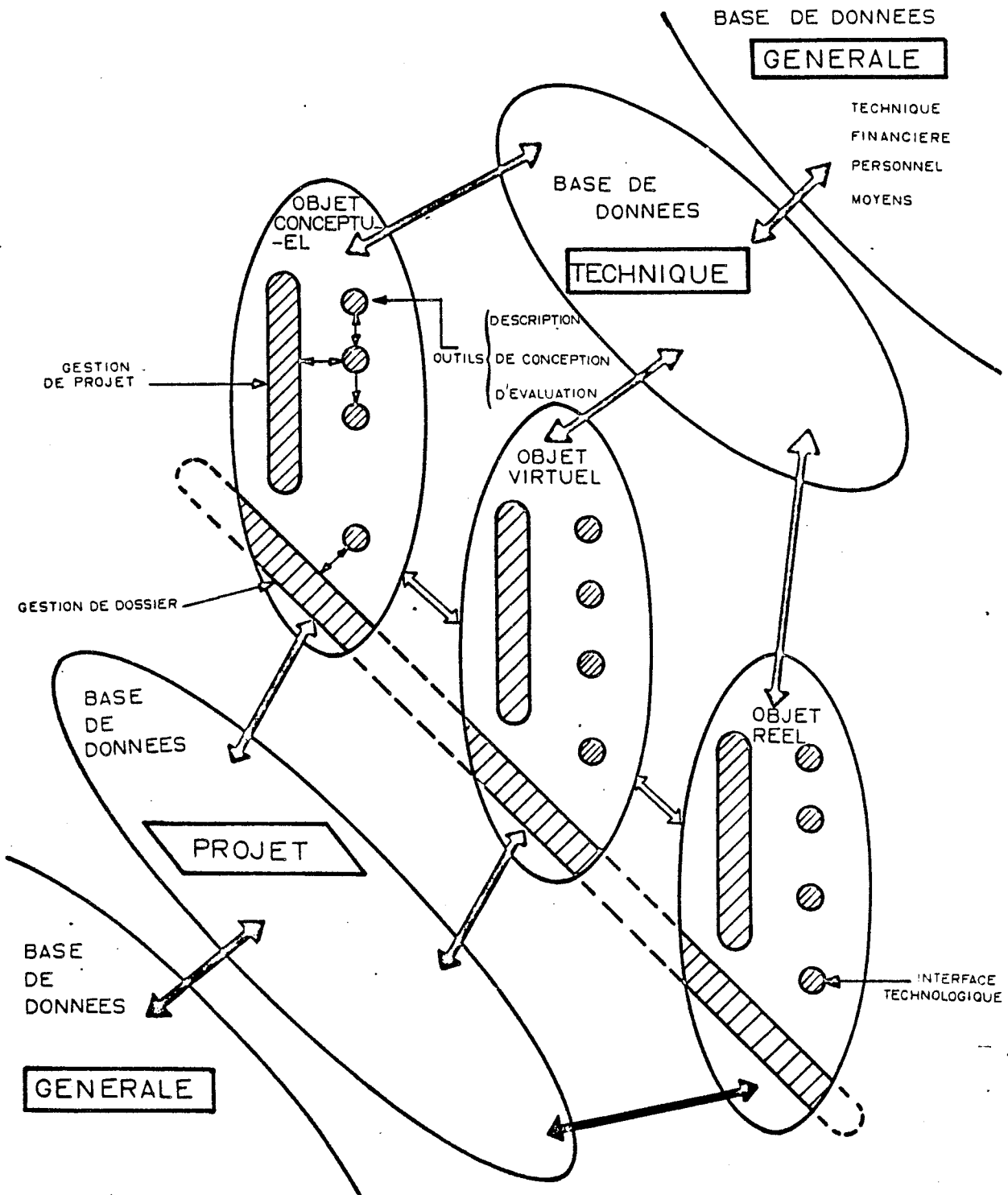


FIGURE II - 3.2 : Schéma global d'un système CAO

De cette façon, on peut espérer améliorer l'efficacité de tous les partenaires participant au processus technique d'élaboration de produits, ou concevoir plus rapidement, ou

concevoir des produits de meilleure qualité, ou des produits moins coûteux, ou enfin tout cela à la fois.

Un tel système aurait l'avantage de pouvoir prendre en compte la politique industrielle de l'entreprise, notamment en assurant une exploitation efficace de l'acquis technique et une maîtrise de l'innovation. La conduite de projets serait ainsi plus précise, ce qui semble être surtout important pour les produits très complexes intégrant de nombreuses contraintes fonctionnelles et techniques.

Finalement une structure d'accueil orientée vers le produit et non plus vers la tâche, permet une organisation de responsabilité plus claire et facilite une vue globale et intégrée du produit, ce qui rend possible une optimisation globale souvent nécessaire.

Pour cela il faut que le système CAO puisse effectivement accéder à toutes les informations, à toutes les décisions et à toutes les tâches qui s'effectuent pendant le processus CED, et qu'il puisse agir à priori et/ou à posteriori dans ce processus.

Il doit donc être un véritable système d'information technique, qui communique avec le système d'information de l'entreprise.

Le schéma global d'un tel système CAO serait le suivant: (Fig.II - 3.2)

3.2 FORMALISATION

Notre objectif est de permettre l'introduction dans une entreprise de systèmes dont nous venons d'exprimer les principales caractéristiques. Nous voulons, non seulement promouvoir ces systèmes, mais également fournir des aides méthodologiques et techniques.

Pour cela nous devons dégager puis modéliser tout ce qui semble fondamental à des applications variées. Nous devons identifier les fonctions de base et les types de données utilisés par un système CAO. Ce travail de modélisation n'est pas simple même s'il s'agit d'un système CAO orienté vers une application spécifique. Pour nous, il s'agit de travailler à un niveau de généralisation plus élevé car nous voulons bâtir un système indépendant de toute application, ou mieux pouvant s'adapter à toute application. Il faut donc faire un effort d'abstraction et de conceptualisation afin de dégager un modèle abstrait, ainsi que le formalisme permettant d'exprimer l'application à réaliser à l'aide de ce modèle.

Le schéma (Fig.II - 3.3) précise nos propos. Ce schéma, dont le principe a été élaboré au séminaire MICADO(MIC.75), présente le cadre conceptuel d'un système CAO.

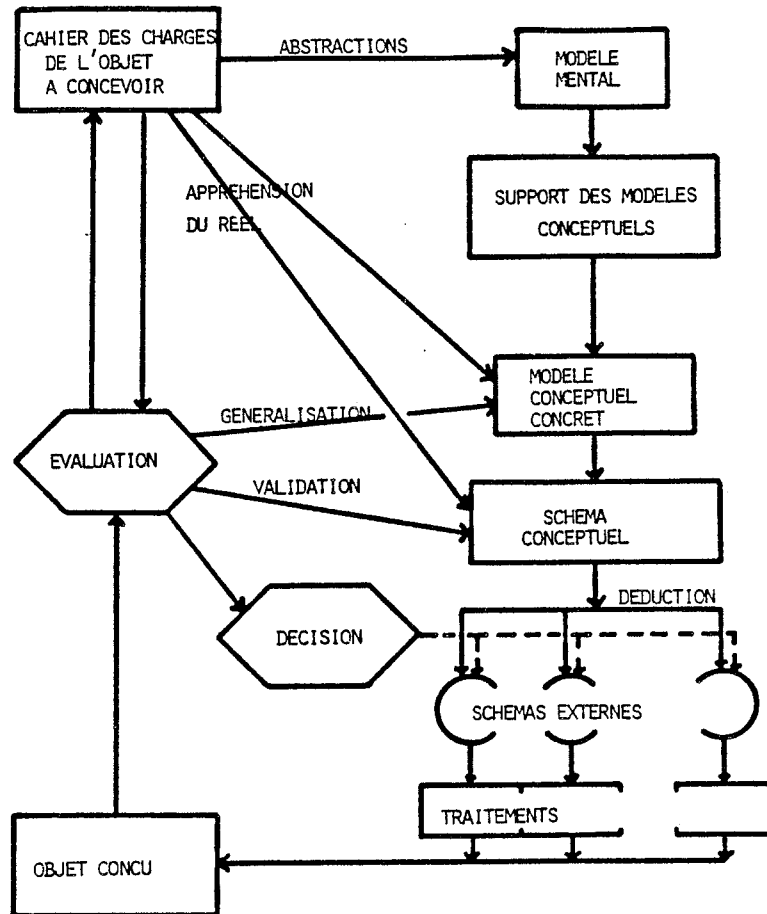


FIGURE II - 3.3 : Cadre conceptuel d'un système CAO.

Quand on bâtit un système CAO orienté vers une application il faut tout d'abord formuler la nature de l'objet perçu, c'est à dire le cahier des charges de l'OBJET A CONCEVOIR. On doit imaginer toutes les formes et variantes possibles que l'on souhaite prendre en compte.

On procède ensuite par abstraction en dégagant un modèle mental contenant toutes les informations souhaitées. Puis à partir de ce modèle mental on bâtit une représentation, informatisable, qui sera le support du système CAO, désigné par le terme de MODELE CONCEPTUEL. Le modèle conceptuel n'appartient pas à un objet précis mais à une famille d'objets (exemple chaise Louis XV). C'est un modèle générique.

Quand on traite un projet précis au sein de ce système CAO appliqué, on construit à partir du modèle conceptuel un schéma

conceptuel qui est valué. Il s'agit d'une valuation du modèle conceptuel, car l'on précise l'ensemble des valeurs (numériques et non numériques) caractérisant le projet. Ce schéma conceptuel suit le déroulement du processus de conception et représente les informations à tous les stades de ce processus d'élaboration. Les différentes transformations s'effectuent à l'aide de traitements (outils informatiques) réalisant de façon automatique ou interactive, les tâches demandées.

Chaque tâche n'a besoin que d'une partie des informations contenues dans le schéma conceptuel. On peut donc les extraire de ce dernier et éventuellement les transformer sous la forme souhaitée par la tâche (changement de représentation) On construit ainsi des vues externes, appelées SCHEMAS EXTERNES (exemple: coupe d'un pied de chaise Louis XV).

Après l'exécution d'une ou d'un ensemble de tâches, on obtient une maquette de l'OBJET CONCU, dont le niveau ou degré de réalisation est de plus en plus grand. Pour s'assurer que cet objet conçu correspond toujours au cahier des charges, il faut effectuer des évaluations.

A la suite de celles-ci, différentes décisions peuvent être prises: Si l'objet est conforme au cahier des charges il est possible de le valider et d'intégrer alors ces nouvelles informations dans le schéma conceptuel; dans le cas contraire deux options sont à considérer: soit il faut remettre en cause la proposition que l'on vient d'élaborer et repartir de l'objet validé à l'étape précédente, soit on considère l'objet obtenu comme très intéressant et on admet qu'il est souhaitable de remettre en cause le cahier des charges.

Nous y reviendrons car c'est une particularité importante de la conception; on élabore certes l'objet conçu, mais les connaissances acquises lors de cette élaboration peuvent servir à une meilleure connaissance de l'objet à concevoir et remettre en cause la définition de cet objet. A la limite, on peut considérer que la définition de l'objet à concevoir est élaborée en parallèle avec l'objet conçu mais de préférence avec une étape d'avance.

Le dernier point important est la faculté de prendre en compte de nouveaux aspects non initialement prévus dans le modèle mental et conceptuel: possibilité de généralisation absolument fondamentale en CAO car les connaissances et les méthodes évoluent très rapidement, et il n'est pas envisageable de reconstruire le système CAO support à chaque modification.

Nous voulons à la fois permettre la construction de systèmes CAO fonctionnant selon ce schéma et les construire à partir des bases communes. Nous devons donc généraliser et formaliser aussi bien le fonctionnement que le contenu informationnel. Cela veut

dire, entre autres, que nous devons élaborer un modèle conceptuel plus général que celui envisagé jusqu'à présent. En effet celui-ci est un modèle conceptuel CONCRET car il contient, non seulement les informations structurelles, mais aussi les informations sémantiques du domaine d'application considéré. Nous devons définir un SUPPORT DE MODELES CONCEPTUELS qui offre tous les éléments de base nécessaires à la construction de modèles conceptuels concrets en vue d'applications variées.

Sans entrer pour l'instant dans les détails, disons que celui-ci doit permettre l'expression de la démarche de conception, laquelle comporte trois composantes:

- le processus de conception et la définition des étapes de ce processus,
- l'objet de la démarche: tout ce qu'il faut prendre en compte dans l'élaboration de l'objet de la conception (passage de l'objet à concevoir à l'objet conçu),
- les traitements manuels, automatiques ou interactifs, appropriés.

On procède en fait à la MODELISATION de la démarche. Cette modélisation est plus ou moins complète et complexe, différents critères interviennent dans le choix de la modélisation et parmi ceux-ci citons:

- la finesse de la modélisation
- la généralité du modèle recherché
- les possibilités laissées aux utilisateurs de faire évoluer le modèle

Selon le niveau de généralité on peut parler de modèles CONCRETS ou de modèles ABSTRAITS.

Les modèles concrets très fins, ayant un degré de généralité réduit ne laissant à l'utilisateur que de très faibles marges d'évolution, se nomment MODELES PARAMETRIQUES.

Par contre, les modèles abstraits proposent en général un FORMALISME à l'aide duquel l'utilisateur peut bâtir sur mesure un modèle concret répondant à un problème (projet) donné. Un modèle abstrait est en fait un générateur de modèles concrets. C'est pourquoi nous l'appelons SUPPORT DE MODELES CONCEPTUELS.

En conception en général, il semble, important, voire indispensable, de ne pas être prisonnier d'un modèle paramétrique ou même d'un modèle concret trop rigide car chaque projet contient des informations spécifiques. Pour un type de projet il est éventuellement possible de bâtir un modèle très concret mais

dont la validité et le champ d'action sont obligatoirement réduits.

Nous devons définir un modèle suffisamment abstrait pour supporter un ensemble de démarches, un formalisme permettant d'exprimer des modèles concrets "personnalisés" pour chaque projet, ainsi que les outils abstraits ou concrets pouvant travailler sur ces modèles.

Ce modèle doit permettre de modéliser de façon abstraite des objets, une démarche et des outils.

L'élaboration d'un modèle concret peut s'effectuer de façon progressive; il n'est en effet pas indispensable de figer tous les degrés de liberté du support de modèles en même temps.

Il faut laisser au concepteur la possibilité de définir pour chaque projet les caractéristiques importantes de l'objet à concevoir, d'affiner le processus de conception et éventuellement de faire appel à des traitements particuliers. Par exemple dans le domaine des disjoncteurs, d'un projet à l'autre l'importance du comportement thermique ou du comportement mécanique peuvent être très différents et le concepteur doit pouvoir en tenir compte dans l'élaboration du modèle concret.

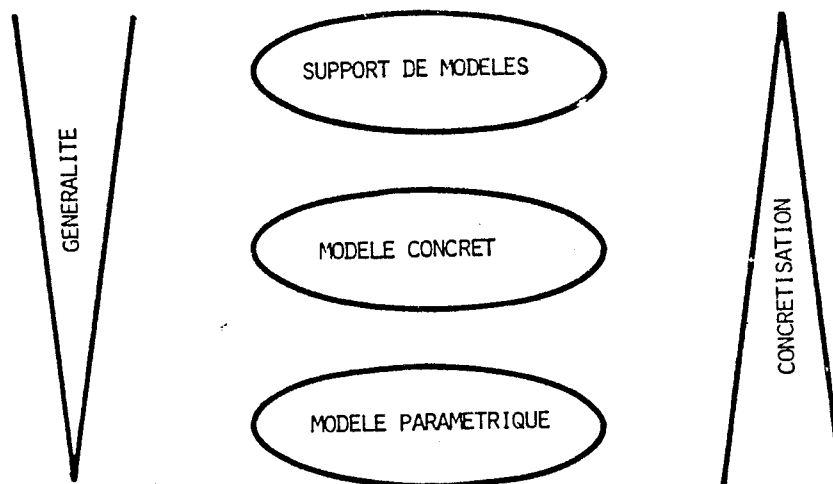


FIGURE II - 3.4 : Généralisation et concrétisation du modèle conceptuel (différents niveaux de généralité et de spécificité).

3.3 RECHERCHE D'INVARIANTS

L'effort de généralisation, de modélisation et de formalisation met en évidence l'ensemble d'invariants qui constitue le cadre d'un système CAO. Nous en avons identifié trois invariants.

- le premier, au niveau INFORMATIQUE, est concerné par le support informatique de systèmes CAO. Il a été mis en évidence par des études sur les systèmes généraux de CAO évoquées au chapitre 1. Nous voulons insister sur l'indépendance possible vis-à-vis des différents supports informatiques envisageables et, pour cela nous introduisons une définition indépendante du contexte, s'appuyant uniquement sur une définition fonctionnelle.
- le deuxième, au niveau CONCEPTUEL a trait à la recherche et la formalisation d'un support de modèles conceptuels, tel que nous venons d'en noter le besoin dans le paragraphe précédent.
- le troisième, au niveau METHODOLOGIQUE a pour objet la recherche d'une méthode pour introduire la CAO en milieu industriel, permettant l'utilisation de la méthode informatique, noyau de notre travail et des résultats de nos investigations.

Nous pouvons schématiser l'organisation générale de ces invariants de la façon suivante (Fig. II - 3.5):

- une application peut être exprimée à l'aide du SUPPORT DE MODELES CONCEPTUELS (INVARIANT CONCEPTUEL),
- ce modèle est supporté par la MACHINE FONCTIONNELLE ABSTRAITE (INVARIANT INFORMATIQUE), à partir de laquelle différentes mises en oeuvre réelles sont possibles,
- le processus de mise en place se déroule selon la METHODE POUR INTRODUIRE LA CAO (INVARIANT METHODOLOGIQUE).

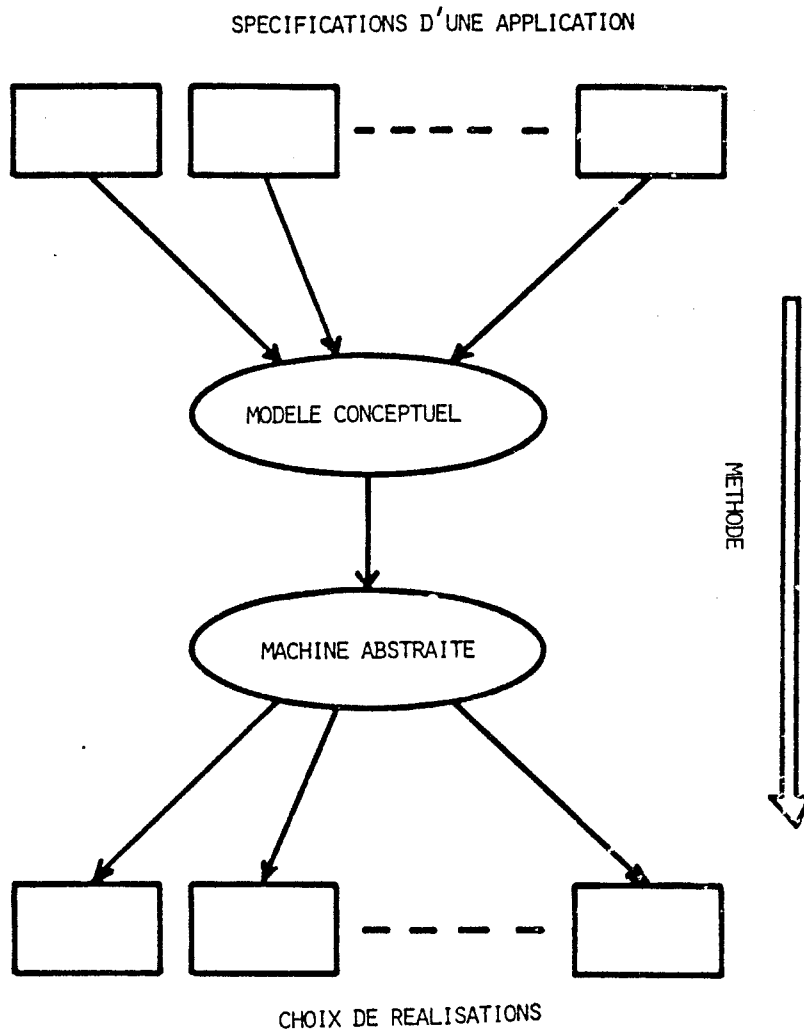


FIGURE II - 3.5 : INVARIANTS SIGMA-CAO et leurs liens.

Dans ce qui suit nous allons approfondir ces différents aspects. Nous avons choisi de commencer par la présentation de la méthode pour introduire la CAO dans une entreprise qui constitue le chapeau de notre démarche. Il s'agira d'une présentation de principes, les exemples précis seront développés dans la partie IV de ce document.

3.4 METHODE POUR INTRODUIRE LA CAO DANS UNE ENTREPRISE

Nous avons essayé de définir une méthode capable de dégager, à partir du fonctionnement actuel de l'entreprise et des possibilités offertes par la CAO, une organisation du processus Conception - Etude - Développement, assurant une efficacité maximum à l'entreprise.

Pour pouvoir effectivement aboutir à un résultat satisfaisant, il faut mener cette opération de façon systématique et exhaustive afin de mettre en évidence tous les aspects importants permettant de réaliser un système C.A.O. complet. Nous avons défini cette méthode à partir d'une confrontation entre une étude théorique et une expérimentation pratique. Nous avons pu vérifier son opérationnalité sur des exemples pratiques.

C'est l'étude de la CAO des équipements électroniques que nous avons menée avec la société CROUZET qui a été la plus importante de ce point de vue. Nous la rapportons de façon détaillée dans l'annexe 2. Nous avons pu confirmer cette méthode par d'autres applications notamment celle pour la société grenobloise MERLIN-GERIN, de la société SPIC en Mayenne et de la société Moreux de Varennes dans l'Allier. Pour la première il s'agit d'une grande entreprise d'environ 7000 personnes qui travaille essentiellement sur la distribution de l'énergie électrique. Elle fabrique surtout l'appareillage de coupure et de protection, principalement des disjoncteurs, et des équipements, à savoir, tableaux et armoires regroupant les différents appareils répondant à un usage particulier.

Pour la conception de ces produits la CAO n'est pas une obligation, car la complexité reste encore maîtrisable à la main. Néanmoins, la CAO peut apporter des gains en temps d'étude et en qualité de conception, ce qui est très intéressant, car il s'agit de produits fabriqués en grandes séries.

Il nous a donc été demandé de participer à l'introduction de la CAO dans cette entreprise. Le domaine d'application choisi était celui des disjoncteurs. Ce thème a été adopté parce que la conception de disjoncteurs nécessite non seulement une part importante de graphisme (schémas, épures, plans), mais également et, conjuguée avec les tracés, une part de calculs dont le volume est lié au degré d'optimisation que l'on souhaite atteindre.

Le travail entrepris a été mené en deux temps. La première phase se compose de l'analyse et de la définition du problème, et de la mise en oeuvre d'un système prototype permettant l'expérimentation et la proposition du matériel d'un système exploitable industriellement. La deuxième étape, très pratique, consiste à installer le système chez l'industriel, l'adapter sur son matériel (notamment graphique) et compléter son fonctionnement pour devenir véritablement opérationnel.

Nous avons donc conçu et réalisé un système permettant la conception progressive de schémas fonctionnels et leur habillage, pour bâtir des formes réelles de pièces, ainsi que la cotation interactive de ces schémas. A partir des données stockées en machine, il est possible d'appeler des programmes notamment celui de calcul de cinématique développé au sein de l'entreprise (programme de test qui a été intégré au système).

Les conclusions que nous pouvons tirer de cette application sont surtout:

- l'intérêt de l'intégration de différents outils (dessin et calculs) au sein d'un système,
- l'insertion de ce système sans bouleversement dans le processus habituel de conception,
- l'ouverture obligatoire du système pour être capable de recevoir de nouveaux outils,
- la première expérimentation partielle de la méthode d'introduction de la CAO en milieu industriel.

Lors de deux autres applications nous avons participé à la phase d'expertise (analyse du processus actuel, définition du processus adapté à la CFAO, proposition d'un système CFAO).

Le cadre de la première de ces études est l'entreprise SPIC à Ernée en Mayenne. Elle emploie 300 personnes, dont 10 cadres et produit environ 2 millions de paires de chaussures d'homme par an. Il s'agit d'une entreprise performante puisqu'elle exporte 65 % de sa production, et que le pourcentage de coût de la main d'oeuvre dans le prix des produits est seulement de 25 % contre 35 % pour le cuir. C'est une entreprise qui est à la pointe de l'automatisation car elle expérimente actuellement une ligne de production automatisée nécessitant seulement quatre personnes. Par ailleurs elle étudie la possibilité d'opérations automatisées (semelle), pilotées par ordinateur (piqûres à plat) et la réalisation de robots divers.

Son orientation résolue vers l'automatisation l'a conduite naturellement à s'intéresser à la CAO. La gestion de la production est très informatisée, ce qui permet d'ores et déjà de disposer d'informations précises sur les gammes de production, la durée des opérations, leur enchaînement, etc...

Nous avons donc effectué l'expertise de cette entreprise en vue de proposer une organisation CAO valable. Ce travail a été réalisé à la fin 1979 et au début 1980. Nous rendons ici compte uniquement du déroulement de cette phase d'analyse et des propositions que nous avons élaborées.

En analysant le processus de conception actuel, nous avons surtout observé la linéarité de la démarche. Elle résulte du manque de temps généralisé compte tenu des délais très précis à respecter. On doit donc à tous les niveaux corriger les imperfections dues au manque de coordination. Il semble que ceci soit la plus grande lacune du processus et il nous a semblé important que le système CAO essaie de l'atténuer.

En effet, nous avons pu constater que l'intérêt d'améliorer ponctuellement telle ou telle action du processus n'est pas assez évident, et ne justifie à lui seul un investissement lourd en matériel et logiciels pour la CAO. Ceci est surtout vrai en conception proprement dite. Par contre si le système CAO permet d'intégrer le processus de conception et de fabrication, accélère certaines étapes de ce processus et dégage du temps pour des boucles de remise en cause, il devient très utile. Notamment s'il permet de remonter certains problèmes plus en amont, par exemple de la fabrication, pour être pris en compte dès la conception, alors le système CAO a rempli son rôle, car il contribue à l'amélioration du processus conception - fabrication.

Cette application montre surtout que la CAO peut être utilisée même dans des domaines où la conception est relativement simple. Mais dans ce cas il est indispensable d'élargir son champ d'action car elle ne serait pas rentable en conception proprement dite. Aussi l'intégration Conception - Etude - Développement - Production devient fondamentale, et les problèmes de conception se rencontrent à tous les stades de ce processus (exemple conception d'outils de production ou problème de placement).

Une autre application que nous avons analysée est celle d'une production de meubles en bois massif. Il s'agit de l'entreprise MOREUX de Varennes qui emploie 600 personnes dont une dizaine de cadres. L'ordinateur est actuellement utilisé uniquement en gestion. D'autres utilisations sont en projet après le changement de celui-ci.

La diversité des produits est assez grande puisque l'entreprise est divisée en 8 lignes de fabrication. Le souci d'une bonne gestion est permanent, et se traduit notamment par la minimisation du nombre de pièces différentes apparaissant dans l'ensemble des produits. On a recours pour cela, à la notion de sous-ensemble qui peut intervenir dans différents produits. Ainsi on minimise les encours et les stocks. L'entreprise est en fait coupée en deux parties très distinctes. La première produit des sous-ensembles, la seconde assemble des produits finis.

C'est à partir du processus de conception-fabrication actuel et des lacunes exprimées, que nous avons envisagé l'organisation d'un processus nouveau utilisant la CFAO.

La nature des problèmes rencontrés, leur diversité, la nécessité de diffuser des informations et de disposer de renseignements de provenances distinctes, nous a conduit à proposer un système intégré.

Ce système doit pouvoir saisir sous forme de schéma l'idée du meuble à concevoir et permettre son élaboration, pour aboutir progressivement à un objet de plus en plus précis en réduisant les délais d'élaboration et augmentant la qualité des informations diffusées.

3.4.0 PRESENTATION DE LA METHODE

Pour mener à bien le travail d'introduction de la CAO dans une entreprise, nous avons dressé une liste très complète des phases obligatoires, si l'on veut avoir des chances raisonnables d'aboutir à un système satisfaisant, c'est à dire remplissant toutes les fonctions possibles ou nécessaires et évoluant dans de bonnes conditions économiques.

· Les principales phases sont les suivantes:

- 1/ ETUDE DU PROCESSUS ACTUEL DE CONCEPTION
- 2/ ETUDE DE POSSIBILITES OFFERTES PAR LA CAO
- 3/ DEFINITION DU PROCESSUS DE CONCEPTION POUR LA CAO
- 4/ ETABLISSEMENT DU CAHIER DES CHARGES FONCTIONNEL DU SYSTEME CAO
- 5/ ETUDE DE FAISABILITE - (MAQUETTE)
- 6 /CAHIER DES CHARGES FONCTIONNEL DETAILLE DU SYSTEME C.A.O.
- 7/ ETABLISSEMENT DU CAHIER DES CHARGES ORGANIQUE DU SYSTEME CAO
- 8/ PLAN D'INVESTISSEMENT et PLAN DE MISE EN OEUVRE
- 9/ MISE EN OEUVRE

Nous allons rapidement détailler le contenu de chacune de ces phases.

3.4.1 ETUDE DU PROCESSUS ACTUEL

Pour pouvoir intégrer valablement la CAO dans le processus Conception - Etude - Développement, il faut partir du processus

actuel. Il ne s'agit pas de le reproduire entièrement dans le nouveau contexte, mais de l'adapter en tenant compte d'une part, de ses lacunes (goulets d'étranglement, impasses, etc...) et d'autre part, des possibilités nouvelles offertes par la CAO.

C'est pourquoi l'analyse du processus actuel doit faire ressortir son organisation générale c'est à dire:

- la structure du processus et ses étapes,
- les personnes concernées, leur nombre et leur structure,
- la nature des informations: informations échangées, informations ou produits reçus, informations ou produits élaborés,
- les outils informatiques déjà utilisés,
- les informations quantitatives sur chaque étape du processus: le temps nécessaire, le nombre d'opérations sur l'objet, la proportion du travail créatif, etc...

Il est souvent intéressant de s'appuyer sur des cas significatifs.

Il est évident que la complexité de l'analyse dépend en grande partie de celle de l'objet à concevoir, du nombre de concepteurs et du degré de couplage production - conception nécessaire ou souhaitable, c'est à dire du niveau d'interaction entre la conception et la production (par exemple conception de séries utilisant au mieux la chaîne de production).

3.4.2 LES POSSIBILITES OFFERTES PAR LA CAO

Le but premier de la CAO est d'associer l'homme et la machine dans le processus de conception en s'appuyant sur leurs complémentarités respectives.

Dans un premier temps on a principalement utilisé les grandes capacités de traitement et de mémorisation de l'ordinateur. On a donc créé des programmes réalisant, de façon ponctuelle, diverses opérations (la plupart du temps non créatives) du processus de conception.

Mais les limitations de cette approche sont apparues très rapidement, surtout dans un contexte de conception complexe nécessitant l'intervention de plusieurs personnes de compétences variées.

Le problème fondamental résidait dans le fait suivant: les informations nécessaires à chaque programme n'étaient que très partiellement différentes mais n'avaient pas le même format, ce qui rendait fastidieuse l'utilisation des programmes car les mêmes informations étaient introduites plusieurs fois.

C'est pourquoi une autre approche fondée cette fois sur les données et non sur les traitements, a été étudiée.

On parle de SYSTEME INTEGRE car il s'agit d'intégrer au sein d'un même système, des données de l'ensemble des phases de conception pouvant aller jusqu'à la production et assurer la création des données qui décrivent l'objet à concevoir, la manipulation de ces données pour aboutir à une forme achevée de conception et souvent même la génération des informations nécessaires à la fabrication de cet objet.

On préfère cette seconde approche dans la plupart des études sur l'utilisation de la CAO.

3.4.3 DEFINITION DU PROCESSUS DE CONCEPTION ADAPTE A LA CAO

En plus de l'automatisation de certaines tâches non créatives, la CAO (dans l'optique d'une approche intégrée) est concernée par la gestion des informations et par l'organisation du projet.

Il est donc indispensable d'adapter le processus de conception à cette nouvelle façon de percevoir le projet. Pour cela il faut bâtir un nouveau processus de conception à partir:

- du processus actuel,
- de l'analyse des outils CAO existant soit dans l'entreprise soit à l'extérieur,
- des souhaits des utilisateurs relatifs aux nouveaux outils,
- des possibilités offertes par l'approche intégrée de la CAO.

Dans ce processus on doit, d'une part introduire de nouvelles phases indispensables à la marche du processus informatisé ou réalisant de nouvelles tâches inconcevables jusqu'à présent sans informatique et, d'autre part supprimer celles devenues caduques.

L'objectif est donc de définir un processus permettant une conception plus globale et par là même plus cohérente.

3.4.4 ETABLISSEMENT DU CAHIER DES CHARGES FONCTIONNEL DU SYSTEME CAO

Si les phases précédentes peuvent s'effectuer de façon informelle, l'établissement du cahier des charges du système CAO doit en refléter tous les aspects. Pour cela, nous avons défini un langage de spécifications (voir plus loin chapitre 8), permettant d'exprimer le cahier des charges. De cette façon il est possible de vérifier la complétude et la cohérence des spécifications, et de faciliter la transmission de ces informations à d'autres personnes.

Le Cahier des Charges fonctionnel doit contenir essentiellement les spécifications suivantes:

- l'organisation du processus de conception qui décrit comment s'effectue la conception, c'est à dire quelles sont ses phases et leurs inter-relations.
- l'organisation de la structure d'exécution qui décrit comment est organisé un projet, qui sont les intervenants et à quelle phase.
- l'organisation du processus de décision qui décrit les prérogatives des personnes participant au projet.
- l'objet en cours de conception qui décrit les caractéristiques de l'objet à concevoir et de l'objet conçu dans les différentes phases du processus, et comment s'effectue la transformation.
- les traitements pouvant intervenir dans le processus de conception qui décrivent les phases dans lesquelles le traitement peut être utilisé, les personnes pouvant s'en servir ainsi que les données qu'elles peuvent manipuler, les résultats, les dialogues associés et l'algorithme du traitement.
- l'acquis technique qui décrit l'ensemble des informations auxquelles on souhaite pouvoir accéder pendant le processus; il peut s'agir de données permanentes (bibliothèques, catalogues), de projets déjà réalisés, de consignes pour la conception, etc...

Ainsi, le Cahier de Charges fonctionnel décrit entièrement le fonctionnement du système CAO souhaité.

3.4.5 ETUDE DE FAISABILITE - MAQUETTE

Il n'est pas toujours possible de produire directement des spécifications détaillées du système CAO complet, notamment dans le cas où il s'agit d'un domaine d'application de la CAO tout à fait nouveau. Il est alors préférable de procéder en deux temps: en premier lieu on spécifie une partie représentative du processus de conception et on réalise la partie correspondante du système, puis, après une étude détaillée, on élargit les

spécifications à l'ensemble du système. Le premier travail démontre la faisabilité du projet et permet un raffinement du cahier des charges. En effet, en ce qui concerne les néophytes, une heure devant le poste de travail remplace avantageusement des jours d'explication mais surtout, ouvre de nouveaux horizons quant aux possibilités offertes par la CAO.

Néanmoins pour que cette étude soit rentable, il faut respecter les quelques règles élémentaires suivantes:

- l'aspect démonstratif de la maquette doit être certain
- les besoins potentiels doivent être largement couverts sauf si la partie choisie est un échantillon complet,
- il faut pouvoir traiter des cas significatifs et pas uniquement des exemples scolaires,
- la réalisation de la maquette doit être rapide pour ne pas retarder la suite des opérations,
- le coût de la maquette doit être réduit car elle sera abandonnée dès qu'elle aura rempli son rôle,
- l'évolutivité de la maquette doit être très grande pour permettre une prise en compte rapide des modifications du cahier des charges.

Les deux derniers points concernent l'organisation du système support de la maquette. Une des solutions est de disposer d'un système général pour la CAO (système fournissant l'ensemble des fonctions de base nécessaires à tout système CAO), et de construire une maquette directement en spécialisant le système général.

3.4.6 REDACTION DU CAHIER DES CHARGES DETAILLE

La rédaction du cahier des charges détaillé correspond à la production des spécifications fonctionnelles du système CAO. En fait on raffine les spécifications produites à la phase 4 en tenant compte des renseignements obtenus lors de l'étude de faisabilité, et on les rédige sous la même forme.

3.4.7 ETABLISSEMENT DU CAHIER DES CHARGES ORGANIQUE

Après avoir déterminé le bon fonctionnement du système CAO, il faut maintenant définir la façon de bâtir effectivement le système, c'est à dire choisir le matériel et le logiciel.

Pour cela, il faut disposer non seulement d'informations qualitatives, mais aussi quantitatives, pour déterminer les besoins (en puissance de calcul, en taille mémoire, en taux d'utilisation des périphériques, etc...) de chaque étape du processus, puis globalement, en tenant compte de la charge du

processus et donc envisager de dédoubler certains postes de travail.

A partir de ces informations, on élabore une stratégie de choix de matériel (aussi bien de ou des ordinateurs que des périphériques) et de logiciels (réutilisation, élaboration, achat, location, service bureau, etc...).

Différentes solutions doivent être envisagées, comparées et proposées.

3.4.8 PLAN D'INVESTISSEMENT et PLAN DE MISE EN OEUVRE

L'étude fonctionnelle et l'étude des choix de réalisation fournissent des éléments fondamentaux, permettant de dégager les moyens de mise en oeuvre du système souhaité.

Il est alors possible de proposer un plan d'investissement s'étalant souvent sur plusieurs années, et parallèlement, un plan de mise en oeuvre tenant compte de l'importance relative des différentes parties du système, de leur possibilité d'installation progressive et de leur rentabilité.

3.4.9 MISE EN OEUVRE

La mise en place s'appuie sur le plan de mise en oeuvre et doit faire apparaître clairement les différentes étapes de l'installation de chacune des parties: mise en place, essai, test, fonctionnement provisoire, fonctionnement définitif et cela aussi bien pour des logiciels que pour des matériels.

C'est particulièrement important car les contraintes économiques sont beaucoup plus pressantes que dans d'autres phases plus éloignées de l'exploitation.

3.4.10 CALENDRIER DES OPERATIONS

Comme nous avons pu le constater, la mise en place d'un système CAO est un travail délicat et de longue haleine. Il dépend évidemment de la complexité des produits fabriqués, de la complexité de leur conception et du niveau souhaité d'intervention de ce système. Il n'est donc pas possible de prévoir de façon absolue les délais de cette étude. Néanmoins, c'est possible de façon relative, en dressant un schéma représentant les durées et surtout les interdépendances des différentes phases de ce processus (Fig.III-3.6).

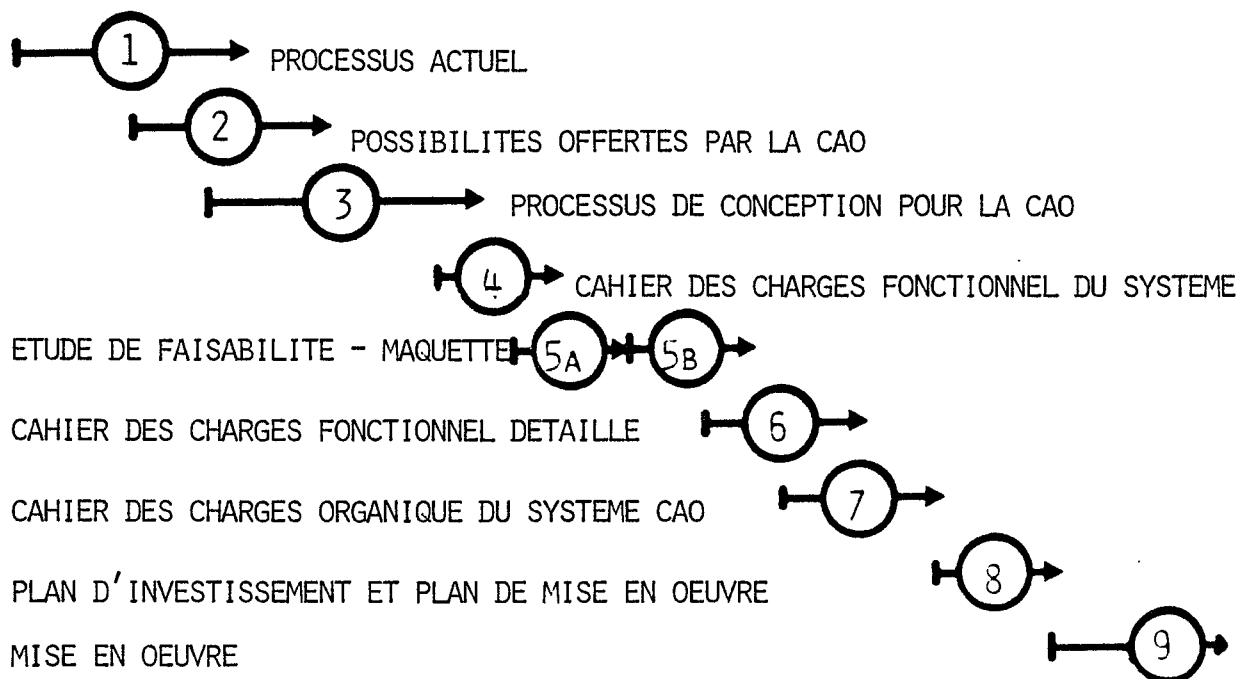


Figure III - 3.6 : Calendrier des opérations pour la mise en place d'un système CAO.

Ce schéma appelle les commentaires suivants:

Le recouvrement relatif entre les trois premières phases est compréhensible: en effet les deux premières phases sont indépendantes, mais fournissent, au cours de leur exploration, des éléments permettant de mener à bien la troisième.

Comment peut-on accélérer ce processus? Ce n'est sans doute pas en réduisant le temps des phases d'analyse, car une analyse incomplète sera payée très cher au niveau de l'exploitation. Par contre, il est possible et souhaitable de réduire le temps de réalisation de la maquette en choisissant une bonne méthode de mise en oeuvre.

A part cela c'est surtout au niveau des décisions politiques (notamment au niveau d'approbation des plans), que l'on peut espérer certains gains.

3.4.11 STRUCTURE DE L'EQUIPE CHARGEE DE L'ETUDE

La bonne organisation et la bonne composition de l'équipe chargée de l'exécution de cette étude est un des facteurs fondamentaux du succès.

Il est indispensable que la structure de cette équipe soit représentative de tout le champ à traiter. Il faut assurer le contact aussi bien à l'extérieur qu' à l'intérieur de l'entreprise. C'est pourquoi cette équipe doit se composer:

- de représentants du bureau d'étude
- de représentants de la production et du contrôle
- d'informaticiens

Il est clair que tous les aspects du processus Conception - Etude - Développement doivent être représentés. Mais pour des questions d'opérationnalité, il ne semble guère réaliste de dépasser le nombre de 5 personnes.

Pour assurer la collaboration de l'ensemble des personnes concernées (car le pire des problèmes est de se croire ou de se sentir isolé) tout en restant opérationnel, il faut constituer une structure à deux étages:

- le premier, réduit, qui constitue le groupe opérationnel en réunissant les gens disponibles pour effectuer le travail, et suffisamment représentatifs pour ne pas être (trop) contestés.
- le second, large, entièrement représentatif des secteurs de l'entreprise et des aspects du problème, mais n'ayant qu'un rôle consultatif créant ou préservant le contact avec la pratique.

Pour le premier groupe, cette activité doit devenir l'activité principale si l'on veut aboutir à un résultat satisfaisant dans un délai raisonnable. Par contre le second groupe composé en grande partie des responsables des différents secteurs concernés, (Fig.III-3.7) devrait se réunir au moins tous les mois.

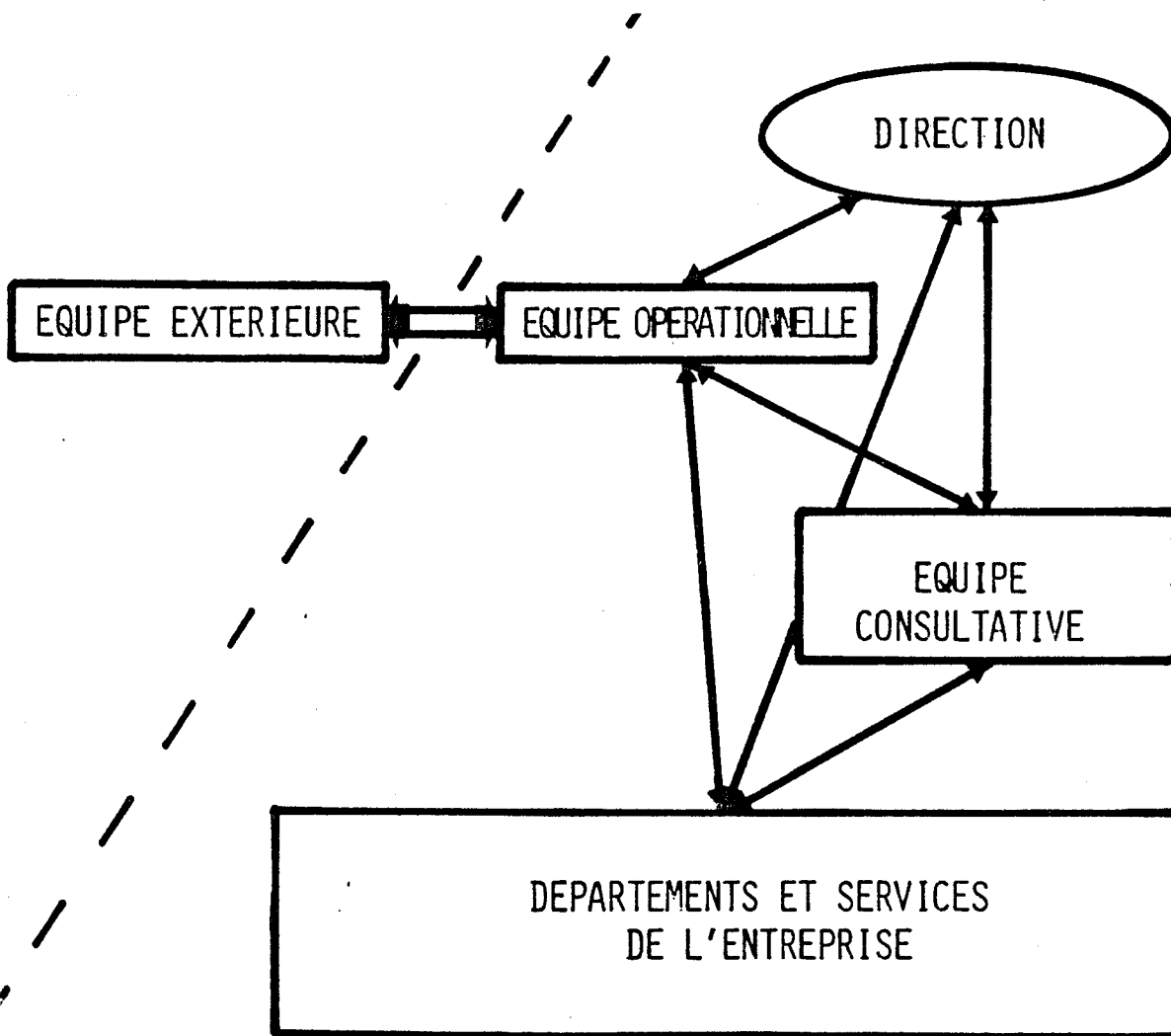


Figure III - 3.7 : Structure de l'équipe chargée de l'étude.

CHAPITRE 4

PROCESSUS DE CONCEPTION

RESUME: Ce chapitre est le premier de ceux consacré à l'invariant conceptuel. Il essaie de répondre à la question fondamentale "qu'est-ce-que la conception ? " en présentant différentes modélisations de l'activité de conception: les démarches explicite ou implicite, l'enchaînement des tâches, la boucle de conception, la conception comme un processus d'optimisation, et le traitement de problèmes bien et mal définis. L'intuition, la créativité, l'induction, la déduction et la compétence sont évoqués comme éléments de l'activité de conception. Un modèle de synthèse est alors présenté: celui-ci modélise l'espace de conception, le cahier des charges de l'objet à concevoir sous forme de contraintes et objectifs pondérés, l'objet en cours de conception; il classifie les opérations et montre la façon d'assurer la cohérence et la traçabilité du projet. Un langage permettant de spécifier ces différents projets est également décrit.

4. PROCESSUS DE CONCEPTION

4.1 QU'EST-CE QUE LA CONCEPTION ?

Nul doute qu'il n'est pas possible de vouloir assister la conception sans savoir en quoi consiste cette conception. Malheureusement les connaissances précises des règles qui régissent la conception sont encore faibles, ce qui rend toute définition discutable et constitue un obstacle que la CAO doit surmonter.

Néanmoins pour pouvoir mettre en place des systèmes CAO il faut modéliser même provisoirement et partiellement ce processus. C'est ce qui est de toutes façons fait dans la pratique opérationnelle de l'industrie où on n'a pas attendu des résultats théoriques pour organiser des processus de conception qui doivent produire des résultats. Dans ces schémas d'organisation on joue en réalité sur la composante la plus adaptable intervenant dans ce processus, l'homme, qui grâce ou malgré des schémas d'organisation, est capable de résoudre le problème qui lui est posé.

On constate finalement en étudiant de près ces structures qu'elles ne permettent pas de réaliser quoi que ce soit mais délimitent le champ d'action laissé au concepteur.

En CAO le problème est différent, car il ne s'agit pas de délimiter mais d'assister le concepteur de façon intime dans toutes les tâches qu'il effectue. Ceci implique une connaissance approfondie de son activité et dépasse donc largement l'aspect exprimé par un schéma d'organisation. L'effort de modélisation doit être nettement plus grand.

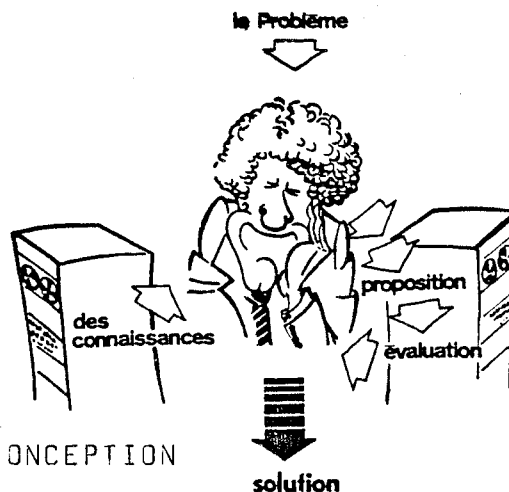


FIGURE II - 4.1 : CONCEPTION

4.2 MODELISATION DE L'ACTIVITE DE CONCEPTION

Les essais de modélisation du processus de conception ont été très nombreux. Chacun tente d'exprimer sous la forme choisie une partie de ce que l'on observe lors de la conception. Ainsi on s'est appuyé sur les mathématiques, l'automatique, l'optimisation, la théorie d'information, etc... Les modèles proposés sont nombreux. Nous pouvons en rappeler quelques uns:

- DEMARCHE IMPLICITE (boite noire) : on procède sans méthode apparente, sans justification, s'appuyant sur l'expérience et sur l'intuition (on synthétise sans savoir quoi),

Cette démarche est couramment utilisée par les architectes, elle leur laisse une entière liberté de comportement, de jugement et de responsabilité. La subjectivité est la pièce maîtresse de cette méthode qui par contre ne permet que très difficilement la justification et la collaboration.

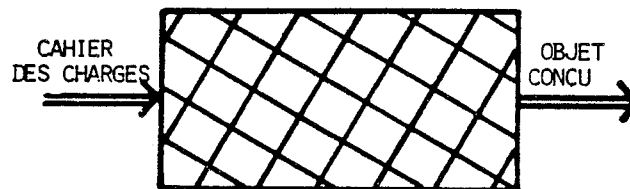


FIGURE II - 4.2 : BOITE NOIRE

- DEMARCHE EXPLICITE (boite transparente) : on procède de façon plus formelle selon un cheminement pré-défini, basé sur un raisonnement logique et en justifiant toutes les étapes.

Cette démarche s'appuie sur l'explicitation du déroulement du processus de conception, c'est à dire sur des schémas de comportement pré-définis selon lesquels le travail doit se dérouler. Ainsi on rend ce travail plus systématique, ce qui facilite le partage du travail et la collaboration entre différentes personnes et permet d'avoir recours aux procédés pouvant assister le concepteur dans son travail.

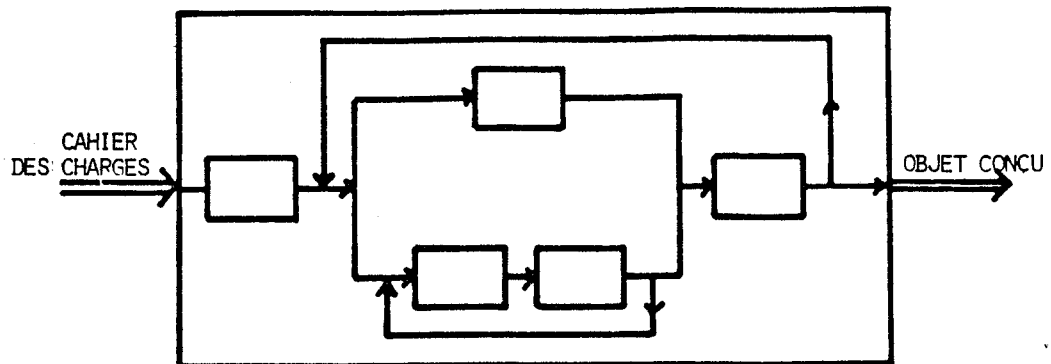


FIGURE II - 4.3 : BOITE TRANSPARENTE

- ENCHAINEMENT: la conception d'un objet, notamment d'un objet complexe n'est jamais faite d'un coup. Il paraît donc intéressant de mettre en évidence des étapes dans l'élaboration de l'objet. Ainsi on parle d'objet CONCEPTUEL, d'objet VIRTUEL, d'objet REEL ou de façon plus détaillée de MAQUETTE VIRTUELLE de l'objet, de la MAQUETTE REELLE, de PROTOTYPE d'objet réalisable, etc... En tout cas, il s'agit d'exprimer un découpage chronologique dans un déroulement du processus de conception ou de façon plus globale, du processus Conception - Etude - Développement (CED), si on ne se limite pas seulement à la conception mais si on prend en compte aussi la fabrication.

A ces différents stades de l'objet correspondent des étapes appelées par exemple: avant-projet, projet, industrialisation, fabrication.

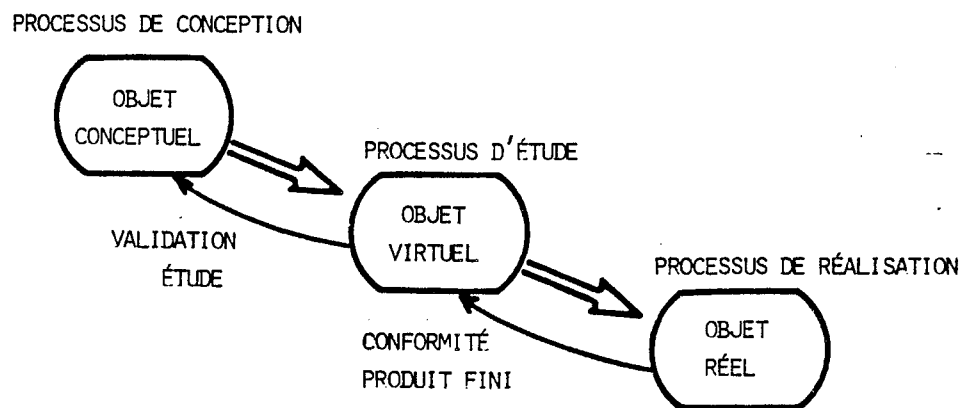


FIGURE II - 4.4 : PHASES DE CONCEPTION

- BOUCLE: il semble aussi important de modéliser le déroulement d'une étape de ce processus. On la représente généralement par la boucle: ANALYSE, SYNTHÈSE, ÉVALUATION, DÉCISION. On commence par analyser la partie concernée du cahier des charges puis on conçoit la solution. Cette solution doit être évaluée puis la décision de l'accepter ou pas doit être prise. Si elle est positive on passe à l'étape suivante si elle ne l'est pas on doit remettre en cause l'objet conçu et envisager une autre solution.

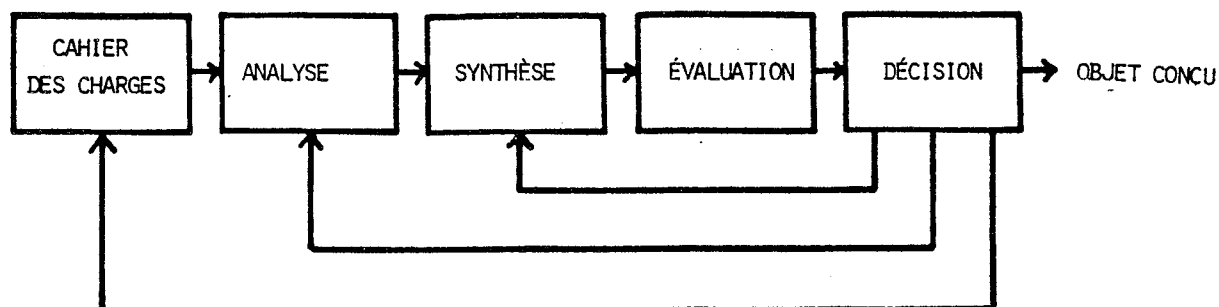


FIGURE II - 4.5 : BOUCLE DE CONCEPTION

- OPTIMISATION: souvent on a essayé de modéliser le processus de conception comme un processus d'optimisation. Dans ce cas, il s'agit d'identifier:
 - + les variables sur lesquelles le concepteur peut agir,
 - + les paramètres que le concepteur ne peut pas influencer,
 - + les contraintes permettant d'exprimer les liens entre les variables et les paramètres,
 - + la fonction-objectif qui exprime ce que l'on cherche.

L'intérêt de ce type de modélisation n'est pas aussi grand que l'on pourrait le croire. Il est très utile dans les ultimes étapes du processus où on peut véritablement parler d'optimisation, mais dans les étapes initiales, où on crée réellement on ne dispose pas, malheureusement, ou peut-être heureusement, d'un tel modèle. Pour pouvoir l'utiliser il faut le concevoir et cette conception fait partie de la conception de l'objet.

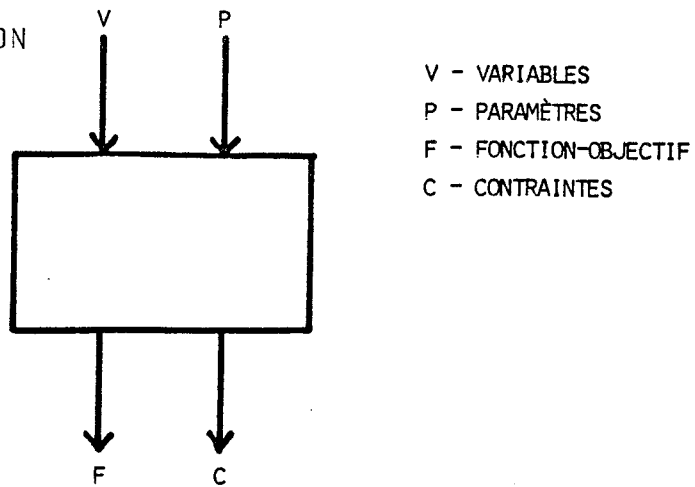


FIGURE II - 4.6 : PROCESSUS DE CONCEPTION COMME UN PROCESSUS D'OPTIMISATION

- PROBLEME BIEN ET MAL DEFINI: le paragraphe précédent met en évidence un aspect important de la conception; si on connaît bien ce que l'on veut obtenir, il est possible d'utiliser des techniques d'optimisation. On est en présence d'un problème bien défini, on peut donc utiliser pour sa résolution un modèle bien défini. Dans ce cas il s'agit souvent d'ajustement de valeurs, c'est à dire de ce que l'on appelle la conception paramétrique. Mais la véritable conception est la conception où on part d'un cahier des charges plus ou moins précis et on conçoit l'objet. Dans ce cas la création proprement dite est véritable car on doit découvrir progressivement l'objet aussi bien du point de vue de sa structure que de sa sémantique. Dans ce cas il s'agit d'un processus de transformation dans lequel on élabore peu à peu des versions de plus en plus précises du cahier des charges de l'objet à concevoir et de l'objet conçu. Les techniques utilisées sont aussi bien heuristiques qu'analogiques. Pour les problèmes mal définis (et tout problème est d'une certaine façon mal défini car sinon on n'a plus besoin de concevoir), on procède par itération avec une reformulation de plus en plus précise.

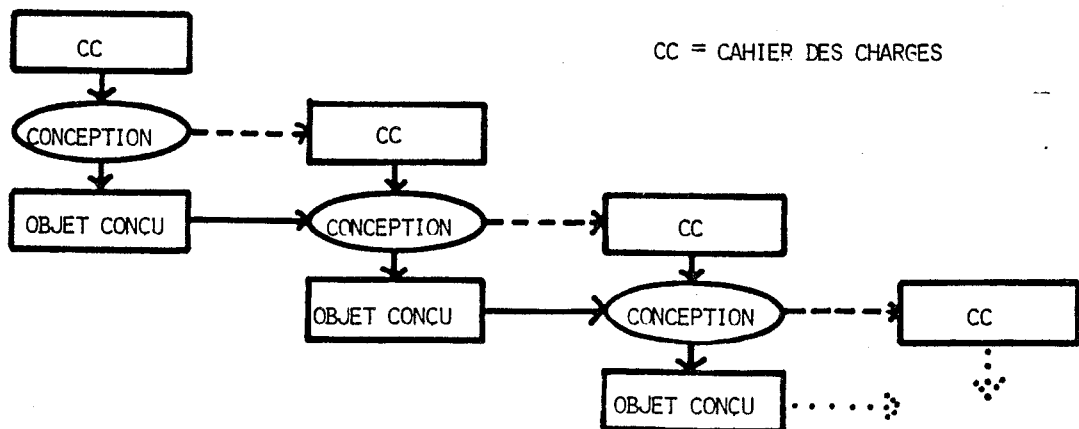


FIGURE II - 4.7 : PROBLEME MAL DEFINI

- INTUITION, CREATIVITE, INDUCTION, DEDUCTION, EXPERIENCE: pour bâtir une nouvelle version de l'objet à concevoir et de l'objet conçu on exploite alternativement diverses méthodes de résolution basées sur l'intuition, la créativité, l'expérience, la compétence, l'induction ou la déduction.

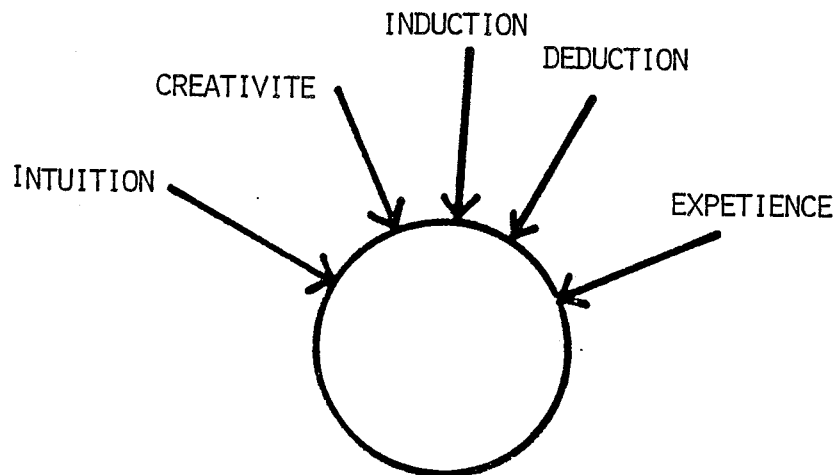


FIGURE II - 4.8 : ACTIVITE DE CONCEPTION

4.3 MODELE DE CONCEPTION

Le modèle de conception doit indiquer le ou les processus de conception selon lesquels la conception peut être effectuée. L'importance du modèle de conception n'est pas la même selon qu'il s'agisse d'une conception individualiste ou collective. Dans le premier cas, le processus de conception n'a pas à être transmis, il peut donc rester uniquement dans la tête du concepteur et celui-ci peut le modifier dynamiquement sans problème majeur, étant seul responsable de ses actes.

A partir du moment où on est en présence d'une conception collective, le problème de qui fait quoi et quand se pose. - Il s'agit donc de définir le déroulement du projet du point de vue des responsabilités d'une part, et du point de vue temporel d'autre part.

La CAO doit être considérée comme étant un cas particulier de la conception collective même si elle est utilisée par un seul concepteur, car il s'agit de partager le travail et les responsabilités entre l'homme et la machine.

Après avoir examiné des diagrammes plus ou moins compliqués, il est important de mettre en évidence les concepts de base qui régissent la conception. Ceci paraît indispensable pour pouvoir

définir des modèles dont le champ de validité est certain et qui remplissent des conditions importantes d'adaptabilité, d'évolutivité et d'extensibilité. Notre approche présentée en détail dans (Riv.77), est la suivante:

LA CONCEPTION est une activité, un processus qui produit un OBJET CONCU à partir d'un OBJET A CONCEVOIR. Elle comporte donc un processeur - le concepteur - et une procédure -méthode de conception - composée d'une séquence d'actions de conception: manipulation des objets par le concepteur.

La conception s'effectue dans un domaine de conception à l'intérieur duquel nous pouvons caractériser séparément deux sous-domaines:

- le domaine des objets,
- le domaine des concepteurs.

Dans le domaine des objets, on décrit le passage de l'objet à concevoir à l'objet conçu (procédure de conception). Nous avons distingué quatre aspects des objets: leur morphologie, leur définition, leur structure et leur résolution. Ces aspects constituent des directions ou critères de voisinage différents du domaine des objets.

- DIRECTION MORPHOLOGIQUE: de la fonction vers la forme, vers la structure, c'est à dire de l'abstrait et qualitatif au concret et quantitatif, ou encore des notions conceptuelles aux notions relationnelles, aux notions dimensionnelles et finalement, aux notions matérielles.
- DIRECTION DE DEFINITION: on s'intéresse à la fonction de la conception en tant que tel. Elle exprime qu'est-ce que l'on conçoit, pourquoi, et elle indique les objectifs sous forme de la définition de l'objet à concevoir, et donc, du problème de conception: le passage de l'objet à concevoir à l'objet conçu. Cette direction prend de l'importance pour les problèmes mal définis où il faut d'abord élaborer la définition de l'OAC. Pour les problèmes bien définis, le déplacement dans cette direction n'a pratiquement pas lieu.
- DIRECTION STRUCTURELLE: étude de la structure des objets pour structurer leur conception (objet, sous-objet, l'articulation des sous-objets). Cette direction est donc liée à la méthode de réduction de la théorie de résolution des problèmes.
- DIRECTION DE LA RESOLUTION: on appelle résolution le degré de détail dans la représentation d'un objet. C'est la quantité et la finesse de l'information contenue dans le modèle de l'objet (définition en profondeur). Cette direction est associée à la notion courante de phase de conception,

associée elle-même à la méthode descendante, ou par affinage successifs: conception préliminaire, conception schématique, conception détaillée.

Sur la Figure II - 4.9 (tirée de Riv.77), nous montrons les différentes directions de conception pour une définition donnée. Un processus particulier dans cet espace de conception est défini par un cheminement dans cet espace.

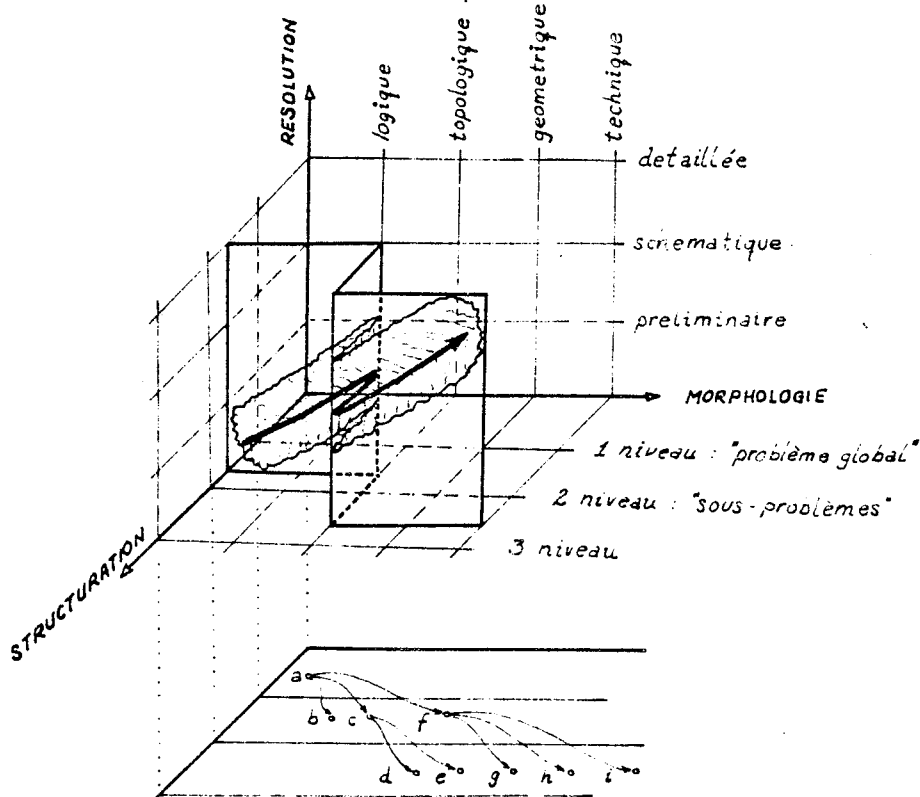


FIGURE II - 4.9 : PROCESSUS DE CONCEPTION DANS L'ESPACE DE CONCEPTION

On retrouve donc un graphe de conception avec les états de conception comme dans les autres modèles de conception. Mais par rapport à ces modèles, dans celui-ci nous pouvons caractériser très exactement les transitions entre états. Leur nature est définie par leur orientation dans l'espace et correspond à une transition (ou à une composition des transitions) appartenant à l'ensemble des transitions définies pour chaque direction. De cette façon, il est possible de gérer le changement d'état de façon cohérente et systématique. Nous avons remarqué que le changement d'état s'effectue la plupart du temps dans une seule direction selon la règle "une chose à la fois". Par contre, à partir d'un état donné, il faut presque toujours pouvoir aller dans les quatre directions.

On obtient donc un support de processus qui peut engendrer des processus très variés.

Dans le domaine des concepteurs, on décrit les trois caractéristiques des concepteurs, à savoir :

- la motivation (très difficile à prendre en compte), la seule qui semble actuellement exploitable est la motivation fonctionnelle qui correspond à la notion de responsabilité,
- l'intuition, qui agit dans le sens de la génération de variété pouvant faire appel aux méthodes de créativité,
- la compétence, qui agit dans le sens de la réduction de variété, et peut s'appuyer sur les méthodes rationnelles et l'acquis technique qui semble important de gérer.

Le lien ou la relation qui existe entre le domaine des concepteurs et le domaine des objets est la compréhension. On peut augmenter la compréhension d'un système-objet par la manipulation du système jusqu'à la détermination des relations de causalité et des lois de variation des différentes variables observées dans le système. C'est un processus d'apprentissage qui est lié aux notions d'analyse de système.

C'est la manipulation de l'objet conçu qui constitue la face visible du processus de conception proprement dit. A partir du cahier des charges, d'un champ de solutions possibles (l'acquis technique ou expérience du concepteur), le concepteur propose des solutions, puis les évalue.

A chaque étape de conception, la définition du produit se précise et passe d'un concept à une réalité. L'ensemble des descriptions retenues constitue le dossier de définition du produit et il importe donc de gérer celui-ci en conséquence. Il existe fréquemment plusieurs types de dossier concernant un même produit: dossier d'étude, de développement, de production, de maintenance

Ce qui rend le métier de concepteur difficile est certainement la multiplication et l'hétérogénéité des objectifs ou performances que doit satisfaire l'objet conçu. Jusqu'alors, ces performances étaient acquises séquentiellement pendant les étapes successives de la conception. Mais aujourd'hui, leurs interdépendances, ou leurs incompatibilités, imposent de leur attribuer un poids relatif et de les prendre en compte globalement. Cette tendance est amplifiée par la recherche de compétitivité et l'obligation plus rigoureuse du respect d'une politique industrielle définie. Cette dernière se traduit notamment par un ensemble de contraintes imposées ou discutées, qui sont à considérer par le concepteur comme autant d'objectifs ou de performances à satisfaire.

Il découle de ceci que le traditionnel "cahier des charges", généralement celui du client, se transforme en ce que l'on appellera "les objectifs et contraintes pondérés".

Nous entendons par:

- OBJECTIF: la définition du but à atteindre décrit de façon fonctionnelle,
- CONTRAINTE: la définition du moyen que l'on doit utiliser dans l'élaboration de l'objet ou dans l'objet lui-même,
- PONDERATION: un moyen d'exprimer le compromis entre l'ensemble des objectifs et des contraintes pouvant être contradictoires.

L'algorithme de conception aura pour entrée un ensemble d'objectifs et contraintes pondérés de niveau n et produira un ou plusieurs ensembles d'objectifs et contraintes pondérés du niveau n+1 (Fig. II - 4.10).

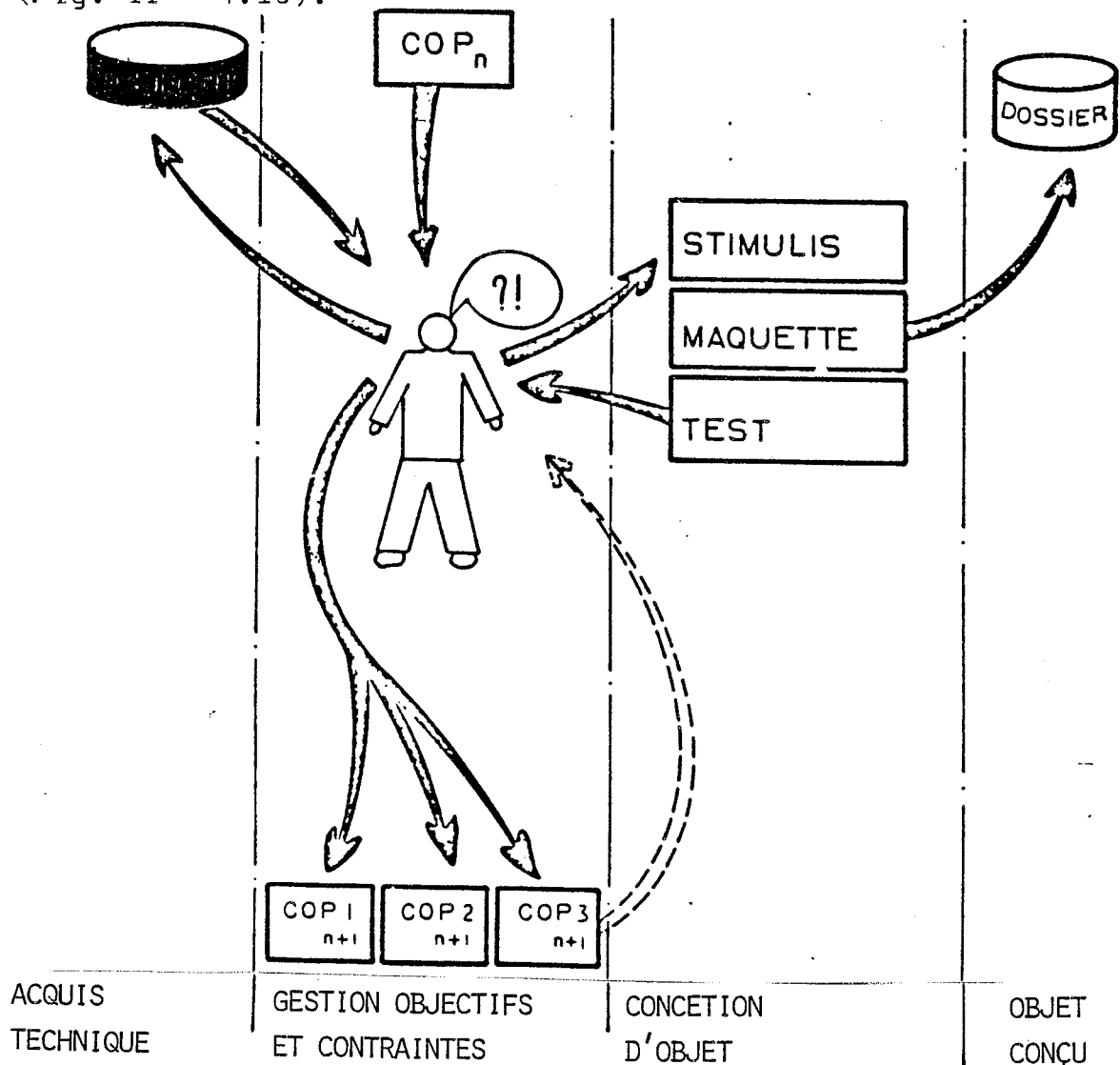


FIGURE II - 4.10 : ALGORITHME DE CONCEPTION

La gestion des objectifs et contraintes pondérés pose deux problèmes importants:

- il faut assurer la conformité et l'intégrité des objectifs et contraintes pondérés à chaque étape de transformation, problème difficile car leur expression doit être adaptée au "métier" du concepteur d'un niveau donné, mais doit aussi refléter exactement la décision du niveau supérieur formalisée dans un autre langage. Les notions de propriétés, associées aux entités conçues et reflétant leur aptitude à satisfaire un ensemble des objets et contraintes pondérés ainsi que leurs relations, devront pouvoir être manipulées et évaluées par des algorithmes logiques ou arithmétiques.
- la "traçabilité" du projet, ou suite cohérente des "pourquoi" et "comment", doit être mémorisée et exploitable. En effet, en cas d'impossibilité de satisfaire l'ensemble des objectifs et contraintes pondérés n au niveau $n+1$, il faut pouvoir remonter au niveau de décision supérieur pour redéfinir un nouvel ensemble.

En plus, il est important de pouvoir prévoir et gérer toutes les conséquences liées aux demandes de modifications.

Afin de donner une solution à ces deux problèmes nous proposons de procéder de la façon suivante: pour assurer la conformité et l'intégrité des ensembles d'objectifs et contraintes pondérés et des objets conçus nous définissons des opérations sémantiquement cohérentes qui assurent la sauvegarde de l'intégrité.

En plus nous les lions aux directions de l'espace de conception, c'est à dire que nous définissons par quelles opérations il est possible de passer d'un état à un autre selon la transition choisie et le sens du déplacement.

| <u>4 DIRECTIONS</u> | <u>MODIFIE</u> | <u>CONCERNE NON MODIFIE</u> |
|------------------------|------------------------------------|-----------------------------|
| . DEFINITION | OBJET A CONCEVOIR | OBJET CONCU |
| . ETAPES (MORPHOLOGIE) | OBJET CONCU | OBJET A CONCEVOIR |
| . PHASE (REALISATION) | OBJET CONCU | OBJET A CONCEVOIR |
| . STRUCTURE | { OBJET CONCU OBJET A CONCEVOIR | |

FIGURE II - 4.11 : OPERATIONS

Ainsi nous pouvons préciser (voir le tableau Fig II - 4.11) le comportement de l'objet à concevoir et de l'objet conçu lors du déplacement dans une direction.

Nous pouvons aussi préciser la nature de l'opération à effectuer lors d'un tel changement (voir Fig. II - 4.12).

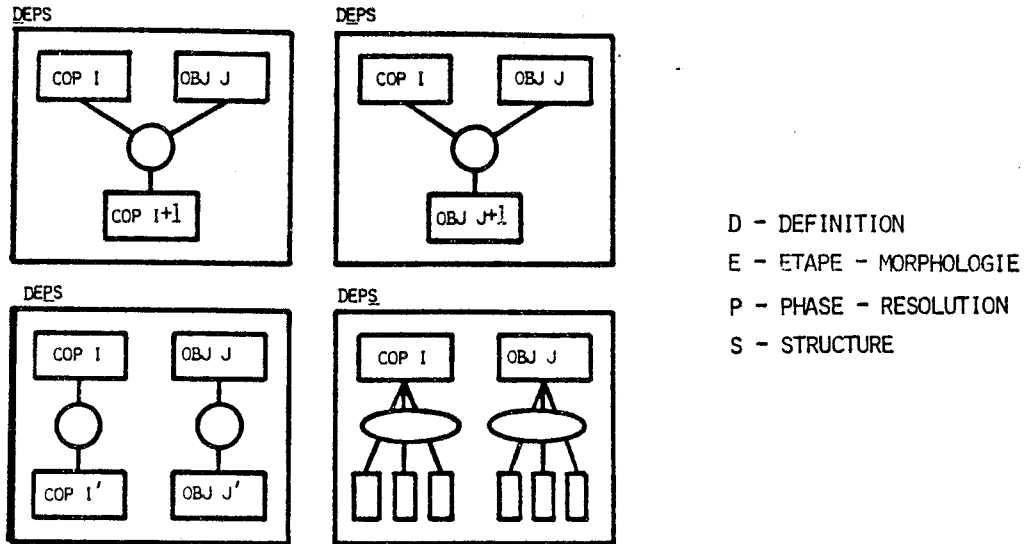


FIGURE II - 4.12 : OPERATIONS ET DIRECTIONS

Il reste à savoir quelles sont les opérations que l'on peut entreprendre au sein d'un état de conception (FIG.II-4.13). Dans notre formalisation toutes les opérations que l'on effectue dans un état de conception sont des opérations de mise à jour, car toutes les autres ont pour conséquence le changement d'état.

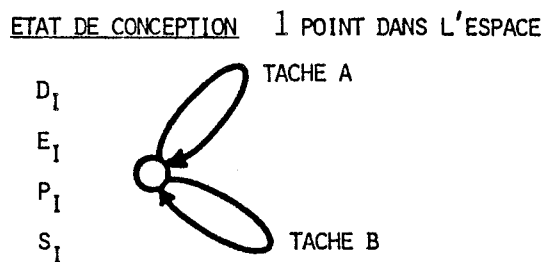


FIGURE II - 4.13 : OPERATIONS DANS UN ENVIRONNEMENT

Pour assurer la "traçabilité" du projet, nous proposons d'utiliser une RELATION DE CAUSE A EFFET (FIG.II-4.14), que le système gère à chaque modification et permet ainsi de retracer le déroulement du projet et éventuellement de revenir en arrière lors des impossibilités.

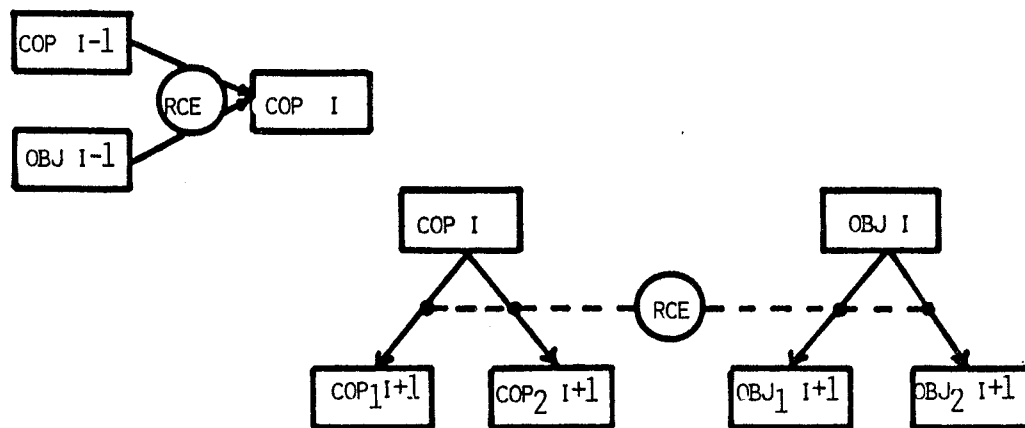


FIGURE II - 4.14 : RELATION DE CAUSE A EFFET

4.4 LANGAGE DE SPECIFICATION DU MODELE DE CONCEPTION

Le langage de spécification du modèle de conception doit permettre d'exprimer le déroulement de la conception concernant une application particulière. Il doit définir la structure et les cheminements dans le domaine de conception, c'est à dire l'organisation du processus de conception, du processus de décision, des manipulations des objets à concevoir et des objets conçus, ainsi que les tâches pouvant intervenir pendant le processus.

Nous nous limiterons ici à la définition des énoncés permettant la spécification de l'organisation du processus de conception et de décision. Les autres aspects seront traités dans les chapitres suivants.

L'énoncé ESPACE DE CONCEPTION (FIG.II-4.15) permet de spécifier statiquement l'espace de conception, c'est à dire les noms des graduations sur les axes de cet espace. On a le choix entre nommer explicitement ces graduations, ce qui est le cas pour les axes de DEFINITION, de RESOLUTION (étape), et de MORPHOLOGIE (étape), ou utiliser une spécification implicite comme pour l'axe de STRUCTURATION. Dans ce cas, on peut utiliser le mot STANDARD à la place d'un nom précis de graduation, c'est la nature de l'objet qui définit si la structuration est possible (un objet de base ne peut pas être décomposé). Il s'agit donc d'une spécification générique.

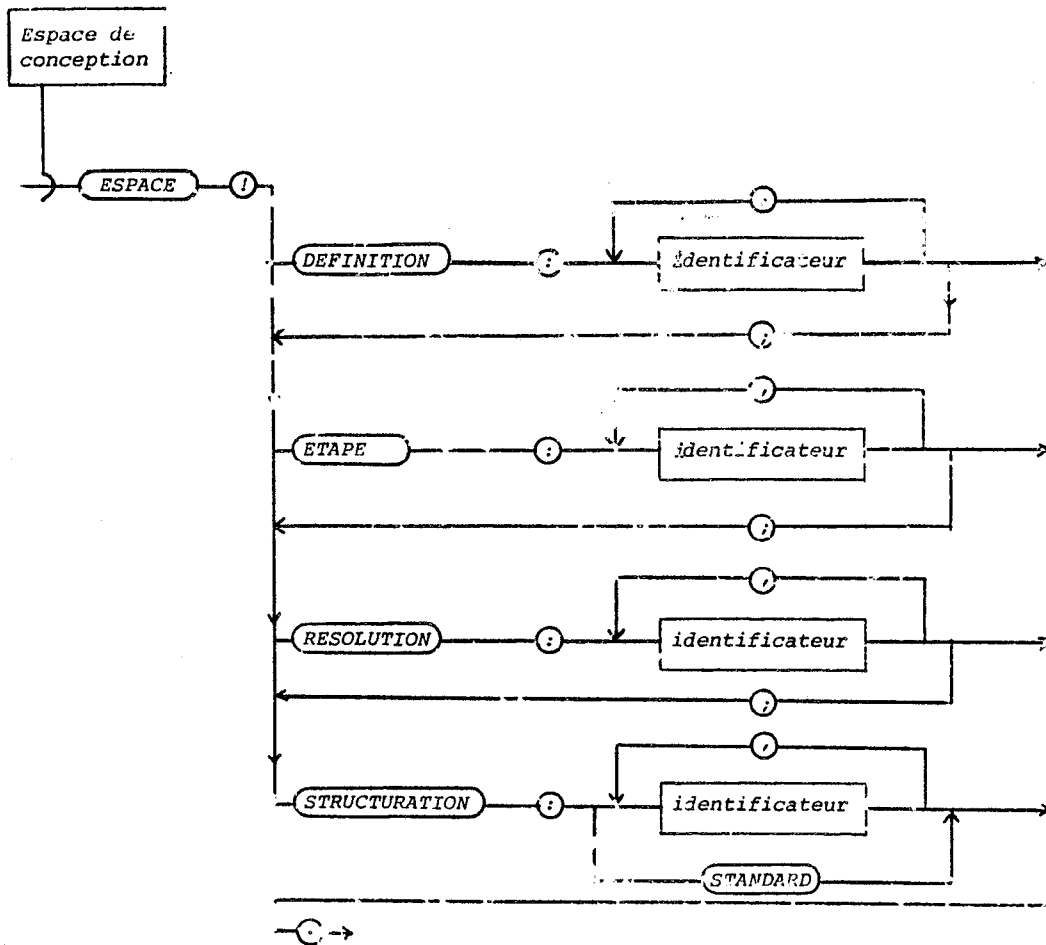


FIGURE II - 4.15 : ENONCE ESPACE DE CONCEPTION

l'énoncé PROCESSUS DE CONCEPTION (FIG.II-4.16) permet de spécifier la dynamique du processus de conception. Pour cela, on définit d'abord les états de conception, c'est à dire les points dans l'espace de conception correspondant aux environnements que l'on veut prendre en compte, puis les transitions entre ces états (environnements).

Pour obtenir un point dans l'espace, il faut donner les coordonnées du point, ce que nous faisons en donnant les références de l'état sur chacun des axes de conception.

Dans la définition des transitions, on nomme des transitions. Pour cela, ou bien on fournit pour chaque transition les noms de l'état initial et de l'état final, que l'on vient de créer, ou bien on ne nomme pas explicitement l'état, mais on fournit ses "coordonnées".

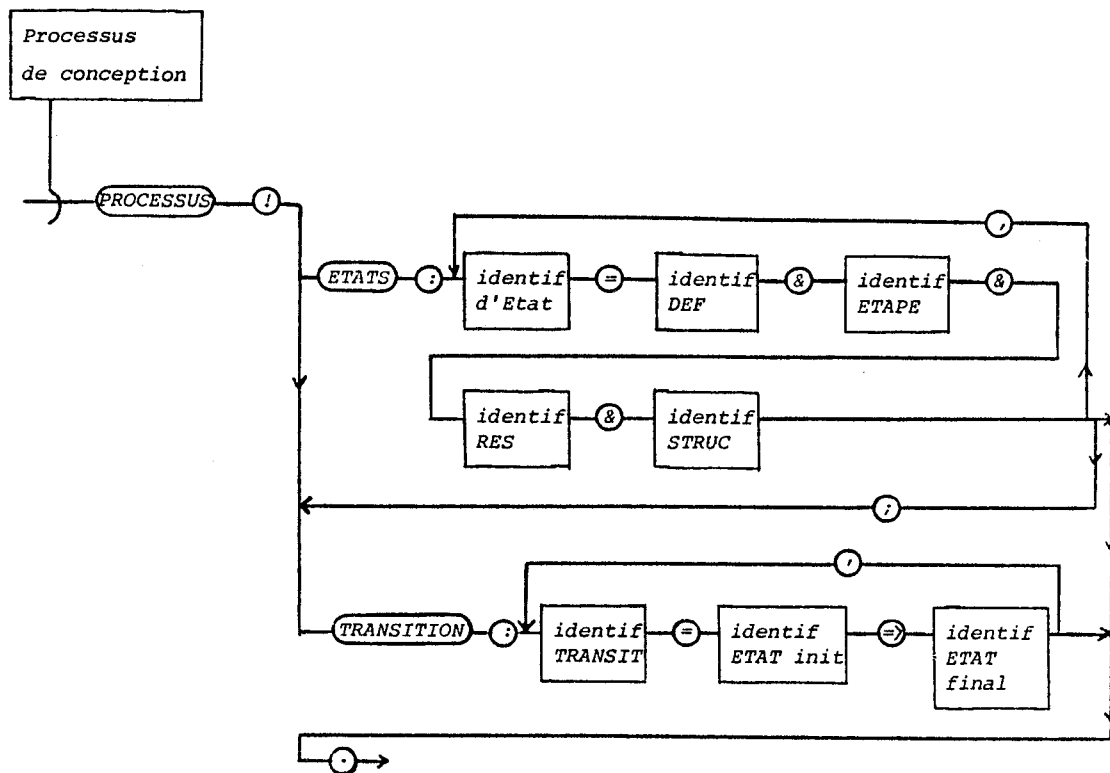


FIGURE II - 4.16 : ENONCE PROCESSUS DE CONCEPTION

Pour mener à bien un projet important nécessitant un travail d'équipe, il est indispensable de définir clairement le rôle exact de chaque participant. Pour mieux cerner les responsabilités de chacun, il est essentiel de distinguer la RESPONSABILITE DE DECISION de la RESPONSABILITE D'EXECUTION. A titre d'exemple, l'ingénieur qui fixe les grandes lignes d'un projet est responsable des décisions, par contre, le technicien réalisant le projet dans le contexte défini par l'ingénieur, est responsable de l'exécution.

Souvent, il est souhaitable de pouvoir disposer d'avis différents et de compétences nouvelles sans pour autant que la responsabilité soit modifiée. C'est le rôle des CONSEILLERS.

Celui du CONTROLEUR est également important car il assure que l'objet - conçu est conforme à l'objet -à - concevoir.

On est donc en mesure de donner à chaque participant un descriptif définissant clairement son rôle à tous les stades du projet.

Ce descriptif servira de filtre d'accès au projet, c'est-à-dire à la définition du projet, aux décisions, à la description de l'objet en cours de conception, aux traitements travaillant sur l'objet, aux données du patrimoine, etc...

Lors de la mise en place du système, on pourra vérifier que toutes les responsabilités ont été prévues et que toutes les décisions sont possibles.

Pendant le déroulement du projet, on pourra vérifier que toutes les responsabilités ont été distribuées et qu'il n'y a pas d'incohérence dans cette répartition: cumul trop important (abus de pouvoir) ou bien décision impossibles par manque d'informations.

L'énoncé AFFECTATION DES FONCTIONS (FIG.II-4.17) permet de spécifier les identificateurs de responsabilité qui définissent d'une façon fonctionnelle les droits de chaque type d'intervenant dans le projet.

Pendant le déroulement du projet, le chef du projet distribue (affecte) ces identificateurs aux personnes (physiques) participant au projet. Les identificateurs d'état sont soit explicites, soit implicites (construits directement comme dans l'énoncé précédent).

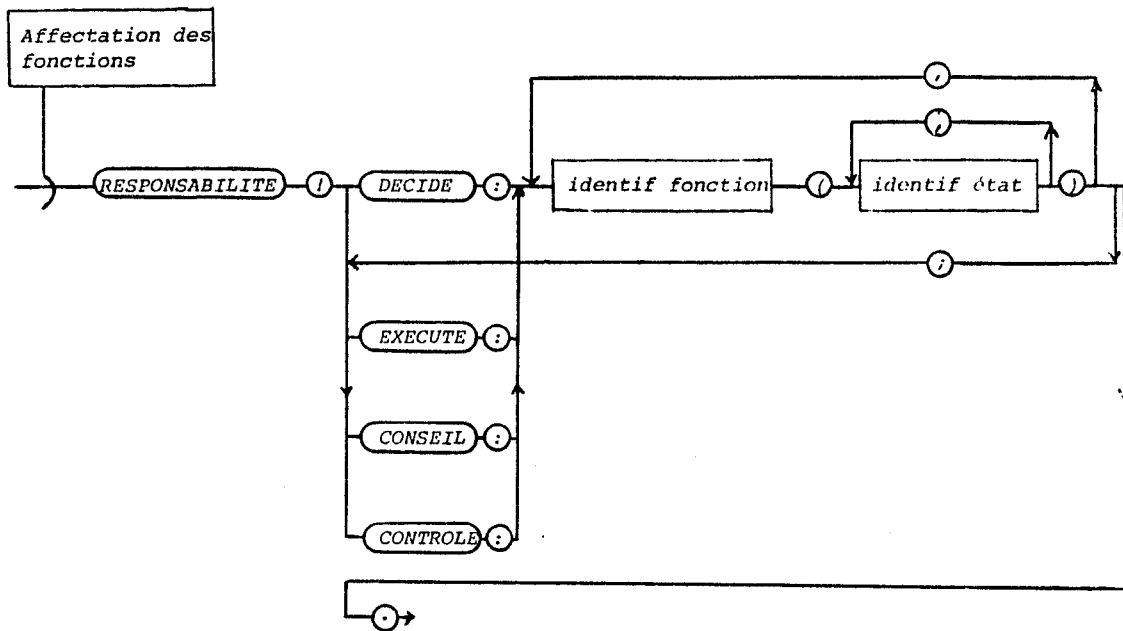


FIGURE II - 4.17 : ENONCE AFFECTATION DES FONCTIONS

Dans le chapitre 5 nous étudions la modélisation de l'objet à concevoir et de l'objet conçu et dans le chapitre 7 les algorithmes manipulant l'objet à concevoir et l'objet conçu. Dans les deux cas nous trouverons dans la spécification les références au processus de conception, afin d'assurer la cohérence de la conception et l'adéquation des objets et de leur manipulations par rapport au processus de conception.

CHAPITRE 5

DONNEES DANS LE PROCESSUS

RESUME: Ce chapitre expose un autre aspect de l'invariant conceptuel, celui des données utilisées dans le processus CED. Il met en évidence l'intérêt d'utiliser un système de gestion de bases de données spécifique à la CAO. Les données sont classées en trois types (générales, techniques ou propres au projet) et il y a une base de données pour chaque type. On propose alors un modèle pour la base de données du projet, en tenant compte de la contrainte la plus forte sur ces données (évolutivité ou extensibilité). Ce modèle s'appuie sur quatre composants fondamentaux: la structure, la sémantique, la cohérence et les opérations. Les essais de mise en oeuvre présentés utilisent soit un ensemble de fichiers soit un système de gestion de base de données. La dernière partie est consacrée à l'étude des accès aux données.

5. DONNEES DANS LE PROCESSUS

Le second aspect important de la CAO (après le processus de conception) concerne la gestion des données dont on a besoin pendant le déroulement du processus de conception. Ces données sont très diverses et au moins pour certaines, leur vie au sein du système CAO est très mouvementée. C'est pourquoi on envisage de plus en plus d'utiliser un système de gestion de base de données pour les gérer.

5.1 - POURQUOI UNE BASE DE DONNEES EN CAO ?

Pour répondre à cette question, il faut étudier l'évolution de la CAO.

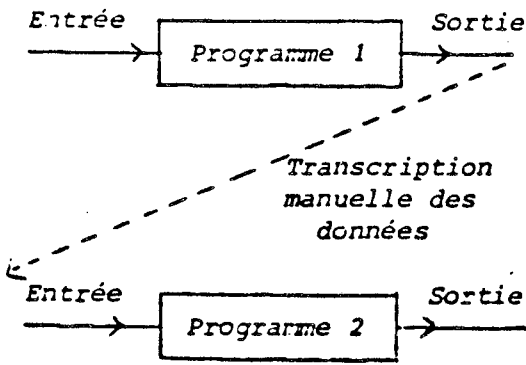
La CAO s'était initialement limitée à la recherche de tâches du processus de conception dans lesquelles l'ordinateur pouvait être utile. Le passage d'un programme à un autre s'effectuait par la transcription manuelle des données (Fig. II - 5.1a). Comme ce travail devenait assez considérable, et que son aspect manuel était une source d'erreurs très courante, on a cherché à constituer des séries de programmes s'enchaînant automatiquement.

La première solution (Fig. II - 5. 1b) raccordait les différents programmes en remplaçant la transcription manuelle des données par une transcription automatique, c'est à dire que l'on produisait des fichiers de données pour chaque programme.

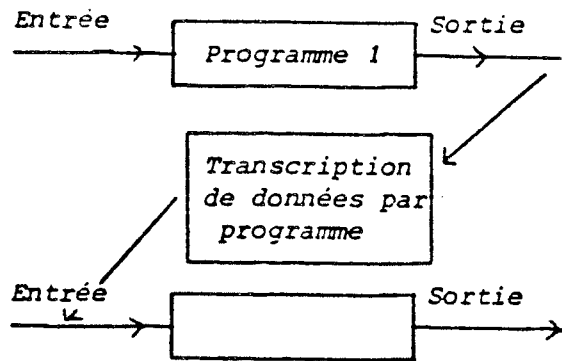
De cette façon, on constituait effectivement des séries de programmes avec un enchaînement prédéfini. Mais les programmes de transcription de données étaient souvent très longs et le temps de transformation devenait insupportable. De plus, l'insertion d'un nouveau programme nécessitait l'écriture de deux programmes d'interface. On a donc cherché à homogénéiser cette démarche en essayant de supprimer les programmes de transcription. Pour ce faire, on a remis en cause l'organisation des données d'une part, et la façon d'effectuer les entrées/sorties dans les programmes, d'autre part.

Pour les données, on a choisi de constituer une base de données intégrée (Fig. II - 5.1c), capable de recevoir toutes les données nécessaires à l'ensemble des programmes. Ces données ont été stockées directement dans les formats adéquats, sans transcriptions importantes.

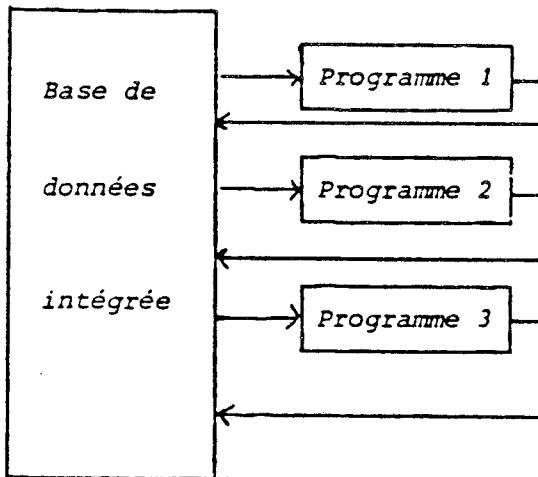
Pour les programmes, on a procédé à l'éclatement de leur structure en séparant les parties d'entrées/sorties et la partie de traitement proprement dite. Les entrées/sorties classiques ont été remplacées par des accès à la base.



a) Programmes individuels

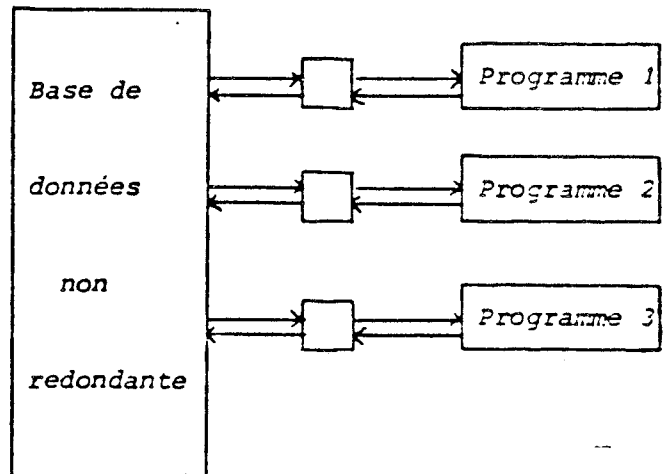


b) Programmes individuels enchaînés



L'ordre d'enchaînement figé

c) Base de données intégrée



L'enchaînement à la demande

d) Base de données non redondante

FIGURE II - 5.1 : DIFFERENTS NIVEAUX D'ENCHAINEMENT

On a constitué ainsi une organisation naturelle des données car les données de chaque programme ont été bien définies et stockées. Néanmoins, cette organisation pose un problème important: beaucoup de données stockées sont liées fonctionnellement. La modification d'une donnée demande une mise à jour des données qui lui sont liées. Maintenir l'intégrité des données est très compliqué et coûteux. En imposant un ordre d'enchaînement, on peut espérer pouvoir maîtriser ce problème.

Une autre solution consiste en une base de données ne contenant que les informations non redondantes; les informations redondantes sont produites (à partir des données stockées) chaque fois que c'est nécessaire. Dans ce cas, les données fonctionnelles liées sont réduites à des parties indépendantes et stockées de façon indépendante. Quand un programme particulier demande des données organisées d'une façon particulière, elles sont calculées à partir des données stockées, fournies aux programmes, puis réintégrées dans la base sous une forme non redondante (Fig. II - 5.1d). De cette façon, le problème de la cohérence devient abordable. De plus, on est en présence d'une base de données plus compacte et plus maniable. Il en découle la possibilité d'enchaînement à la demande, et une extensibilité plus grande.

5.2 SPECIFICITE DES BASES DE DONNEES POUR LA CAO

Une autre question à laquelle nous voulons répondre au moins partiellement, est de savoir s'il existe une différence et de quelle nature entre les SGBD classiques et ceux dont a besoin la CAO.

Il semble utile de distinguer trois bases de données dans un système CAO:

- BASE DE DONNEES GENERALE (B.D.G.): base d'informations générales de l'entreprise, concernée par la gestion du personnel, par la gestion budgétaire et financière etc...
- LA BASE DE DONNEES TECHNIQUE ET DE CONNAISSANCES (B.D.T.): contient l'ensemble des informations techniques de l'entreprise, c'est à dire la description de tous les composants utilisables, le parc du matériel disponible, les fournisseurs, les délais de livraisons, la charge du matériel, etc... ainsi que des informations sur la conception: comment concevoir, méthodes de conception disponibles, des projets déjà résolus, etc...
- BASE DE DONNEES DU PROJET (B.D.P.): contient toutes les informations sur le projet, pendant toute sa durée. c'est à dire d'une part l'objet à concevoir (cahier des charges du

projet) et d'autre part l'objet conçu (description de l'objet réalisé par rapport à ce cahier des charges).

Si nous nous intéressons maintenant à ces trois bases de données, nous pouvons constater que la base de données générale est une base de données classique pouvant être utilisée de façon habituelle, elle n'est ni spécifique à la CAO, ni utilisée uniquement en CAO. Un utilisateur CAO (le concepteur) va donc accéder à cette base de données comme n'importe quel autre utilisateur (que ce soit directement ou par l'intermédiaire du système CAO). Par contre, la base de données du projet (BDP) est une base de données très spécifique à la CAO. Nous soulignerons les points suivants:

- VOLUME D'INFORMATION: pour une BDP le volume d'information est en constante progression. En effet, si dans la base de données classique on travaille surtout en mise en jour, dans la BDP on est constamment en création: à partir d'un cahier des charges réduit on élabore l'objet conçu.
- DUREE DE VIE: pour une BDP la durée de vie est relativement courte, allant de quelques jours à un an, ce qui est très peu par rapport aux bases de données de gestion. On peut donc se permettre de ne pas faire une récupération de la place perdue aussi efficace que dans l'autre cas.
- COMPLEXITE DES OBJETS: les informations manipulées sont souvent très complexes. Par exemple, la description géométrique d'une pièce mécanique représente un volume d'information assez important, qu'il faut manipuler dans son ensemble.
- STRUCTURE EVOLUTIVE: par rapport aux problèmes traités en gestion, la réalité en CAO est beaucoup plus mouvante. Le concepteur lui-même définit le découpage de l'objet en sous-objets, et ce découpage peut varier pendant la durée de vie de l'objet. Il n'est limité ni en nombre de niveaux, ni en nombre de parties.
- EXTENSIBILITE: il est indispensable de permettre d'introduire de nouvelles caractéristiques de l'objet, et de les prendre en considération lors de divers traitements sans devoir reconstruire la base à chaque fois.
- ACCES A LA BDP: la manipulation des données ne s'effectue presque jamais à l'aide d'un langage d'interrogation de la base, mais pratiquement toujours de façon transparente pour l'utilisateur et par l'intermédiaire de modes d'accès programmés, activées directement depuis le programme que l'utilisateur veut exécuter. C'est ainsi qu'un programme de simulation électrique va demander toutes les informations sur l'organisation et le fonctionnement électrique de

l'objet. De la même façon un programme de dessin va demander les informations géométriques nécessaires à la réalisation du dessin.

- TYPES D'ACCES: selon la nature du travail, il faut aller chercher des informations homogènes (description graphique de l'objet) ou bien des informations ponctuelles. Il semble qu'en CAO le premier cas soit le plus fréquent.
- MULTI-HIERARCHIE DANS LA STRUCTURATION DES OBJETS: la structure n'est pas unique mais varie selon les responsabilités de l'utilisateur, les phases de conception, ce qui nécessite une organisation multi-hiérarchique évoluant dynamiquement.
- POINT DE VUE: l'utilisateur doit avoir la faculté d'exprimer son angle de vision du projet. C'est ainsi que le concepteur du schéma de principe du fonctionnement logique d'un calculateur, doit avoir accès à des informations différentes de celles du concepteur de l'armoire dans laquelle le calculateur sera implanté.
- COHERENCE SEMANTIQUE: le souhait d'assurer la cohérence sémantique dans la Base de Données du Projet (BDP) semble être légitime.

Cette liste n'est pas exhaustive, mais correspond aux performances les plus importantes qu'un système de gestion de base de données doit fournir en tant que BDP.

D'autres problèmes se posent, notamment celui du choix du lieu d'exécution des programmes : faut-il travailler directement sur la base de données, donc intégrer les fonctions d'accès à la base directement dans le déroulement de l'algorithme, ou bien est-il plus intéressant d'amener les informations nécessaires à l'algorithme avant le traitement puis de les réécrire dans la base après celui-ci? La réponse n'est pas simple, et elle varie selon la nature du traitement et la façon dont les modes d'accès sont réalisés.

Pour la base de l'acquis technique, la spécificité est beaucoup moins marquante. Elle doit en effet, pouvoir vivre très longtemps, contenir des informations assez stables, en cela elle est proche d'une base de données classique, par contre elle doit contenir des résultats de conception, donc des objets de complexité certaine, mais elle doit surtout offrir des modes d'accès très évolués, permettant des recherches par contexte, ou des recherches analogiques (pièces ressemblant à une nouvelle pièce), etc... Il semble qu'il ne faut pas sous-estimer sa complexité.

5.3 RECHERCHE D'UN MODELE POUR LA BASE DE DONNEES DU PROJET

Nous allons d'abord étudier comment définir et réaliser une BDP en tenant compte des objectifs présentés ci-dessus.

Il semble que les contraintes les plus fortes soient celles de l'évolutivité et de l'extensibilité. En effet dans les applications classiques des SGBD, notamment en gestion, il est possible de définir complètement au préalable l'univers dans lequel cette application va se dérouler. On peut donc figer la structure et la sémantique des informations qui seront stockées et manipulées.

En CAO ceci n'est possible ni pour la structure, ni pour la sémantique, car la structure est conçue pendant le processus de conception au même titre que les autres aspects de l'objet de la conception. Ceci est aussi vrai pour la sémantique que l'on veut prendre en compte, car celle-ci varie d'un projet à l'autre et dépend également des compétences du concepteur.

Nous définissons l'EVOLUTIVITE comme une possibilité de faire progresser dynamiquement la STRUCTURE de l'objet décrit et la SEMANTIQUE attachée à cette structure.

Nous définissons l'EXTENSIBILITE comme une possibilité de prendre en compte de nouvelles caractéristiques (propriétés sémantiques et relations), de nouvelles organisations structurelles.

La prise en compte de ces possibilités d'évolution et d'extension modifie totalement le comportement du modèle que nous devons dégager, et qui sans cela ressemblerait beaucoup aux modèles classiques utilisés dans les SGBD.

A partir du moment où on prend en compte l'évolutivité et l'extensibilité des données, il n'est plus possible de travailler avec des organisations de gestion basées sur des structures déclaratives et on doit utiliser des structures interprétatives, c'est à dire des structures adaptables pouvant être facilement modifiées.

La démarche que nous avons choisie est la suivante: dégager un modèle abstrait, le formaliser et fournir des moyens et des outils pour construire sur celui-ci des modèles concrets, voire des modèles paramétriques, comme nous l'avons expliqué dans le chapitre 3.

Le modèle abstrait ne peut plus être monolithique, et nous devons donc mettre en évidence des éléments à l'aide desquels il est possible de bâtir le squelette de l'organisation et du fonctionnement de la BDP.

Ce modèle doit pouvoir respecter le but du processus de conception qui est d'élaborer l'objet conçu correspondant à la définition exprimée par l'objet-à-concevoir. Cette définition est représentée par les objectifs et les contraintes, et s'affine au fur et à mesure de l'avancement du processus de conception. L'objet lui-même suit cette évolution. Pour faire face aux contraintes d'évolutivité et d'extensibilité nous avons mis en évidence quatre composants fondamentaux du modèle. Il s'agit de:

- LA STRUCTURE : ayant pour rôle d'exprimer l'organisation structurelle du projet,
- LA SEMANTIQUE : exprimant la signification des éléments organisés structurellement,
- LA COHERENCE : assurant l'évolution cohérente de l'objet pendant la conception,
- LES OPERATIONS : fournissant des outils à l'aide desquels on peut manipuler les informations, et les faire évoluer pendant le processus.

5.3.1 Structure

La structure doit permettre de matérialiser différentes méthodes de conception (descendante, ascendante, mixte), c'est pourquoi on se ramène la plupart du temps à une structure hiérarchique (ou arborescente). Elle doit aussi pouvoir exprimer les différents points de vue. Une telle structure a reçu, dans la littérature (Eas.78), le nom de hiérarchie abstraite.

Cette structure multi-hiérarchique doit être construite et modifiée dynamiquement (Fig. II - 5.2). Elle ne peut donc pas être définie une fois pour toutes. Il faut partir des éléments de la structure et des liens.

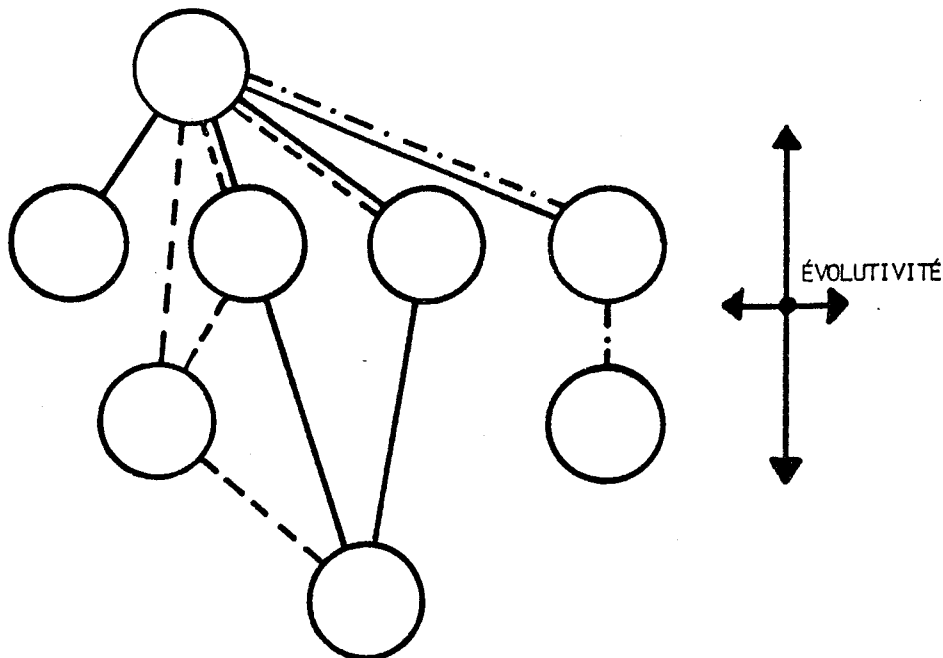


FIGURE II - 5.2 : STRUCTURE MULTI-HIERARCHIQUE EVOLUTIVE

La première notion de base de la description structurale est la notion d'ENTITE. Une "ENTITE" est un contenant qui peut recevoir une description d'"objet" de complexité quelconque, mais qui est toujours un objet "à concevoir" (cahier des charges), ou "conçu".

Les "ENTITES" doivent pouvoir être composées entre elles et permettre ainsi une description hiérarchisée.

Pour exprimer cette organisation il faut utiliser des liens. La deuxième notion de base de la description structurale est la notion de RELATION. Une "RELATION" représente un lien entre entités. Nous faisons la distinction entre des relations verticales (de structuration) et des relations horizontales (de fonctionnement).

5.3.2 Sémantique

La structure n'est qu'un aspect de la description, la signification (la sémantique) en est un autre. Pour exprimer cette signification nous introduisons la notion de "PROPRIETE".

Les objets et les relations qui se trouvent dans la hiérarchie sont donc caractérisés par les propriétés. Ces propriétés doivent pouvoir évoluer dans le temps.

Une entité est décrite par les "PROPRIETES" qui lui sont attachées. Ces propriétés permettent de qualifier et quantifier l'objet, avec un degré de définition évoluant au cours de la conception. Les propriétés peuvent être classées par familles (ex.: secteur d'activité, spécialité,...).

Ayant présenté comment on peut exprimer la structure et la sémantique, nous devons maintenant en décrire l'utilisation.

Comme il n'est pas possible de figer des entités, des relations et des propriétés en les énumérant, nous devons introduire un moyen pour travailler de façon plus progressive. Ceci est réalisé grâce à la notion de GENERICITE connue aussi sous l'appellation de type ou de modèle.

Le principe en est le suivant: on crée d'abord les modèles ou les types d'objet que l'on souhaite manipuler, puis à l'utilisation on choisit des valeurs précises et on crée ainsi des occurrences particulières d'objets. De cette façon il est possible de prévoir les types d'objet que l'on veut manipuler et de les valuer à l'utilisation.

L'évolutivité structurelle est ainsi assurée, car l'utilisateur peut à l'aide de ces éléments bâtir une structure de son choix.

Mais ceci n'est pas suffisant, car en CAO, il faut pouvoir prendre en compte de nouvelles caractéristiques de l'objet de façon tout à fait dynamique. Il n'est donc pas toujours possible de prévoir, dans la phase de création du modèle conceptuel concret, toutes ses caractéristiques. Il faut donc définir un moyen encore plus général et plus souple que la genericité classique, pour introduire de nouveaux types d'éléments dans la phase d'utilisation du modèle conceptuel concret. Ceci est en effet nécessaire pour assurer l'évolutivité sémantique.

Une solution consiste en l'utilisation d'une structure interprétative.

Pour cela on peut partir d'une seule entité pour l'ensemble des objets traités. Cette entité est tout à fait générale car elle est capable de recevoir des informations de toutes natures. (Nous restons pour l'instant au niveau purement conceptuel et nous laissons de côté le problème de la réalisation.) Cette entité (Fig. II - 5.3) doit pouvoir contenir toutes les informations nécessaires, c'est à dire le nom de l'entité, son contenu sémantique décrit par un ensemble de propriétés et son

comportement structurel décrit par un ensemble de relations hiérarchiques et fonctionnelles.

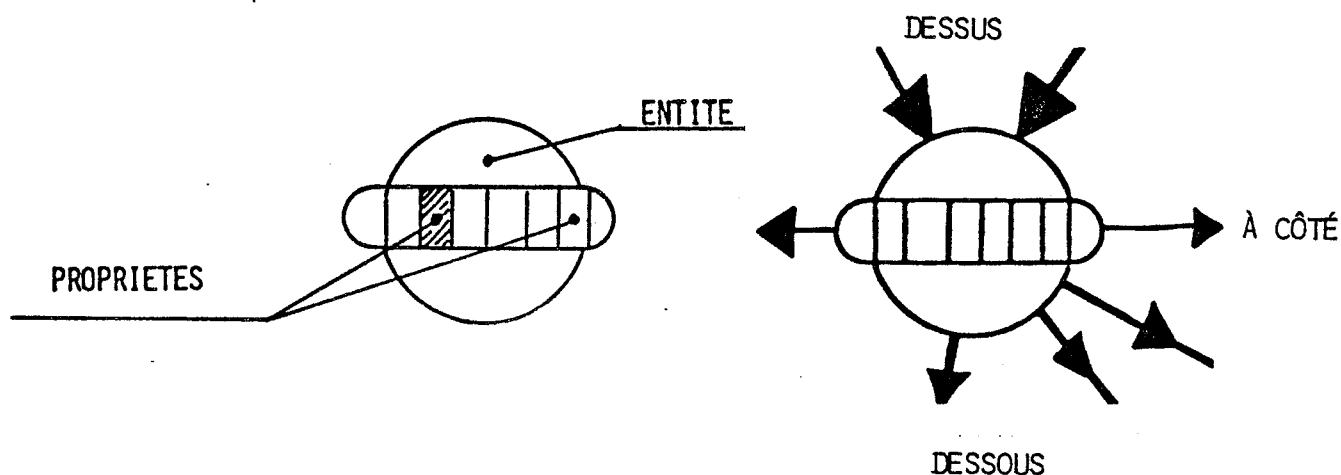


FIGURE II - 5.3 : ENTITE GENERALE

Malheureusement cette définition est insuffisante, car nous n'avons aucun renseignement sur la nature des informations stockées dans l'entité. Afin d'y accéder et de les manipuler il nous faut avoir des caractéristiques plus précises tout en gardant la souplesse d'utilisation. Pour cela au lieu de figer des entités, nous allons commencer par figer les éléments constitutifs des entités. Nous allons d'abord définir des types de propriétés. Nous indiquons de façon générique le comportement d'une propriété d'un type donné. Ce comportement se traduit par le type de valeur que la propriété peut prendre et par un ensemble d'opérations permettant de manipuler cette propriété.

Pour faciliter la mise en œuvre de types de propriétés, il convient de permettre une construction de types de propriétés à partir de types de base. Dans ce cas il est possible de définir dynamiquement des propriétés composées en composant des types de valeurs et des opérations pouvant les manipuler. Cette méthode de construction s'avère très utile pour assurer l'évolutivité et l'extensibilité (Fig. II - 5.4).

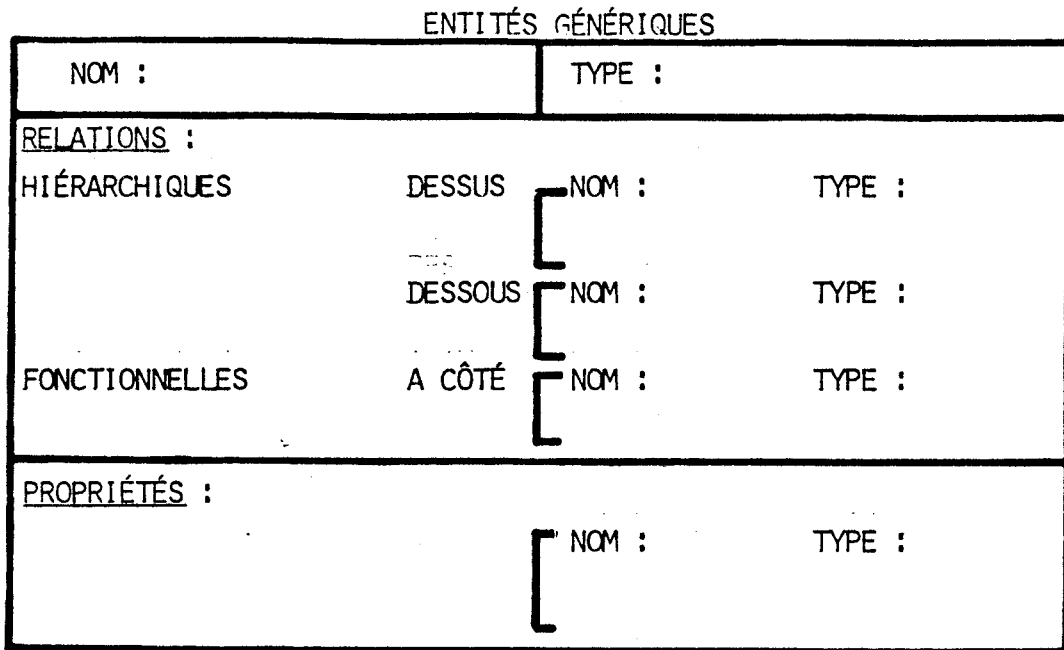


FIGURE II - 5.4 : ENTITE GENERIQUE - DEFINITION DETAILLEE

La mise en place d'une entité peut s'effectuer de différentes façons:

- de façon DIRECTE : on prend l'entité générique et on définit en même temps les types de caractéristiques souhaitées et les valeurs associées. Cette entité n'est connue que par son nom propre,
- de façon INDIRECTE : on prend l'entité générique et on définit les types de caractéristiques que l'on souhaite prendre en compte, puis on sauvegarde ce modèle d'entité comme un modèle particularisé. Lors de l'utilisation on peut valuer directement ce modèle d'entité et créer des occurrences,
- de façon PROGRESSIVE: on procède de façon indirecte mais en plus il est possible de sauvegarder des modèles partiellement valués.

L'utilisateur peut définir dynamiquement une nouvelle entité en précisant l'ensemble de caractéristiques propres et des relations qui la définit. Cette entité peut être manipulée comme telle, ou bien elle peut être nommée et conservée comme un type d'entité. De cette façon il n'est plus nécessaire de la reconstruire entièrement lors de la prochaine utilisation, mais il suffit de la valuer.

Aussi bien pour les entités typées que non typées, il faut supposer que la définition courante n'est que partielle, et que l'utilisateur voudra et devra pouvoir introduire de nouvelles

caractéristiques propres ou des relations, et modifier celles existantes.

5.3.3 Cohérence

Le rôle de la BDP n'est pas seulement de stocker des informations du projet, elle doit aussi assurer et garantir leur cohérence. Cela doit être fait tout au long du déroulement du processus de conception (Fig. II - 5.5). Il ne s'agit pas seulement de la cohérence physique (protection contre le mal fonctionnement du système de gestion) que doit assurer tout SGBD, mais surtout de la cohérence logique liée à un projet précis. Cette cohérence logique doit être maintenue pendant tout le déroulement du projet.

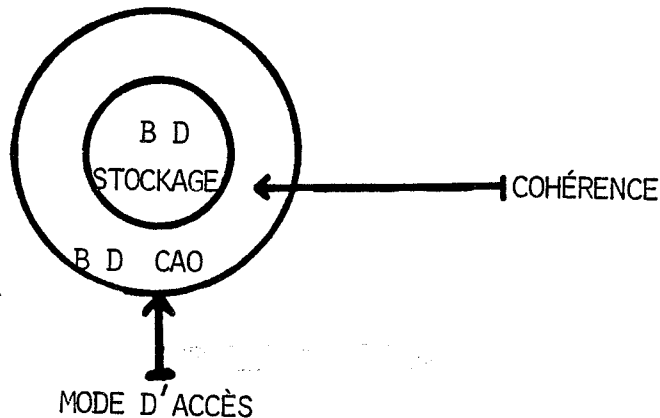


FIGURE II - 5.5 : MAINTIEN DE COHERENCE DE LA BDP

L'évolution des informations peut s'effectuer selon deux grandes classes de manipulations:

- manipulations locales: modifications ponctuelles, souvent interactives, d'une partie de la description,
- manipulations globales: changement global et systématique de la représentation.

Il faut donc assurer la maintien de la cohérence dans ces deux types de manipulations.

Pour assurer l'évolution de l'objet aussi bien du point de vue de sa structure que de ses caractéristiques, et pour garantir la cohérence de cette évolution, nous avons défini cinq types de contraintes de cohérence (Fig. II - 5.6):

- les liens horizontaux entre les propriétés du même composant: nous parlerons de la dépendance des propriétés,

- les liens verticaux entre les propriétés de différents composants: nous parlerons de la correspondance des propriétés dans la hiérarchie,
- les transformations de représentations lors de changement de phase dans le processus de conception: nous parlerons de transformations, (manipulations globales),
- les contraintes d'utilisation par rapport au déroulement du processus de conception: nous parlerons de l'utilisation,
- les contraintes d'accessibilité par rapport aux fonctions attribuées au sein de l'équipe de conception: nous parlerons d'accessibilités.

| | |
|---------------------|---------------------------------|
| VERTICALE | CORRESPONDANCE |
| HORIZONTALE | DÉPENDANCE |
| PROGRESSIVE | TRANSFORMATION |
| UTILISATION-QUAND | ACCESSIBILITÉ CONCEPTUELLE |
| UTILISATION-PAR QUI | ACCESSIBILITÉ ORGANISATIONNELLE |

FIGURE II - 5.6 : CINQ TYPES DE COHERENCE LOGIQUE

Les propriétés d'une entité peuvent être dépendantes. Dans ce cas, les relations de DEPENDANCE entre propriétés définissent ces liens (ex.: surface d'une pièce et son contour). Ces relations de dépendance entre propriétés (contraintes d'intégrité), sont des invariants pendant la conception. (Fig. II - 5.7).

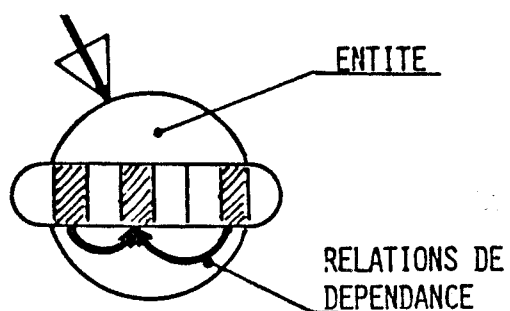


FIGURE II - 5.7 : RELATION DE DEPENDANCE

Pour chaque propriété il faut donc savoir si elle est liée dans une relation de DEPENDANCE et de quelle façon (source ou cible, relations symétrique ou assymétrique, formule de dépendance calculable, inhibante ou spéciale).

Les propriétés des entités dans la structure hiérarchique peuvent aussi être en relation (ex: la surface de l'appartement est la somme des surfaces des pièces qui le composent). On a donc une relation structurelle entre propriétés, que nous nommons CORRESPONDANCE des propriétés. Ces attributs sont aussi des contraintes d'intégrité, et sont donc invariants pendant la conception (Fig. II - 5.8).

L'attribut de correspondance peut être:

- hérité
- synthétisé
- spécial

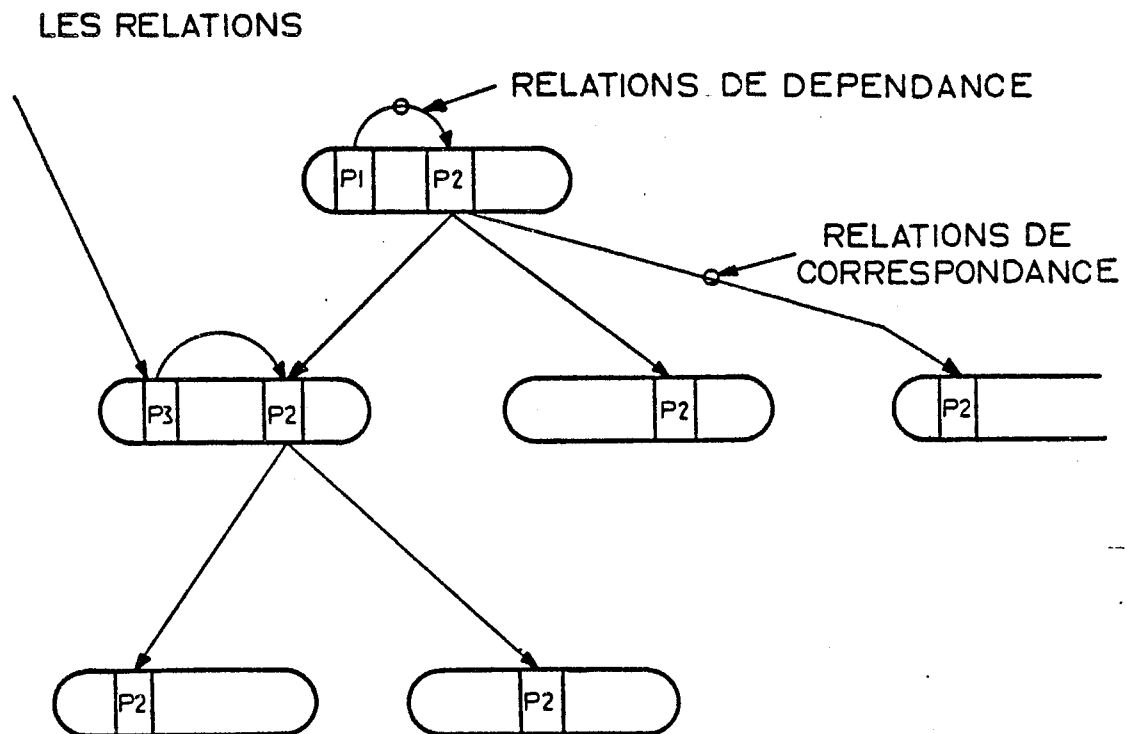


FIGURE II -5.8 : RELATION DE CORRESPONDANCE

La description du déroulement du processus de conception fournit la règle d'utilisation, dans une étape donnée du

processus, (accède, crée, modifie) des informations décrivant l'objet. On spécifie donc les différents filtres d'accès aux données, par rapport à la phase de conception (contrainte d'utilisation).

Cette description indique aussi comment cette vision se modifie lors d'un changement d'étape. Les TRANSFORMATIONS qu'il faut effectuer concernent aussi bien des propriétés (sémantiques) que la structure.

De plus on associe à l'objet des droits d'accès et d'utilisation, selon les responsabilités du concepteur dans le projet donné.

5.3.4 Opérations

Les opérations sur la description de l'objet doivent permettre la manipulation de cette description: création d'entités, structuration de l'objet, définition et mise à jour de propriétés et de relations.

Les opérations sur la description du processus doivent permettre à l'utilisateur d'effectuer son projet, donc de parcourir le processus de conception de la façon qui lui semble utile (avec des retours en arrière, ...).

A partir du moment où on veut assurer une cohérence sémantique, il n'est plus possible d'autoriser des opérations quelconques, il est indispensable de fournir des opérations sémantiques garantissant par elles-mêmes la continuité de la cohérence.

Les contraintes d'évolutivité et d'extensibilité rendent ce maintien de cohérence plus difficile. C'est pourquoi il faut raisonner séparément sur la structure et sur la sémantique. Il est possible de définir un ensemble d'opérations structurelles:

- ajouter une entité dans la structure,
- supprimer une entité au niveau d'une feuille de la hiérarchie ou au niveau intermédiaire,
- créer ou supprimer les liens fonctionnels,
- contracter deux ou plusieurs entités,
- éclater une entité en deux ou plusieurs,
- etc...

Tout ce qui porte uniquement sur la structure peut être défini une fois pour toutes. La partie structurelle des entités dans la structure peut donc être manipulée quel que soit le contexte.

La transformation structurelle de la partie de l'entité contenant des caractéristiques sémantiques dépend évidemment de cette sémantique et varie donc selon la nature des informations contenues.

Pour pouvoir effectuer les opérations structurelles sur cette partie sémantique, il faut définir le comportement structurel de ces caractéristiques.

Cela veut dire que lors de la définition de types (ou de modèles) de propriétés, on définit non seulement les types et les types de valeurs mais aussi l'ensemble d'opérations que l'on peut effectuer sur la propriété. De cette façon nous retrouvons la définition complète de type abstrait (LiZ.75).

L'extensibilité, l'évolutivité et la cohérence sont ainsi proprement prises en compte et gérées.

5.3.5 Modèle conceptuel

Récapitulons maintenant comment nous pouvons, avec le modèle proposé, satisfaire le contexte CAO dans lequel il est important de permettre l'évolution de l'ensemble des informations (extension des données manipulées, du traitement, du processus de conception), tout en gardant la maîtrise de l'intégrité du système.

Pour cela au lieu de partir de la structure globale, nous avons identifié l'ensemble des notions de base et des règles d'organisation.

Nous avons associé à chaque élément de base, l'ensemble d'opérations que l'on peut lui faire subir. De cette façon on assure la cohérence locale de chaque élément. La cohérence globale est exprimée à l'aide des relations d'intégrité (Correspondance, Dépendance et Transformation).

Le choix d'une structure interprétative permet de manipuler des informations incomplètes et de prendre en compte au fur et à mesure des compléments d'information.

Ces informations peuvent être construites soit à partir des types d'informations prédéfinis dans la phase de création du modèle conceptuel concret, soit directement de toutes pièces à l'utilisation.

ÉLÉMENTS DE BASE

ENTITÉS

- RELATIONS - STRUCTURELLES (HIÉRARCHIQUES)
 - FONCTIONNELLES (HORIZONTALES)

- PROPRIÉTÉS - PROPRES (DES ENTITÉS)
 - DE RELATION (DES RELATIONS)

- STRUCTURE - MULTI-HIÉRARCHIE DES ENTITÉS

- COHÉRENCE - DÉPENDANCE
 - CORRESPONDANCE
 - TRANSFORMATION
 - ACCESSIBILITÉ CONCEPTUELLE
 - ACCESSIBILITÉ ORGANISATIONNELLE

- OPÉRATIONS SÉMANTIQUEMENT COHÉRENTES :
 - SUR LA STRUCTURE
 - SUR LES PROPRIÉTÉS

EXTENSIBILITÉ ET ÉVOLUTIVITÉ

TYPE DE PROPRIÉTÉ

TYPE DE RELATION

TYPE D'ENTITÉ

TYPES DE BASE EXTENSIONS PAR ADJONCTION

TYPES CONSTRUITS EXTENSIONS PAR COMPOSITION DE TYPES
 (DE BASE OU CONSTRUITS)ÉLÉMENT GÉNÉRIQUE ÉLÉMENT TYPE NON VALUÉ ÉLÉMENT TYPE PARTIELLEMENT VALUÉ
 OCCURRENCE (COMPLÈTEMENT VALUÉE)

FIGURE II - 5.9 : ELEMENTS FONDAMENTAUX DU MODELE CONCEPTUEL DES DONNEES

5.4 LANGAGE CONCEPTUEL

Le langage conceptuel des données a pour rôle d'exprimer le modèle conceptuel concret d'une application. Il s'agit d'une particularisation du modèle conceptuel abstrait.

Ce langage s'appuie sur les six énoncés suivants:

- définition des propriétés
- définition des objets (objet à concevoir, objet conçu)
- dépendance des propriétés
- définition des correspondances
- définition des transformations
- transformations de l'objet à concevoir

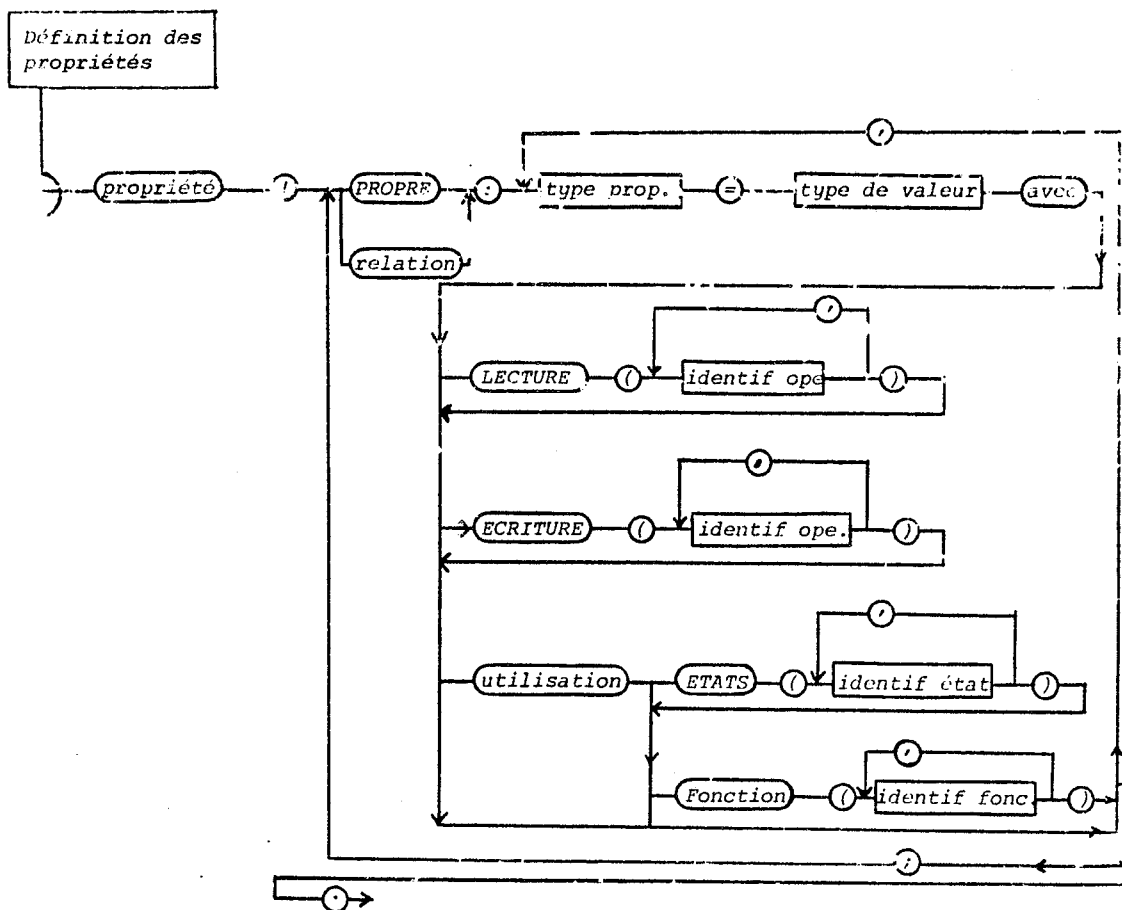


FIGURE II - 5.10.a : LANGAGE CONCEPTUEL DES DONNEES - ENONCE DEFINITION DES PROPRIETES

L'énoncé DEFINITION DES PROPRIETES permet de spécifier des propriétés propres et de relation. En fait, il s'agit de spécifier les types de propriétés. Lors de l'exploitation, l'utilisateur créera des occurrences de propriétés de ce type.

Pour la spécification, nous avons choisi l'approche par type abstrait, nous donnons les opérations sur la propriété, en distinguant si les opérations modifient ou pas les informations. En plus il est possible de spécifier l'accessibilité conceptuelle (par rapport au processus de conception) et l'accessibilité organisationnelle (par rapport à l'équipe de conception) de ce type de propriété.

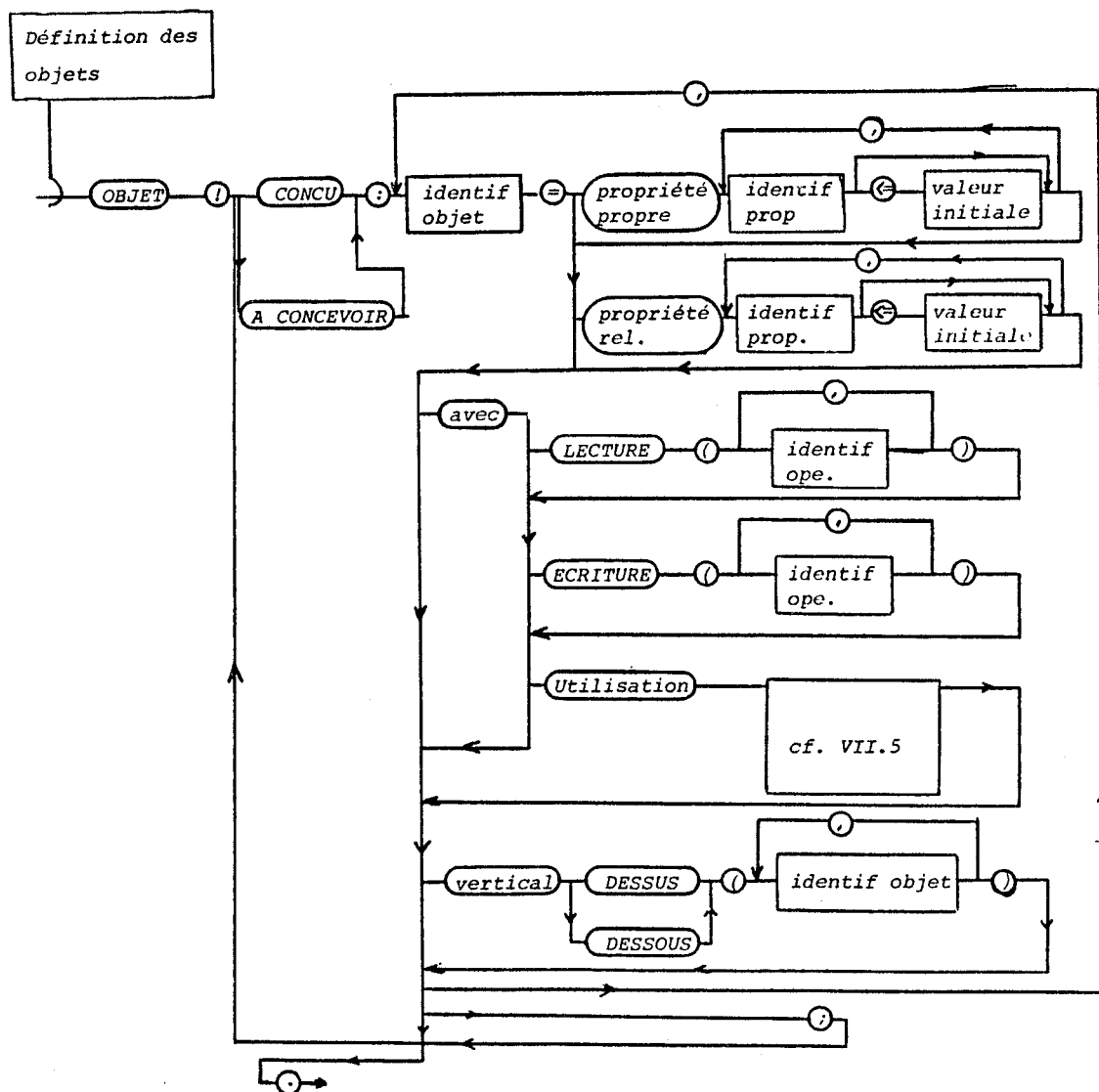


FIGURE II - 5.10.b : LANGAGE CONCEPTUEL DES DONNEES - ENONCE DEFINITION DES OBJETS (objet à concevoir, objet conçu)

L'énoncé DEFINITION DES OBJETS (objet à concevoir, objet conçu) permet de spécifier les types d'objets (objets génériques) pouvant être manipulés par le système. Il s'agit aussi bien des objets à concevoir que des objets conçus. On doit spécifier les propriétés propres et de relation qui caractérisent l'objet en leur fournissant éventuellement les valeurs initiales, puis les opérations que l'on autorise (dans quels états de conception et par qui) et finalement le comportement de l'objet dans la structure verticale.

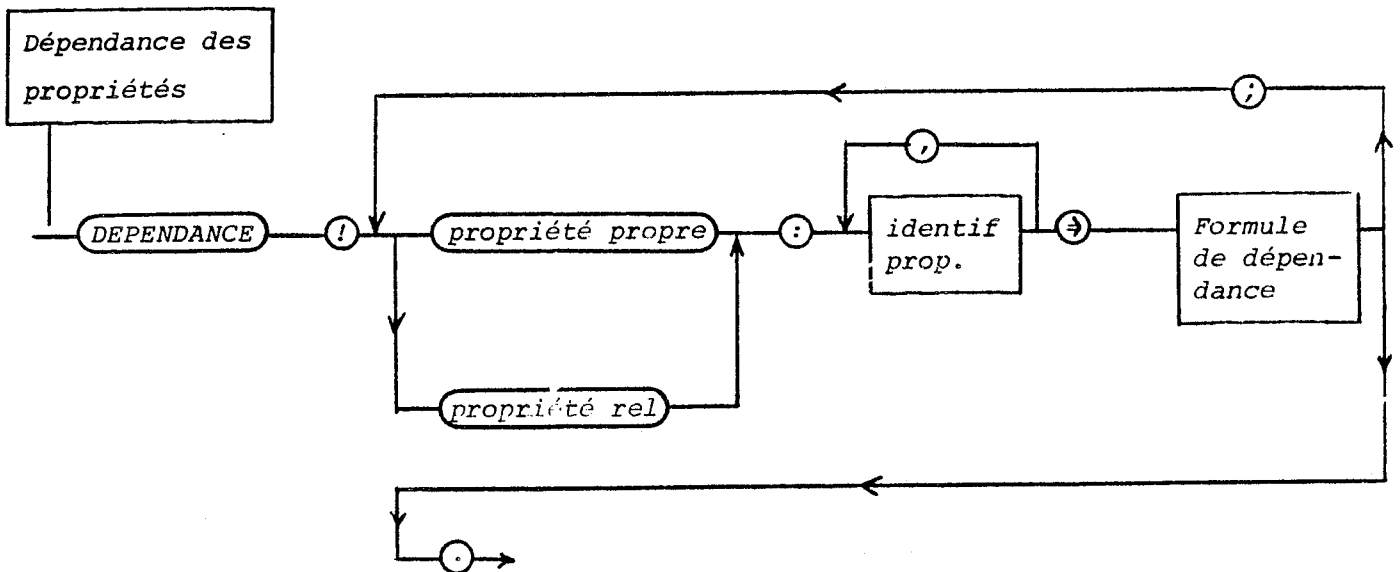


FIGURE II - 5.10.c : LANGAGE CONCEPTUEL DES DONNEES - ENONCE DEPENDANCE DES PROPRIETES

L'énoncé DEPENDANCE DES PROPRIETES permet d'exprimer les dépendances entre propriétés du même objet. La formule de dépendance est en réalité un programme qui, selon le cas: vérifie la satisfaction de la formule et appelle un diagnostic si elle

n'est pas vérifiée, ou bien calcule la valeur de la propriété manquante s'il le peut. Cette formule peut être une expression complexe utilisant différentes opérations des propriétés et délivrant un résultat logique.

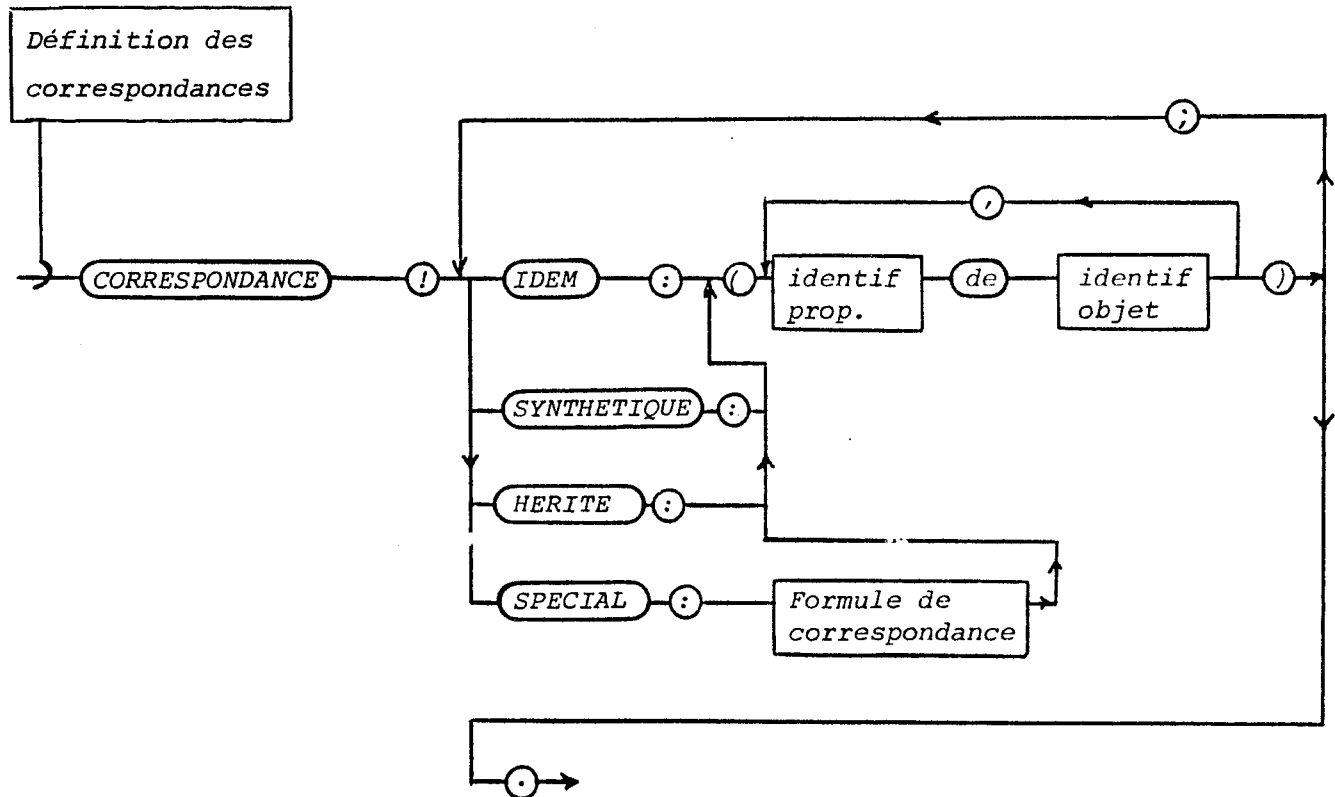


FIGURE II - 5.10.d : LANGAGE CONCEPTUEL DES DONNEES - ENONCE DEFINITION DES CORRESPONDANCES

L'énoncé DEFINITION DES CORRESPONDANCES permet de spécifier les liens verticaux entre propriétés de différents objets et fournit donc un moyen d'assurer l'évolution de l'objet dans la hiérarchie abstraite. Les principales correspondances sont : l'IDENTITE entre propriétés des objets sur différents niveaux; la SYNTHESE qui assure la remontée des propriétés et l'HERITAGE pour la descente. La correspondance de type SPECIAL permet la prise en compte d'autres types de correspondances par exemple la synthétisation cumulative (pour les surfaces par exemple).

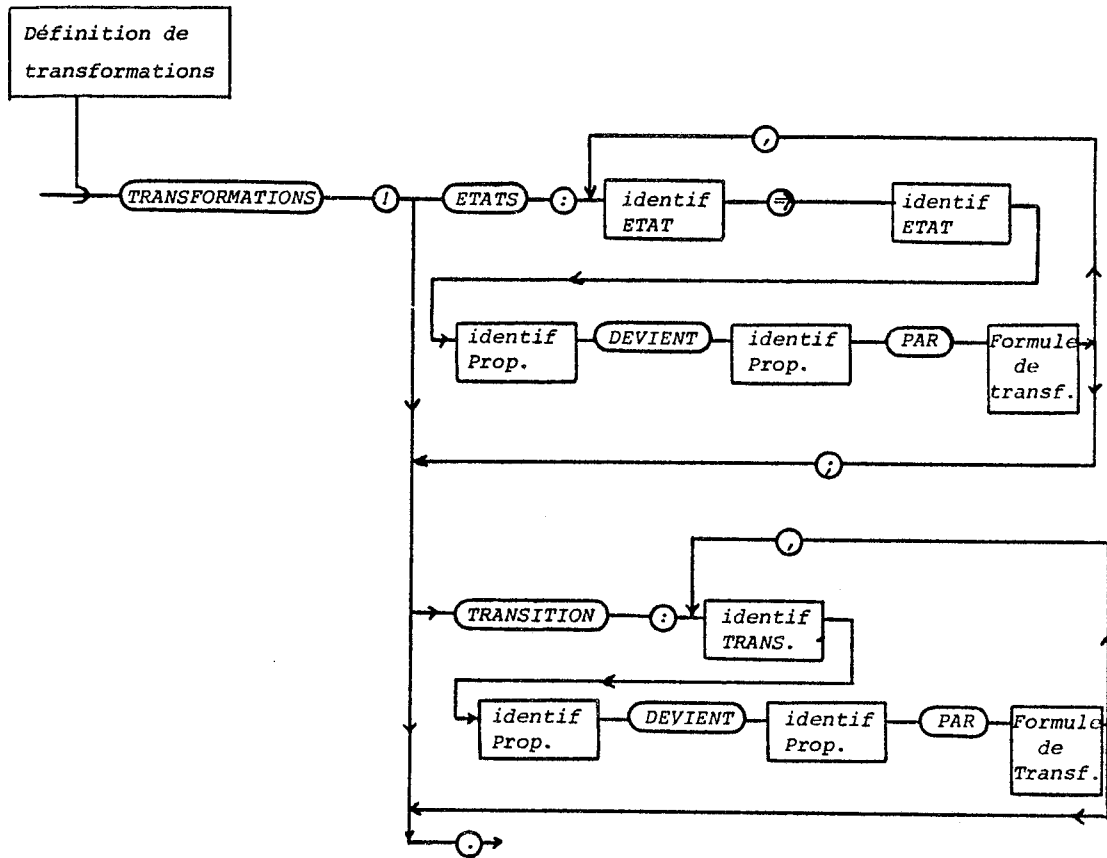


FIGURE II - 5.10.e : LANGAGE CONCEPTUEL DES DONNEES - ENONCE DEFINITION DES TRANSFORMATIONS

L'énoncé DEFINITION DES TRANSFORMATIONS permet de spécifier le changement des propriétés pendant les transitions dans l'espace de conception. La formule de transformation indique comment on passe de la propriété existante à la propriété qui suit le mot-clé DEVIENT.

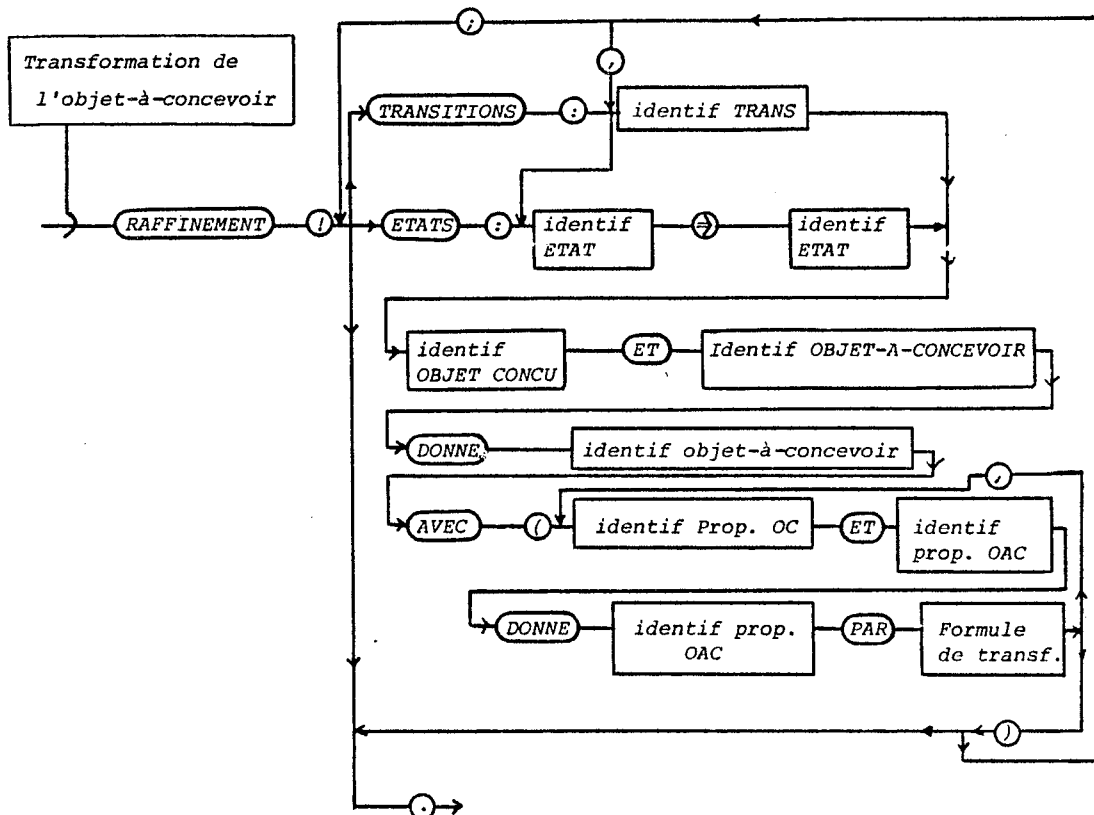


FIGURE II - 5.10.f : LANGAGE CONCEPTUEL DES DONNEES - ENONCE TRANSFORMATION DE L'OBJET A CONCEVOIR (raffinement de la définition)

L'énoncé TRANSFORMATION DE L'OBJET A CONCEVOIR permet de spécifier comment s'affine la définition de l'objet à concevoir pendant le processus de conception. On spécifie la transition et les objets conçus et à concevoir, qui interviennent dans la transformation, puis on fournit les formules de transformation de chaque propriété.

5.5 ESSAIS DE MISE EN OEUVRE - EXPERIMENTATION

Nous avons travaillé sur différents essais de mise en oeuvre dans des contextes variés, principalement l'utilisation de fichiers séquentiels, de fichiers à accès direct et d'un SGBD.

Le choix entre ces différents types de gestion des données dépend d'une part de la complexité du problème à traiter et d'autre part du contexte (matériel et logiciel) dans lequel il faut le traiter. Ainsi pour des problèmes relativement simples, on peut faire appel aux fichiers séquentiels. La simplicité d'un problème se définit surtout par la simplicité de sa structure. Le problème est donc simple s'il s'appuie sur une mono-hiérarchie avec une dynamique réduite.

En ce qui concerne le contexte d'utilisation, il est évident que plus l'ordinateur est petit (mini, micro) moins est varié le choix de solutions et plus strictes sont les contraintes techniques.

Nous allons donc présenter les différentes solutions que nous avons étudiées.

5.5.1 Organisation du rangement des données

Le premier aspect qu'il faut considérer est celui de l'organisation du rangement des données sur le support externe.

Deux solutions extrêmes sont envisageables:

- rangement par objet: on stocke ensemble toutes les caractéristiques d'un objet,
- rangement par type de données: on stocke séparément d'une part la structure globale du projet et d'autre part toutes ses caractéristiques, chacune de façon autonome et groupée pour tous les objets.

Ces deux organisations ont évidemment des propriétés différentes. Si la première est beaucoup plus rapide d'accès (on accède globalement et séquentiellement à toutes les informations d'un objet), elle n'est pas assez évolutive, car la description d'objet doit être figée. La deuxième par contre nécessite des temps d'accès plus grands (car il faut chercher séparément différentes caractéristiques stockées indépendamment), mais elle offre des possibilités d'extension plus grandes. On peut, en effet, ajouter facilement de nouveaux types de propriétés et les prendre en compte.

Nous avons naturellement opté pour cette deuxième solution et cela aussi parceque: un accès se fait la plupart du temps selon un point de vue, c'est à dire, une vision particulière du projet. On doit accéder à tous les objets et pour ces objets à un sous-ensemble précis de caractéristiques. C'est ce type d'accès, que nous voulons privilégier.

Dans ce cas l'organisation physique que l'on peut mettre en oeuvre aussi bien à l'aide des fichiers séquentiels, que des fichiers à accès direct, ainsi que d'un SGBD, est la suivante (Fig. II - 5.11):

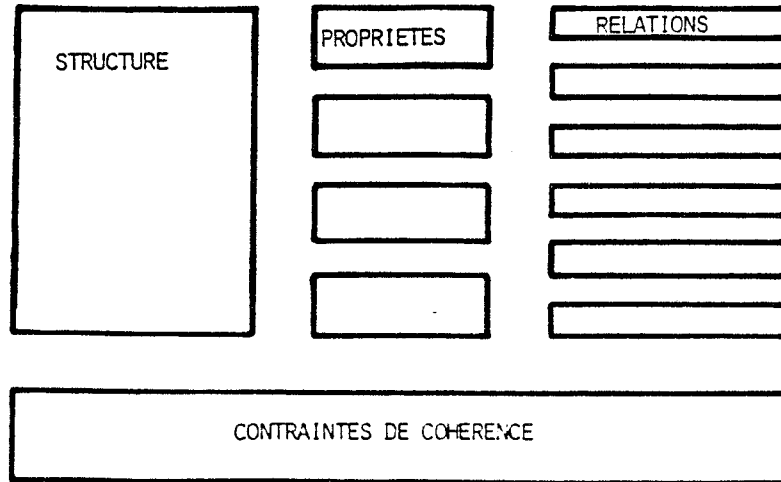


FIGURE II - 5.11: L'ORGANISATION PHYSIQUE DES DONNEES

5.5.2 Expérimentation à l'aide des fichiers séquentiels

Quand on utilise des fichiers séquentiels on associe un fichier à chaque pavé. Ainsi:

- F.DEF contient N: nombre d'objets, NOMS(N): les noms associés aux objets, et la structure hiérarchique. Cette structure est représentée par un arbre dans lequel à chaque noeud on associe les informations suivantes:
 - + numéro du noeud
 - + Degré du noeud (nombre de branches sortantes du noeud)
 - + NF(d) - numéro du fils
 - + PF(d) - pointeur sur le fils
 - + PP - pointeur sur le père
- F.Rxx contient N et la matrice $N \times N$ représentant la relation
- F.Pxx contient N et le vecteur Pxx(N) représentant les valeurs de la propriété.

L'accès aux informations s'effectue de la façon suivante: on choisit l'objet auquel on veut accéder et le numéro d'objet fournit le rang de toute propriété et de toute relation, donc si NO est le numéro d'objet, Pxx(NO) est une propriété et Rxx(NO,*) est la relation.

Cette organisation permet d'ajouter dynamiquement de nouvelles propriétés et de nouvelles relations. Elle permet aussi

d'augmenter dynamiquement le nombre d'objets manipulés. Par contre le problème majeur se situe dans la manipulation structurelle de ces informations et le maintien de la cohérence, car il faut travailler dans un ensemble de fichiers.

Le regroupement de toutes les informations dans le même fichier, auquel on doit accéder de façon aléatoire (accès direct), offre des avantages supplémentaires en ce qui concerne la manipulation de la structure et le maintien de cohérence, mais rend plus difficile la gestion d'espace mémoire pour les extensions dynamiques.

Finalement un SGBD doit marier les avantages de ces deux solutions, car il s'occupe de façon transparente pour l'utilisateur de la gestion de place.

5.5.3 Expérimentation d'un système de gestion de bases de données

Nous travaillons actuellement sur une réalisation expérimentale en utilisant le système de gestion de base de données SOMINE (Gai.76, Mar.76, Say.76). C'est un système de type réseau, dont la conception s'appuie sur le système SOCRATE. Néanmoins, il s'agit d'un système plus pauvre, possédant un langage de citation rudimentaire, mais dont l'avantage essentiel est d'être programmé en FORTRAN, donc portable, facilement modifiable et pouvant fonctionner sur une petite machine.

Dans ce cas une entité générique la plus générale est définie de la façon suivante (Fig. II - 5.12):

```

ENTITE 50 STRUCTUR DEBUT
      TYP+ENT      MOT 4
      T+DESSUS    MOT 4 DIM 5
      DESSUS      NUM E DIM 5
      T+ACOTE     MOT 4 DIM 5
      ACOTE       NUM E DIM 5
      T+DESSOU   MOT 4 DIM 5
      DESSOU     NUM E DIM 5
      NOM+PTE    MOT 4 DIM 10
      PTE        NUM E DIM 10
      VAL        NUM E DIM 10
      REL        NUM E DIM 10
      FIN

```

FIGURE II -5.12 : ENTITE GENERIQUE EXPRIMEE EN SOMINE

Il est possible de définir et d'ajouter dynamiquement des nouveaux types de propriétés et de relations (Fig. II - 5.13).

```

ENTITE 50 PTE+NUM DEBUT
  NOM MOT 4
  VAL NUM E DIM 50
  FIN

ENTITE 10 PTE+NOM DEBUT
  NOM MOT 4
  ENTITE 50 PTE DEBUT
    NOM MOT 4
  FIN FIN

ENTITE 5 PTE+COORD DEBUT
  VAL NUM E DIM 100
  FIN

ENTITE 10 REL+NUM DEBUT
  NOM MOT 4
  REL NUM E DIM 2500
  FIN

ENTITE 10 PTE+FB DEBUT
  PTE NUM E DIM 1600
  FIN

ENTITE 5 OBJET2D DEBUT
  ENTITE 50 F2D DEBUT
    PTS NUM E DIM 40
    FAS NUM E DIM 20
  FIN FIN

ENTITE 5 OBJET3D DEBUT
  ENTITE 50 F3D DEBUT
    PTS NUM E DIM 60
    FAS NUM E DIM 300
  FIN FIN

ENTITE 5 TRANSFO DEBUT
  ENTITE 50 REF3D DEBUT
    IDN NUM E
    MAT NUM R DIM 16
  FIN FIN

```

FIGURE II - 5.13 : DEFINITION DE TYPES DE PROPRIETES ET DE RELATIONS

L'organisation, que nous venons de présenter est purement interprétative. Elle est utile dans une exploitation prototype. Lors d'une utilisation opérationnelle, il n'est pas souhaitable de supporter une aussi grande généralité évidemment pénalisante. Il est donc utile de figer la sémantique globale du domaine considéré (correspondant au modèle conceptuel concret) et de laisser des possibilités d'évolution et d'extension au delà de ce modèle conceptuel concret.

Dans l'organisation classique d'un SGBD réseau, on travaille sur deux parties différentes:

- la structure: contient la définition de la structure des objet à manipuler
- les données: contient des occurrences valuées des objets.

Pour notre réalisation nous nous appuyons sur cette organisation que nous garnissons de la façon suivante (Fig. II - 5.14):

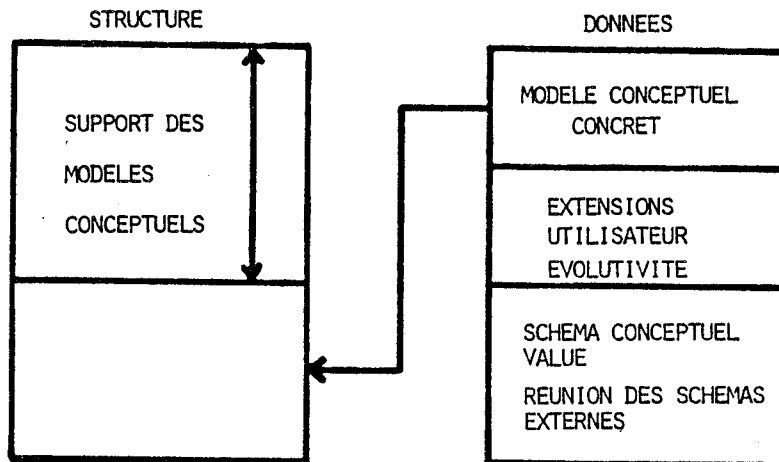
- dans l'organisation prototype:
 - + dans la partie structure: on stocke une fois pour toutes le modèle conceptuel abstrait,
 - + dans la partie données: on stocke d'abord la spécialisation qui définit un modèle conceptuel

concret, puis des extensions dynamiques et enfin les données proprement dites du projet précis (schéma conceptuel valué, qui est une réunion des schémas externes),

- dans l'organisation finale:

+ il est possible de transférer (et donc figer) une partie de la définition du modèle concret dans la partie structure.

De cette façon on remplace l'interprétation de la partie figée par la déclaration et on accélère le traitement.



POSSIBILITE PLUS TARD DE TRANSFERER LE MODELE CONCEPTUEL CONCRET DANS LA STRUCTURE DE LA BASE (ET LE FIGER)

FIGURE II - 5.14 : REPARTITION DES INFORMATIONS DANS UN SGBD DE TYPE RESEAU

Le schéma sur la Fig. II - 5.15 montre de façon globale la mise en oeuvre de la modélisation SIGMA-CAO sur un SGBD de type réseau.

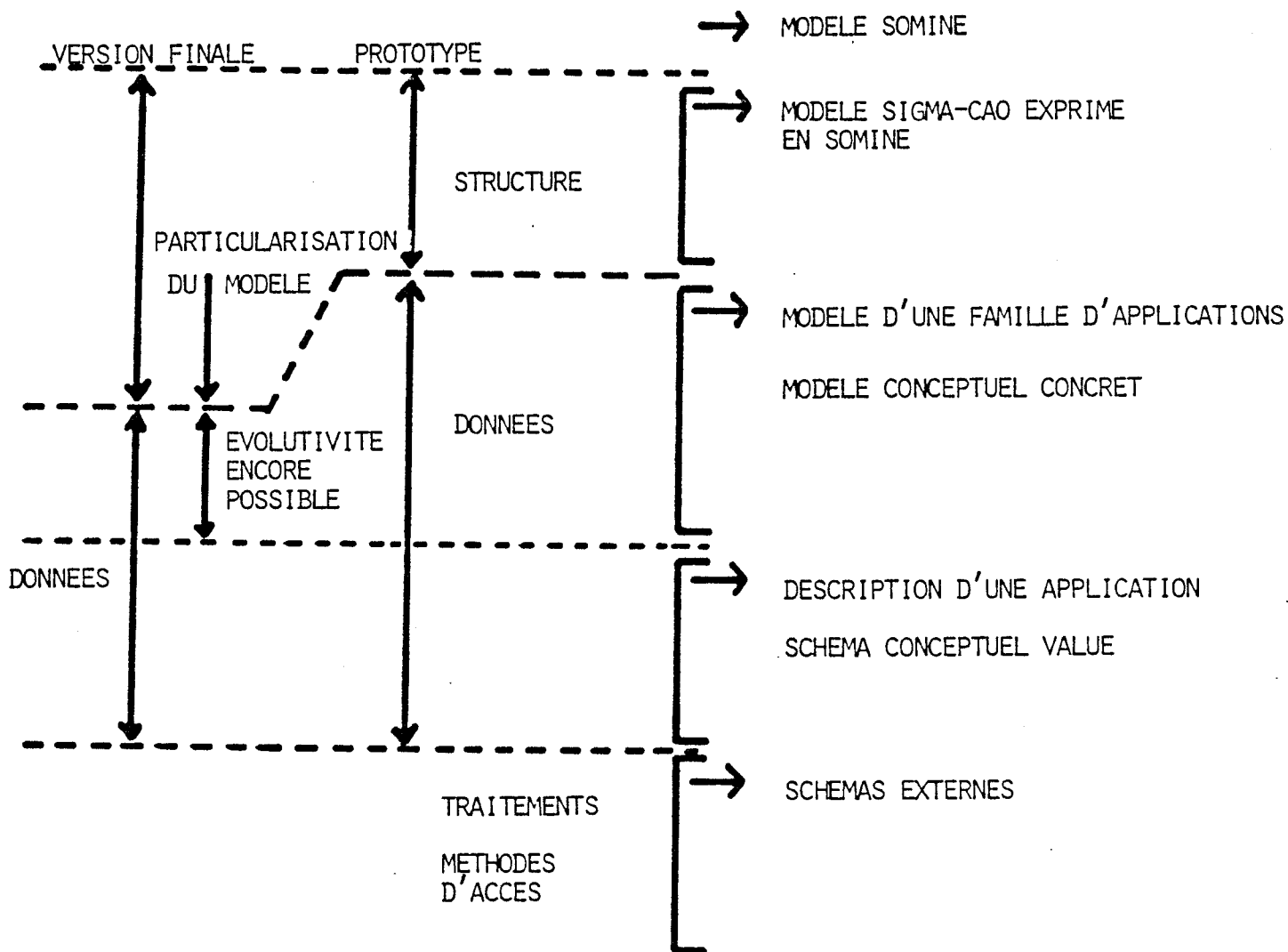


FIGURE II - 5.15 : MISE EN OEUVRE DE LA MODELISATION SUR UN SGBD DE TYPE RESEAU

Nous nous intéressons particulièrement à la mise en place d'un prototype de notre approche afin de la valider et de répondre, éléments techniques à l'appui, au problème du type de base de données nécessaire à la CAO.

A ce jour, nous avons défini et mis en oeuvre le langage de spécifications (COLA), ainsi que les premiers traitements de ces spécifications. Nous avons défini une méthode de construction de schémas conceptuels (modèle logique), à partir du modèle conceptuel et de sa projection sur une organisation standard (non optimisée) de structuration de SOMINE, ce qui correspond au passage spécification - prototype; nous avons également commencé à adapter le système SOMINE.

5.6 MODES D'ACCES

Nous distinguons deux types de données: données globales (catalogues) utilisées en lecture seulement et données du projet pouvant être modifiées par les traitements. Voyons maintenant les modes d'accès à ces données fournis par un système CAO, leurs réalisations et leurs utilisations (Fig. II - 5.16).

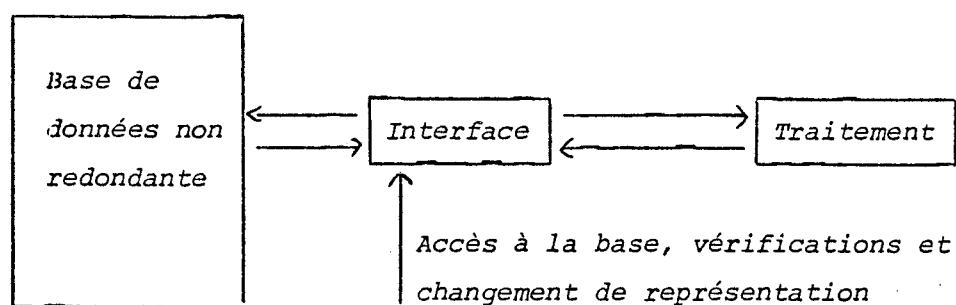


FIGURE II - 5.16 : ACCES AUX DONNEES

5.6.1 contexte

Nous voulons surtout respecter les trois contraintes suivantes:

- le système CAO doit pouvoir évoluer dans des contextes différents de matériels et de logiciels.
- l'indépendance des traitements par rapport à la structure globale des données est primordiale. En effet, pour des questions d'évolutivité, d'extensibilité, ainsi que de portabilité, les traitements doivent se contenter d'une vision partielle des données, limitée aux données directement utilisables par ces traitements et réduite à l'aspect logique. L'organisation physique doit toujours être cachée.
- pour des questions de rapidité il est utile de transporter les données dans l'espace de travail du traitement et de les retourner après usage. Il est aussi souhaitable de minimiser la transformation de ces données.

5.6.2 Solution proposée

Pour chaque traitement il est indispensable de faire apparaître la partie concernée par les entrées/sorties et celle qui assure les opérations proprement dites. La partie interne travaille sur une structure de données propre au traitement. La partie entrées/sorties doit remplir cette structure à partir de la BDP et éventuellement remettre les données nouvelles ou modifiées dans la BDP. Le schéma suivant (Fig. II - 5.17) représente cette organisation.

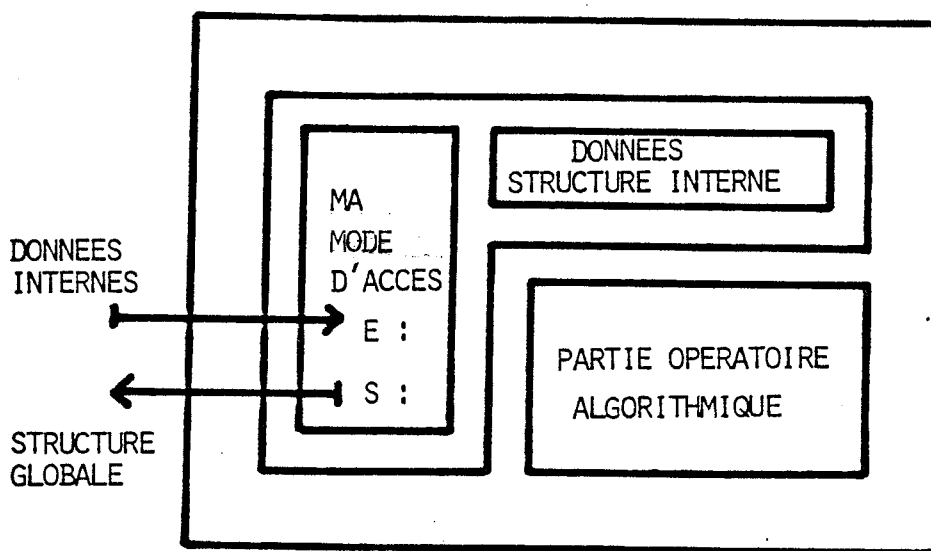


FIGURE II - 5.17 : STRUCTURE D'UN TRAITEMENT

Différentes solutions ont été envisagées et nous les présentons rapidement. Le problème principal est d'assurer l'indépendance du mode d'accès vis à vis de la structure externe ET de la structure interne.

A/ Mode d'accès et structure interne.

Etudions d'abord le lien du mode d'accès avec la structure interne du traitement. On peut évidemment éviter ce problème en imposant une représentation unique (et donc standard) de cette structure. Dans ce cas le mode d'accès peut ranger les informations directement dans la structure interne, qui est plus globale que la portée du traitement (Fig. II - 5.18).

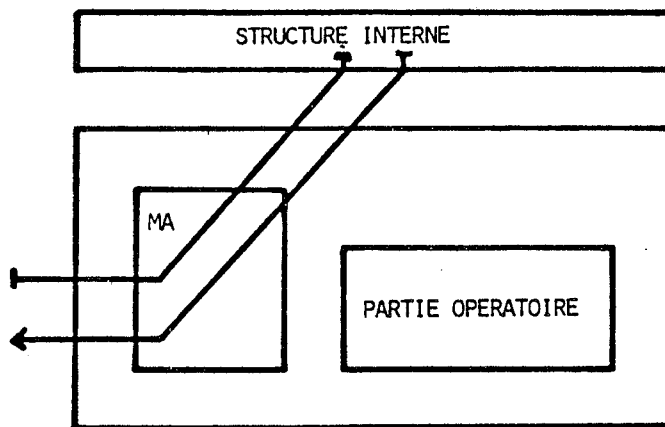


FIGURE II - 5.18 : STRUCTURE INTERNE STANDARD

Si l'on veut ou doit avoir pour des raisons d'efficacité, une structure interne spécifique il est nécessaire d'interfacer le mode d'accès, c'est à dire prendre les informations du mode d'accès et les traduire dans la structure interne spécifique et vice versa.

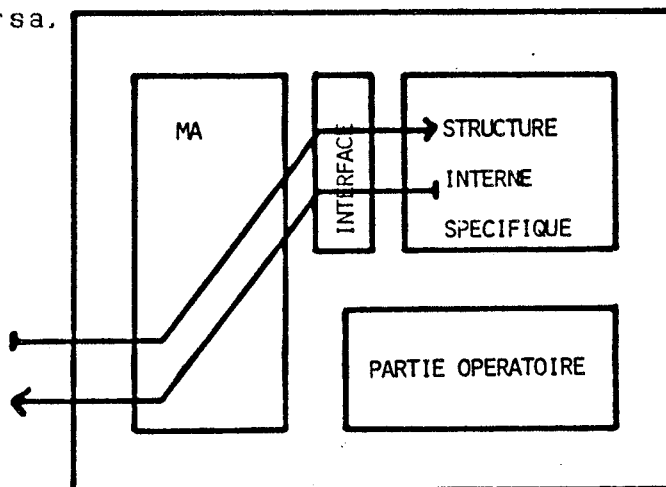


FIGURE II - 5.19 : STRUCTURE INTERNE SPECIFIQUE

Organisation du mode d'accès.

Pour le mode d'accès nous pouvons choisir une des deux solutions suivantes:

- la première est un mode d'accès STANDARD, c'est à dire étant capable d'accéder à toutes les données externes. Dans ce cas il se situe évidemment en dehors d'un traitement spécifique et on lui fournit la liste des données à manipuler (Fig. II - 5.20).

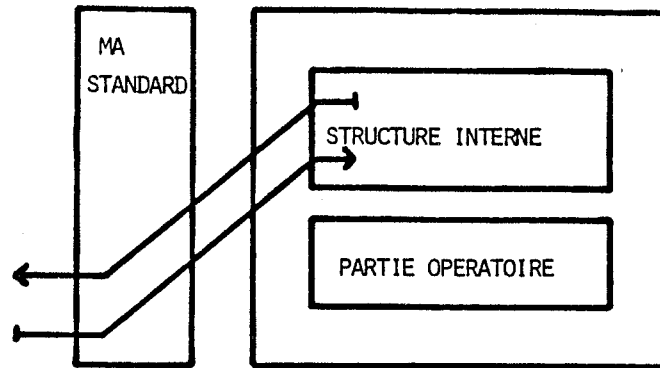


FIGURE II - 5.20 : MODE D'ACCES STANDARD

- la deuxième est un mode d'accès SPECIFIQUE, faisant partie du traitement et pouvant être produit automatiquement à partir du mode d'accès standard et de la liste des accès souhaités (Fig. II - 5.21).

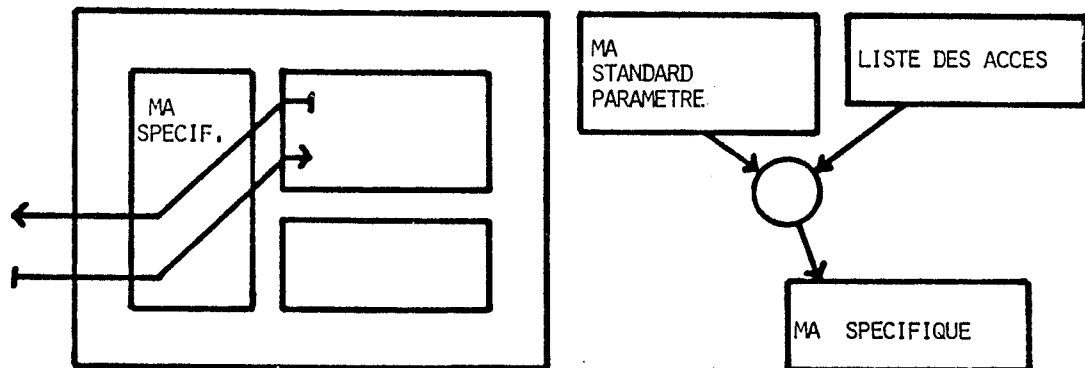


FIGURE II - 5.21 : MODE D'ACCES SPECIFIQUE ET SA PRODUCTION A PARTIR DU MODE D'ACCES STANDARD

L'intérêt du mode d'accès spécifique est d'une part la réduction de sa taille (utile pour les petites machines), et d'autre part la possibilité de prendre en compte, lors de sa génération, la structure interne spécifique et produire ainsi directement un mode d'accès spécifique d'une structure interne. Cela permet de supprimer l'interface de la Fig. II - 5.14. Dans ce cas le schéma de production est le suivant: Fig. II - 5.22.

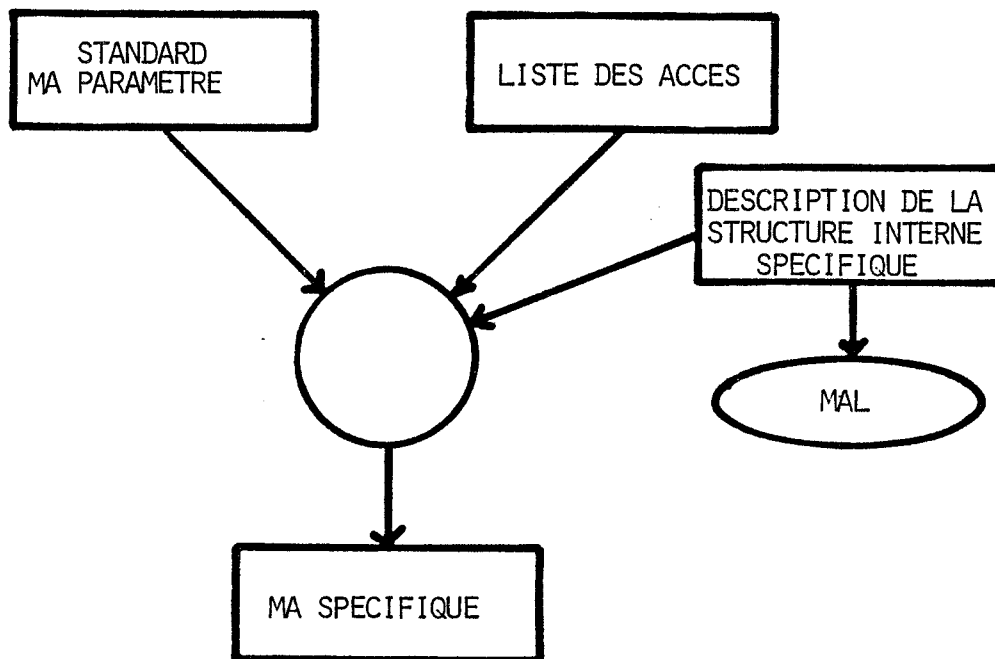


FIGURE II - 5.22 : PRODUCTION D'UN MODE D'ACCES SPECIFIQUE POUR UNE STRUCTURE INTERNE SPECIFIQUE

B/ Mode d'accès et la structure externe

Le rôle du mode d'accès est d'assurer aussi l'accès à la structure externe. Cette structure externe peut se trouver sur des supports différents exploités de façon différentes (accès direct, accès séquentiel, base de données). Pour pouvoir faire face à toutes ces éventualités, le mode d'accès est structuré en trois niveaux (Fig. II - 5.23):

- le MODE D'ACCES STANDARD constitue le moniteur d'accès aux données. Il ne traite pas directement des données mais dirige et aiguille uniquement les opérations. Il effectue donc une gestion globale et assure la protection des données,
- le MODE D'ACCES LOGIQUE travaille sur un type logique de données, initialise, traite les transferts et effectue les passages vers et depuis la structure spécifique,
- le MODE D'ACCES PHYSIQUE effectue les échanges physiques d'un bloc précis d'information; ce mode d'accès physique peut être séquentiel, direct ou utiliser un SGBD.

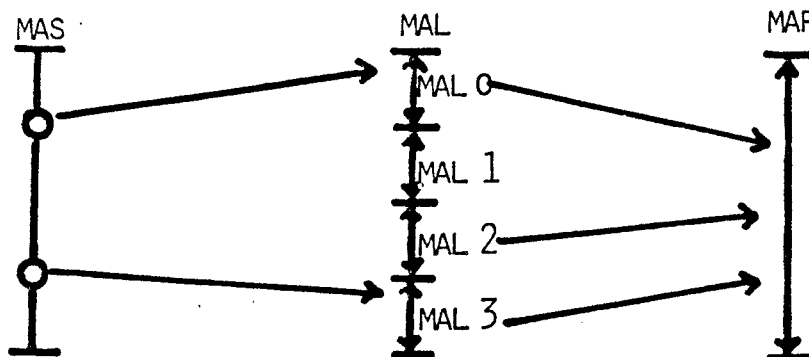


FIGURE II - 5.23 : ORGANISATION DU MODE D'ACCES

Nous pouvons montrer ici un exemple du fonctionnement de ce découpage du mode d'accès, utilisant les fichiers séquentiels.

Les informations se trouvent sur un support externe constitué par un ensemble de fichiers séquentiels homogènes (contenant chacun un type de données). Pendant l'activation du traitement on fournit la liste des données sur lesquelles le traitement va travailler. On constitue alors un fichier standard d'entrée par composition de fichiers de base. Ce fichier standard d'entrée est exploité par le mode d'accès standard (Fig. II - 5.24), qui transmet les informations dans la structure interne. A la fin de l'exécution, on donne au mode d'accès standard les informations qui doivent être transmises à la structure externe. Le mode d'accès standard constitue alors un fichier standard de sortie qui est éclaté pour retrouver la forme des fichiers homogènes (Fig. II - 5.25).

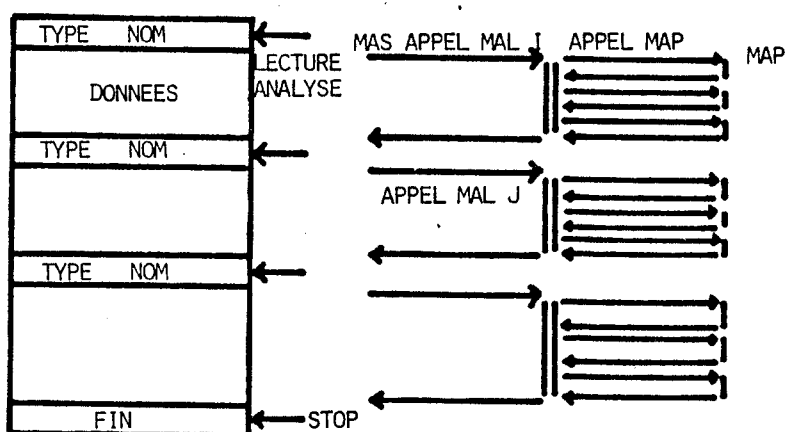


FIGURE II - 5.24 : EXPLOITATION DU FICHIER STANDARD D'ENTREE PAR LE MODE D'ACCES STANDARD

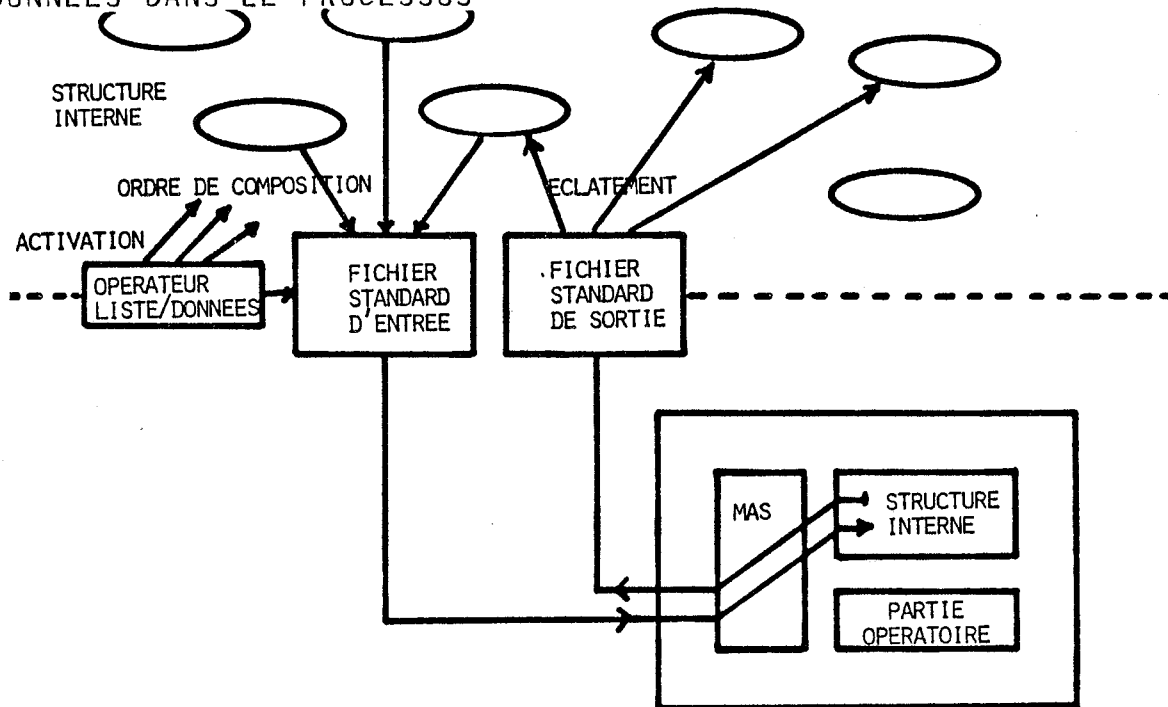


FIGURE II - 5.25 : EXEMPLE DU FONCTIONNEMENT D'UN MODE D'ACCES STANDARD UTILISANT LES FICHIERS SEQUENTIELS

Le même fonctionnement peut être obtenu en indiquant dans le mode d'accès le fichier de base auquel il faut accéder. De cette façon on peut obtenir une première illusion d'accès direct.

C/ Methode d'accès standard et l'accès direct

La notion d'accès direct peut être comprise de deux façons différentes:

- accès à l'information souhaitée sans devoir accéder aux informations qui la précèdent,
- accès de façon dynamique à cette information.

Dans le premier cas le fonctionnement du mode d'accès n'est pas modifié, c'est à dire que l'on fait l'acquisition des données à l'initialisation de la tâche et on rend des résultats à la fin de cette tâche. Le mode d'accès standard et le mode d'accès logique ne sont pas modifiés, le changement s'effectue uniquement au niveau du mode d'accès physique.

Dans le deuxième cas (accès dynamique) on ne fait plus l'acquisition globale mais de façon progressive à la demande. Dans ce cas le lien entre l'utilisation des informations est plus étroit car à chaque manipulation on fait appel à la fonction d'accès qui consulte si l'objet est présent, sinon elle le fait venir. Dans ce cas les modes d'accès standard, logique et physique sont les mêmes que dans le cas précédent mais l'appel au

mode d'accès standard se fait dynamiquement. L'algorithme de remplacement est un algorithme classique qui consiste à sortir l'objet le moins utilisé et si possible non modifié, sinon modifié. Il s'agit d'un mécanisme de pagination fonctionnelle car il manipule des objets.

D/ Mode d'accès standard et la base de données

Pour le fonctionnement avec un SGBD nous nous retrouvons dans le même contexte qu'un accès direct. Nous pouvons donc l'utiliser de la même façon (deux types d'utilisation). La différence se situe au niveau du mode d'accès physique qui est remplacé par un appel au SGBD qui se charge de résoudre tous les problèmes d'adressage.

5.7 BASE DE DONNEES TECHNIQUE

La base de données doit contenir l'ensemble des informations d'intérêt général représentant l'acquis technique de l'entreprise. Elle doit pouvoir contenir les données techniques sur les éléments utilisés, les bibliothèques de nomenclatures, etc... Il paraît aussi important d'y trouver les résultats, bons ou mauvais, des différents projets pour pouvoir s'en inspirer.

La structure de cette base doit être essentiellement relationnelle, car elle doit servir dans les recherches de réponses à des questions très variées.

Lors de la spécification des informations à mettre dans cette base, il est important d'indiquer quels seront les accès les plus couramment utilisés et de prévoir les conditions d'accès à ces informations, c'est-à-dire qui a le droit de les connaître, éventuellement de les modifier, et à quel moment du processus de conception.

CHAPITRE 6
DIALOGUES HOMME-MACHINE

RESUME: Le troisième aspect de l'invariant conceptuel concerne la communication homme-machine. Après une classification des dialogues, une méthode de mise en oeuvre de langages de commandes est proposée.

6. DIALOGUES HOMME-MACHINE

6.1 DEFINITION ET CLASSIFICATION

La communication homme - machine est une des composantes importantes de la Conception Assistée par Ordinateur. C'est compréhensible puisque le but de la CAO est d'associer l'homme et la machine dans la résolution d'un problème de conception.

Le premier aspect important des dialogues homme-machine concerne le degré d'association de l'homme et de la machine, qui peut être plus ou moins étroit selon la nature de la coopération. Pour certains problèmes ou tâches du processus de conception, les participations de l'homme ou de la machine sont bien distinctes, et le transfert d'informations de l'un vers l'autre nécessite rarement une grande durée ou un important volume d'informations. Dans d'autres problèmes, l'association homme - machine est plus étroite et a pour objectif une symbiose parfaite; au sein d'un contexte interactif la fréquence d'échanges est alors très élevée et le volume d'informations réduit.

Le second aspect important concerne la forme du dialogue. Il faut que le concepteur - utilisateur puisse échanger les informations sous une forme pour lui habituelle, souhaitable et agréable. Ceci pour deux raisons: d'une part il n'est pas forcément un spécialiste de l'informatique, et il faut donc lui cacher toute la "cuisine" informatique; d'autre part le bon déroulement du processus créatif ne supporte pas d'être dérangé par les perturbations dues au mode de dialogue, par la lenteur des échanges, etc... C'est pourquoi il est indispensable de bâtir le poste de travail correspondant à la nature du travail avec tous les périphériques nécessaires à un bon déroulement des travaux.

Le troisième aspect important concerne la mise en place des dialogues. Autrement dit comment concevoir des programmes interactifs de façon systématique et efficace.

Ces problèmes ont été évidemment abordés dans la réalisation de tout système CAO, c'est pourquoi les solutions apportées pour organiser les dialogues dans des systèmes CAO sont très nombreuses, plus ou moins originales, satisfaisantes et générales.

Pour échanger des volumes importants d'information on a d'abord utilisé des bordereaux où l'on définissait de façon précise les champs physiques sur le support d'entrée (essentiellement des cartes perforées). Cette solution très commode pour la machine (format de lecture figé) s'est avérée très insatisfaisante pour l'utilisateur car il devait tenir compte des cadrages. Quant à la relecture et la correction, elles étaient fastidieuses. C'est pourquoi on a cherché d'autres méthodes d'introduction des données. On a alors principalement essayé d'utiliser les langages de description et les langages de commande.

Le but du langage de description est de décrire des données. Par opposition au bordereau, l'utilisateur est privilégié par rapport à la machine. L'utilisateur n'a plus de champs physiques ou de cadrages à respecter, mais il dispose d'une syntaxe et d'une sémantique à l'aide desquelles il exprime sous une forme textuelle les données caractérisant le problème à traiter.

Cette technique de langage de description est surtout utilisée en électronique et en logique. Il s'agit de décrire à l'aide d'un langage artificiel, la structure et le fonctionnement d'un objet (réel, virtuel ou fonctionnel) sans ambiguïté et de façon compréhensible aussi bien par l'homme que par la machine.

Nous ne rentrerons pas ici dans les détails techniques car il s'agit d'un domaine très vaste et très complexe.

A titre d'exemple nous pouvons citer les travaux effectués dans notre laboratoire sur ce sujet: il s'agit de langage de description pour l'électronique IMAG (LeR.77), pour les systèmes logiques CASSANDRE (BGL.71, Mer.73) et pour les systèmes informatiques LASCAR et LASSO (Bor.75, BoG.79). D'autre part il existe un groupe de travail international qui étudie la possibilité d'homogénéiser ces différents langages, le projet porte le nom évocateur CONLAN pour "CONsensus LANguage" (BBD.80).

L'utilisation d'un langage de description se justifie pour des volumes d'information importants préparés puis soumis à la machine. On peut également résoudre ce problème en utilisant un langage de commande qui permet d'exprimer l'activation des commandes: actions plus ou moins complexes que l'utilisateur désire entreprendre. Pour fournir des données volumineuses il est possible de découper la saisie et de procéder par morceaux. Dans cette démarche, on s'approche d'une utilisation de la machine en mode interactif dans lequel le langage de commande constitue le moyen de dialogue.

Les réalisations de langages de commandes sont nombreuses et varient selon le contexte d'utilisation: les personnes concernées (utilisateurs avertis ou occasionnels), le travail à assurer (systématique ou aléatoire), et le matériel disponible

(console alpha-numérique, terminal graphique, poste de travail complexe, ...).

GERO dans (GeJ.77) distingue six méthodes différentes de dialogues homme-machine:

- 1/ langue naturelle
- 2/ langue naturelle limitée
- 3/ mnémonique
- 4/ réponses directes aux questions
- 5/ sélection de menu
- 6/ langage de programmation

Cette classification énumère effectivement les modes classiques de dialogue. Il semble que les deux premiers et le dernier ne soient pas très utilisables en CAO. La communication en langue naturelle nécessite encore un volume de programmes considérable et ne paraît pas pouvoir satisfaire les contraintes d'efficacité et de rapidité sur les configurations industrielles actuelles. Ceci est presque vrai pour la langue naturelle limitée même si ici, l'applicabilité future peut être assez proche.

L'utilisation de la langue naturelle ne se justifie pas toujours en CAO si la conception est une affaire de spécialistes. Il s'agit donc d'un contexte dans lequel, après un apprentissage initial, il est surtout important d'avoir un outil efficace et rentable. Cependant en s'élargissant à tous les domaines de la création, avec des concepteurs de moins en moins techniciens, la CAO peut progressivement requérir une langue naturelle limitée.

De même la communication à l'aide d'un langage de programmation n'est pas non plus satisfaisante car trop proche de l'aspect informatique de l'outil; donc loin de la sémantique particulière à l'application; donc difficilement assimilable par des non spécialistes de l'informatique.

Il semble donc que les modes mnémoniques, les réponses directes aux questions ou les sélections de menus, constituent les moyens de communication les plus souhaitables et utilisables dans les systèmes directement opérationnels en l'état actuel des matériels et des logiciels de CAO.

Parmi les objectifs d'un langage de commande, nous mettons l'accent sur les points suivants:

- le système doit détecter les erreurs, les signaler et surtout ne pas tomber en panne,
- l'utilisateur doit pouvoir utiliser le vocabulaire de tous les jours pour communiquer avec le système,
- l'utilisateur doit pouvoir utiliser le langage technique du domaine considéré,

- le système doit pouvoir être utilisé par les utilisateurs non expérimentés,
- le système doit répondre en fonction de l'expérience de l'utilisateur,
- le système doit aider l'utilisateur dès que cela est nécessaire,
- il doit être possible d'étendre dynamiquement les possibilités de communication.

Si les premiers points sont souhaitables dans tous les langages de commande quel que soit leur domaine d'application, le dernier point concernant l'extensibilité est bien plus important en CAO car il doit permettre aux concepteurs - utilisateurs de faire évoluer les outils pour pouvoir concevoir de nouveaux objets et prendre en compte de nouvelles caractéristiques.

Un autre aspect dont il faut tenir compte est le moyen de dialoguer. Historiquement limité à l'échange d'informations textuelles, les moyens se diversifient et permettent non seulement les dialogues graphiques mais aussi les premières utilisations opérationnelles de commandes vocales qui ont vu jour pour le dessin des circuits imprimés par exemple. Ceci introduit une nouvelle dimension dans la communication homme - machine et augmente ainsi la complexité de la mise en oeuvre.

La mise en place de langages de commande satisfaisant aux objectifs énumérés plus haut, n'est pas un travail facile. Il faut, en effet, à tout prix assurer la clarté, le style et l'évolutivité des dialogues. Il s'est avéré que ceci est très difficile à assurer dans le contexte habituel de la programmation où l'on mélange au sein d'un seul programme le dialogue et les manipulations algorithmiques des données. C'est pourquoi on a vu naître une approche dans laquelle les interactions ne sont plus noyées dans les programmes mais organisées en un module de dialogues chargé uniquement de la communication homme - machine. Les liens avec la partie algorithmique s'expriment à l'aide des FONCTIONS SEMANTIQUES. Les séquences de dialogue permises sont étudiées séparément, spécifiées à l'aide d'un langage de spécification de dialogues puis produites de façon automatique. Cette organisation permet de systématiser la mise en oeuvre et de satisfaire ainsi les objectifs (clarté, style, évolutivité, ...). En plus elle permet une souplesse dans le choix de la mise en oeuvre, par exemple différentes exploitations sont possibles (interpréteur, traducteur, ...). Cette démarche a été utilisée initialement dans ICES (Roo.67), puis présentée et discutée par WARMAN (War.77) et un exemple concret a été proposé par K.BO (Bo..77) dans le système GRANADA.

Cette démarche fondée sur l'expression des objectifs à atteindre au lieu de l'expression des moyens à utiliser est de

plus en plus utilisée et porte l'appellation de VERY HIGH LEVEL LANGUAGES.

Il nous semble important de constater qu'elle n'a été appliquée qu'aux langages de commande textuels.

6.2 LDC - UNE METHODE UNIFIEE POUR LA MISE EN OEUVRE DE LANGAGES DE COMMANDES

Nous nous sommes donc posé le problème de la mise en oeuvre d'une méthode pour définir et mettre en oeuvre des langages de commande pouvant exploiter indifféremment les trois modes de dialogues que nous avons retenu, mais aussi la lunge naturelle limitée. Cette méthode devra également vérifier l'ensemble des objectifs caractérisant un langage de commande satisfaisant tout en s'appuyant sur divers moyens de dialogues (textuel, graphique, vocal, ...).

Pour cela nous avons d'abord déterminé le modèle conceptuel de langages de commande, puis nous avons défini le langage de spécification de langages de commande. Ensuite nous avons étudié la réalisation effective du support de cette méthode. Nous rendons rapidement compte de ces différentes étapes dans ce qui suit.

6.2.1 SPECIFICATION DES DIALOGUES D'UNE APPLICATION

Tout d'abord nous devons définir ce qu'est un langage de commande.

Une commande est un nom suivi d'une liste composée de mots obligatoires ou pas, de paramètres obligatoires que l'on retrouve par position, de paramètres facultatifs que l'on retrouve par mot-clé et de fonctions sémantiques. A chaque commande on peut associer un ou plusieurs messages d'aide portant un numéro de niveau définissant les utilisateurs qui pourront les activer.

Le passage de valeurs de paramètres vers les fonctions sémantiques s'effectue selon un mode bien défini à l'aide d'une zone de communication.

Nous nommerons environnement un regroupement de commandes constituant un ensemble logique.

Un environnement est décrit par l'ensemble des commandes qui le constitue, par leur comportement (changement ou non d'environnement après l'exécution de la commande), ainsi que par les messages d'aide associés à l'environnement.

Pour obtenir une souplesse d'expression souhaitée il est possible de mélanger de façon récursive commandes et environnements.

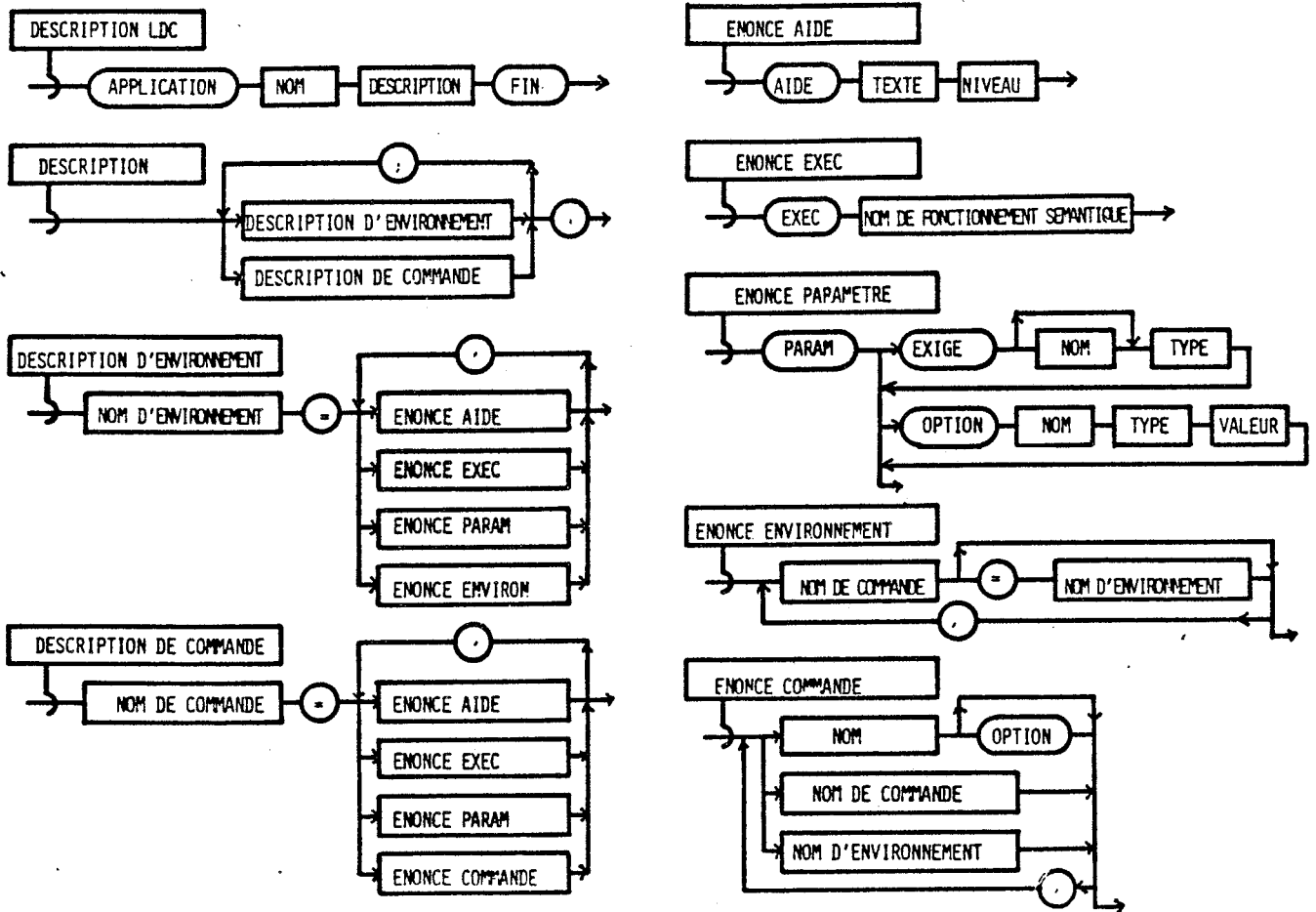


FIGURE II - 6.1 : GRAMMAIRE DU LANGAGE DE SPECIFICATION DE DIALOGUES

Ainsi il est possible de décrire:

- une organisation "plate" des dialogues, c'est à dire à un seul niveau (avec un seul environnement),
- le réseau des environnements,

- une organisation "hiérarchique" à plusieurs niveaux, c'est à dire pouvant contenir une hiérarchie des environnements.

De cette façon on couvre l'ensemble des possibilités d'organisation d'un processus de conception.

Tous les paramètres (obligatoires ou optionnels) sont typés et les paramètres optionnels peuvent recevoir une valeur par défaut.

Les types de données utilisés sont des types de base classiques (entier, réel, booléen, caractère, mot, chaîne) avec les possibilités de construire des tableaux et des structures. Pour pouvoir mieux exprimer la sémantique des paramètres nous permettons aussi des intervalles pour lesquels on précise le type de l'élément et les bornes, et des ensembles qui peuvent être énumérés ou construits. A cela s'ajoutent des types graphiques qui sont des ensembles ou des intervalles particuliers auxquels on a précisé dans le type, les caractéristiques graphiques telles que les noms de la figure, de la fenêtre et de la clôture.

La grammaire du langage de spécification de langages de commande se trouve sur la figure II - 6.1. Une description de dialogues est une suite de descriptions d'environnements et de commandes. Dans ces descriptions on utilise principalement les énoncés suivants:

- énoncé AIDE: indique le texte permettant d'expliquer à l'utilisateur ce qu'il doit faire. Le niveau associé indique à partir de quel niveau de compétence ce texte doit être présenté,
- énoncé EXEC: indique le nom de la fonction sémantique à exécuter,
- énoncé PARAM: indique le ou les paramètres à saisir. Il peut s'agir soit des paramètres obligatoires (EXIGE) repérés par position, soit des paramètres optionnels (OPTION), repérés par mot-clé et pouvant recevoir une valeur par défaut,
- énoncé environnement: indique les noms des environnements et des commandes accessibles,
- énoncé COMMANDE: indique la structure de la commande.

La variable COMPETENCE que l'utilisateur peut initialiser, (c'est à dire qu'il peut choisir son niveau de compétence), est modifiée soit par lui même soit de façon automatique vers le haut s'il ne fait pas de fautes, soit vers le bas s'il commet des erreurs.

La personne chargée de la définition et de la mise en oeuvre d'un programme interactif doit d'une part, définir les dialogues et les exprimer à l'aide de ce langage de spécifications et d'autre part, fournir les fonctions sémantiques évoquées dans la spécification.

La mise en place de ce programme s'effectue de la façon indiquée par la figure II - 6.2. Dans un premier temps les spécifications de dialogues sont analysées par un traducteur qui s'assure de leur justesse, puis en produit une version exécutable. Dans un second temps on associe à cette version exécutable les fonctions sémantiques (actions associées) et on est prêt à satisfaire les commandes de l'utilisateur.

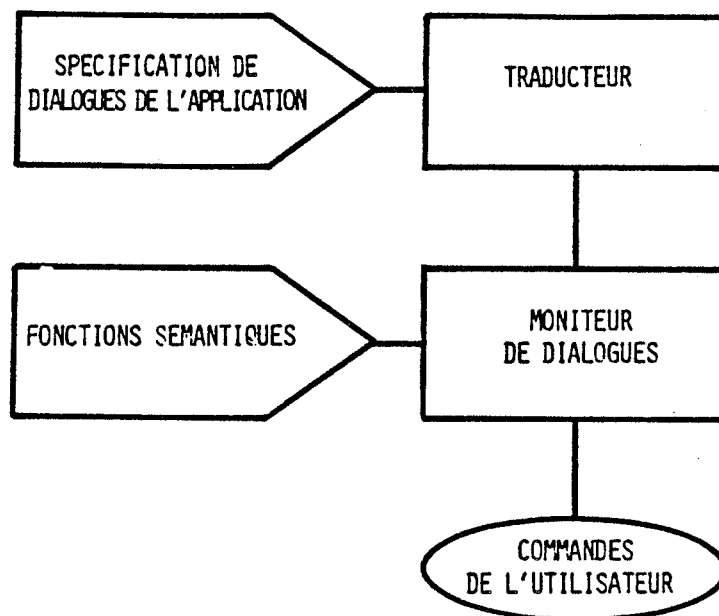


FIGURE II - 6.2 : SCHEMA DE PRODUCTION D'UN LANGAGE DE COMMANDE

```

APPLI      'CREATION DES HISTOGRAMMES' DEBUT=EXEC(1);
HISTOGRA = FILS(CREERCHR,VISUALISER,SUPPRIMER,FIN) REPETER SAUF POUR
           (FIN),EXEC(1),ALT;
CREERCHR=FILS(HORLOGES,SIGNAUX),ALT,EXEC(2);
HORLOGES=FILS(COMPLETE,PERIODE,CONTEXTE),ALT,EXEC(3);
SIGNAUX=FILS(COMPLETE,PERIODE,CONTEXTE,NUMERIQUE),ALT,EXEC(4);
COMPLETE=AIDE('DUREE GLOBALE DE LA SIMULATION' 0),
           PARAM(1 ENTIER 1),
           EXEC(5),FILS(RELEVE);
PERIODE=AIDE('DUREE DE LA PERIODE,NOMBRE DE PERIODES' 0),
          PARAM(2 ENTIER 1),
          EXEC(6),FILS(RELEVE);
RELEVE=AIDE('RELEVER LES PAS DU CHRONOGRAMME' 0),EXEC(7);
CONTEXTE=AIDE('FACTEUR D ESPACEMENT' 0),
          PARAM(1 ENTIER 1),EXEC(8);
NUMERIQUE=FILS(SERIECCM,SERIEPER,PARALCOM,PARALPER),
           ALT,AIDE('DUREE GLOBALE DE LA SIMULATION OU PERIODE'
                   0),
           EXEC(9),
           PARAM(1 ENTIER 1,1 ENTIER 2 =BASE 16);
SERIECCM=EXEC(10);
SERIEPER=EXEC(11),AIDE('NOMBRE DE PERIODES' 0),
          PARAM(1 ENTIER 1);
PARALCOM=EXEC(12);
  
```

FIGURE II - 6.3 : Exemple d'utilisation de LDC

Il semble important de souligner qu'il est possible selon ce même schéma, de produire différentes versions réalisant chacune différents objectifs:

- production directe d'un moniteur de dialogues, dans ce cas le traducteur compile la description et génère un moniteur directement exécutable,
- production de tables qui dirigent un moniteur de dialogues général, dans ce cas le moniteur procède par interprétation.

Ces deux versions sont tout à fait différentes car la première est très efficace mais très figée, tandis que la seconde est évidemment plus lente car interprétée mais plus facilement modifiable.

Nous avons préféré cette dernière car d'une part nous pouvons et voulons (cf. objectifs plus haut) faire évoluer les possibilités du langage de commande de l'application, et d'autre part la solution d'interprétation a l'avantage de permettre de réaliser le traducteur de spécifications et le moniteur de dialogues à l'aide du même programme dirigé par des tables différentes.

Selon que le moniteur de dialogues utilise une méthode syntaxique descendante ou ascendante, il peut soit réaliser les dialogues déterministes selon les modes 3, 4, 5 de notre classification, soit traiter le non déterminisme correspondant au mode 2, c'est à dire à la formulation en langue naturelle réduite comme par exemple (Ber.79). Nous avons préféré la méthode descendante qui fournit des dialogues plus précis.

6.2.2 STRUCTURE SUPPORT DE LA GESTION DES DIALOGUES

Jusque là nous avons vu comment on peut spécifier les dialogues d'une application. Mais dans une application on n'utilise pas seulement des commandes spécifiques. Pour diriger le fonctionnement il faut un ensemble de commandes que nous pouvons classer de la façon suivante:

- CA : les commandes spécifiques à chaque application,
- CS : les commandes système pouvant être utilisées quelque soit l'application. Il s'agit principalement des commandes de lancement d'un dialogue ou de l'arrêt, de déclaration de variables dynamiques, d'affectation de valeurs à ces variables, de la libération de ces variables, ainsi que de la gestion du poste de travail (voir plus loin),
- CC : les commandes de contrôle servant à étendre le fonctionnement. Ces commandes testent les valeurs des paramètres ou des variables dynamiques et selon le résultat activent ou pas les commandes CA, CS ou MC. Nous avons introduit deux commandes de contrôle: une conditionnelle

(SI) et l'autre répétitive (TANTQUE),
 - MC : les macro-commandes ayant pour rôle d'assurer l'évolutivité du fonctionnement.

La structure support de gestion des dialogues est la suivante (FIG.II-6.4):

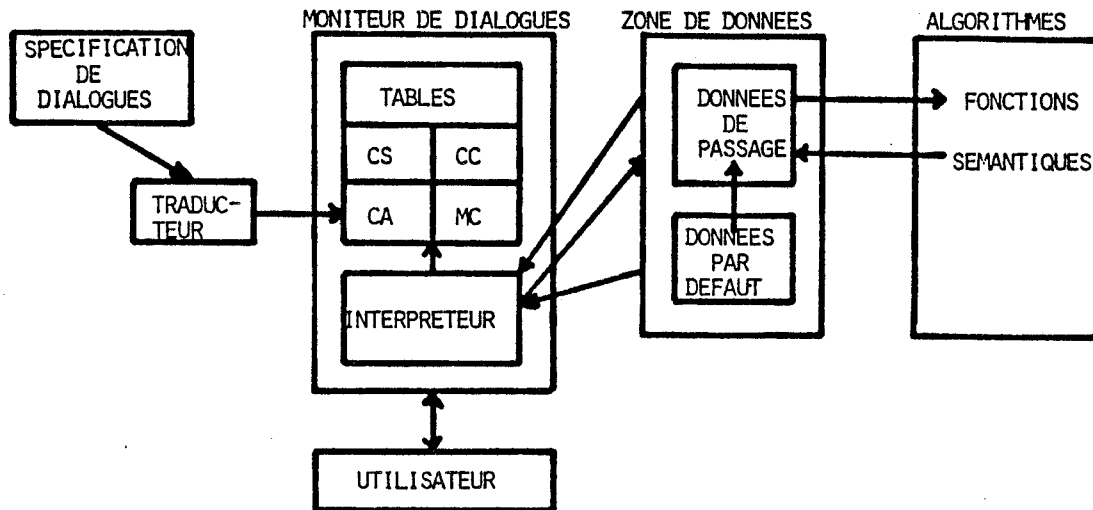


FIGURE II - 6.4 : STRUCTURE-SUPPORT DE LA GESTION DES DIALOGUES

L'utilisateur accède au moniteur de dialogues (nous allons voir plus loin comment). Ce moniteur peut être réalisé par exemple par un interpréteur général et des tables décrivant l'application. Les actions associées aux dialogues sont exprimées de façon algorithmique dans des fonctions sémantiques. Pour véhiculer des informations entre l'utilisateur et les algorithmes une zone de données a été créée entre le moniteur de dialogues et les algorithmes. Dans cette zone on stocke d'une part les valeurs par défaut les paramètres optionnels et d'autre part les valeurs effectives des paramètres. Si les premières données ne peuvent être manipulées que par le moniteur de dialogues (pour les recopier dans les paramètres effectifs chaque fois que c'est nécessaire), les secondes peuvent être accédées aussi bien par le moniteur que par les algorithmes.

Ce découpage précise l'organisation et contribue à définir une méthode de programmation.

6.2.3 LES EXTENSIONS

En ce qui concerne les extensions nous mettons en évidence différents niveaux selon leur complexité:

- le premier niveau est celui des extensions directement utilisables pendant l'exécution d'un programme. Il est possible de définir de nouvelles variables dans lesquelles on stocke les résultats obtenus. Sur ces variables on peut alors appliquer des opérations de comparaison, c'est à dire les situer dans les conditions d'exécution (si - alors, si - alors - sinon), ou d'exécution itérative (pour, tantque). La notion de définition passive d'une macro complète ce niveau. Il s'agit d'enregistrer au fur et à mesure du déroulement des dialogues, des commandes dans un fichier et de pouvoir les rappeler par la suite en évoquant uniquement le nom de la macro-commande ainsi constituée.
- le deuxième niveau est celui où l'utilisateur doit, toujours au sein du programme, indiquer les extensions avant de les utiliser. Ce travail peut être très simple: création classique de groupement de commandes (macros) avec un enchaînement séquentiel, conditionnel ou itératif; ou bien plus complexe: pour les macros paramétrées, modification de la liste des paramètres de la commande ou de l'appel des fonctions sémantiques. Ce travail reste néanmoins à la portée d'un utilisateur normalement expérimenté dans l'environnement qui est le sien.
- le troisième niveau est celui des modifications très importantes concernant notamment les changements d'algorithmes et plus généralement, les modifications dans les fonctions sémantiques. Le contenu de ces fonctions n'est pas géré par le moniteur de dialogues, il est donc nécessaire de reprogrammer ces modifications. Il paraît évident qu'un effort devrait être fait dans un sens permettant de spécifier également le contenu des fonctions sémantiques.

6.2.4 GESTION DES MODES ET DES MOYENS DE DIALOGUE

Pour montrer que cette démarche permet de remplir l'ensemble des objectifs que nous nous sommes fixés, il faut étudier plus finement l'organisation d'un système CAO et la façon dont le moniteur de dialogues s'insère dans un tel système.

Un schéma très simple, représenté à la figure II-6.5, montre que le poste de travail constitue le seul moyen de l'utilisateur d'accès au système CAO. On constate que selon l'application le poste de travail peut être plus ou moins élaboré. Il semble utile d'associer au poste de travail, le moniteur de communications dont le rôle est de gérer les informations.

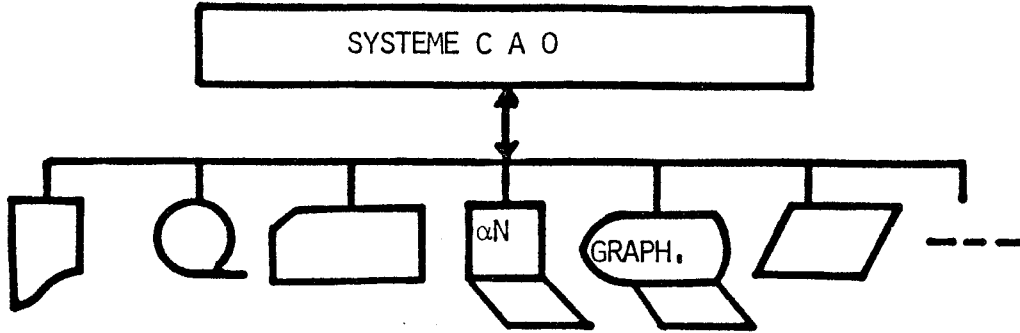


FIGURE II - 6.5 : POSTE DE TRAVAIL - LE MOYEN D'ACCES AU SYSTEME CAO

Un algorithme interactif est, comme nous l'avons vu précédemment, constitué par le moniteur de dialogues, des tables définissant les dialogues spécifiques et des fonctions sémantiques associées.

L'information qui vient de l'utilisateur, via le poste de travail, n'est pas présentée au moniteur de dialogues sous la forme brute, mais elle a été déjà prétraitée: elle a été mise sous une forme standard (canonique) par le moniteur de communication. Cela signifie que le moniteur de dialogues n'a pas à s'occuper de la provenance de l'information (du terminal graphique, du terminal alpha-numérique, du digitaliseur, d'une entrée vocale, etc...). Le deuxième pré-traitement est fait au niveau lexicographique et le moniteur de dialogues traite ces informations uniquement en tant qu'entité syntaxique.

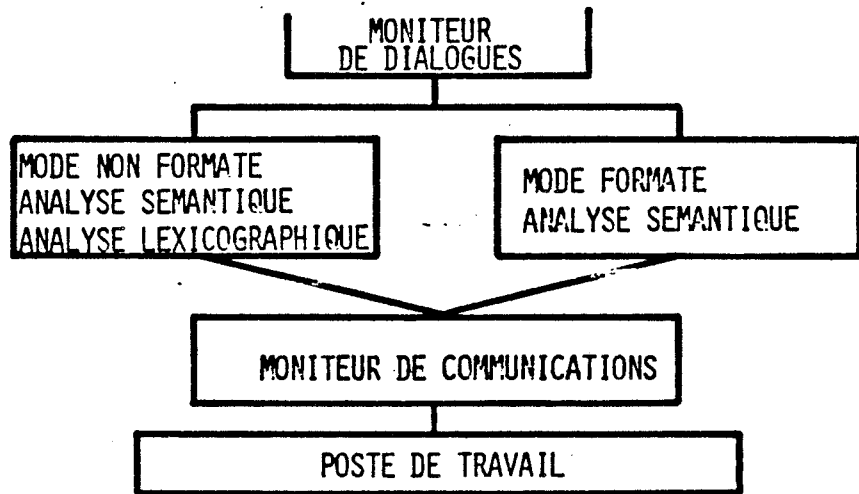


FIG. II - 6.6 : CHEMINEMENT D'INFORMATION ENTRE L'UTILISATEUR ET LE MONITEUR DE DIALOGUES

De la même façon on peut présenter à l'utilisateur une information sous la forme qu'il a choisie. Ce découpage plus fin, schématisé sur la figure II - 6.6, permet de prendre en compte des postes de travail très variés et textuels tout en laissant de côté des dialogues graphiques, vocaux, etc. et sans être obligé de créer des dialogues spécifiques pour chaque périphérique particulier. Avec l'approche que nous proposons (et que nous avons expérimentée), il est possible de spécifier tous les dialogues et de fournir à partir de la même spécification des modes aussi différents que le mode libre, le mode tutorial, le mode graphique, le mode différé, etc... Le mode du dialogue peut-être:

- LIBRE: dialogue où l'utilisateur dirige les dialogues, la machine lui fournit éventuellement des aides selon le niveau de compétence choisi,
- TUTORIAL: dialogue guidé où la machine présente les choix possibles à un moment donné,
- GRAPHIQUE: dialogue s'appuyant sur les outils graphiques de dialogue et s'effectuant de façon tutoriale,
- DIFFERE: dialogue enregistré au préalable et dont l'exécution peut s'effectuer en traitement par lot.

Avec cette organisation il est possible par exemple d'envoyer toutes les aides du système sur un écran de contrôle, de décider que les commandes activables à un moment donné seront proposées par un menu sur le terminal graphique, ou bien sur le clavier de touches de commandes, ou bien encore sur le menu du digitaliseur.

Indépendamment de l'application nous pouvons assurer la gestion de l'écran graphique ou de l'écran alpha-numérique intelligent, en indiquant les emplacements où se situent les zones d'interaction.

Par exemple nous avons pu prendre en compte le logiciel graphique de base GRIGRI (LLM.78). Dans ce cas, au lieu de nous fonder sur la saisie d'une ligne de texte contenant en forme libre la commande, nous utilisons les primitives de dialogues qui sont les suivantes:

RVAL, NVAL, CVAL pour l'introduction de données alpha-numériques (réel, entier, chaîne)
 POSITION pour la collecte de coordonnées
 IDENTIFICATION pour l'identification d'un objet graphique
 MENU pour le choix d'une alternative dans un menu

et nous générons le MENU pour choisir les commandes, les RVAL, NVAL, CVAL pour les paramètres de type alpha-numérique, et

POSITION et IDENTIFICATION pour les paramètres de type graphique. Pour ces paramètres graphiques nous pouvons utiliser soit l'écran soit le digitaliseur.

En conclusion nous pouvons constater, après expérimentation, que cette démarche complète très utilement la démarche textuelle utilisée jusqu'à présent et permet de tirer profit d'un poste de travail complet.

Il est néanmoins évident qu'étant fondée sur la notion de commande, cette démarche ne couvre ni les dialogues en langue naturelle ni le dessin à main levé.

CHAPITRE 7

TACHES DANS LE PROCESSUS

RESUME: Le dernier volet de l'invariant conceptuel concerne les tâches (ou les algorithmes) intervenant dans le processus. Après la présentation de différentes classifications de tâches, on s'intéresse à la spécification de tâches et on montre l'importance d'une description du point de vue interne et externe: l'aspect externe définit le comportement de la tâche vis-à-vis du système CAO, l'aspect interne caractérise les principes de fonctionnement. Le langage de spécification proposé permet d'exprimer ces deux aspects de la tâche. Ce chapitre se termine par la présentation de principes d'activation des tâches.

7. TACHES DANS LE PROCESSUS

7.1 DEFINITION ET CLASSIFICATIONS

La dernière des quatre composantes principales d'un système CAO (processus de conception, gestion des données, dialogues homme-machine, tâches de conception) est la composante TACHES DE CONCEPTION. En effet dans un système CAO intégré le concepteur - utilisateur a la possibilité d'effectuer à sa demande diverses tâches enrichissant l'objet en cours de conception. La notion de tâche est une notion de base d'un système CAO, il est donc indispensable de la définir.

Dans le travail manuel le concepteur effectue un certain nombre de tâches pour élaborer son objet. Lors du passage à l'informatisation, il n'est pas souhaitable de reproduire exactement les mêmes tâches mais il vaut mieux étudier la façon de parvenir à un meilleur résultat tout en tenant compte du nouveau contexte (nouvelles possibilités, capacités différentes, ...). Il faut donc reconsidérer le contenu de ces différentes tâches, les redéfinir et dégager une nouvelle organisation. Pour marquer cette différence on introduit les notions de TRAITEMENT, d'OPERATEUR et d'OUTIL, qui correspondent chacune à un travail identifiable du point de vue informatique: TRAITEMENT - tout travail informatique, OPERATEUR - toute tâche destinée ou pouvant servir à un utilisateur, OUTIL - toute tâche plus spécifiquement orientée vers le fonctionnement du système.

Une tâche manuelle peut être réalisée à l'aide d'un ou plusieurs traitements informatiques. Elle peut aussi être réalisée par une partie d'un traitement si dans le nouveau contexte, l'identité de (ou l'intérêt d'identifier) cette tâche disparaît.

En fait il s'agit d'un remodelage du découpage en tâches et souvent par abus de langage nous utiliserons indifféremment tâche, traitement, opérateur et outil car les différences s'estompent.

Pour définir complètement la notion de tâche dans un processus CAO nous devons mettre en évidence les trois aspects suivants:

- description externe de la tâche: tout ce qui caractérise une tâche,
- description interne de la tâche: comment réalise-t-on une

- tâche, quelles en sont les règles de mise en oeuvre,
- activation et enchaînement des tâches: comment active-t-on une tâche et de quelles façons peut-on enchaîner différentes tâches.

Avant de détailler ces différents aspects, envisageons les différentes possibilités de classification des tâches:

- par rapport au type d'activité au sein d'un projet. On peut distinguer notamment:
 - + conduite de projet
 - + exploitation de l'acquis technique
 - + outils de description: symboles et dessins
 - + évaluation de solutions
 - + outils d'interface avec des traitements annexes
- de façon plus générale on peut caractériser les outils de la façon suivante:
 - + les vérificateurs (statiques ou dynamiques, visualisation, simulation, ...)
 - + les traitements documentaires (production de documents définitifs ou de travail)
 - + aides à la conception:
 - = étude et précision de problèmes
 - = analyse, aide à la décision
 - = résolution de problèmes (optimisation, implantation, ...)
 - = stimulation de la créativité (génération de configurations, production de variantes, ...)
- par rapport à l'action sur des données:
 - + productrice d'informations
 - + consommatrice d'informations
 - + productrice et consommatrice d'informations
- par rapport au contact avec l'utilisateur:
 - + tâche automatique (batch)
 - + tâche interactive (dialogues homme-machine)
- par rapport à la façon dont une tâche est réalisée et en dépassant la distinction élémentaire entre une tâche à contenu général et une tâche à contenu spécifique:
 - + outils de base dont le rôle est d'assurer la réalisation des fonctions de base du système CAO (voir chapitre 9):
 - = gestion des données
 - = gestion des algorithmes
 - = gestion du poste de travail
 - = gestion du moniteur de conception
 - = etc...
 - + outils généraux logiques: indépendants des applications et assurant des fonctions logiques telles que:

- = conduite de projet
- = exploitation de l'acquis technique
- = production du dossier
- = etc...
- + outils technologiques généraux: qui permettent d'exécuter des tâches dans les différents domaines d'application:
 - = conception de dessins
 - = évaluation de coûts
 - = simulations
 - = etc...
- + outils technologiques spécifiques: qui permettent d'exécuter des tâches spécifiques dans le domaine concerné:
 - = tracé de circuits
 - = insertion automatique de composants sur une carte
 - = etc...

Nous pouvons donner maintenant notre définition de TACHE:

une tâche est un traitement autonome (donc identifiable) d'une certaine importance, qui s'effectue à la demande de l'utilisateur (donc à la suite d'une commande).

Nous laissons donc volontairement de côté les tâches système (outils de base), car elles ne sont pas directement accessibles à l'utilisateur et contribuent uniquement au bon fonctionnement du système CAO.

7.2 DESCRIPTION D'UNE TACHE

Après avoir défini ce que nous entendons par tâche dans un contexte de CAO, nous devons maintenant déterminer par quoi une tâche est définie.

Il semble utile de distinguer deux aspects:

- description externe qui caractérise la tâche vis à vis de l'ensemble du système CAO,
- description interne, qui est orientée vers la façon dont la tâche est ou doit être réalisée.

7.2.1 Description externe d'une tâche

Décrire une tâche de façon externe c'est préciser son comportement vis à vis du système CAO. Ce comportement se caractérise par différents aspects qui sont principalement:

- étapes du processus de conception: il s'agit de préciser dans quelle partie du processus de conception cette tâche peut intervenir et notamment à quel niveau de résolution et à quel niveau morphologique (étape et phase du processus),
- type d'utilisateur pouvant s'en servir: il s'agit de préciser un autre aspect organisationnel, celui des utilisateurs et d'indiquer qui peut accéder à cette tâche et dans quelles conditions. Il ne s'agit pas de la personne elle même, mais de la fonction dans le projet (chef de projet, conseiller, exécutant, ..., d'une étape et phase du projet).
- ressources nécessaires: il s'agit de préciser quelles sont les ressources dont la tâche a besoin. Ces ressources sont d'une part liées au poste de travail nécessaire (poste graphique par exemple) et d'autre part indiquent le potentiel indispensable au niveau du processeur: taille mémoire, puissance de calcul, ressources particulières comme certains catalogues, etc...
- données nécessaires et produites: il s'agit de préciser le comportement de la tâche vis à vis des données globales, c'est à dire son action sur le contexte informationnel. Il faut préciser notamment ses liens avec la BDP (objet à concevoir, objet conçu, relation de cause à effet), avec la BDT (catalogues techniques) et éventuellement avec la BDG (informations générales) (Fig. II - 7.1). Il ne faut pas oublier non plus les données que l'utilisateur fournit à l'activation de la tâche.
- liens avec d'autres tâches: pour bien situer la tâche dans le système il s'agit de préciser ses liens avec d'autres tâches pour mettre en évidence (Fig. II - 7.2):
 - + celles produisant des données nécessaires à cette tâche,
 - + celles consommant des résultats de cette tâche.

Sur la figure II - 7.2 nous pouvons observer le comportement d'une famille d'outils, celle de manipulations de symboles-schémas dans un système CAO des équipements électroniques. On trouve au centre du tableau la liste des fonctions que doivent assurer différents outils de cette famille. Pour chaque outil nous définissons son domaine d'application (conduite de projet, fabrication,...). Les liens entre l'outil et les autres types d'outils CAO sont symbolisés par:

- X - les résultats de l'outil seront utilisés dans d'autres types d'outils,
- O - l'outil doit utiliser les résultats des autres types d'outils.

7.2.2 Description interne d'une tâche

Décrire une tâche de façon interne c'est préciser son comportement intrinsèque, c'est à dire exprimer comment la tâche est effectivement réalisée.

Pour rendre ce travail utile et efficace, il est indispensable de définir une structure standard de tâche. C'est ce que nous allons faire, incomplètement, car nous devons intégrer certains aspects qui seront abordés dans le chapitre 9.

Pour permettre une grande réutilisation de programmes et minimiser ainsi les efforts de programmation nous essayons d'appliquer le plus strictement possible la règle suivante:

ORGANISER LA TACHE AFIN DE PERMETTRE UNE GRANDE INDEPENDANCE ENTRE LES DONNEES, LES TRAITEMENTS ET LES DIALOGUES.

De cette façon nous sommes dans un contexte de développement de logiciel bien structuré.

Une tâche est donc intrinséquement définie par quatre parties (Fig. II - 7.3):

- dialogues
- données globales manipulées
- fonctions spécifiques
- fonctions système

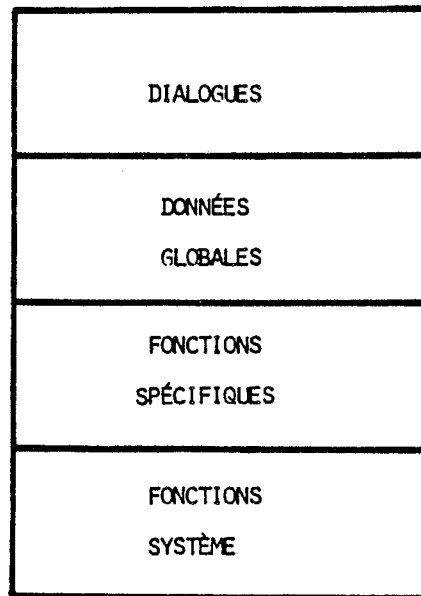


FIGURE II - 7.3 : STRUCTURE D'UNE TACHE

La partie "dialogues" a été traitée dans le chapitre 6, celle concernant les données dans le chapitre 5 et celle sur les "fonctions système" le sera dans le chapitre 9. Nous présentons ici la partie "fonctions spécifiques".

La partie "fonctions spécifiques" contient l'ensemble des fonctions demandées par la tâche et activées par le moniteur de dialogues. Le dialogue pilote donc les différents traitements effectués au sein d'une tâche. Cet ensemble de fonctions est regroupé dans un MODULE.

Par exemple pour la visualisation d'un objet tridimensionnel nous disposons des fonctions suivantes:

- lecture de la représentation de l'objet de la BDP,
- translation,
- rotations,
- changement d'échelle,
- choix de projection (axonométrie, perspective normale, cylindrique, sphérique, ...),
- visualisation sur le support choisi (terminal graphique, table traçante, ...),
- sauvegarde de l'objet dans la BDP,
- ...

En général un MODULE est constitué par un ensemble de fonctions ayant un but commun et travaillant sur des données communes. Tous les modules ont la même structure. Pour des raisons évidentes de commodité de mise en œuvre, nous avons fait la distinction entre les modules systèmes (appelés modules logiques) et les modules utilisateurs (appelés modules technologiques). Les premiers assurent les fonctions de la machine abstraite MACAO (chapitre 9), les seconds les développements effectués sur cette machine. Ils peuvent être soit d'intérêt général et donc utilisables dans plusieurs applications, soit spécifiques à une application.

Lors de la mise en place d'une nouvelle application on ne réalise que les modules technologiques spécifiques.

Pour faciliter cette tâche, ces modules auront le même squelette et obéiront à une même méthode de programmation.

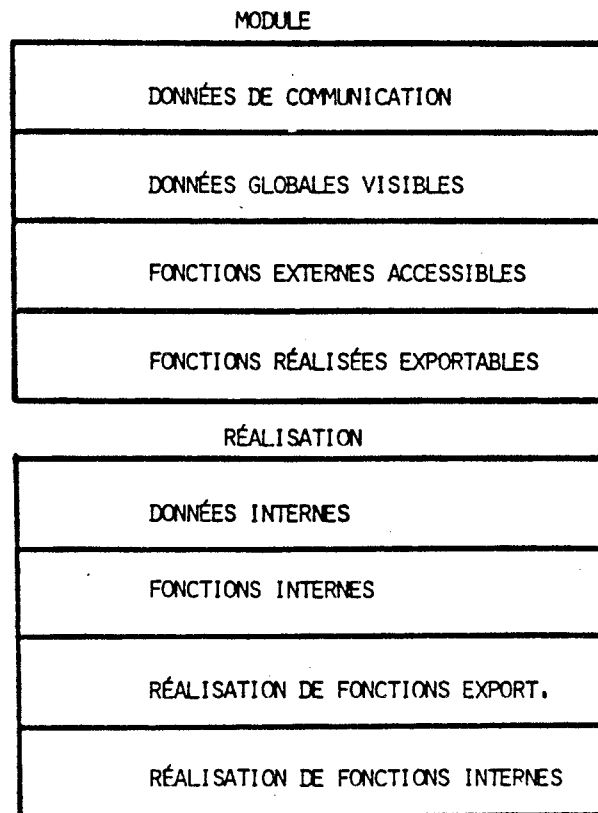


FIGURE II - 7.4 : STRUCTURE D'UN MODULE

La structure d'un module est schématisée sur la figure II - 7.4.

Parmi les nombreux problèmes liés à cette définition de module, le plus important est celui de l'accessibilité des variables depuis l'extérieur. L'utilisation directe de ces variables dans les programmes, pose un problème de visibilité de la représentation, qui n'est pas bien vu dans les réalisations modernes (types abstraits) où on se force à cacher la réalisation; l'accès s'effectue par des fonctions associées. Nous nous sommes imposé cette règle sans malheureusement pouvoir la contrôler car nous n'avions pas à notre disposition un langage de mise en oeuvre permettant la définition de modules. Notre objectif n'était pas de définir un langage de ce style. Et comme les langages actuellement disponibles (CIVA(Der.74), SESAME(Mos.77), AIM(Der.79),..) ne sont pas suffisamment portables, nous n'avons pu les utiliser car c'est un objectif essentiel de notre projet. Une solution possible et intéressante aurait pu être le projet FORTISH(Del.79), auquel nous avons participé, notamment pour sa portabilité, car il s'agissait de compiler le langage en FORTRAN. Malheureusement ce projet n'était pas suffisamment avancé quand nous avons commencé notre réalisation.

Peut être que dans le futur, les langages comme MEFIA ou surtout le GREEN de DOD devenu ADA, marieront ces deux atouts indispensables: modernité et portabilité.

Nous avons donc bâti un premier prototype à l'aide d'une version modulaire d'ALGOLW, puis nous avons procédé à sa traduction en FORTRAN.

Une version en BASIC nous permettrait d'atteindre les utilisateurs de micro-ordinateurs; une version APL pourrait être aussi intéressante.

Nous avons délibérément oublié le langage assembleur car les gains en temps d'exécution ne nous semblaient pas compenser les difficultés accrues lors de la mise en oeuvre, de la maintenance et de la portabilité.

Pour plus de détails voir le chapitre 9.

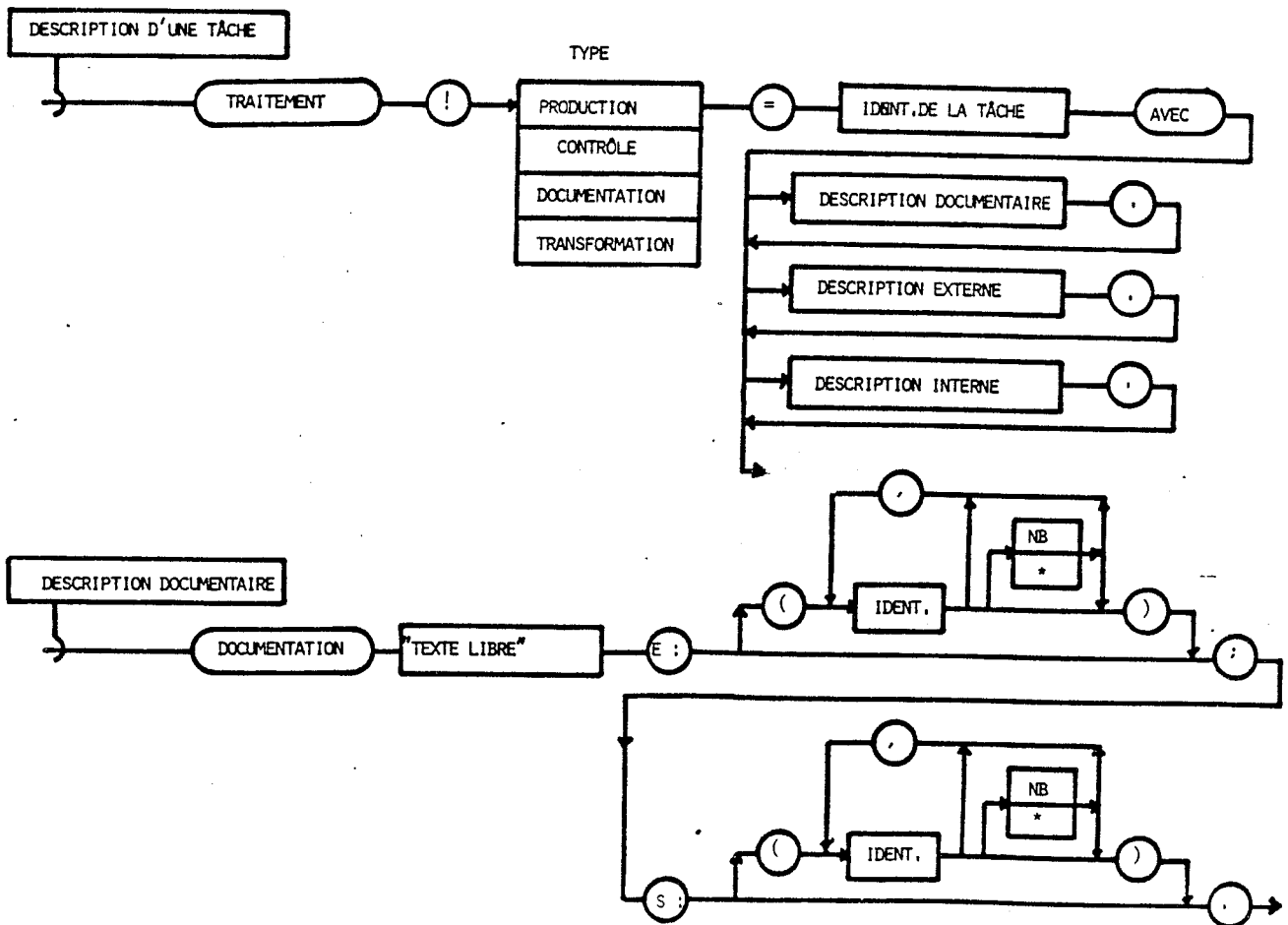
7.3 SPECIFICATION D'UNE TACHE

Dans le paragraphe précédent nous avons déterminé comment il est possible de décrire une tâche aussi bien de façon externe qu'interne. Nous pouvons maintenant définir un langage à l'aide duquel cette description peut être faite.

La grammaire de ce langage est représentée sur la figure II - 7.5. A l'aide de ce langage nous pouvons décrire soit des tâches demandées par l'utilisateur (tâches de PRODUCTION, de CONTRÔLE ou de DOCUMENTATION) et activables par celui-ci à la suite d'une commande (TACHES C), soit des tâches de TRANSFORMATION (TACHES T), activables par le système dans le cas de l'impossibilité d'activer directement la tâche demandée (insatisfaction de la condition nécessaire mais satisfaction de la condition suffisante).

La description d'une tâche se compose de trois parties:

- une description documentaire: contenant quelques renseignements sur la tâche dont une explication en texte libre de son but,
- une description externe: précisant les conditions d'utilisation,
- une description interne: contenant des informations sur la réalisation, l'organisation et l'activation de la tâche.



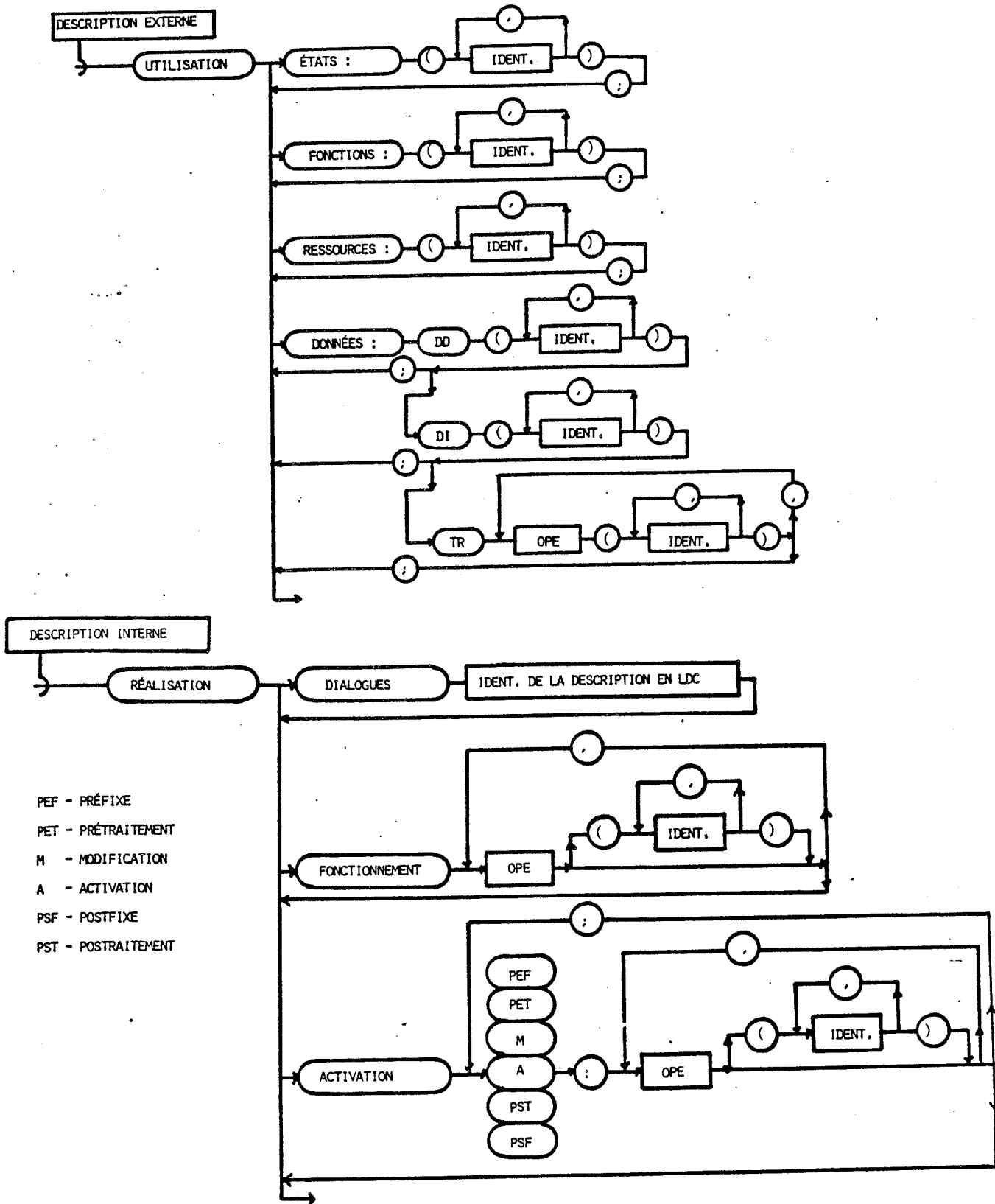


FIGURE II - 7.5 : LANGAGE DE SPECIFICATION DE TACHES

La documentation de la commande est effectuée essentiellement à partir de la description externe décrivant l'ensemble des conditions d'utilisation. Pour compléter cette documentation il faut notamment une explication libre des objectifs de la tâche et les indications sur les paramètres que l'utilisateur doit fournir à cette tâche.

Le contenu des descriptions externe et interne a été précisé dans le paragraphe 7.2.

La Fig.II-7.6 montre un exemple de description d'une tâche.

| | |
|--|-------------------------------|
| $C = \sqrt{2}D$ | |
| DOCU "VISU EN 2 DIM" |] DESCRIPTION DOCUMENTAIRE |
| E : N, T, R *; S : . | |
| UTILISATION |] DESCRIPTION EXTERNE |
| RESSOURCES : STGR; | |
| DONNEES : DD N, T, R. |] DESCRIPTION INTERNE |
| REALISATION ACTIVATION | |
| PEF : PARMS(T, 1), PARMS(R,*); | |
| M : SPEV $\sqrt{3}D(2)$; | |
| A : EFF(7), LOAD(ZT $\sqrt{3}$), START; | |
| PSF : ECLAT. | |

FIGURE II - 7.6 : EXEMPLE DE SPECIFICATION D'UNE TACHE

Il s'agit d'un programme de visualisation d'un ensemble d'objets en deux dimensions. A l'entrée ce programme a besoin de connaître:

- N - dimension de l'ensemble,
- T - description des objets à visualiser,
- R* - une ou plusieurs relations entre objets.

Ce programme ne produit pas d'autres résultats que la sortie graphique. La ressource indispensable pour ce programme est un poste de travail graphique standard (STGR). Toutes les données sont les données directes, c'est à dire qu'elles doivent être fournies au programme. Il n'existe pas de données indirectes,

c'est à dire des données à partir desquelles il serait possible de déduire automatiquement, par un programme de transformation, des données directes.

L'activation commence par le préfixe qui saisit le nom de l'ensemble d'objets à visualiser et des noms des relations, puis la séquence de modification indique à l'algorithme qu'il s'agit des objets en deux dimensions. L'appel de l'algorithme s'effectue en trois étapes: l'effacement d'un fichier temporaire, chargement de l'algorithme et son activation.

Le postfixe termine l'exécution par la procédure ECLAT.

La description d'une tâche est analysée pour vérifier si elle ne comporte pas d'anomalies internes ou externes, c'est à dire par rapport à l'ensemble des spécifications globales. Une documentation est alors éditée. Pour pouvoir organiser convenablement l'activation des tâches nous nous imposons un schéma de lancement unique. Une tâche se déroule toujours selon le schéma suivant:

- la saisie de la demande d'activation,
- le préfixe - travail préparatoire pour l'activation,
- le pré-traitement - travail préparatoire non standard,
- l'activation,
- le post -traitement - travail de terminaison non standard,
- le post-fixe - travail de terminaison standard.

De façon plus précise:

- le préfixe standard doit assurer
 - = la saisie de paramètres fournis par l'utilisateur,
 - = la vérification de l'applicabilité physique (conditions d'utilisation),
 - = la vérification de l'applicabilité sémantique (phase, étape du processus, données nécessaires),
 - = l'appel aux transformations éventuelles,
- le lancement assure
 - = la mise en place physique de la tâche,
 - = l'initialisation de la tâche,
 - = l'activation de la tâche,
- le postfixe standard assure
 - = le retour des données dans la BDP,
 - = la mise à jour de l'état d'avancement du processus,
 - = etc...

Pour pouvoir intégrer dans le système, des programmes complets conçus séparément, c'est à dire ne respectant pas notre organisation et notre méthode de programmation, nous introduisons deux traitements d'interface (pré-traitement et post-traitement), qui ont pour rôle d'assurer le lien entre l'organisation globale du système et le fonctionnement spécifique de ce programme particulier (Fig.II - 7.7).

En plus, pour des facilités de manipulation, il est possible d'insérer à tous les niveaux une séquence dite de modification permettant de prendre en compte l'imprévu et de garder une souplesse d'utilisation plus grande, importante surtout lors d'intégration de programmes existants.

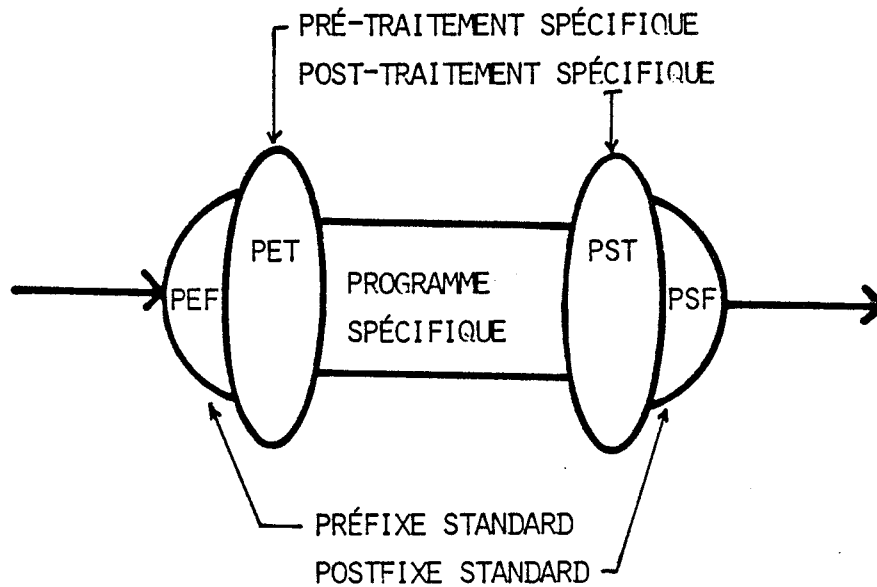


FIGURE II - 7.7 : INTEGRATION D'UN PROGRAMME SPECIFIQUE

Pour les détails de cette démarche et notamment tous les problèmes liés à la collaboration de la tâche avec le système nous devons nous reporter au chapitre 9 où nous introduisons la machine abstraite et au chapitre 10 où une réalisation particulière est présentée, ainsi qu'au chapitre 13 où nous étudions des problèmes de transport.

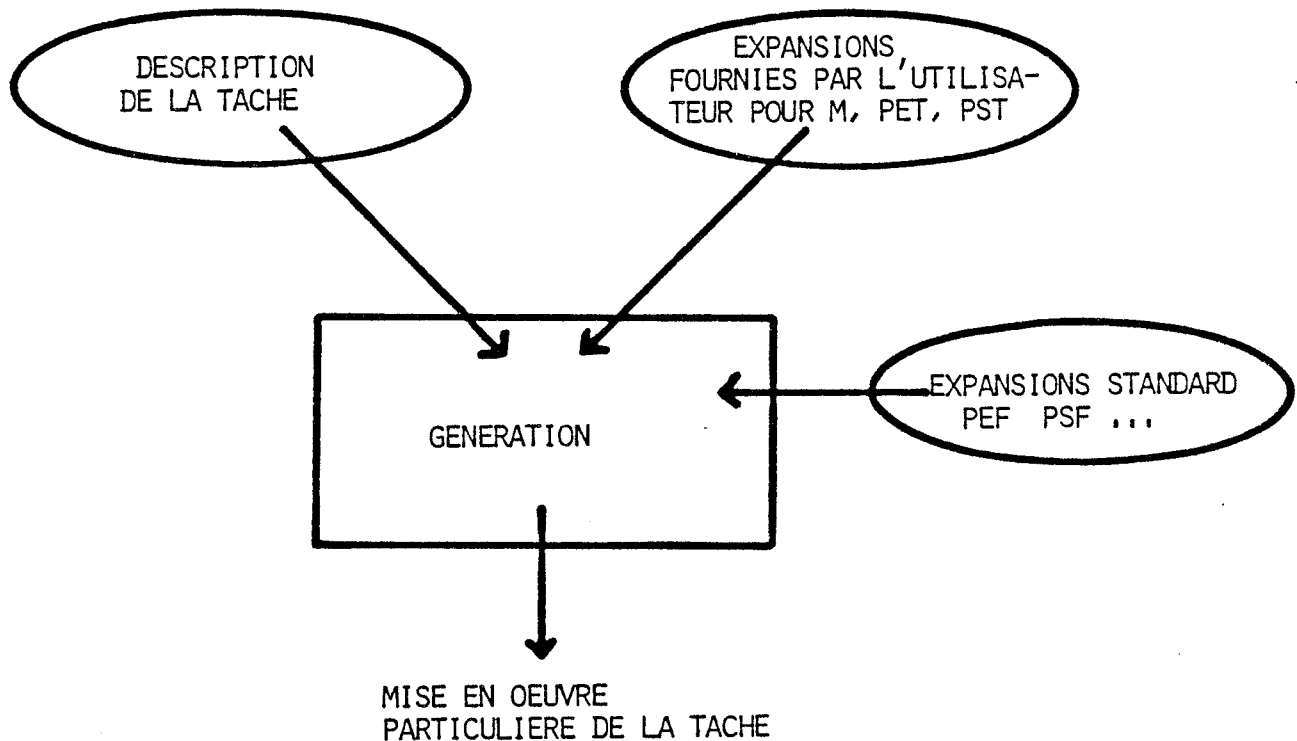


FIGURE II 7.8 : SCHEMA DE FONCTIONNEMENT DE TRAITEMENT DU LANGAGE DE SPECIFICATION DE TACHES

Pour des raisons de portabilité, que nous avons déjà discutées, nous devons, dans la mesure du possible, rester indépendant d'un contexte particulier et fournir un outil de production pouvant s'adapter à différents contextes.

Les phases de spécification, de vérification et de documentation sont les mêmes quelque soit ce contexte.

Après avoir vérifié que la spécification de la tâche est correcte, il faut construire une réalisation exploitable dans le contexte spécifique (matériel et logiciel). Nous pouvons pour cela nous appuyer sur la méthode travaillant selon le schéma désormais classique (Fig. II - 7.8) et produisant une réalisation particulière de la tâche dans le contexte demandé.

Pour nous, cela signifie que nous devons associer à chaque élément de spécification (PEF, PSF, A, M,..) une fonction de génération. Ces fonctions, appelées FONCTIONS DE SERVICE, ont chacune un rôle précis, mais la façon dont elles le remplissent dépend du contexte particulier.

Fondamentalement nous pouvons adopter l'une des deux solutions suivantes:

- produire des ordres directement exécutable dans le contexte choisi,
- produire des appels à d'autres fonctions de service et procéder donc par interprétation.

Chacune de ces solutions a évidemment ses avantages et ses inconvénients. Pour la première l'avantage majeur se situe dans la rapidité d'exécution et l'inconvénient dans sa rigidité. Pour la seconde les rôles sont inversés.

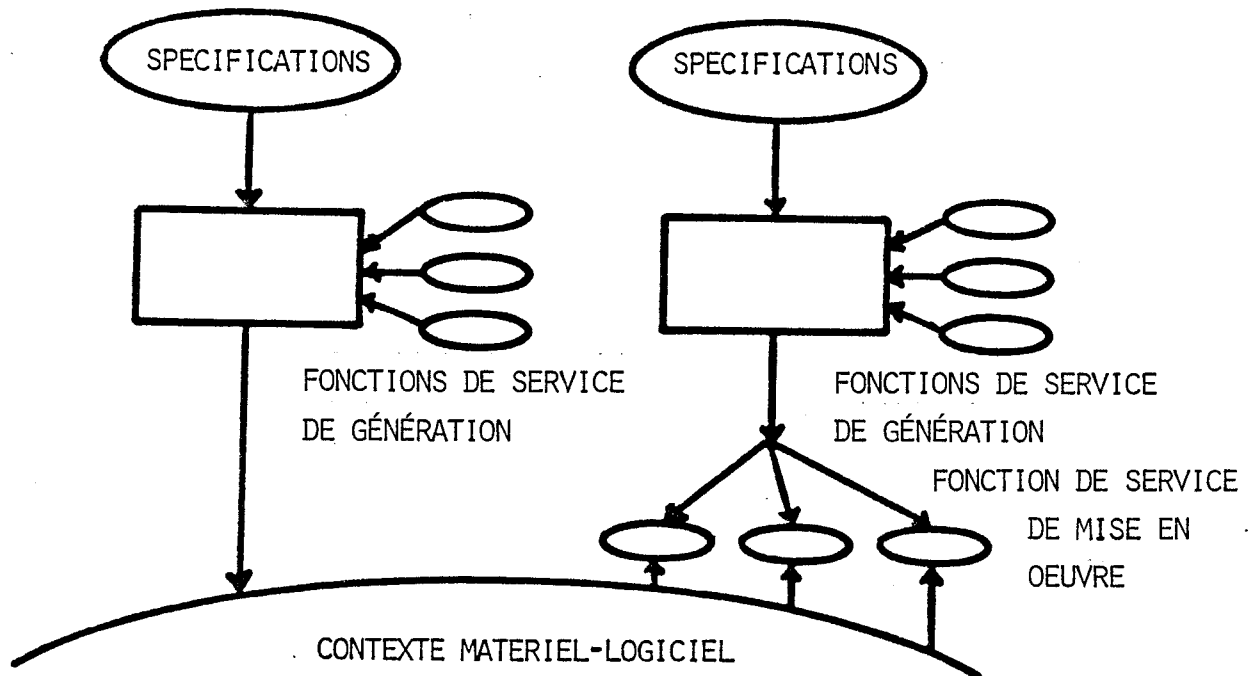


FIGURE II - 7.9 : DEUX SOLUTIONS DE MISE EN OEUVRE

Dans notre démarche nous sommes en mesure d'assurer ces deux types de solution.

Il semble intéressant d'étudier comment on peut prendre en compte les différents niveaux de changement. Lors du changement de contexte (support matériel et/ou logiciel) on doit modifier le comportement des fonctions de service. Selon la solution choisie (Fig.II - 7.9) et la complexité des changements on peut traiter :

- au niveau FONCTIONS DE SERVICE DE MISE EN OEUVRE: la modification du contenu de ces fonctions permet de tenir compte des changements intervenus dans le comportement du support; utile par exemple pour l'intégration d'un nouveau logiciel support,
- au niveau FONCTIONS DE SERVICE DE GENERATION: la modification du contenu de ces fonctions permet de tenir compte des changements souhaités dans le code généré; utile

par exemple pour le transport vers un nouveau système support.

On peut ainsi prendre en compte des contextes différents sans modifier les spécifications.

Si on veut permettre l'utilisation des nouvelles possibilités de spécification, on peut procéder de deux façons différentes selon la nature et la complexité de ces changements:

- niveau MODIFICATIONS: grâce à la clause "MODIFICATIONS" l'utilisateur peut introduire dynamiquement les extensions qui lui semblent nécessaires et lui sont propres. Pour qu'elles puissent être prises en compte il doit fournir la procédure associés,
- niveau GRAMMAIRE: la modification de la grammaire permet une prise en compte de nouvelles possibilités de spécification d'intérêt général. Ce dernier cas doit évidemment être évité autant que possible.

La Fig. II - 7.10 montre la hiérarchie de ces changements.

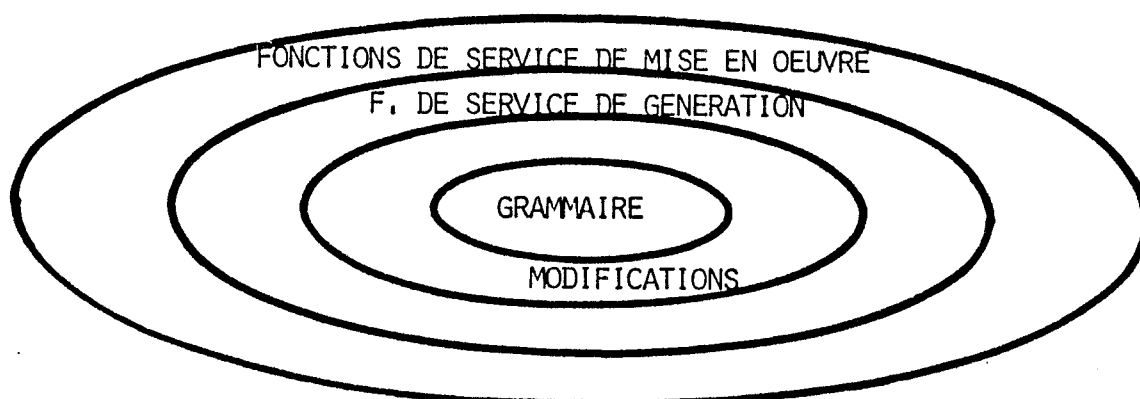


FIGURE II - 7.10 : HIERARCHIE DES CHANGEMENTS

7.4 ACTIVATION ET ENCHAINEMENT DE TACHES

Pour effectuer l'activation et l'enchaînement de tâches il faut mettre en place un moniteur de tâches.

Ce moniteur doit avoir à sa disposition la description de l'ensemble des tâches et leurs schémas d'activation, ainsi qu'un

schéma global d'enchaînement, correspondant au processus de conception.

Cette description du processus de conception peut être soit explicite, donc décrite de façon globale et synthétique, soit implicite, donc décrite de façon diffuse dans la description d'activation de chaque tâche. Chacune de ces deux solutions a ses avantages. Il est certain que la première permet une vision globale mais peut être plus contraignante, la seconde par contre est beaucoup plus souple, mais ne donne pas une vision synthétisée du déroulement du processus.

Pour des questions d'évolutivité, nous préférons cette deuxième solution. Le schéma de principe est donc le suivant (Fig. II - 7.11):

Le moniteur de tâches prend le contrôle à l'activation du système CAO. Il se charge de saisir les demandes de l'utilisateur et active si c'est possible la tâche demandée, après avoir saisi l'ensemble de paramètres interactifs nécessaires à la tâche.

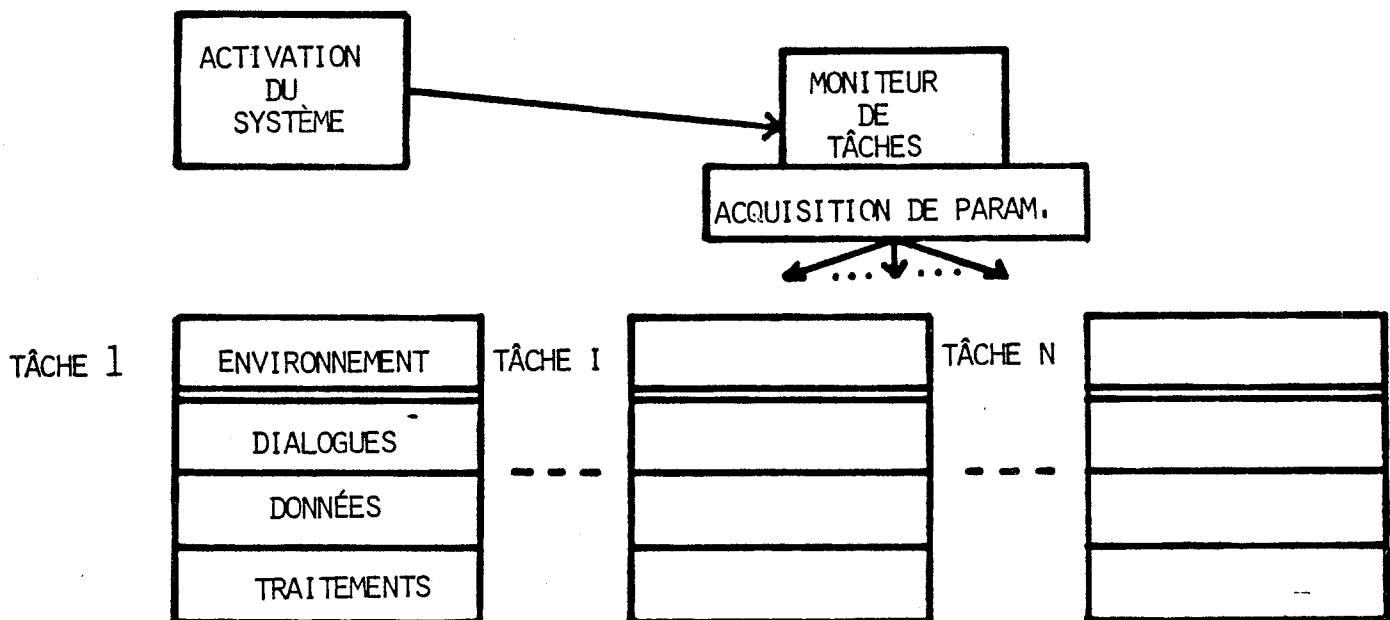


FIGURE II - 7.11 : SCHEMA DE FONCTIONNEMENT DU MONITEUR DE TACHES

Le schéma de la Fig. II - 7.11 montre une organisation à un seul niveau. En effet, nous avons préféré obliger l'utilisateur à finir une tâche avant d'en commencer une autre, et cela pour deux raisons: d'une part pour la clarté du processus de conception - deux tâches qui peuvent s'entremêler n'en constituent en réalité qu'une seule - et d'autre part pour la complexité technique de réalisation dans des contextes (matériels et logiciels) différents.

Un autre aspect dont il faut tenir compte est celui de la réalisation du moniteur de tâches. Deux solutions fondamentalement différentes se distinguent:

- réalisation du moniteur de tâches dans le même environnement et particulièrement à l'aide du même langage de programmation que celui des tâches,
- réalisation du moniteur dans un environnement différent, notamment celui du système.

La première solution s'appuie sur un chargement dynamique transparent pour l'utilisateur. La définition de tâche devient moins précise, car la frontière entre les tâches et le moniteur est floue. Un exemple de ce type de réalisation est le fonctionnement du Système Intégré pour des Ensembles Electroniques et Logiques. Ce système est en cours de mise en oeuvre sur HB68 travaillant sous MULTICS, avec comme langage de programmation unique FORTRAN (Fig. II - 7.12).

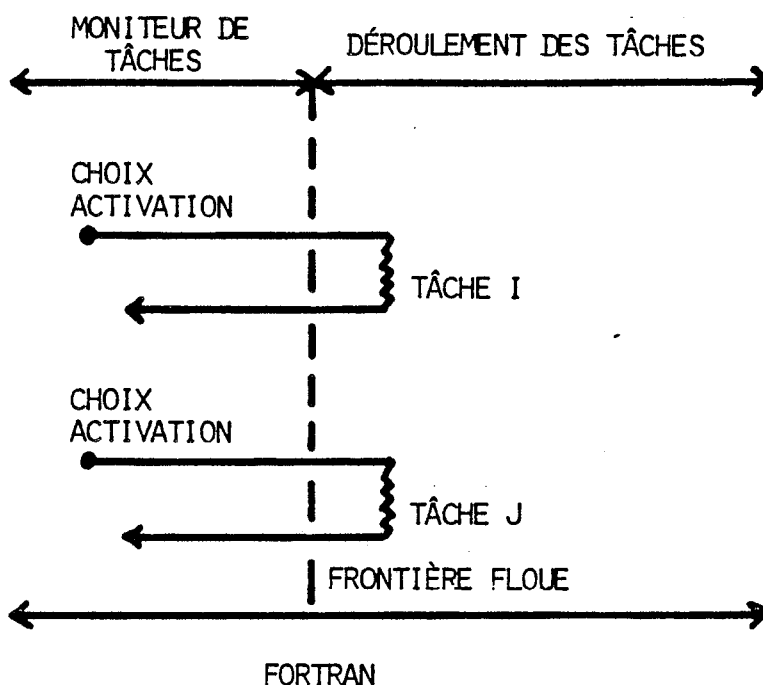


FIGURE II - 7.12 : CONTEXTE MONO LANGAGE

La deuxième solution s'appuie sur un chargement à la demande. La frontière est plus stricte car à la fin de chaque tâche on quitte l'environnement de la tâche pour revenir dans l'environnement du moniteur de tâches. Un exemple de ce type de réalisation est le fonctionnement du système MINI.SIGMA-ARCHI, réalisé sur IBM 360/67 sous système CP/CMS. Dans cette

réalisation les langages de programmation étaient FORTRAN et ALGOLW pour les tâches et EXEC (langage de commande du système CMS) pour le moniteur de tâches (Fig. II - 7.13).

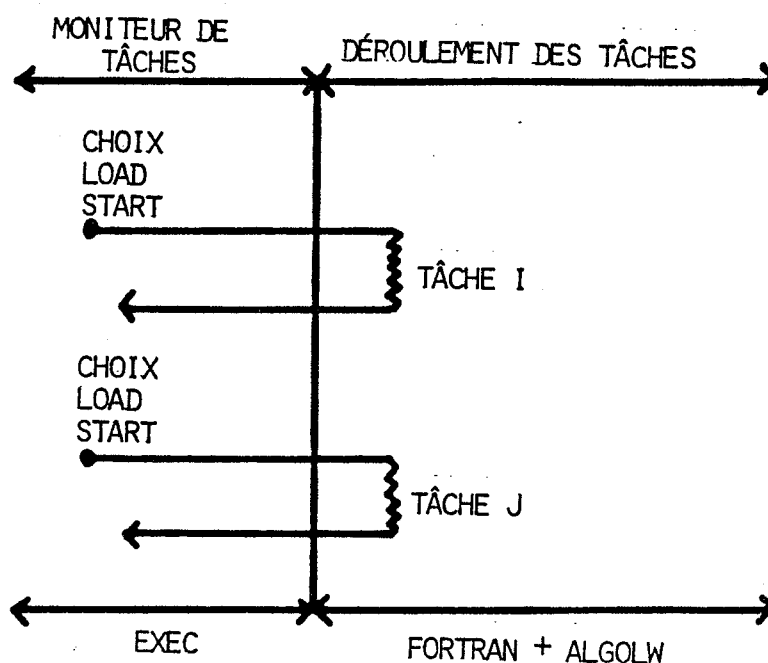


FIGURE II - 7.13 : CONTEXTE MULTI LANGAGE

Notre objectif est de pouvoir supporter ces différentes solutions et de choisir la solution la plus adaptée au contexte particulier.

Nous devons donc prévoir dans le traitement des spécifications la possibilité de générer différentes réalisations selon le contexte choisi. Cela s'effectue grâce à la réalisation différente des fonctions de service (de génération et de mise en oeuvre) utilisées dans le traitement des spécifications.

Il faut d'autre part réaliser les moniteurs de tâches correspondant à chaque type de contexte. Nous verrons des cas concrets dans le chapitre 10.

Ayant décomposé le problème en un ensemble de parties indépendantes, nous pouvons faire face plus facilement à un changement de contexte. Le système devient donc adaptable.

CHAPITRE 8

SPECIFICATION D'UNE APPLICATION CAO

RESUME: Ce chapitre commence par une étude bibliographique des méthodes de spécification et se poursuit par une discussion concernant l'intérêt de la spécification des systèmes CAO. La place d'un langage de spécifications dans la modélisation CAO est ensuite abordée. Après un rappel sur le modèle conceptuel des applications CAO, on définit le langage de spécifications COLA qui réunit les énoncés présentés dans les chapitres précédents. Le traitement de ce langage facilite la documentation, permet l'étude de la cohérence et de la plénitude des spécifications et contribue à la mise en oeuvre du prototype et de la version définitive. Différentes approches pour l'utilisation des spécifications complètent ce chapitre.

8. SPECIFICATION D'UNE APPLICATION CAO

8.1 METHODES DE SPECIFICATION

Les méthodes de spécifications de systèmes ont débuté dans le domaine de la gestion. C'est là que le besoin de la spécification fonctionnelle s'est fait le plus sentir. Nous commençons par une étude des projets de systèmes d'informations.

8.1.1 SPECIFICATION DE SYSTEMES D'INFORMATIONS

Le projet le plus important dans le domaine des systèmes d'information est le projet ISDOS (Information System Design and Optimisation System) (TeH.78), qui a débuté à l'Université de Michigan sous la direction de D. TEICHROEW, en 1968. Ce projet se concrétise par : Program Statement Language (PSL) / Problem Statement Analyzer (PSA).

- PSL est un langage pouvant être traité par machine, permettant d'exprimer sous une forme rigoureuse et uniforme des spécifications fonctionnelles et des performances. PSL contient un ensemble d'énoncés déclaratifs permettant de nommer les objets conceptuels des systèmes à spécifier, de décrire les propriétés de ces objets et de faire apparaître les relations entre elles.
- PSA est un ensemble de programmes qui accepte les énoncés en PSL, les analyse chacun statiquement (correction syntaxique) vérifie la consistance et la complétude de la base de données et génère une documentation variée ainsi que les spécifications du système. Le langage de commande associé à PSA est très simple et orienté vers l'utilisateur.

A Grenoble, le projet MACSI (Méthode d'Aide à la Conception des Systèmes d'Informations) (Pec.75), a étudié ce problème entre 1973 et 1977, sous la direction de F. PECCOUD. Les deux caractéristiques fondamentales de MACSI sont: d'une part une méthode, assistée par ordinateur, d'analyse fonctionnelle d'un projet (dCh.75) et d'autre part la fabrication de prototypes d'applications informatiques de gestion (Gir.77).

C'est le deuxième point qui est particulièrement intéressant et nouveau (par rapport à ISDOS par exemple). La construction de prototype permet à la plupart des critiques relatives à la

conception, de s'exprimer avant la réalisation de la version opérationnelle. On procède donc à une vérification complète des spécifications fonctionnelles, indépendamment des contraintes organiques particulières. Nous reviendrons plus en détail sur la notion de prototype car nous la considérons comme un des points clé de la réussite dans la mise en oeuvre des systèmes CAO.

Un autre projet français a attiré notre attention, il s'agit du projet REMORA(RFR.79), qui se déroule à l'Université de Nancy sous la direction de Madame C. ROLLAND. Dans ce projet, une approche par la conception assistée par ordinateur est exploitée vraiment à fond. On partage véritablement le travail entre l'homme et la machine. A chaque moment, on étudie qui entre l'homme et la machine est le mieux à même de résoudre le problème posé. Nous sommes particulièrement attachés à cette approche de collaboration très étroite entre l'homme et la machine.

8.1.2. SPECIFICATION EN GENIE LOGICIEL

Les méthodes qui spécifient le fonctionnement des systèmes sont déjà relativement nombreuses. Dans leur article, publié en 1977(DaV.77), DAVIS et VICK font état d'une quinzaine de méthodes, mais leurs conclusions ne sont pas tendres pour autant. Finalement, ils limitent leur comparaison aux quatre méthodes suivantes: ISDOS (déjà vu au paragraphe précédent), SADT (Structured Analysis and Design Technique) de SofTech Inc. (SAD.76), HOS (Higher Order Software) de C.S. Draper Lab. (HaZ.76) et SREM (Software Requirements Engineering Methodology) de TRW (BBD.77). Nous reproduisons ici le tableau permettant une comparaison assez détaillée de ces quatre systèmes.

Les principaux points de comparaison sont:

- les aides de vérifications statiques et dynamiques des spécifications,
- la production automatique de la documentation associée,
- la prise en compte (dans le formalisme), d'un changement de niveau de description (description de plus en plus détaillée),
- la capacité de traiter les informations concernant la conduite du projet,
- la possibilité de simuler le système spécifié.

COMPARAISON DE 4 METHODOLOGIES DE SPECIFICATIONS

| | SREM | ISDOS | HOS | SADT | |
|-------------------------------|---|--|---|---|--|
| Attributs généraux | Utilisation dans le cycle de développement | Système/sous-système spécifications | définition de système | définition de système et de sous-systèmes | Définition de systèmes et de sous-systèmes |
| | Principale orientation | Applications militaires | Systèmes d'informat. | Systèmes à interface critique, applications spatiales | Conduite de systèmes informatiques |
| | Degrés d'automatisation | Grand | Grand | Prévu | Manuel |
| | Langage de communication | RSL | PSL | AXES | Anglais |
| | Saisie de spécifications sous la forme traitable en machine | OUI | OUI | Prévu | NO* |
| | Etat d'avancement | marche en évaluation | marche, grand nombre d'utilisations | Conceptualisé, en développement | Utilisé dans un grand nombre de projets |
| Méthodologie | Approche définie pour passer d'un niveau à un autre | OUI | NON | NON | OUI |
| | Support de documentation | automatisé, format flexible | automatisé, format fixe | pas encore | Bibliothecair manuel * |
| | Aspects de conduite | définis, étapes mesurables, supportés par le langage | supportés par le langage | pas encore | Un ensemble de procédures d'examens |
| | Aides à l'analyse . Vérification de consistance et de perfection. . Support de simulation | Analyse automatique OUI | Rapports automatiques pour l'analyse NON | pas encore NON | manuel * Manuel |
| description de spécifications | Mécanismes formels de structuration | Réseaux de spécification (R-Nets) | Arbres ou réseaux de fonctions | Arbre de processus | SADT diagrammes structurés |
| | Forme de spécification fonctionnelle | Description du flux logique | Interactions fonctionnelles | Abstraction des données et structure de contrôle | graphes de données et des activités avec texte |
| caractéristique du langage | Supporte la description de : | | | | Description en anglais via SADT |
| | . Structure | OUI | OUI | OUI | . diagrams |
| | . Données | OUI | OUI | OUI | . textes |
| | . Spécifications de performances | OUI | NON | NON | . glossaire |
| | . Modèles de simulation | OUI | NON | NON | |
| . Traçabilité | OUI | OUI | OUI | | |

* Support ISDOS pour SADT prévu.

FIGURE II - 8.1 : COMPARAISON DE QUATRE METHODOLOGIES DE SPECIFICATIONS

A la NCC 78 (NCC.78), une session a été consacrée à l'expérience des utilisateurs concernant ces nouvelles méthodes de construction de logiciels. On a pu mesurer la pratique des utilisateurs industriels avec ISDOS, SADT et d'autres méthodes souvent très directement couplées aux méthodes de production de logiciels comme par exemple Structured Design (SMC.74).

En France, le travail récent de J.R. ABRIAL (Abr.78) entre partiellement dans ce domaine. Dans sa démarche à contre-pied des autres méthodes utilisées dont le mérite est jugé surtout par rapport à la capacité d'assister par ordinateur l'ensemble des travaux d'analyse. En Z, le langage de spécifications ZO est un langage non interprétable par la machine: les spécifications et les vérifications de cohérence se font en interaction de l'homme avec le papier sur lequel il a écrit les spécifications en ZO. A l'aide de Z1, on peut produire une description plus détaillée contenant déjà les notions algorithmiques. De cette façon, Z permet un raffinement des spécifications.

8.1.3. CONCLUSIONS

En conclusion de cette étude bibliographique, nous pouvons souligner:

- l'intérêt très grand d'un support de spécifications (langage d'expression) traitable par la machine, c'est à dire associant la machine au processus de spécifications. Cela permet de produire la documentation et d'analyser la cohérence et la complétude de façon automatique.
- l'intérêt d'une vérification dynamique des spécifications, soit sous forme de simulation, soit sous forme de prototype. Ceci afin mieux saisir les spécifications pour les raffiner,
- l'intérêt d'une approche intégrée permettant le raffinement des spécifications, et le passage naturel à la mise en oeuvre provisoire ou définitive, fondée sur des méthodes rigoureuses de programmation (manuelle ou automatique).

Dans ce qui suit, nous présentons notre méthode de spécifications des systèmes CAO. Nous nous sommes évidemment inspirés des divers concepts présentés dans les systèmes que nous venons d'examiner, et nous les avons complétés par des concepts spécifiques de la Conception Assistée par Ordinateur.

8.2 SPECIFICATION D'UN SYSTEME C.A.O.

8.2.1 INTERET ET PRINCIPES

Dans ce chapitre, nous nous intéressons aux dialogues homme-machine lors de la définition et la mise en place d'un système CAO. C'est le dialogue du concepteur de systèmes CAO avec la machine.

L'idée de base est très simple et découle directement de la démarche pour la définition et la mise en oeuvre des langages de commande: au lieu d'exprimer tout de suite une description détaillée afin de réaliser un programme particulier, on exprime d'abord les objectifs à atteindre sans se soucier de la réalisation.

L'intérêt de cette démarche est de plus en plus grand et en rapport direct avec la diversification des systèmes CAO et leur domaines d'application.

L'approche par spécification est également utilisée dans les applications scientifiques ou de gestion.

La mise en place de programmes informatiques n'est pas un travail simple, c'est pourquoi on a vu naître un thème de recherche très important portant le nom de génie logiciel (software engineering), dont la finalité est de pouvoir définir, produire et maintenir les logiciels de façon systématique, en s'appuyant sur des procédures industrielles et non plus sur des activités artisanales. Il nous semble que l'article (RGI.75) présente assez complètement les principaux concepts. Les études et les réalisations essaient d'apporter soit des solutions pour diverses tâches du processus d'élaboration de logiciels comme par exemple la programmation structurée, les constructions modulaires de systèmes, les nouveaux langages de programmation, les langages de spécifications, les techniques de programmation automatique, etc ..., soit des méthodes globales réunissant les solutions précédemment évoquées pour bâtir des processus complets de spécification, de mise en oeuvre et de maintenance de logiciels.

Ces dernières tentatives sont beaucoup moins nombreuses et on peut considérer essentiellement deux types d'approche:

- la première est une démarche manuelle éventuellement assistée par ordinateur; spécifications, découpage en modules, programmation structurée, ...
- la seconde est fondée sur les techniques de l'intelligence artificielle: les programmes sont produits automatiquement à partir des spécifications.

Il semble intéressant de remarquer que les problèmes de génie logiciel ont aussi été abordés dans divers domaines de l'informatique, comme par exemple les Bases de Données et la CAO et des solutions intéressantes ont été élaborées. En ce qui concerne les bases de données le projet le plus important est sans aucun doute le projet ISDOS de l'Université de Michigan. En France nous constatons une activité importante dans ce domaine. Les deux projets les plus significatifs sont MACSI et REMORA.

En CAO on a vu apparaître le terme SYSTEME GENERAL pour la CAO il y a déjà plus de dix ans (ICES (Roo.67), GERMINAL (CER.74),...). Ce terme exprime la volonté de dégager les concepts et les fonctions de base d'un système CAO et de fournir une méthode et des outils de haut niveau permettant de bâtir une application sur ce système général.

Néanmoins on constate une certaine stagnation au niveau de cette démarche: on a l'impression que les nouveaux outils, beaucoup plus orientés vers la sémantique des problèmes, n'ont pas été pris en compte.

Il semble que l'analyse du problème soit encore effectuée de façon empirique. Le rôle du langage de spécifications est justement de faciliter et de systématiser cette phase du processus. Il fournit en effet un support formel des spécifications du processus de conception dans un domaine d'application, c'est à dire l'univers de conception, les caractéristiques de l'objet à concevoir, les stades de la conception, les états de l'objet pendant le processus, les divers traitements sur l'objet, etc ... Le langage de spécifications est un outil d'expression d'un modèle conceptuel de cet univers.

8.2.2 PLACE DU LANGAGE DE SPECIFICATIONS DANS LE SCHEMA CONCEPTUEL D'UN SYSTEME CAO

En reprenant le schéma figure II - 3.3 du chapitre 3 nous pouvons situer la place du langage de spécifications dans le schéma de modélisation de la CAO. Le langage de spécifications est un outil d'expression d'un modèle conceptuel concret basé sur le support des modèles conceptuels. Sa place se situe donc entre le support des modèles conceptuels et le modèle conceptuel concret, spécifique à une famille d'applications (Fig. II - 8.2).

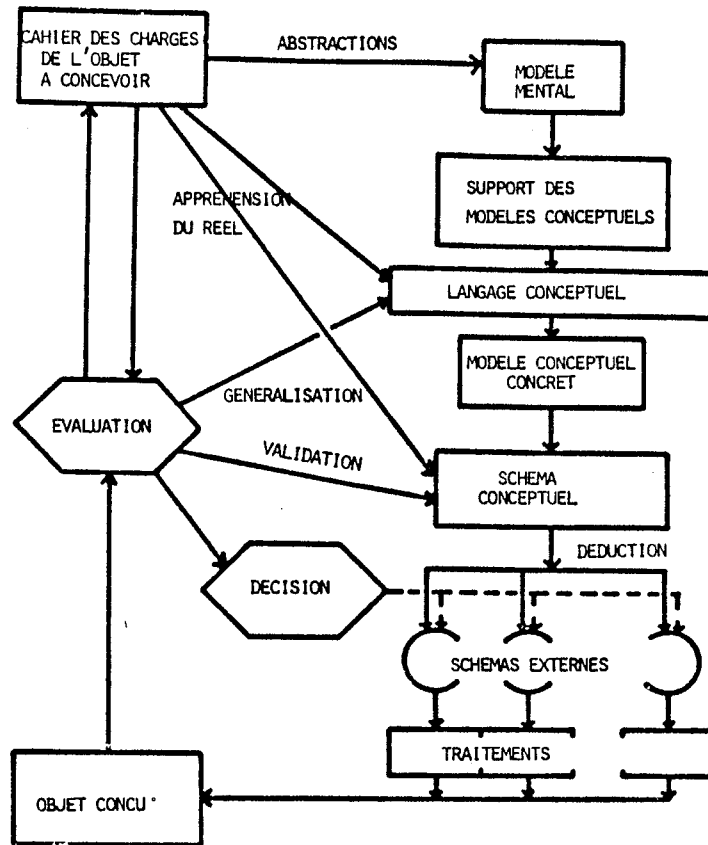


FIGURE II - 8.2 : PLACE DU LANGAGE DE SPECIFICATIONS DANS LA MODELISATION CAO

8.3 COLA - UN LANGAGE DE SPECIFICATION FONCTIONNELLE D'APPLICATIONS CAO

Nous avons défini et mis en oeuvre un langage de spécifications: COLA (COnceptual LAnguage). Pour cela nous avons dégagé un modèle conceptuel de l'univers CAO, nous avons défini un langage et mis en oeuvre l'ensemble des outils travaillant sur la description exprimée à l'aide de ce langage. Nous présentons brièvement ces trois aspects.

8.3.1 MODELE CONCEPTUEL DES APPLICATIONS CAO

Ce modèle conceptuel fournit un cadre de description de l'ensemble des informations qui caractérisent un système CAO.

Nous ne présentons pas ici complètement ce modèle mais nous décrivons brièvement toutes les informations qu'il renferme. Nous avons présenté et discuté tous ces aspects dans les chapitres précédents et notamment dans les chapitres 4, 5, 6 et 7. Nous récapitulons seulement le contenu de ce modèle. (Nous donnons à la fin de chaque paragraphe la référence du chapitre contenant la discussion détaillée.)

- l'organisation de la structure d'exécution dans laquelle le processus de conception prend place. Il s'agit de spécifier le fonctionnement et la répartition des responsabilités, c'est à dire la distribution du travail (qui fait quoi), et des responsabilités (qui décide quoi), (voir chapitre 4).
- l'organisation du processus de conception. Il s'agit de spécifier le déroulement du processus de conception, les étapes et phases de ce processus, (aspect statique), et l'enchaînement de ces différentes étapes et phases (aspect dynamique), (voir chapitre 4).
- l'organisation du processus de décision. Il s'agit de spécifier comment la définition de l'objet-à-concevoir, caractérisé par les objectifs et les contraintes de conception, évolue pendant le processus de conception, la façon dont s'effectue les vérifications de la satisfaction de ces objectifs et contraintes, et ce que l'on fait dans le cas des impossibilités (voir chapitre 4).
- l'objet en cours de conception. Il s'agit de définir comment l'objet conçu est matérialisé pendant la conception, comment on le décrit, et comment cette description évolue (voir chapitre 5).
- les traitements pouvant intervenir dans le processus de conception. Ils peuvent soit effectuer automatiquement certaines tâches, soit aider à l'accomplissement d'autres tâches dans un contexte interactif. Dans tous les cas, il est indispensable de préciser la fonction du traitement, ses connexions avec les autres parties du système (notamment les entrées et les sorties des données) et l'environnement dans lequel le traitement peut s'insérer (phase et étape du processus, responsabilité, etc...), (voir chapitre 3).
- l'acquis technique: il décrit le contexte général de l'étude. Il s'agit de représenter l'expérience acquise dans le domaine considéré, les résultats obtenus, les solutions préférentielles, les échecs rencontrés et analysés, etc... Tout ceci afin de faciliter le travail actuel, l'insérer dans un contexte plus général et d'introduire un style de conception et de production (Sim.75), (voir chapitre 5).

8.3.2 LANGAGE DE SPECIFICATIONS COLA

COLA est le langage de spécifications fondé sur le modèle conceptuel que nous avons évoqué dans le paragraphe précédent.

Du point de vue lexicographique, COLA admet les caractères suivants:

- LETTRES : a b c x y z

- CHIFFRES: 0 1 2 3 4 5 6 7 8 9
- CARACTERES SPECIAUX: * ? + - () < = > , ; . ! ? % & " ' "

Les unités lexicographiques sont les suivantes:

- COMMENTAIRE: un texte quelconque entre << et >>
- MOT-CLE: une suite de lettres , chiffres et caractère "_", commençant par une lettre, les 8 premiers caractères sont significatifs
- CONSTANTE: entière, réelle (fixe ou flottante), caractère et chaîne de caractères (entre deux caractères " ")
- IDENTIFICATEUR: une suite de lettres, de chiffres et de caractère "_", commençant par une lettre, les 8 premiers caractères sont significatifs.

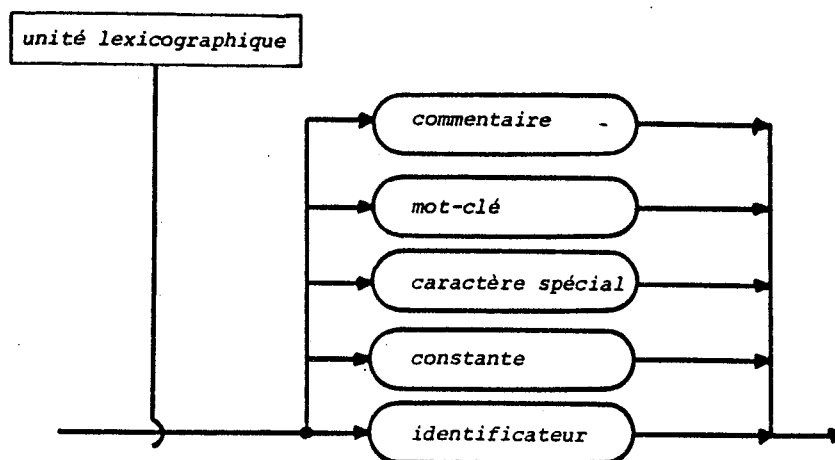


FIGURE II - 8.3 : STRUCTURE LEXICOGRAPHIQUE DE COLA

Du point de vue syntaxique, le langage est composé d'une suite d'énoncés. Afin de rendre l'apprentissage du langage plus aisé et pour en faciliter le traitement, nous avons adopté une structure unique pour tous les énoncés (FIG.II - 8.4).

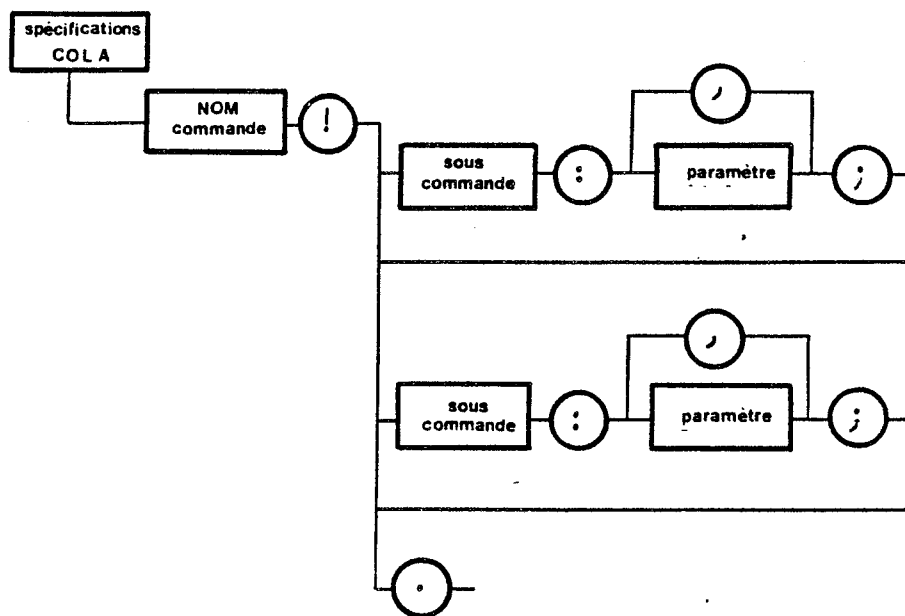


FIGURE II - 8.4 : STRUCTURE SYNTAXIQUE DE COLA

Un nom est à choisir parmi onze noms identifiant chacun un type d'énoncé. Puis les noms du deuxième niveau fournissent une distinction plus fine dans chaque énoncé.

Une description en COLA est une suite d'énoncés commençant obligatoirement par l'énoncé PROJET et se terminant par le mot-clé FIN_DES (fin de la description).

Cet énoncé PROJET permet de spécifier la nature de la description. C'est donc un moyen de gestion des spécifications. Nous distinguons trois cas:

- la CREATION, avec les qualificatifs nom de projet (un identificateur), version (chaîne de caractères), et la date facultative (une chaîne de caractères),
- la MISE A JOUR, à partir des spécifications déjà enregistrées, dont la désignation se trouve après le mot-clé DEPUIS,
- l'ADJONCTION, avec ou sans copie selon que la source et la cible sont différentes ou pas (présence ou absence de DEPUIS).

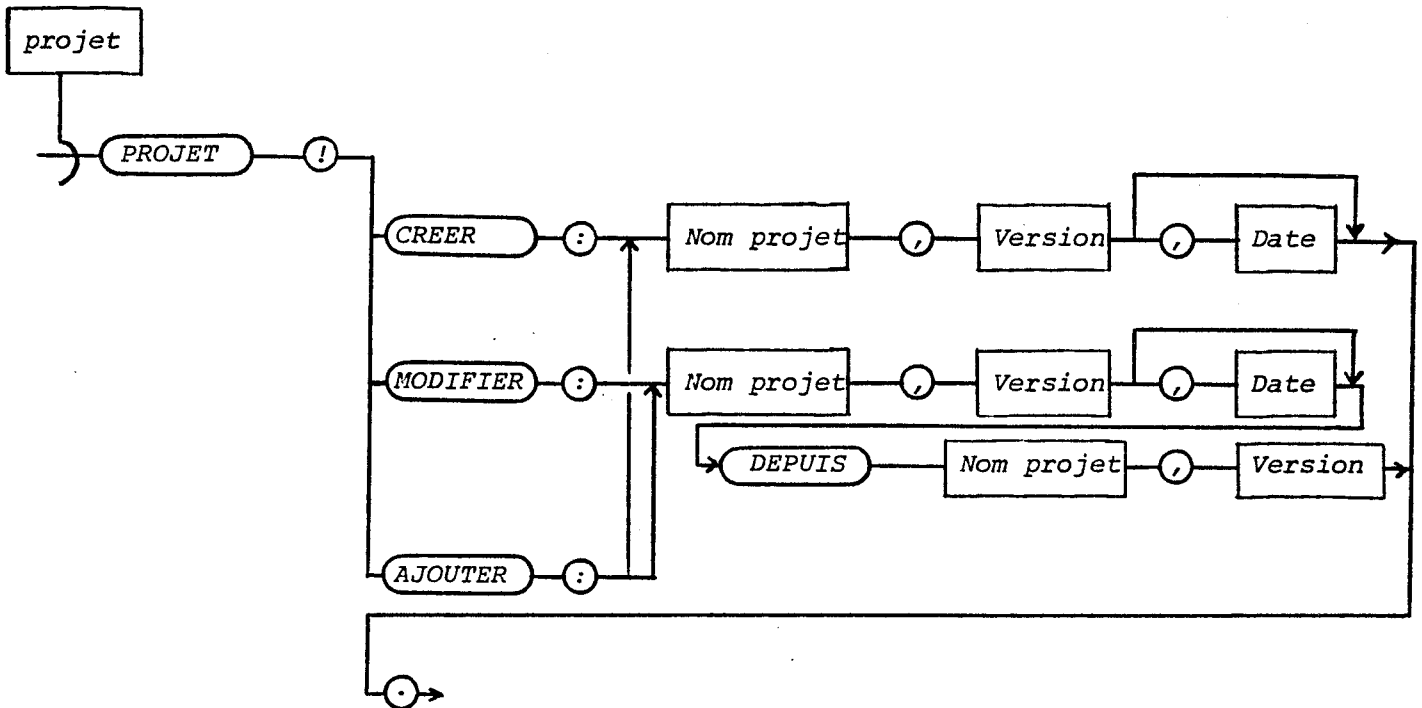


FIGURE II - 8.5 : COLA - ENONCE PROJET

Nous rappelons tous les types d'énoncés: pour chacun d'entre eux nous décrivons brièvement son but sans donner sa syntaxe. Celle-ci se trouve présentée dans les chapitres précédents, nous donnons ici uniquement la référence de la figure où la syntaxe a été décrite.

PROJET - (Fig. II - 8.5) - cet énoncé fournit les indications sur la description: nom du projet, version, création, mise à jour, complément, etc ...

ESPACE DE CONCEPTION - (Fig. II - 4.15) - cet énoncé permet de définir l'espace de conception: définitions, phases, étapes, résolutions et structurations autorisées.

PROCESSUS DE CONCEPTION - (Fig. II - 4.16) - cet énoncé permet de décrire l'ensemble des possibilités du déroulement d'un processus de conception: cheminement dans l'espace de conception.

RESPONSABILITE - (Fig. II - 4.17) - permet de définir les responsabilités et les fonctions des différents participants au processus de conception.

DEFINITION DE PROPRIETES - (Fig. II - 5.10a) - permet de définir les types de propriétés propres et de relations qui seront utilisées dans le projet ainsi que les opérations que l'on pourra effectuer.

DEFINITION D'OBJET - (Fig. II - 5.10b) - permet de définir les types d'objet (objet conçu et objet à concevoir) utilisés pendant le processus ainsi que les opérations possibles.

DEPENDANCE DES PROPRIETES - (Fig. II - 5.10c) - exprime les dépendances de propriétés du même objet et assure donc un type de cohérence.

CORRESPONDANCE DES PROPRIETES - (Fig. II - 5.10d) - exprime les liens verticaux entre les mêmes propriétés des différents constituants de l'objet et fournit donc un moyen pour assurer la cohérence sémantique lors de la manipulation de la structure hiérarchique de l'objet

DEFINITION DES TRANSFORMATIONS - (Fig. II - 5.10e) - exprime le changement de propriétés pendant les transitions dans l'espace de conception.

TRANSFORMATION DE L'OBJET A CONCEVOIR - (Fig. II - 5.10f) - exprime la façon dont se raffine la définition de l'objet à concevoir pendant le processus de conception.

DEFINITION DE TRAITEMENT - (Fig. II - 7.5) - spécifie le contenu de l'algorithme, principalement en deux parties: les dialogues à l'aide du LDC et les fonctions sémantiques.

8.3.3 TRAITEMENT DU LANGAGE DE SPECIFICATIONS

Après avoir présenté la façon d'exprimer les spécifications d'une application en COLA, il paraît important de montrer à quoi une telle description peut servir et ainsi, justifier l'intérêt de cette méthode de spécifications.

Examinons l'ensemble des traitements que l'on peut effectuer sur la description.

Le premier type de traitement est un TRAITEMENT DOCUMENTAIRE. Il est en effet possible et souhaitable de produire une documentation complète et bien structurée à partir de la description fragmentaire, partielle et évolutive en COLA. De

cette façon, on peut avoir une vision globale de l'état des spécifications.

Grâce à cette documentation, l'utilisateur peut chercher dans quelle mesure ces spécifications sont cohérentes et complètes. C'est un travail qui peut devenir assez long et fastidieux, surtout pour les applications complexes. A partir de l'ensemble des informations stockées, il est possible d'effectuer cette étude de cohérence et de complétude des spécifications, qui peut être entreprise par un traitement automatique.

Le deuxième type de traitement est donc l'étude de COHERENCE et de PLENITUDE des spécifications. Comme résultat, on fournit la liste des ambiguïtés et des lacunes des spécifications.

Il semble utile de disposer d'un système interactif de modifications des spécifications afin de raccourcir sensiblement la boucle: écriture en COLA, traduction, étude de cohérence, modifications, et de nouvelles vérifications, etc..

A partir des spécifications complètes et d'un niveau suffisant (notamment pour les traitements), on peut se poser le problème de la mise en oeuvre effective.

Deux approches sont à étudier pour cette mise en oeuvre: soit on estime avoir obtenu des spécifications vraiment significatives, et on pense pouvoir entreprendre directement la mise en oeuvre définitive sur une configuration organique finale (existante ou à définir), soit il paraît plus sûr de construire un prototype sur une configuration intermédiaire pour s'assurer en grande nature de la justesse des spécifications.

Souvent, on choisit l'approche par prototype, surtout pour les applications complexes dans un environnement complexe. Dans les deux cas, il est possible d'assister ce processus de mise en oeuvre par un ensemble de traitements automatiques. Nous discuterons ce point en détail dans le chapitre 11.

Le troisième type de traitements est donc un ensemble d'outils permettant une PRODUCTION rapide et assistée par ordinateur, de PROTOTYPE du système CAO orienté vers l'application spécifiée. A partir des spécifications, nous pouvons structurer automatiquement la base de données, générer les méthodes d'accès à cette base. Les dialogues utilisateur peuvent être aussi construits à partir des spécifications, en utilisant la méthode LDC (chapitre 6).

Le problème des traitements est un peu plus délicat. Selon leur niveau de spécification et leur nature, il est possible soit de les générer automatiquement, soit de les remplacer par des modules standards de la bibliothèque, soit encore de les réaliser manuellement.

Le quatrième type de traitement est un ensemble d'outils d'assistance à la MISE EN OEUVRE DEFINITIVE d'un système CAO. Il s'agit, selon le cas, soit de déterminer une structure organique correspondant le mieux aux spécifications de l'application, soit d'étudier comment mettre en oeuvre l'application sur une structure imposée. Pour cela, il faut étudier les spécifications pour trouver une bonne organisation des données selon leur importance et leur variété, la méthode de stockage et d'activation des traitements, etc... Les problèmes de portabilité, de connexion avec les systèmes d'exploitation, etc... sont traités ici. Parmi les problèmes abordés, il ne faut pas oublier les études sur la répartition (distribution) d'un système CAO, et sur l'utilisation de processeurs spécialisés (processeur fichier par exemple) dans de tels systèmes. Cet aspect de projection du système spécifié vers une configuration cible sera détaillé plus loin (chapitre 13).

En conclusion, soulignons que l'utilisation d'un langage de spécifications fonctionnelles semble revêtir un intérêt considérable car il facilite la spécification d'un système CAO en la rendant systématique et complète.

Il permet une vérification poussée des spécifications et assure une documentation automatique du projet. De plus, il fournit des résultats qui peuvent être directement utilisés dans le processus de mise en oeuvre, lui aussi automatisable.

8.4 DIFFERENTES APPROCHES POUR L'UTILISATION DES SPECIFICATIONS

Spécifier une application CAO n'est pas une tâche facile, surtout si on s'attaque au processus Conception - Etude - Développement dans son ensemble, ce qui est et semble être de plus en plus demandé.

Dans ce cas, saisir en même temps tous les aspects du problème tels que nous les avons structuré dans le modèle conceptuel pour la CAO, et bâtir un modèle conceptuel concret ne semble pas évident. Et cela d'autant plus que cette étude peut introduire, et le plus souvent introduit, des changements assez profonds dans l'organisation utilisée jusqu'à présent.

C'est pourquoi il est important d'organiser la démarche de spécification en essayant de porter l'intérêt sur un aspect du problème à la fois. Il ne faut pas négliger les discussions libres souvent créatives, mais il est indispensable de structurer le travail pour pouvoir aboutir à un résultat satisfaisant dans un délai raisonnable.

C'est pourquoi, nous avons envisagé quatre approches différentes pour la spécification, orientées chacune vers un des

pôles fondamentaux des applications CAO. Il s'agit d'étudier l'application du point de vue:

- du processus CED,
- des données utilisées et manipulées,
- des traitements auxquels on peut faire appel,
- des intervenants dans l'application.

Ces différents aspects n'ont pas la même importance dans chaque application. Pour certaines les intervenants, par exemple, peuvent se limiter à une seule personne, pour d'autres, le processus CED est très rudimentaire. Toutefois il semble indispensable d'aborder ces différents aspects, au moins une fois, pour pouvoir assurer que les spécifications obtenues sont significatives.

Pour des applications complexes (la complexité peut être très variée), il est indispensable de traiter ces différents aspects plusieurs fois afin de stabiliser les spécifications. Ceci est notamment vrai quand l'introduction de la CAO entraîne des modifications importantes dans l'organisation de l'entreprise.

8.4.1 PAR LE PROCESSUS

Le premier aspect important de l'application, qui nécessite un intérêt tout particulier, est l'étude de la structure du processus CED selon lequel l'application doit se dérouler.

Pour le dégager il faut s'inspirer du processus actuel et lui apporter les modifications que l'introduction de la CAO et d'autres changements de structure provoquent. On définit donc les différents états (ou environnements) du processus ainsi que le cheminement qui les relie.

De façon accessoire il est possible de noter tous les autres aspects de l'application, mais le processus CED doit rester le fil conducteur de l'étude.

8.4.2 PAR LES DONNEES

Le deuxième aspect important de l'application est celui des données intervenant dans l'application. Les données sont ici prises au sens large, il s'agit aussi bien des informations que des produits.

Ce qui nous intéresse c'est d'inventorier toutes les données de l'application, afin de les classer pour dégager les atomes d'informations et les types de structures qui les lient. Il s'agit de définir le comportement structurel et sémantique de ces

données ainsi que leur évolution pendant le déroulement du processus CED.

Cette démarche par les données fournit, non seulement un inventaire des données mais aussi une certaine vision du processus CED. Ces résultats doivent donc être confrontés avec ceux obtenus par la démarche basée sur le processus.

8.4.3 PAR LES TACHES

Le troisième aspect de l'application est celui des tâches que l'on entreprend dans l'application. Il s'agit de formuler la liste des traitements que l'on effectue dans l'application. Quand on dresse cette liste, il est important de spécifier pour chaque tâche l'ensemble des informations sur lesquelles elle travaille. De cette façon il est possible de retrouver d'une manière indirecte, les données sur lesquelles l'application s'appuie, ainsi que le processus CED selon lequel ces tâches s'enchaînent. On reçoit ainsi une autre vision de l'application.

8.4.4 PAR LES UTILISATEURS

Le dernier aspect dont il faut tenir compte pour les applications dont le processus CED fait intervenir des personnes variées, de compétences différenciées, est celui des utilisateurs. Il est important de suivre aussi ce fil conducteur. En effet, souvent dans ce cas, c'est le seul qui permette de reconstruire à partir des visions fragmentaires de chaque participant, une vision globale de l'application.

Chaque participant doit donc fournir sa vision du processus CED, c'est à dire son champ d'intervention, les données qu'il manipule et les tâches qu'il doit entreprendre. A partir de ces informations de tous les participants, on obtient une spécification globale de l'application.

Finalement, on peut constater que c'est la concordance de ces quatre aspects de l'application qui assure la cohérence des spécifications. Le langage COLA doit donc permettre non seulement la rédaction finale des spécifications, mais doit surtout servir d'outil dans l'élaboration de ces spécifications

Ceci nous a guidé dans le choix des possibilités de description incomplète et progressive de ces spécifications et dans leur traitement, notamment en ce qui concerne la détection systématique et détaillée des incohérences.

8.5 EXEMPLE DE SPECIFICATIONS

A titre d'exemple, nous allons spécifier ici une partie du système SIGMA-ARCHI. Il s'agit d'un système CAO pour l'architecture, qui permet de prendre en compte un processus de conception architecturale. Nous allons nous limiter à quelques notions, une présentation détaillée et une spécification plus complète seront faites dans l'annexe 1.

```

projet ! creer : sigma_archi, '16/12/31'.
espace ! definition : programme_archi, spatiale, habillage, d_standard;
    etape : formulation, organisation, modularisation, allocation,
    instrumentation, dossier, e_standard;
    resolution : avant_p, projets, r_standard;
    structuration : s_standard .

processus ! etats : proj_archi_ap = programme_archi et formulation
    et avant_p et s_standard,
    tout = j_standard et e_standard et r_standard et s_standard,
    proj_archi_proj = programme_archi et formulation et projets et standard,
    allocation_spa = programme_archi et allocation et
    avant_p et s_standard;
    transition : repartition_spa = proj_archi_ap en allocation_spa .

responsabilite ! decide : architecte (tout),
    client(tout).

propriete ! propre : surface = numerique avec
    lecture (lec_surf) ecriture (ecr_surf)
    utilisation etats(tout)
        fonction(architecte),
    contour = tdf avec
    lecture (lec_ctour, surf_ctour, nb_cote_ctour)
    ecriture (ecr_ctour)
    utilisation etats(allocation_spa)
        fonction(architecte),
    schema_plan = tdf avec
    lecture (lec_sch, surf_sch, nb_uc_sch)
    ecriture (ecr_sch)
    utilisation etats (tout);
relation: fonc = entier avec
    lecture (lec_fonc) ecriture (ecr_fonc)
    utilisation etats(proj_archi_ap, proj_archi_proj,
        allocation_spa) fonction(architecte),
    resultante = entier avec
    lecture(lec_res, degre_res) ecriture (ecr_res)
    utilisation etats (allocation_spa) fonction(architecte).

```

```

objet ! conçu : piece = propriete propre (surface, contour)
                    propriete relation (fonc, resultante)
                    avec lecture (lsurf_piece, lcout_piece, lec_fonc,
                                lec_result)
                    ecriture (esurf_piece, ectour_piece, ecr_fonc,
                                ecr_result),
                    appart = propriete propre( surface, app_ctour)
                    avec vertical dessus (appart)
                    vertical dessous (appart, piece);
a concevoir : bat = propriete propre( cout, nb_piece, nb_appart)
               propriete relation ( rel_eau, rel_hauteur)
               avec lecture (lcout, lnb_piece, lnb_appart).

dependance ! propriete propre : depsc (surface, contour) fleched 'fdepsc'.

correspondance ! syntetique: corrsur (surface de piece, surface de appart).

transformations ! etats : tr_spa allocation_spa fleched proj_archi_proj
                    contour devient contour_detail
                    par 'proj_change_rep_spatiale'.

traitement ! production : spa avec
                    lecture (lcout_piece, lec_result)
                    ecriture (ecr_sca)
                    utilisation etats(allocation_spa)
                    fonction (architecte)
                    algorithme psa 'allocation spatiale avec contour de
pieces et une seule relation a satisfaire' .
eof

```

FIGURE II - 8.6 : SPECIFICATIONS DE SIGMA-ARCHI EN COLA

CHAPITRE 9

MACAO : MACHINE ABSTRAITE POUR LA CAO

RESUME: Ce chapitre aborde le troisième invariant, l'invariant informatique. L'introduction justifie la notion de machine abstraite (machine fonctionnelle) et le passage à une réalisation concrète. Le contenu de la machine abstraite est détaillé: base de données, base d'algorithmes, moniteur de conception, moniteur de communication. L'accent est porté sur le moniteur de communication, englobant la notion d'unités physiques ou logiques, la gestion du poste de travail et le cheminement des informations. Ce chapitre se termine par la présentation des schémas fonctionnels de tous les composants de la machine abstraite.

9. MACAO : MACHINE ABSTRAITE POUR LA CAO

9.1 MACHINE ABSTRAITE: DEFINITION ET JUSTIFICATION

Après avoir déterminé dans les chapitres précédents tout ce qu'un système CAO doit prendre en compte et surtout tout ce que le futur utilisateur doit fournir comme informations nécessaires au système souhaité, nous allons maintenant étudier la façon dont ces informations sont utilisées et traitées par la machine.

Nous nous mettons donc du côté du constructeur de système CAO orienté vers une application (ou vers un domaine d'application) et nous allons étudier comment se structure l'environnement dans lequel ce constructeur travaille.

Pour des questions d'efficacité il est important de dépasser le stade du travail à façon, où de manière tout à fait artisanale le constructeur bâtit de toutes pièces le système demandé.

Il est donc indispensable de dégager les différentes fonctions qu'un système CAO doit remplir et d'estimer la généralité de ces différentes fonctions par une étude comparative des divers systèmes CAO réalisés pour des applications variées. On en déduit une liste de fonctions et leur niveau de généralité. Ces fonctions peuvent être réalisées une fois pour toutes.

Lors de la mise en place d'un nouveau système on choisit d'abord le niveau sur lequel on veut s'appuyer; les concepteurs ne réalisent que les fonctions non encore élaborées.

Cette démarche est une démarche classique de conception par affinement successif: de nombreux développements, notamment en génie logiciel, lui ont été consacrés. Il est important de remarquer qu'elle permet une indépendance ou mieux une dépendance clairement définie entre les différents niveaux.

L'expression purement fonctionnelle laisse le choix des mécanismes assurant la mise en oeuvre de ces différentes fonctions.

Ce n'est pas seulement la définition vers le haut qui est intéressante mais également celle vers le bas, car elle permet (par le choix judicieux de mécanismes sous-jacents) de réaliser les fonctions de façon appropriée. De cette façon, la

portabilité et surtout l'adaptabilité du système aux différents contextes d'utilisation sont assurées.

On procède à la construction par assemblage de morceaux (modules) pré-existants et on ne programme que les parties spécifiques à l'application; le problème étant alors de déterminer les différentes fonctions à assurer pour constituer des niveaux homogènes.

Chaque niveau correspond à une MACHINE ABSTRAITE pour laquelle on a défini les comportements:

- vers le haut, c'est à dire comment elle peut être utilisée
- vers le bas, c'est à dire ce qu'elle doit réaliser et sur quoi elle s'appuie.

De cette façon la structure du système reflète uniquement un découpage fonctionnel qui se veut indépendant de toute contrainte du contexte de réalisation. Nous montrons par la suite comment nous pouvons prendre en compte l'environnement effectif.

Le problème fondamental paraît être celui du choix des niveaux donc des machines abstraites. Il est indispensable de dégager nettement les niveaux et leurs frontières et de spécifier les liens verticaux entre ces différents niveaux.

Il semble important de minimiser le nombre de "points d'appui", c'est à dire de liens verticaux entre niveaux.

9.2 PASSAGE A UNE REALISATION CONCRETE

Le passage à une réalisation concrète s'effectue de la façon suivante: à partir de la définition fonctionnelle de l'application et la description des choix des composants du support, on cherche dans la bibliothèque de modules les occurrences les mieux adaptées au contexte et on compose un système satisfaisant au mieux les contraintes fonctionnelles et de réalisation.

Pour pouvoir effectuer cette démarche nous avons élaboré une gamme de modules interchangeables réalisant les fonctions de la machine abstraite dans des contextes structurels (organiques) de matériel ou de logiciel différents.

Nous présentons quelques exemples dans le chapitre 12 et nous étudions le processus de mise en oeuvre dans le chapitre 13.

9.3 CONTENU DE LA MACHINE ABSTRAITE POUR LA CAO (MACAO)

L'étude de l'ensemble des informations nécessaires à un système CAO et la façon dont elles sont traitées nous a permis de dégager le contenu de la machine abstraite: nous avons mis en évidence un niveau cohérent de fonctions que n'importe quel système CAO doit assurer. Ce niveau constitue une machine abstraite la plus macroscopique et comprend (fig. II - 9.1):

- la base de données
- la base d'algorithmes
- le moniteur de conception
- le moniteur de communication

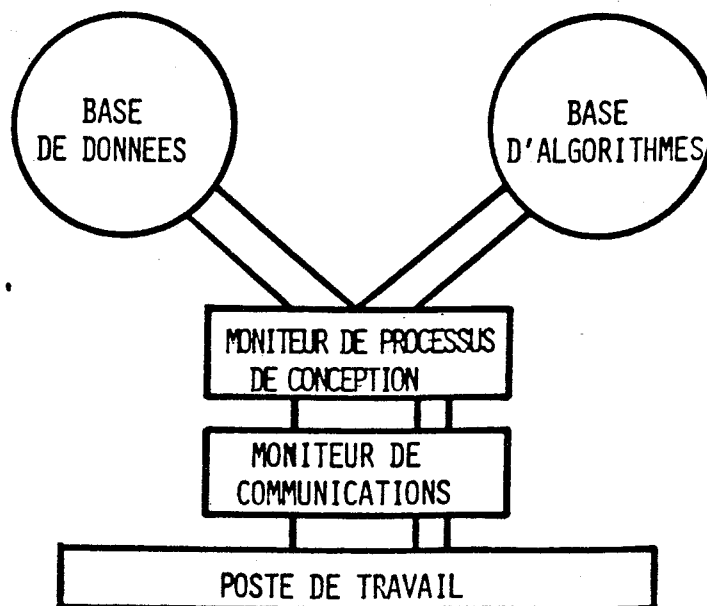


FIGURE II - 9.1 : MACAO : MACHINE ABSTRAITE POUR LA CAO

La Machine Abstraite pour la CAO, MACAO, assure l'association homme machine. Le processus de conception s'effectue à partir du poste de travail approprié. La machine prend en compte les données du problème et permet leur évolution pendant les différentes phases de conception. Elle fournit au concepteur les outils pour manipuler ces données.

La BASE D'ALGORITHMES contient les algorithmes que nous avons classés de différentes façons dans le chapitre 7.

La BASE DE DONNEES contient les données de deux types:

- données permanentes (connaissances)
- données temporaires (description d'un projet)

Le MONITEUR DE PROCESSUS DE CONCEPTION, basé sur un modèle explicite de conception associé à la requête de l'utilisateur (qui doit être conforme à ce modèle de conception), les algorithmes et les données et fait effectuer le travail demandé.

Le POSTE DE TRAVAIL permet à l'utilisateur de dialoguer avec le système. Il comporte divers organes d'entrées/sorties dont la définition dépend notamment de l'application traitée: console alphanumérique, console graphique, digitaliseur, table interactive, traceur, lecteur de cartes, imprimante, ...)

Le MONITEUR DE COMMUNICATION gère le poste de travail. En entrée, il met les commandes ainsi que les données sous une forme standard et les transmet au moniteur de conception; en sortie, il restitue les données sous la forme demandée.

Du point de vue structurel cette machine abstraite n'a rien de spécifique à la CAO, mais correspond à la structure souhaitable pour tout système interactif appliqué. Elle assure en effet des fonctions de communication homme-machine sous toutes les formes, de gestion des données et de leur manipulation par des algorithmes variés. A ce niveau de détail si on remplace l'appellation moniteur de conception par celle de moniteur de gestion ou encore moniteur de documentation on obtiendrait un autre système appliqué tout a fait satisfaisant. On peut constater que la spécificité se trouve dans les différentes parties de cette machine abstraite.

Une description plus détaillée de cette machine abstraite doit faire ressortir un découpage plus fin.

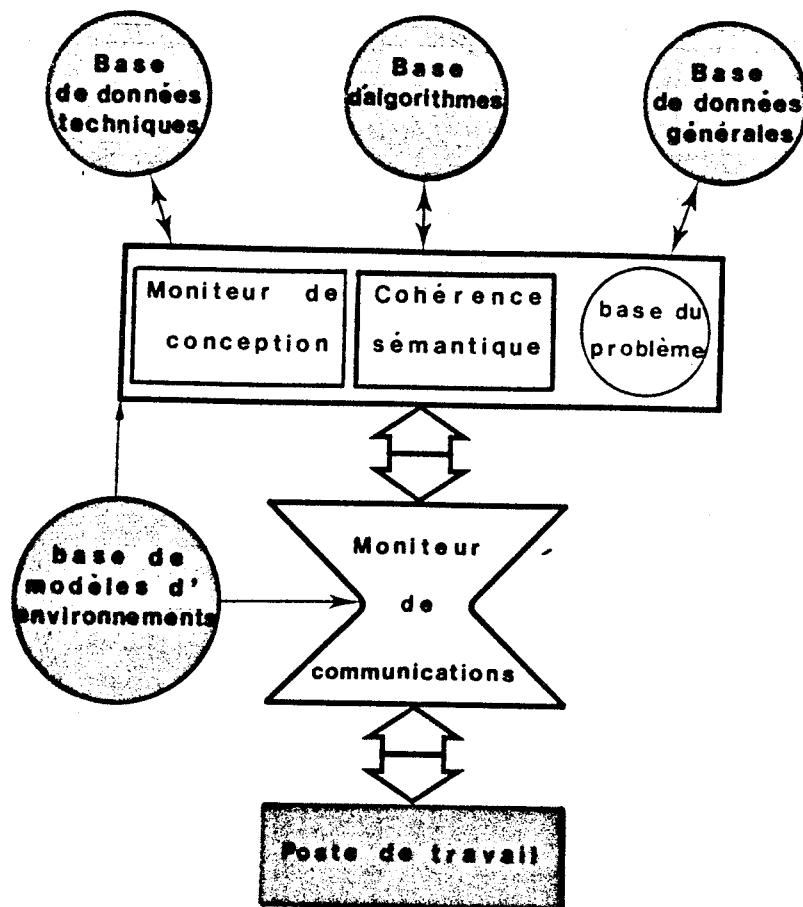


FIGURE II - 9.2 : MACAO - VUE PLUS DETAILLÉE

Si nous examinons plus en détail cette machine abstraite (FIG.II-9.2), nous découvrons trois bases de données (BDG, BDI, BDP), un moniteur de conception devant gérer la cohérence sémantique et la BDE qui fournit les indications sur le comportement possible du système à un moment donné et pour un utilisateur donné.

La spécificité de la CAO, du système CAO support et donc de la MACAO se situe dans la nature des données manipulées et dans la façon dont elles sont manipulées aussi bien par l'utilisateur que par les programmes.

La machine abstraite doit donc intégrer tous les aspects que nous avons présentés dans les chapitres précédents, c'est à dire la gestion du processus de conception, la gestion des données, la

gestion des tâches et la communication homme-machine. Elle doit intégrer de façon cohérente tous les mécanismes proposés tout en restant indépendante d'une réalisation particulière. Avant d'aborder successivement tous les aspects que nous avons déjà traité nous allons d'abord présenter le moniteur de conception dont le rôle est primordial dans la gestion d'un poste de travail complexe et que nous n'avons qu'esquissé dans le chapitre 6.

9.4 MONITEUR DE COMMUNICATION

Le moniteur de communication assure les rôles suivants:

- effectuer les entrées et les sorties d'informations sous la forme demandée et gérer le poste de travail,
- préparer les commandes pour le moniteur de conception, les mettre sous une forme standard indépendante du mode de travail utilisé, c'est à dire soulager le moniteur de conception de cette tâche,
- effectuer la traduction représentation externe -> représentation interne et fournir par conséquent les données au système,
- effectuer la traduction représentation interne -> représentation externe et présenter par conséquent les données à l'utilisateur sous la forme souhaitée.

Le moniteur de communication travaille donc comme un traducteur bidirectionnel: d'une part il transforme les commandes et les données provenant de l'utilisateur sous une forme standard et d'autre part, il met les données provenant du module de conception sous une forme souhaitée par l'utilisateur.

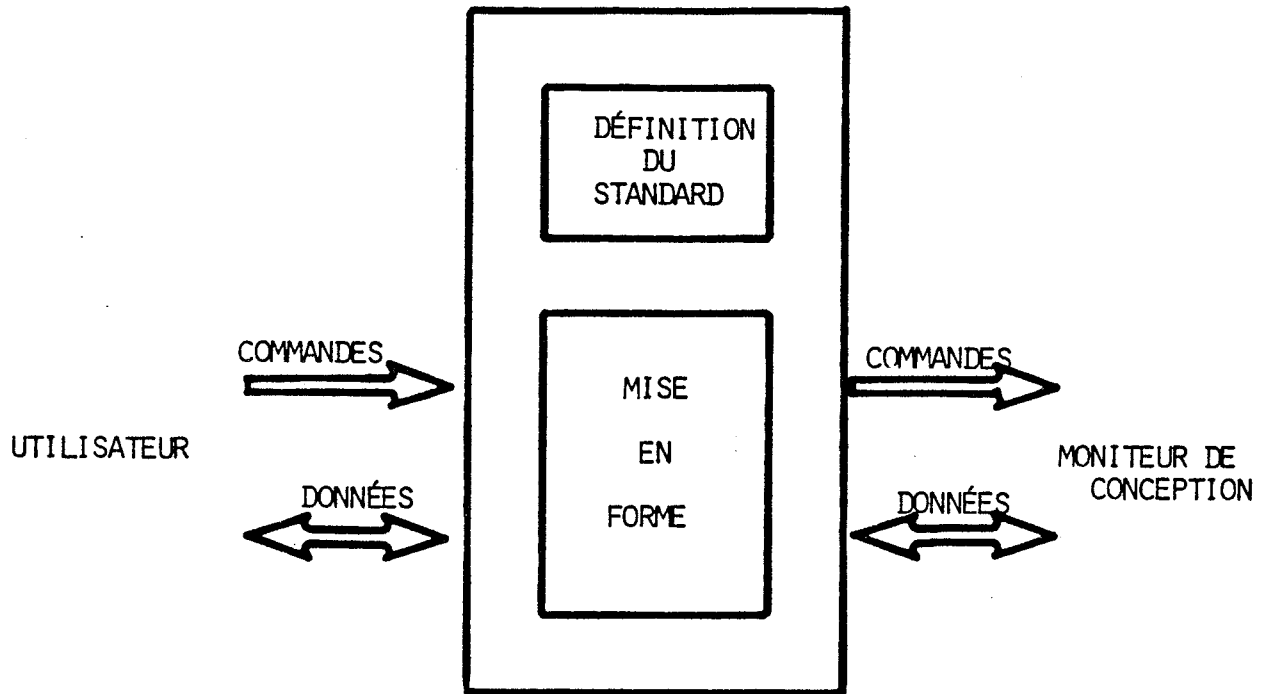


FIGURE II - 9.3 : FONCTIONNEMENT DU MONITEUR DE COMMUNICATION

9.4.1 MECANISME DES UNITES LOGIQUES

Nous avons choisi d'utiliser les concepts d'unité logique et d'unité physique. Il s'agit de définir les différents types d'informations qui transitent à travers le moniteur. A chaque type d'information on associe une UNITE LOGIQUE. Cette unité logique assure de façon unidirectionnelle le passage de l'information. La communication peut s'effectuer à l'aide de différents dispositifs physiques du poste de travail. A chaque dispositif on associe une UNITE PHYSIQUE. L'établissement de la communication s'effectue par l'attachement d'une unité logique à une unité physique. Une même unité physique peut être affectée à plusieurs unités logiques ce qui correspond à une réduction du poste de travail. Ainsi on peut éventuellement travailler avec un seul dispositif: le terminal alphanumérique.

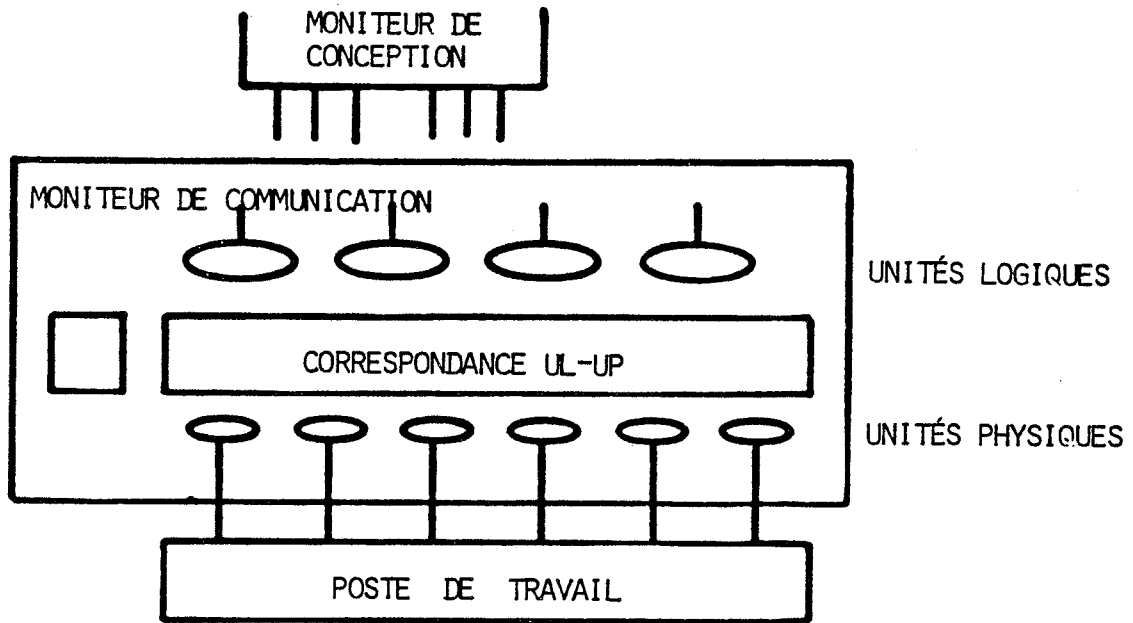


FIGURE II - 9.4 : PASSAGE UNITE LOGIQUE - UNITE PHYSIQUE

Il semble également intéressant de pouvoir attacher en même temps plusieurs unités physiques à une unité logique, notamment pour des questions ergonomiques. Ceci pose un problème de déterminisme qui ne paraît pas simple à résoudre. Les solutions possibles nécessitent soit un attachement dynamique à la demande, qui alourdit considérablement le dialogue, (on peut le réduire en définissant des liens prioritaires (options par défaut) comme dans GRIGRI (LLM.78)), soit une grande démultiplication des unités logiques afin de permettre les attachements unité logique - unité physique à chaque occurrence de la même unité logique. C'est plus rigide car le partitionnement en occurrence d'unités logiques s'effectue à la conception de l'application mais cela simplifie le dialogue et permet une organisation ergonomique très intéressante.

Nous avons retenu cette deuxième solution avec trois comme facteur de démultiplication.

Les tableaux suivants présentent respectivement les unités logiques et les unités physiques que nous avons définies:

TABLEAU D'UNITES LOGIQUES

| NOM | NB | SENS | ROLE |
|-----|----|------|-----------------------------|
| COM | 3 | E | envoi de commandes |
| ERR | 3 | S | retour de messages d'erreur |
| ENT | 3 | E | envoi de données |

| | | | |
|-----|---|---|--|
| SOT | 3 | S | retour de résultats |
| TRA | 3 | S | impression de messages de trace |
| HIS | 3 | S | impression de l'historique des commandes |
| HID | 3 | S | impression d'historique des données |
| ENS | 3 | E | envoi de données système (catalogues) |
| EMS | 3 | E | lecture de macro-commandes |
| SMC | 3 | S | écriture de macro-commandes |

TABLEAU D'UNITES PHYSIQUES

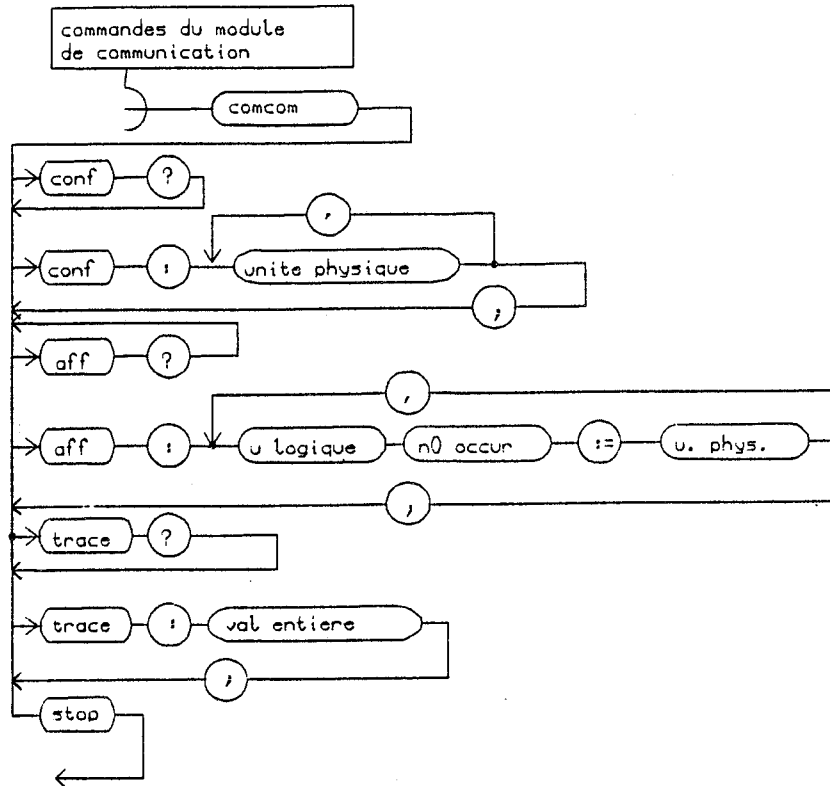
| NOM | NB | SENS | ROLE |
|-------|----|------|-----------------------------------|
| CONS | 3 | E/S | console - terminal alphanumérique |
| IMP | 1 | S | imprimante |
| CARTE | 1 | E/S | lecteur-perforateur de cartes |
| FICH | 10 | E/S | fichiers |
| RIEN | 1 | E/S | absence de périphérique |
| TG | 3 | E/S | terminal graphique |
| TT | 1 | S | table traçante |
| TAB | 1 | E | tablette de digitalisation |
| RUBAN | 1 | E/S | ruban perforé |
| BANDE | 1 | E/S | bande magnétique |

FIGURE II - 9.5 : TABLEAU DES UNITES LOGIQUES ET PHYSIQUES

Un langage de commandes complète ce mécanisme; celui-ci permet à l'utilisateur:

- de définir dynamiquement la configuration du poste de travail (conf:),
- de connaître l'état du poste de travail à un instant donné (conf?),
- d'affecter des unités logiques aux unités physiques (aff:),
- de connaître l'état d'affectation des unités (aff?),
- de demander l'impression d'une trace à partir d'un certain niveau (trace:),
- de connaître le niveau de trace qui a été demandé (trace?).

Cet ensemble de commandes doit être disponible dans n'importe quel environnement, c'est à dire pratiquement pendant toutes les opérations. La figure II-9.6 symbolise la grammaire de cet ensemble de commandes. La trace a été introduite essentiellement pour aider à la mise au point des programmes. La variable "niveau de trace" est utilisée dans une procédure de trace que l'on peut insérer n'importe où dans les programmes et obtenir ainsi des traces sélectives du déroulement des programmes.



no.occur: 1,2 ou 3
 val.entiere: 0 à 10

FIGURE II - 9.6 : LANGAGE DE GESTION DU POSTE DE TRAVAIL (COMCOM)

Au début de chaque programme interactif on peut fixer à l'aide d'une routine d'initialisation la configuration initiale du poste de travail et les affectations des UL à des UP. Si ce n'est pas fait, le système l'initialise automatiquement avec une configuration standard. L'utilisateur peut modifier à tout moment et dynamiquement ces deux caractéristiques à l'aide de l'ensemble de commandes que nous venons de présenter. La FIG.II-9.7 montre quelques exemples d'utilisation de ce langage.

```

aff?
aff?
  com
  1 : consl
  2 : fichier.data
  3 : carte
  ent
  1 : carte
  2 : consl
  3 : consl
  ens
  1 : carte
  2 : carte
  3 : carte
  err
  1 : consl
  2 : consl
  3 : consl
    
```

L'ETAT DES AFFECTATIONS DES UNITÉS LOGIQUES
 AUX UNITÉS PHYSIQUES

```

sor
1 : cons1
2 : cons1
3 : cons1
tra
1 : cons1
2 : cons1
3 : cons1
his
1 : cons1
2 : fichier9.data
3 : cons1
hid
1 : bande
2 : bande
3 : bande
trace= 0

```

L'ETAT DU POSTE DE TRAVAIL (UNITÉS PHYSIQUES DISPONIBLES)

```

conf?
conf?
rien      carte  fich      cons1    rien      carte    fich      cons1

```

ON AJOUTE CONS2 DANS LE POSTE DE TRAVAIL

```

conf: cons2;
conf: cons2;
conf?
conf?
rien      carte  fich      cons1    cons2    rien      carte    fich      cons1    cons2
NOUVEL ETAT DU POSTE DE TRAVAIL

```

```

aff:com(3):=fich commande.ldc s      AFFECTATION DE L'UNITÉ LOGIQUE COM3 AU FICHER
aff:com(3):=fich commande.ldc s
nouvelle affectation : com 3 := commande.ldc s      52
SEQUENTIEL COMMANDE.LDC

```

```

trace:3;
trace:3;      AFFECTATION DE LA VARIABLE TRACE

```

FIGURE II - 9.7 : EXEMPLES D'UTILISATION DU LANGAGE DE GESTION DU POSTE DE TRAVAIL COMCOM

L'interprétation et la transformation des informations entre les UL et les UP s'effectuent de la façon suivante: le système acquiert les informations par l'intermédiaire des unités logiques. Ces unités sont monodirectionnelles. Les transformations à envisager sont donc (FIG.II-9.8).

| ENTREES | | SORTIES | |
|---------|-------|---------|-------|
| UL | UP | UL | UP |
| COM | CONS | ERR | CONS |
| ENT | CARTE | SOR | IMP |
| ENS | FICH | TRA | CARTE |
| | RIEN | HIS | FICH |
| | TG | HID | RIEN |
| | TAB | | TG |
| | BANDE | | TT |
| | RUBAN | | BANDE |
| | | | RUBAN |

FIG.II - 9.8 : AFFECTATIONS POSSIBLES ENTRE UL ET UP

En étudiant de plus près les unités physiques on constate que pour obtenir une forme canonique à l'entrée il faut assurer trois types de transformations:

- 1/ entrée alpha-numérique pour les UP: CONS, CARTE, FICH, BANDE et IG utilisé en mode alpha-numérique
- 2/ entrée graphique pour IG en mode graphique et TAB pour une partie des informations seulement
- 3/ entrée par RUBAN perforé

Pour les sorties on doit transformer la forme canonique selon les trois types d'unités physiques:

- 1/ sortie alpha-numérique pour les UP: CONS, IMP, CARTE, FICH, BANDE et IG utilisé en alpha-numérique
- 2/ sortie graphique pour IG en mode graphique et TT
- 3/ sortie sur RUBAN perforé

Nous laissons de côté le cas du ruban car son utilisation est de plus en plus rare. Ces transformations forme canonique, forme spécifique et réciproquement, s'effectuent de la manière indiquée sur la Figure II-9.9.

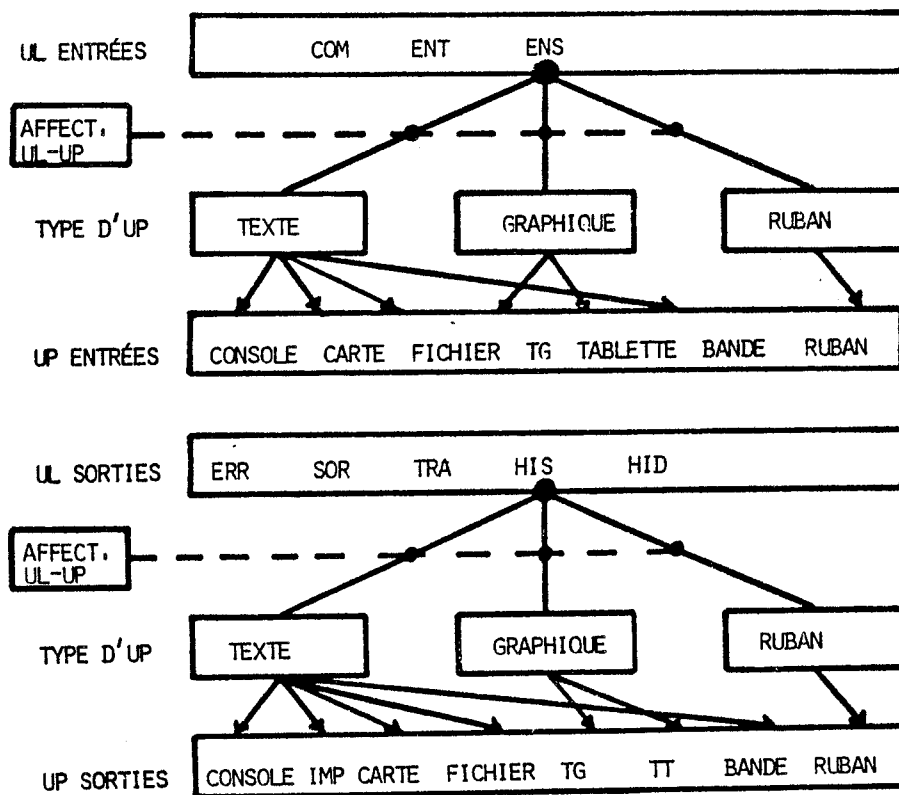


FIGURE II - 9.9 : CHEMINEMENT DES INFORMATIONS

9.4.2. ANALYSE LEXICOGRAPHIQUE

Le rôle de l'analyse lexicographique est de déterminer en analysant la chaîne d'entrée les unités syntaxiques que l'on utilisera dans l'analyse syntaxique. Dans notre cas cette analyse syntaxique s'effectue principalement dans le moniteur de dialogues qui est dirigé par la syntaxe. Cette analyse lexicographique peut être de différents niveaux, soit elle est purement structurelle, c'est à dire qu'elle détermine uniquement la structure de la chaîne d'entrée (elle peut reconnaître un entier, un réel, un identificateur, une chaîne de caractères), soit elle dispose déjà d'une première description sémantique et est donc capable de reconnaître un mot-clé d'un identificateur, le type d'un identificateur, un opérateur, etc... Dans ce deuxième cas elle doit disposer des informations sémantiques. Ceci est généralement fait par des tables (de mot-clés, d'opérateurs,...).

Nous avons choisi d'entreprendre ces deux niveaux d'analyse, mais de ne pas les mélanger et de les effectuer l'une à la suite de l'autre. La première nous l'appelons LEXIC et la seconde SEMANT car elle fait une première analyse sémantique même si celle-ci se limite à une unité syntaxique à la fois. Nous avons opté pour cette solution essentiellement pour la raison suivante. Dans le poste de travail nous pouvons avoir principalement deux types de dispositifs: d'une part des dispositifs à entrée/sortie continue, c'est à dire sous forme de lignes de texte, et d'autre part des dispositifs à entrée/sortie structurée, c'est à dire sous forme d'écran structuré avec des zones et des champs prédéfinis.

Dans le premier cas nous devons effectuer l'analyse lexicographique puis l'analyse sémantique, dans le second nous pouvons nous contenter de l'analyse sémantique car l'analyse lexicographique a déjà été faite. Nous sommes donc en présence soit du mode formaté soit du mode non formaté (FIG.II-9.10).

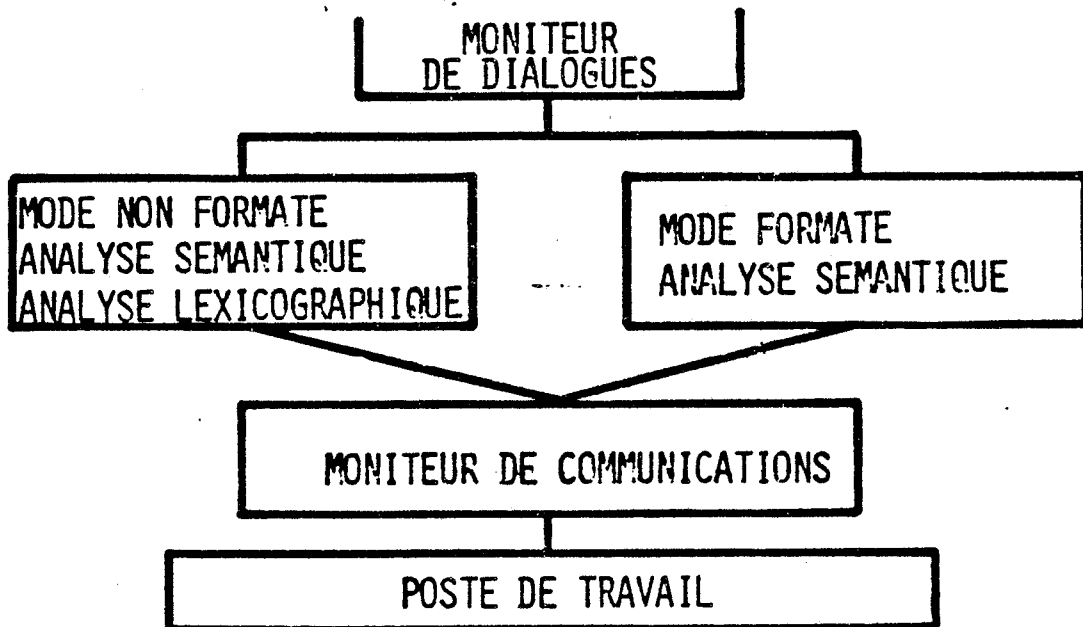


FIGURE II - 9.10 : STRUCTURE DE L'ANALYSE

La figure II-9.11 montre le déroulement de l'analyse. L'information provient du poste de travail. Si elle suit le cheminement des informations non formatées elle passe par la lecture sur un dispositif d'entrée non formaté, par le découpage de la ligne en caractères (CARSUIV), la reconstruction d'unité syntaxique

- entier
- réel
- caractère spécial
- texte
- identificateur pointé

pour finir par l'analyse sémantique (SEMANT) qui détermine sa sémantique. Pour cela SEMANT dispose des tables de mot-clés, des identificateurs et des opérateurs, et selon la nature lexicale de l'information il cherche à déterminer la nature sémantique de l'unité syntaxique. Il dispose pour cela d'un conseil de la part de l'analyseur syntaxique, qui peut lui communiquer par exemple le type d'identificateur qu'il attend. Si ce n'est pas le cas et SEMANT ne peut pas déterminer le type de l'identificateur, il le garde temporairement comme identificateur non typé et il le complètera quand il aura l'information nécessaire.

Si l'information suit le cheminement des informations formatées, elle passe directement par des procédures de saisie

d'entier, de réel, de mot, de texte, et elle peut donc être transmise directement à SEMANT pour une analyse sémantique.

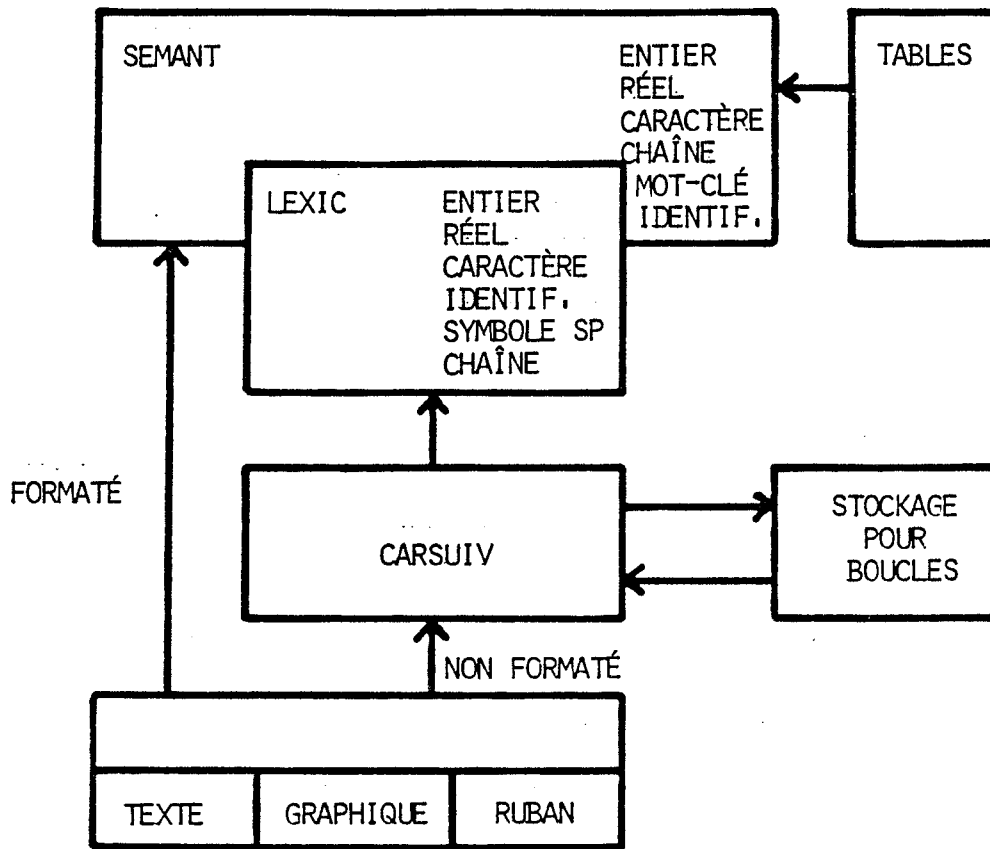


FIGURE II - 9.11 : ORGANISATION DE L'ANALYSE LEXICOGRAPHIQUE

9.4.3 SORTIES SOUS FORME CANONIQUE

Compte tenu de la séparation entre les unités logiques (UL) et les unités physiques (UP) nous ne pouvons pas utiliser les fonctions d'entrée/sortie disponibles dans les langages de programmation (WRITE de FORTRAN par exemple), car ces fonctions travaillent directement sur les unités physiques et leur format diffère selon la nature de l'unité.

C'est pourquoi nous avons été obligés de développer notre propre mécanisme afin d'adresser sous une forme unique toutes les unités physiques.

Ce mécanisme est constitué par:

- un ensemble de fonctions permettant de signaler le transfert de tout type d'information,
- une fonction standard assurant le transfert effectif vers le tampon de l'unité physique concernée et activant cette unité si nécessaire,
- la gestion des tampons associés aux unités physiques.

La figure II-9.12 montre le schéma de principe de ce mécanisme.

Dans le langage de programmation on dispose des fonctions suivantes:

| | |
|------------------|---|
| inserR (UL,R) | insérer dans UL un réel R |
| inserI (UL,I) | un entier I |
| inserC (UL,C) | un caractère C |
| inserB (UL,B) | un logique C |
| inserM (UL,M) | un mot M de 8 caractères |
| inserS (UL,S,NB) | une chaîne S de NB caractères (78 au plus) |
| insFIN (UL) | terminer le tampon courant pour UL et l'envoyer |

Ces procédures font appel à une procédure TRANSFERT qui transmet l'information codée dans le tampon de l'unité physique associée à l'unité logique et s'il le faut active la procédure de sortie du tampon.

A chaque unité de sortie est associé un tampon dont la longueur dépend du type de cette unité. Quand un tampon est plein il est automatiquement vidé par la procédure de sortie de tampon. Il est possible de redéfinir la taille des tampons des UP.

Pour compléter ce mécanisme initialement conçu pour des sorties non formatées, nous avons ajouté de nouvelles fonctions permettant de définir des tabulations (deftab(nom_tab,desc_tab)) et de les utiliser dans les insetX(UL,X,tab,nom_tab).

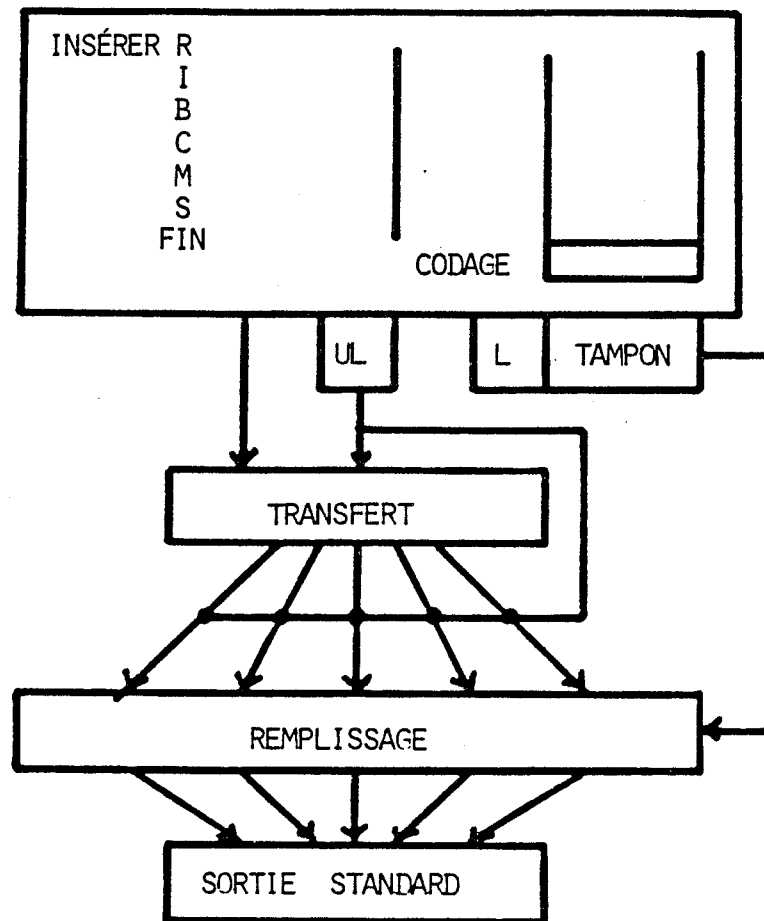


FIGURE II - 9.12 : PRINCIPES DU MECANISME UNIQUE DE SORTIE

9.4.4 STRUCTURE DU MONITEUR DE COMMUNICATION

La structure du moniteur de communication est donc la suivante: les périphériques composant le poste de travail communiquent avec le module CLP dont le rôle est d'assurer la correspondance entre les unités Logiques et les unités Physiques. Selon que l'information est formatée ou non, elle passe ou-elle ne passe pas par LEXIC, avant d'être traitée par SEMANT. Une fois sous la forme canonique (sémantiquement et syntaxiquement correcte), elle est transmise au niveau supérieur généralement dans un module de dialogues. En sortie les informations sont transmises au module SORTIE, qui les transmet à CLP pour une sortie effective sur le ou les périphériques demandés. Le module ERREUR se charge d'effectuer les sorties de messages d'erreur.

Le module DPT (Définition du Poste de Travail) a pour rôle de gérer la définition du poste de travail exprimée à l'aide du langage COMCOM.

La figure II-9.13 schématise la structure du moniteur de communication.

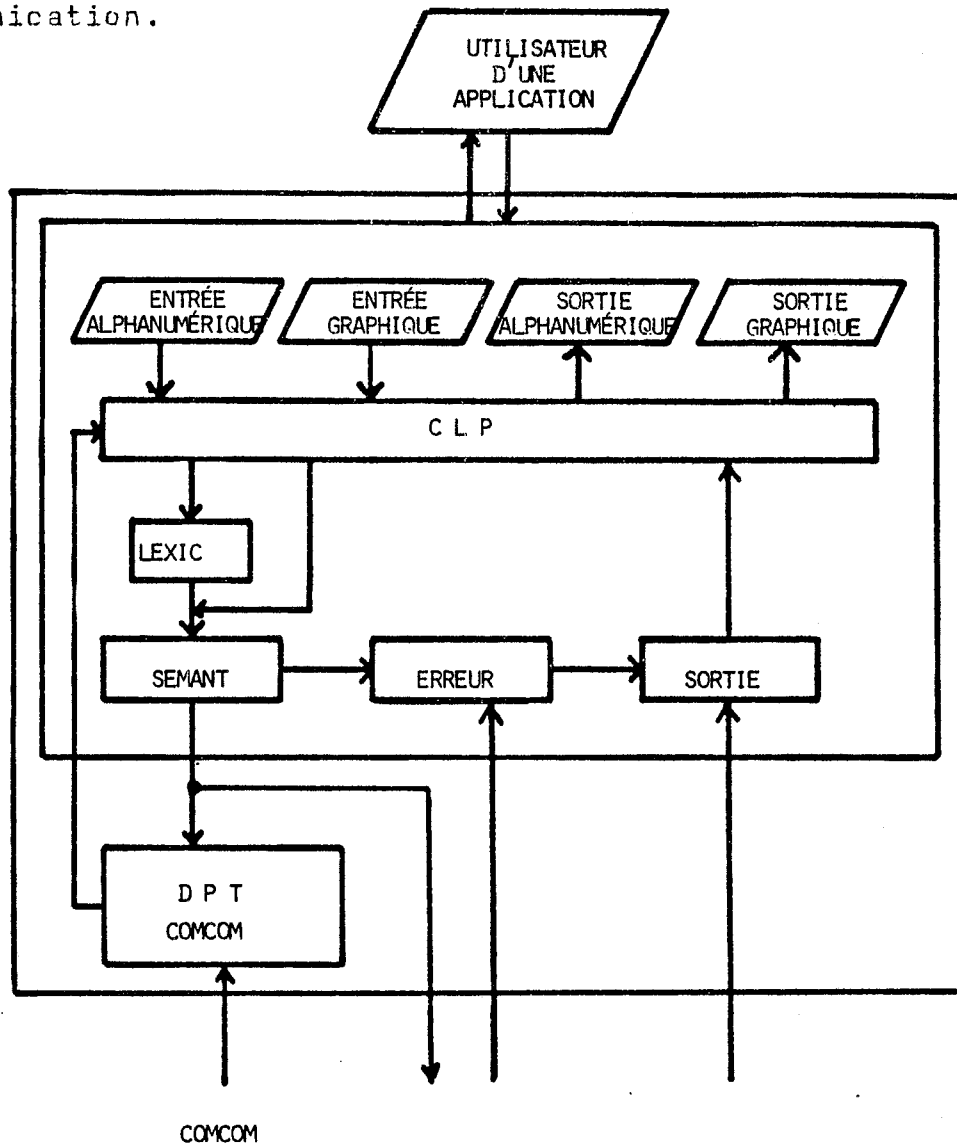


FIGURE II - 9.13 : STRUCTURE DU MONITEUR DE COMMUNICATION

9.5 MONITEUR DU PROCESSUS DE CONCEPTION

Le moniteur du processus de conception est le chef d'orchestre d'un système CAO. Son rôle est de permettre à un utilisateur de travailler dans les meilleures conditions. C'est à dire en lui permettant de faire aussi aisément que possible tout ce qu'il a le droit de faire mais en lui interdisant strictement ce à quoi il n'a pas droit.

Ce moniteur de processus de conception est donc une tâche interactive située au dessus de toutes les autres tâches et

jouant un rôle "logique" (par opposition au rôle "technologique" dépendant d'une application).

Il gère donc principalement des responsabilités, des points de vue, des accès et des opérations et ceci conformément à la description du processus de conception représenté par les états et les transitions.

Du point de vue de sa structure il s'agit d'un moniteur de dialogues (voir chapitre 6), auquel on a fourni comme description de dialogues, la description du processus de conception et comme fonctions sémantiques associées, les fonctions standards de gestion des responsabilités, des points de vue, des opérations, ainsi que celles de gestion du projet et d'activation de tâches de la base d'algorithmes et d'accès à la BDP.

La structure du moniteur de processus de conception est représentée sur la figure II - 9.14.

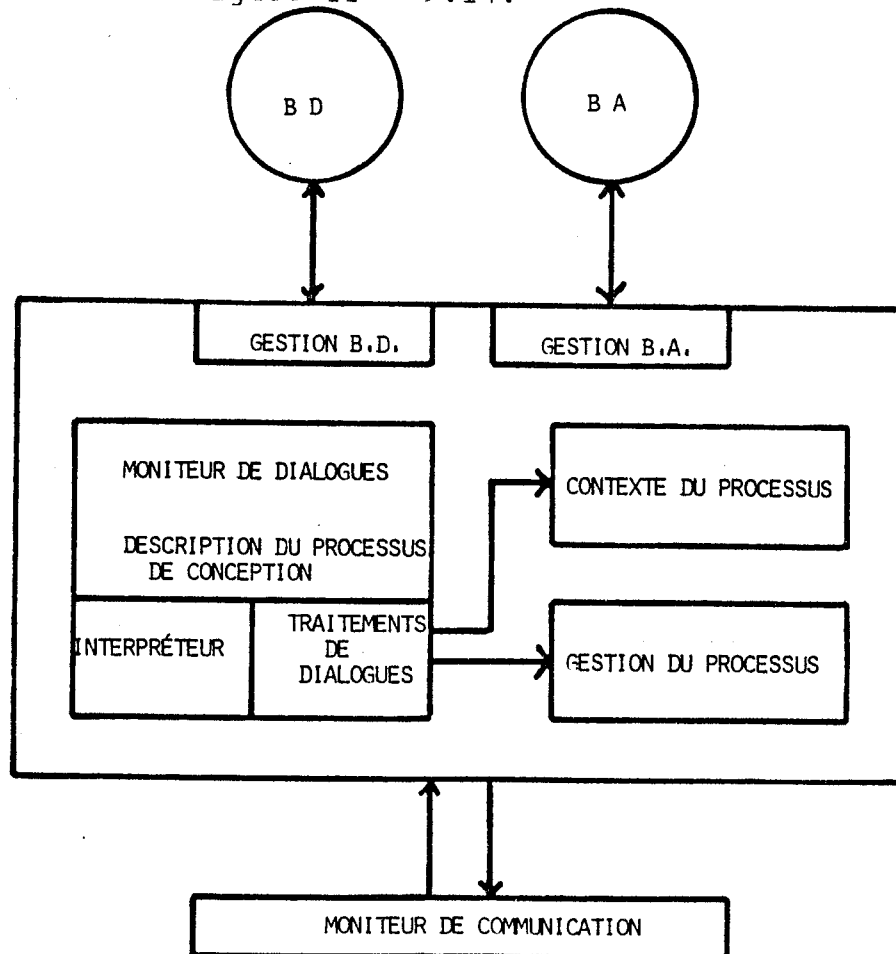


FIGURE II - 9.14 : STRUCTURE DU MONITEUR DE PROCESSUS DE CONCEPTION

9.6 BASE DE DONNEES

L'accès à la base de données ou plus précisément aux bases de données (BDP, BDT, BDG), s'effectue toujours par l'intermédiaire d'une tâche du système CAO. Cette tâche peut être soit une tâche "logique" permettant à l'utilisateur de dialoguer avec les bases de façon interactive quelque soit la base et quelque soit le but de l'opération, soit une tâche "technologique" assurant un rôle précis dans le système CAO.

Dans les deux cas l'accès s'effectue selon le mode d'accès expliqué dans le chapitre 5. Il s'agit du mode d'accès spécifique puis de différents modes d'accès logiques. L'accès à la BDT et à la BDG s'effectue de façon classique car il s'agit des bases de données classiques. Pour la BDP le problème est plus complexe, car on peut soit accéder aux données, soit au modèle des données et dans le cas de modifications des données ou du modèle la gestion de la cohérence doit être effectuée.

La structure du moniteur de gestion des données est schématisée sur la figure II -9.15.

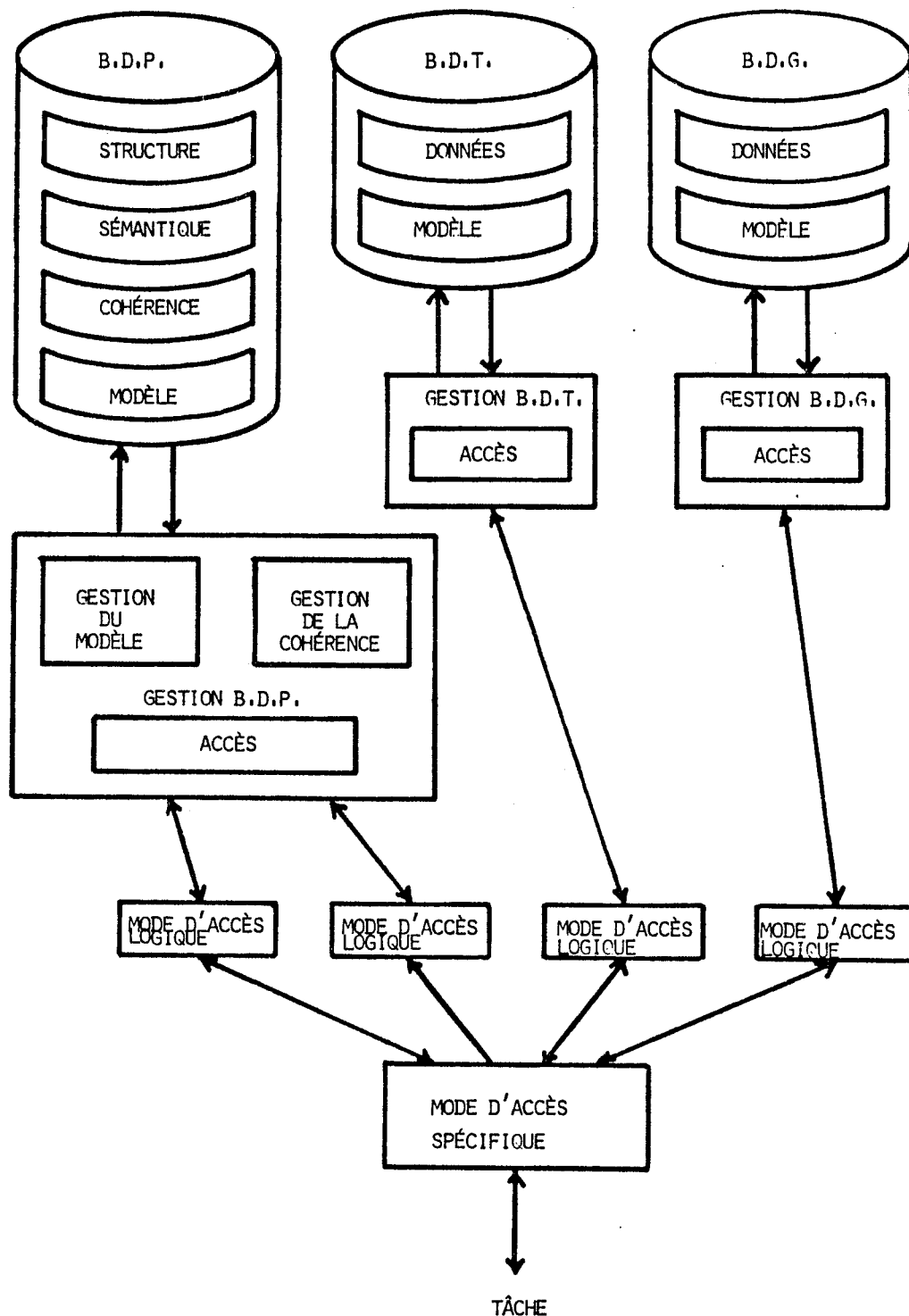


FIGURE II - 9.15 : STRUCTURE DU MONITEUR DE GESTION DES DONNEES

9.7 BASE D'ALGORITHMES

Dans un système CAO il faut gérer et activer des traitements. La base d'algorithmes est constituée par un

ensemble de modules. Ces modules sont soit des modules "logiques" (algorithmes d'intérêt général), soit des modules "technologiques" (algorithmes spécifiques, soit encore des modules de "contexte" décrivant le contexte d'utilisation. Un éditeur graphique se trouvera dans la bibliothèque des modules logiques, un simulateur asynchrone de systèmes logiques dans une des bibliothèques des modules technologiques, quant aux modules de contexte, ils décrivent des conditions physiques (nature du poste de travail, puissance de l'ordinateur, ...) et logiques (nature et type du travail, ...) d'utilisation des autres modules. Cette séparation permet une meilleure gestion des modules (création des bibliothèques publiques ou privées).

Une tâche est décrite par un descripteur qui indique les conditions d'utilisation de la tâche ainsi que l'ensemble des modules "logiques", "technologiques" et de "contexte" qui la composent.

L'activation d'une tâche est formulée par le moniteur de conception qui transmet en même temps à la tâche le contexte, c'est à dire les paramètres associés et le contexte d'activation.

Le moniteur de gestion de tâches cherche dans la base d'algorithmes le descripteur de la tâche et effectue la mise en place (chargement, initialisation, activation). Quand la tâche se termine elle rend le contrôle au moniteur du module de conception avec un descripteur de comportement.

La figure II - 9.16 schématise la structure du moniteur de gestion des tâches.

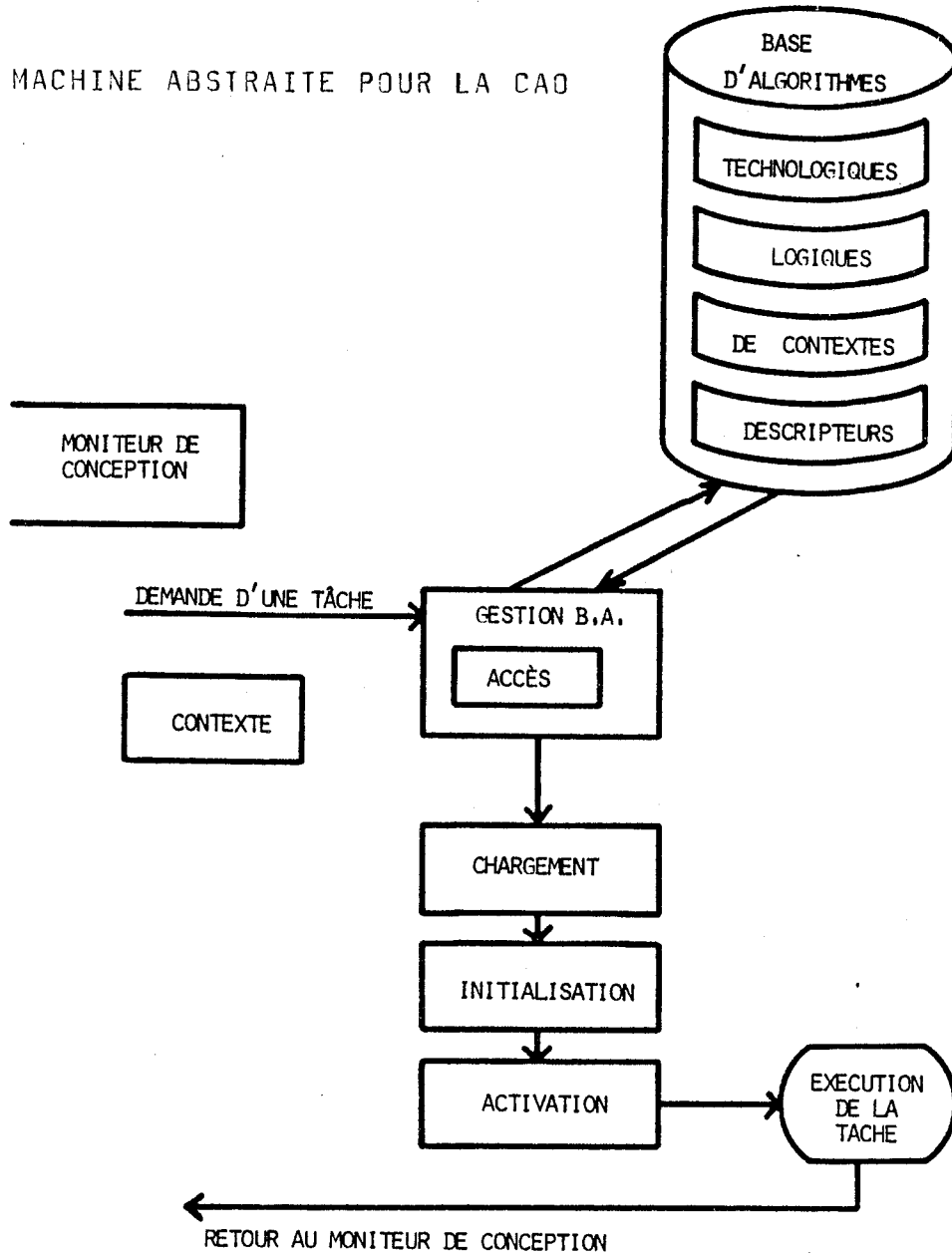


FIGURE II - 9.16 : STRUCTURE DU MONITEUR DE GESTION DES TACHES

CHAPITRE 10

PROTOTYPE, ETUDE DE FAISABILITE, METHODOLOGIE DE CONSTRUCTION

RESUME: Ce chapitre met en évidence les liens entre l'invariant conceptuel représenté par le langage de spécification et l'invariant informatique caractérisé par la machine abstraite. On commence par définir et justifier l'intérêt d'une méthode de faisabilité et de sa construction sous forme de prototype; on schématise ensuite le passage des spécifications au prototype, et on détaille la façon dont les différents énoncés de COLA garnissent toutes les parties de la machine abstraite. On termine par la présentation de deux structures du système pour la construction de prototypes.

10. PROTOTYPE, ETUDE DE FAISABILITE, METHODOLOGIE DE CONSTRUCTION

Nous avons étudié dans les chapitres précédents deux aspects fondamentaux des systèmes CAO, d'une part l'aspect fonctionnel où nous avons décrit par quoi toute application CAO est caractérisée et nous avons défini un langage de spécifications de ce fonctionnement et d'autre part l'aspect support, c'est à dire sur quoi une application spécifiée peut ou doit s'appuyer.

C'est la mise en oeuvre de l'application spécifiée, en COLA, sur la machine abstraite MACAO, qui nous intéresse ici.

Il existe deux techniques de mise en oeuvre, la première nous l'appelons construction de prototype et son but est de bâtir rapidement une réalisation expérimentale de l'application pour vérifier la faisabilité du système, la seconde nous l'appelons construction définitive du système CAO appliqué, tient compte de tous les aspects spécifiques de l'application et de son contexte matériel et logiciel.

Dans ce qui suit il sera surtout question de la construction du prototype.

10.1 DEFINITION ET JUSTIFICATION D'UN PROTOTYPE

La construction du prototype a pour but de vérifier et affiner les spécifications fonctionnelles du système CAO. Comme nous l'avons déjà précisé il est très difficile de produire directement des spécifications précises, justes et détaillées d'une application CAO. Le problème est beaucoup trop vaste et des interactions entre différentes parties peuvent être tellement complexes, qu'il n'est souvent pas possible d'espérer aboutir directement à des spécifications stables. Dans ce cas il serait utopique et économiquement inadmissible de vouloir consacrer un effort considérable à la réalisation d'un système qui ne sera sans doute pas retenu mais servira uniquement de point de départ à une définition plus précise. C'est pourquoi les notions de prototype et d'étude de faisabilité ont été introduites.

La phase ETUDE DE FAISABILITE a pour rôle de démontrer que le système spécifié est faisable, c'est à dire qu'il correspond dans les grandes lignes à ce que l'on souhaite. Pour démontrer la faisabilité on construit un PROTOTYPE du système. Ce prototype représente en partie le système final. Comme il s'agit

de construire une maquette du système on peut faire un certain nombre de simplifications dans les choix de réalisation. Ainsi il est possible de ne pas tenir compte du contexte précis dans lequel ce prototype est développé et de ne pas optimiser le système dans ce contexte (matériel et logiciel). On peut aussi utiliser des programmes fonctionnellement équivalents (composants fonctionnels préexistants) et finalement utiliser des techniques de construction assistée par ordinateur (construction semi-automatique).

Les objectifs du prototype sont principalement:

- de constituer une maquette de la structure fonctionnelle de l'application,
- de permettre, à l'aide d'un nouveau processus de développement, d'obtenir avant la réalisation finale une version dans laquelle la plupart des critiques de fonctionnement puissent s'exprimer.

Il semble important de préciser dans quel contexte la construction de prototype se justifie. Il s'agit principalement:

- d'application au coût élevé admettant de longs délais de réalisation,
- d'application qui modifie profondément les méthodes de travail,
- d'application informatique floue.

Les applications CAO et notamment celles conduisant vers une approche intégrée de la CAO-FAO, réunissent ces trois caractéristiques. L'approche par prototype se justifie donc d'autant plus.

Pour que le prototype remplisse le rôle que l'on lui attribue, il doit pouvoir être réalisé rapidement pour ne pas retarder la phase de mise en oeuvre définitive. Il doit aussi être facilement modifiable pour prendre en compte aisément de nouvelles spécifications et pouvoir tester différentes alternatives de fonctionnement.

Il doit gérer sa propre documentation car à la fin de l'étude de faisabilité il est pratiquement abandonné, ce qui reste uniquement ce sont les spécifications détaillées du système souhaité.

La technique de mise en oeuvre du prototype doit donc tenir compte de ces contraintes particulières et utiliser au maximum des techniques automatiques.

Pouvoir travailler rapidement et faire éventuellement appel à des techniques automatisées, nécessite un cadre d'action bien modélisé. Pour la construction de prototype on s'appuie sur le quadruple suivant:

modèle + langage + méthode + outils

10.2 PASSAGE DES SPECIFICATIONS AU PROTOTYPE

La figure II-10.1 schématise le quadruple

MODELE + LANGAGE + METHODE + OUTILS

pour la construction de prototype du système CAO dans le contexte de SIGMA-CAO. Il s'agit pour:

- la modèle: le modèle conceptuel des applications CAO et la machine abstraite pour la CAO, modèle fonctionnel du support des applications CAO,
- le langage: le langage conceptuel COLA, permettant d'exprimer les spécifications fonctionnelles de l'application CAO,
- la méthode: le processus de projection des spécifications écrites en COLA sur la machine abstraite MACAO, que nous allons expliciter ci-après,
- les outils: les outils de traitement des spécifications en COLA que nous présentons en détail dans le chapitre 11.

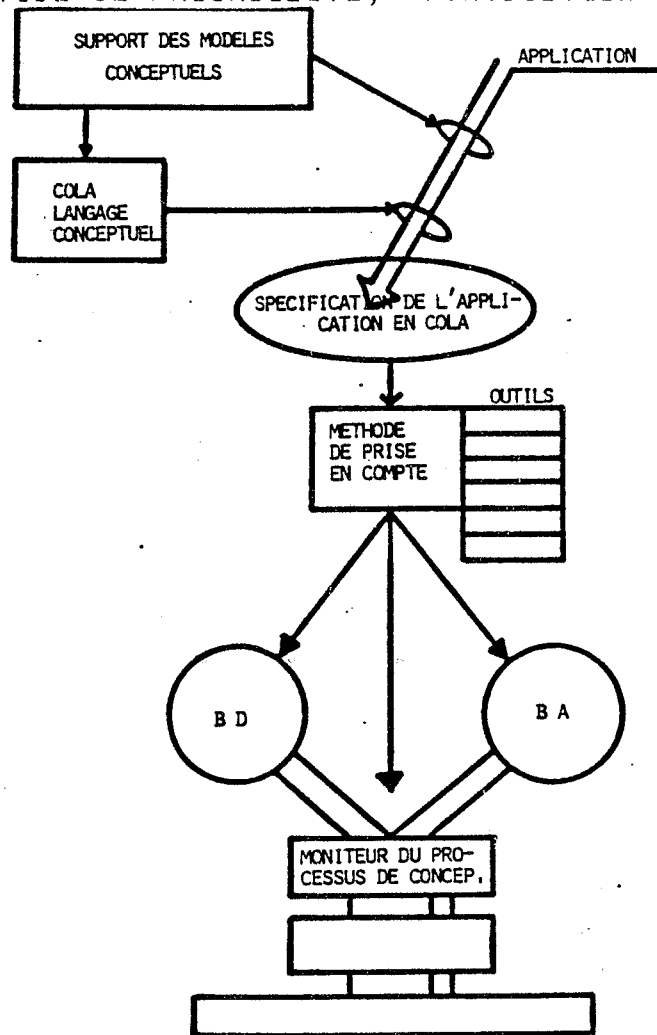


FIGURE II - 10.1 : SCHEMA DE PRINCIPE DE LA CONSTRUCTION DU PROTOTYPE D'UNE APPLICATION CAO

De façon plus précise, il s'agit d'analyser l'ensemble des énoncés et de garnir la machine abstraite par les informations fonctionnelles contenues dans les spécifications.

Pour montrer ce processus examinons les différents énoncés et indiquons pour chacun où les informations qu'il porte seront rangées:

- PROJET - énoncé de gestion des spécifications qui ne porte pas d'informations opératoires,
- ESPACE DE CONCEPTION - énoncé définissant une partie du processus de conception, garnit le moniteur de conception,
- PROCESSUS DE CONCEPTION - énoncé définissant une partie du processus de conception, garnit le moniteur de conception,
- RESPONSABILITE - énoncé définissant l'organisation du processus de conception, garnit le moniteur de conception,
- DEFINITION DE PROPRIETES - énoncé définissant un aspect des données manipulées, garnit la base de données,

DEFINITION DES OBJETS - énoncé définissant un aspect des données manipulées, garnit la base de données,
 DEPENDANCE DES PROPRIETES - énoncé définissant un aspect des données manipulées, garnit la base de données,
 CORRESPONDANCE DES PROPRIETES - énoncé définissant un aspect des données manipulées, garnit la base de données,
 DEFINITION DES TRANSFORMATIONS - énoncé définissant un aspect des données manipulées, garnit la base de données,
 TRANSFORMATION DE L'OBJET A CONCEVOIR - énoncé définissant un aspect des données manipulées, garnit la base de données,
 DEFINITION DES TRAITEMENTS - énoncé définissant des algorithmes utilisés, garnit la base d'algorithmes.

La figure II-10.2 schématise la projection des énoncés de COLA sur la machine abstraite MACAO du niveau le plus bas. Pour mieux montrer comment ces informations garnissent la machine abstraite, nous devons raisonner sur la vision plus détaillée de la MACAO. C'est ce que nous présentons sur les figure II-10.3-6 en reprenant les figures du chapitre 9.

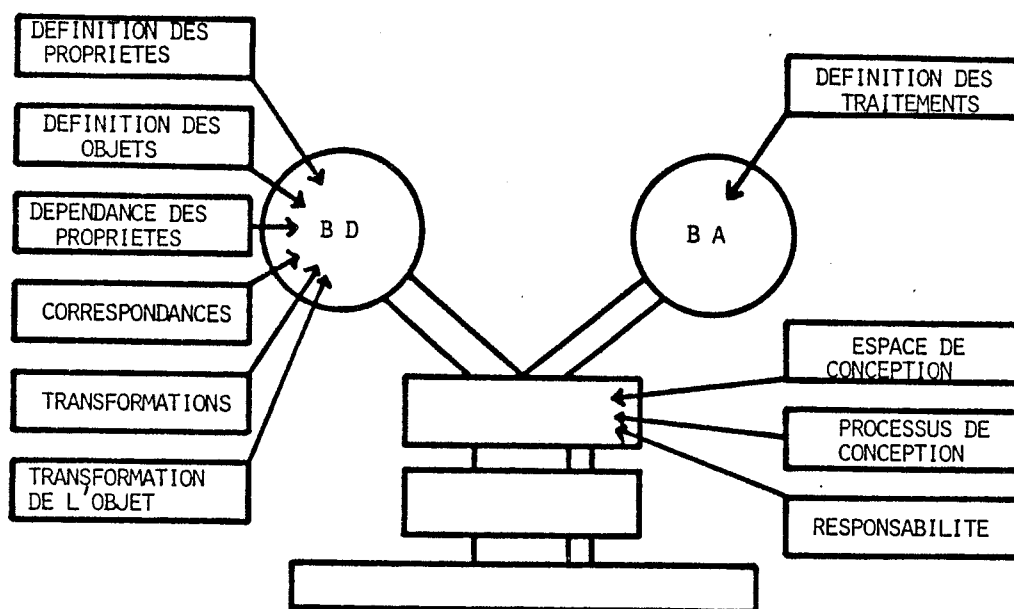


FIGURE II - 10.2 : PROJECTION DES ENONCES DE COLA SUR LA MACAO

Le moniteur de conception reçoit les informations sur la structure et le déroulement du processus de conception. Ce moniteur est constitué essentiellement d'un moniteur de dialogues auquel on fournit la description de dialogues bâtis à partir des énoncés suivants:

- espace de conception,
- processus de conception,
- responsabilité,
- définition des traitements.

Les deux premiers indiquent la structure et l'enchaînement des étapes et des phases du processus et les deux autres la répartition des responsabilités et la situation de l'ensemble des traitements dans ce processus, ainsi que les paramètres nécessaires pour leurs activations.

Ce moniteur de dialogues travaille de façon systématique sur la description du processus et active quand c'est nécessaire des transformations définies par l'énoncé DEFINITION DES TRANSFORMATIONS.

Ce sont ces informations qui constituent le contexte du processus et le module de gestion les gère.

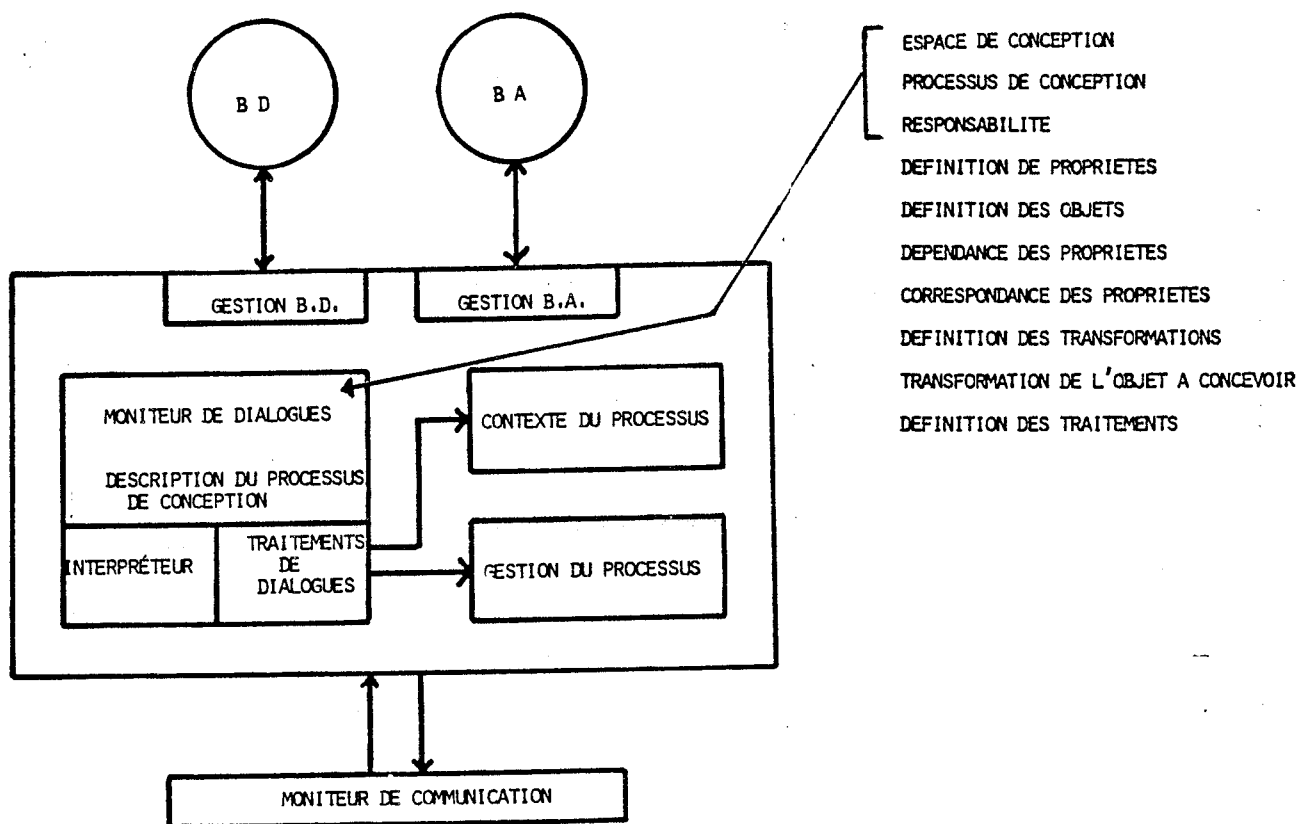


FIGURE II - 10.3 : MISE EN PLACE DU MONITEUR DE CONCEPTION

La mise en place de la gestion des données pour une application doit prendre des informations nécessaires

pratiquement dans tous les énoncés de COLA. En effet, les énoncés ESPACE DE CONCEPTION, PROCESSUS DE CONCEPTION et RESPONSABILITE fournissent des informations nécessaires pour la gestion de la cohérence. L'énoncé DEFINITION DE PROPRIETES spécifie la sémantique de l'application et donne lieu à la construction des modes d'accès logiques pour chaque type de propriété. L'énoncé DEFINITION DES OBJETS permet de prévoir des structures d'objets.

Les énoncés DEPENDANCE DES PROPRIETES, CORRESPONDANCE DES PROPRIETES, DEFINITION DES TRANSFORMATIONS et TRANSFORMATION DE L'OBJET A CONCEVOIR interviennent dans la gestion de cohérence et dans la définition des opérations sémantiquement complètes.

Quant à l'énoncé DEFINITION DES TRAITEMENTS il est utilisé notamment pour la construction des modes d'accès spécifiques.

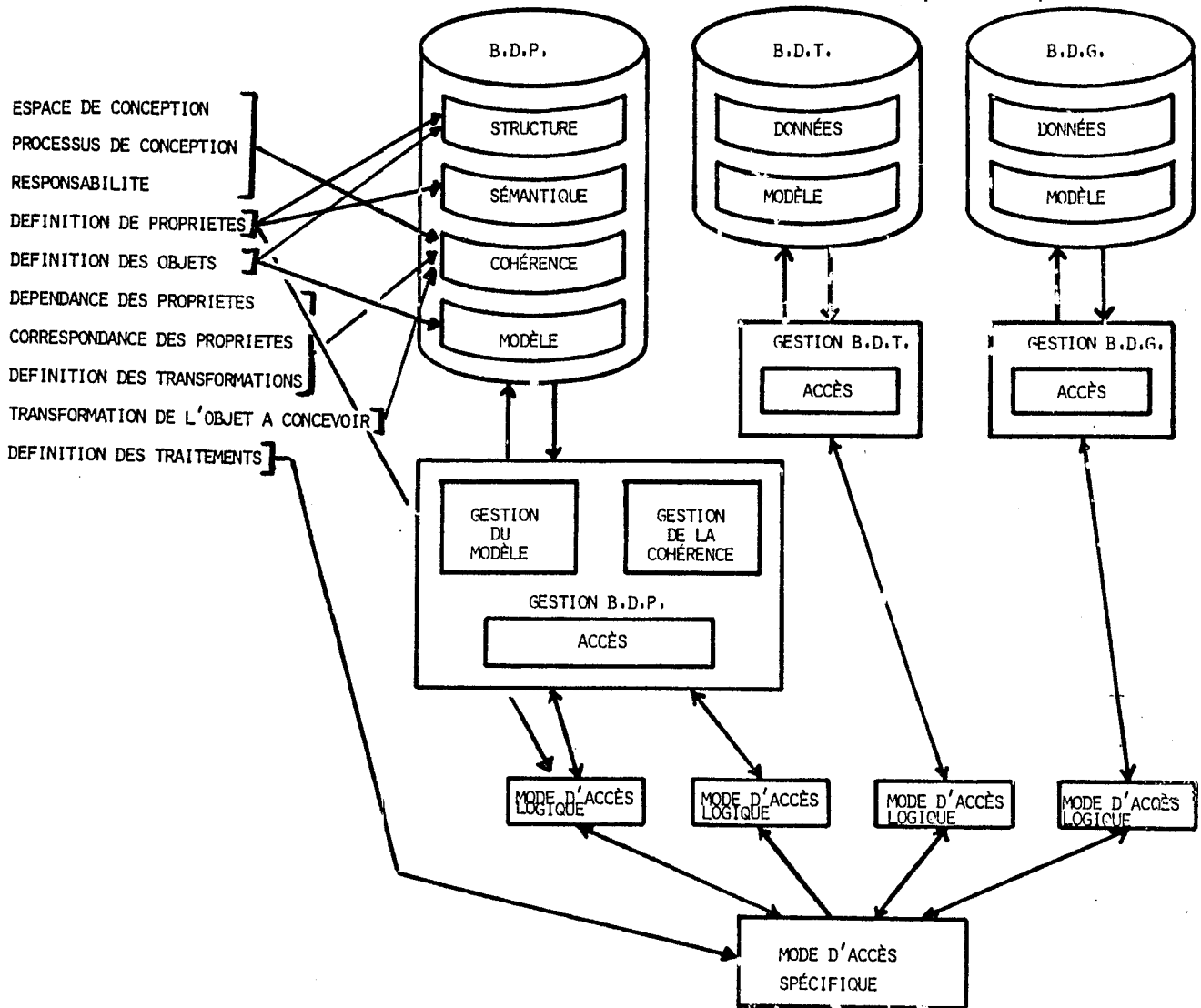


FIGURE II - 10.4 : MISE EN PLACE DE LA GESTION DES DONNEES

La mise en place de la base d'algorithmes s'appuie surtout sur les énoncés décrivant le processus de conception (ESPACE DE CONCEPTION, PROCESSUS DE CONCEPTION, RESPONSABILITE) précisant le contexte d'utilisation des traitements et sur l'énoncé DEFINITION DES TRAITEMENTS lui-même.

Dans la base d'algorithmes nous trouvons des algorithmes logiques ou généraux pouvant être utilisés quelque soit le contexte, des algorithmes technologiques, dont le domaine d'application est plus restreint. En plus la description du contexte permet de préciser le contexte d'utilisation de tel ou tel algorithme. Le descripteur a pour but d'organiser l'ensemble des parties (description du contexte, algorithmes logiques, algorithmes technologiques) en un traitement complet pouvant constituer une tâche du système et suivre donc le déroulement normal d'activation d'une tâche.

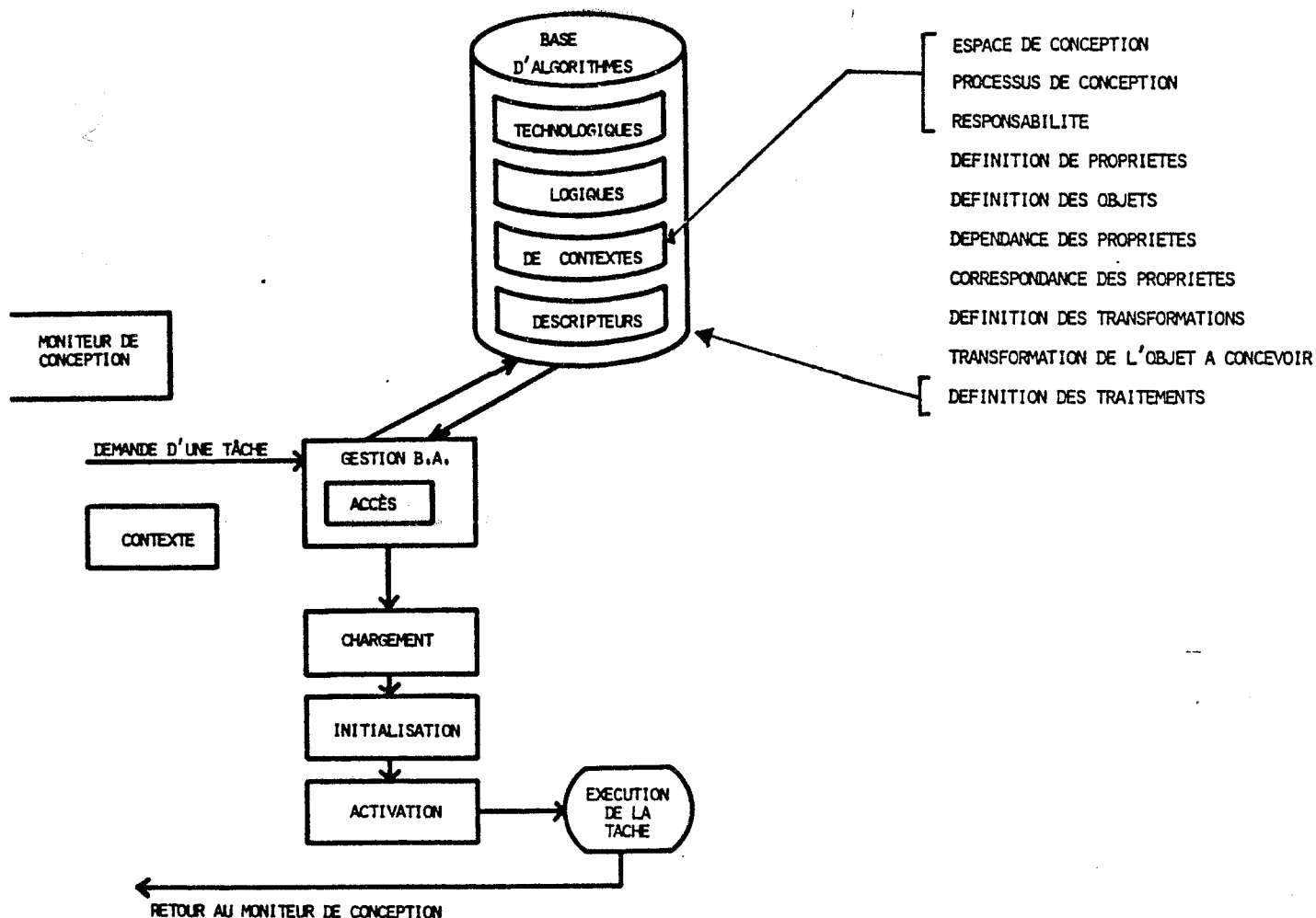


FIGURE II - 10.5 : MISE EN PLACE DE LA BASE D'ALGORITHMES

Précisons mieux comment on bâtit une tâche (FIG.II-10.6). Une tâche interactive s'appuie principalement sur deux modules de base: le moniteur de communication et le moniteur de dialogues. Pour mettre en place une tâche il faut fournir à ces deux moniteurs des informations qui les spécialisent et programmer des parties spécifiques de la tâche.

Le moniteur de communication doit tout d'abord connaître la configuration du poste de travail et les affectations initiales des unités logiques. Puis il faut lui fournir la table de spécialisation de l'analyse sémantique ainsi que la table des messages d'erreurs propres à la tâche.

Le moniteur de dialogues doit recevoir la description des dialogues de la tâche, ainsi que l'ensemble des fonctions sémantiques de la tâche, que le moniteur doit appeler.

Quand on active une tâche, on doit saisir la demande d'activation avec les paramètres associés (c'est le rôle du moniteur de conception), puis on doit initialiser le moniteur de communication et le moniteur de dialogues (c'est le rôle du moniteur de tâches). Ce n'est qu'après cela que l'on peut donner le contrôle au moniteur de dialogues de la tâche qui commande son exécution.

Cette mise en place de se fait pas entièrement manuellement, car elle peut être effectuée au moins en partie par des programmes. La figure II-10.7 montre comment à partir des spécifications en COLA on peut bâtir une tâche.

Ces différents outils seront présentés dans le chapitre suivant.

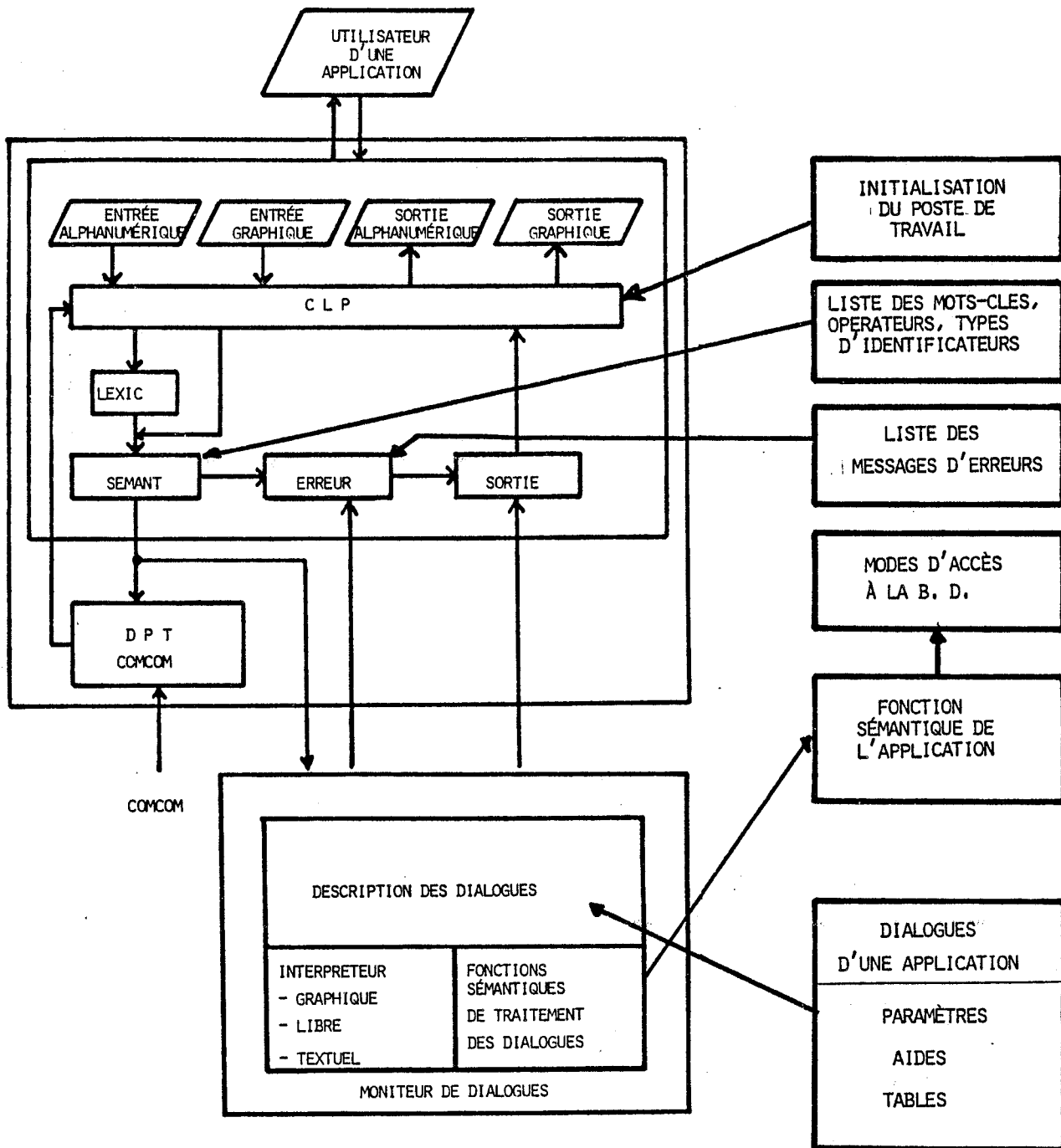


FIGURE II - 10.6 : MISE EN PLACE D'UNE TACHE

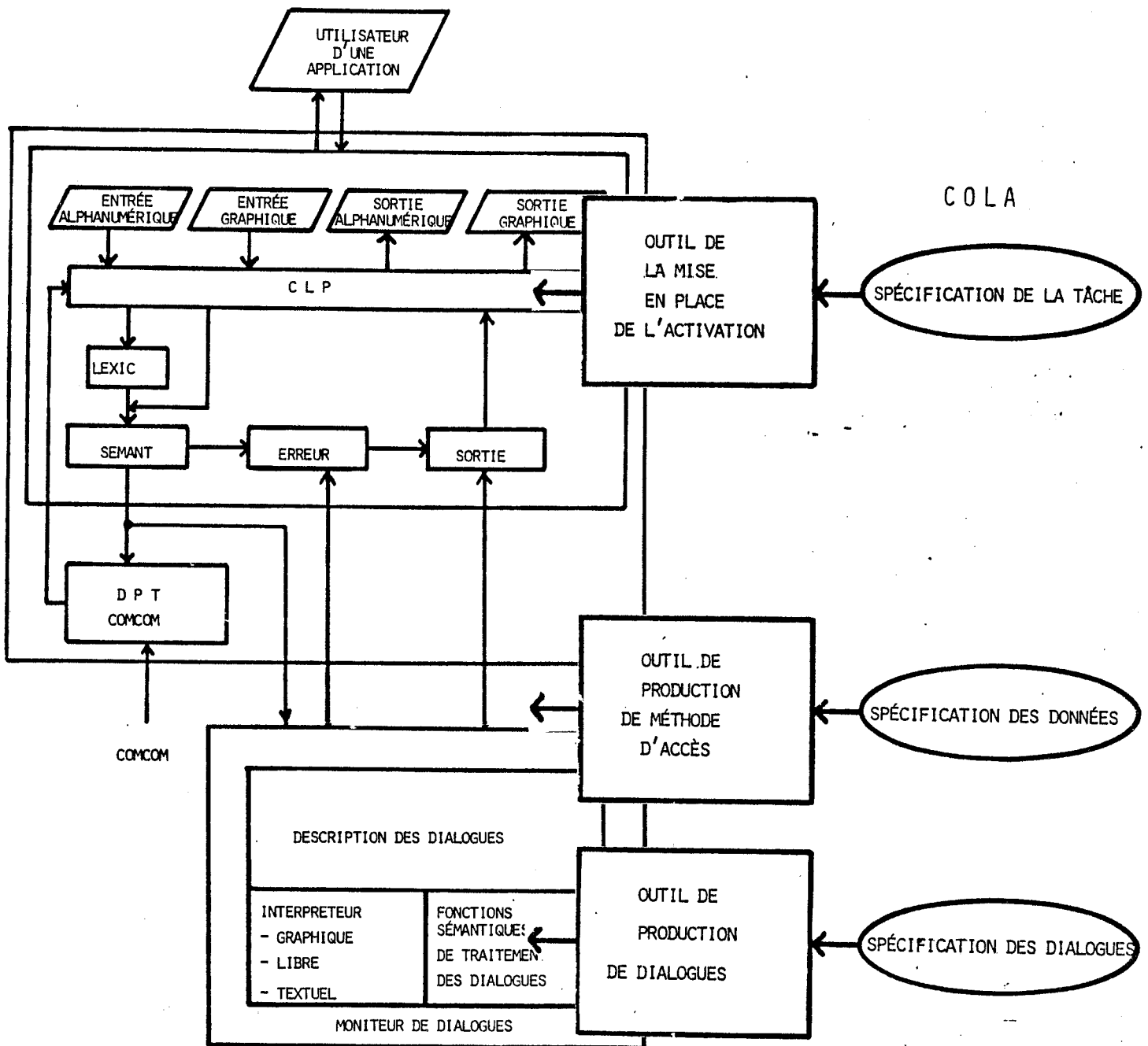


FIGURE - 10.7 : MISE EN PLACE D'UNE TACHE ET DES OUTILS ASSOCIES

10.3 SYSTEME POUR LA CONSTRUCTION DE PROTOTYPES

La construction du prototype peut être assistée par ordinateur. Il semble important de créer un environnement dans lequel on puisse mener cette opération de façon efficace.

Ceci nous conduit à construire un système CAO plus spécialement orienté vers la construction de prototypes. Pour cela deux organisations sont possibles:

- la première - PLATE - se traduit par l'ajout des outils de construction dans la base d'algorithmes et d'un environnement de construction dans le processus de conception,
- la seconde - PLUS STRUCTUREE - traduit beaucoup plus précisément le but du système "conception assistée par ordinateur des systèmes pour la conception assistée par ordinateur". Dans ce cas le système CAO appliqué constitue le but de la conception, dans ce cas le schéma qui traduit le mieux cette démarche est représenté sur la figure II-10.3.

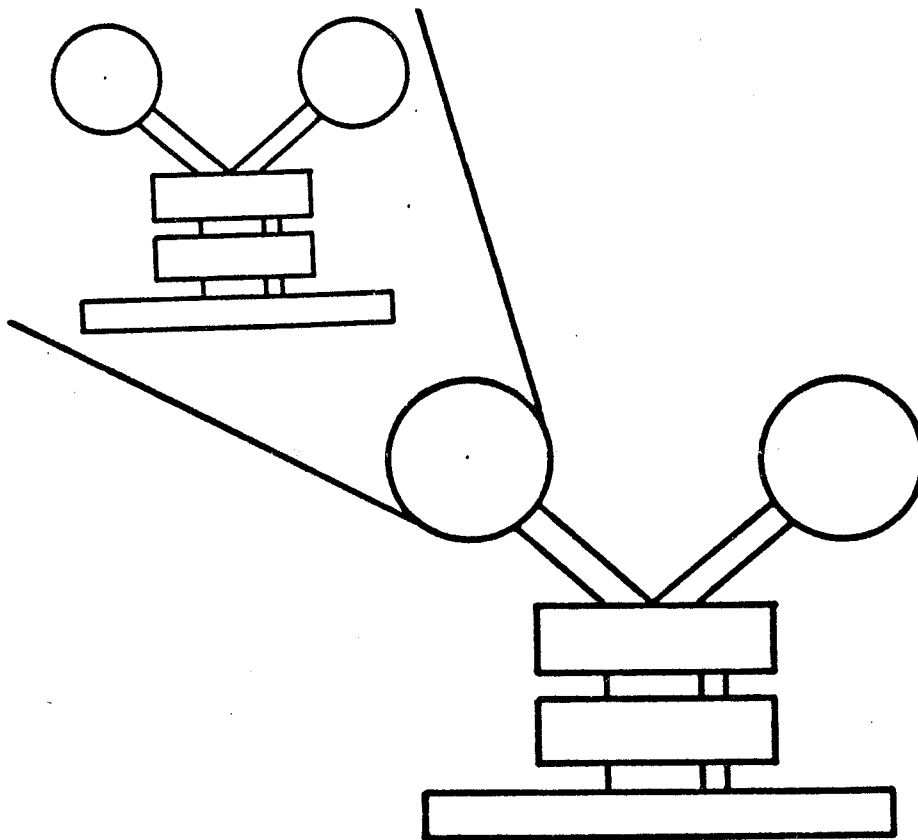


FIGURE II - 10.3 : SYSTEME POUR LA CAO DES SYSTEMES CAO

10.4 LES AUTRES OPERATIONS

La construction d'un prototype à partir des spécifications ne représente qu'une partie de la construction d'un système CAO. Les autres sont:

- le passage du prototype à une réalisation finale. C'est une suite logique de notre méthode de construction. Après avoir vérifié la validité des spécifications fonctionnelles, il s'agit de déterminer le contexte de mise en oeuvre définitive et implanter le système sur l'organisation proposée. C'est l'objet du chapitre 13.
- le passage direct des spécifications (en COLA) à la mise en oeuvre définitive. Pour certaines applications, notamment celles qui semblent simples il est possible d'envisager directement une mise en oeuvre définitive. Il est aussi possible de considérer le système prototype comme un système définitif, si ses performances sont satisfaisantes.

PARTIE III

ASPECTS TECHNIQUES

RESUME: Après avoir étudié dans la première partie les raisons de notre travail et dans la seconde la démarche méthodologique et la solution que nous avons choisi, nous allons présenter maintenant l'aspect technique de la méthode utilisée. Dans un premier temps nous présentons quelques outils que nous avons réalisés. Il s'agit d'une part des outils devant servir dans la construction du prototype, d'autre part des outils généraux qui prendront place dans la base d'algorithmes d'un prototype. Puis nous esquissons les différentes réalisations de la machine abstraite que nous avons effectuées. Finalement nous discuterons de la transition vers une réalisation définitive, c'est à dire le choix approprié des composants de base, ainsi que du problème de la prise en compte de la répartition d'un système CAO.

CHAPITRE 11

LES OUTILS

RESUME: Ce chapitre présente les outils développés afin de compléter la méthode proposée. Il comporte des outils de construction: les différentes phases de traitement de COLA, l'analyse de spécifications, la mise en place des dialogues, des tâches et des données. La notion d'outil logique est introduite: cet outil participe au fonctionnement de la machine abstraite MACAO (par exemple, l'outil assurant la cohérence des données). La notion d'outil général abstrait ou concret est ensuite mise en évidence à l'aide d'exemples (outils graphiques ou non). Le développement de la notion d'outil technologique termine ce chapitre.

11. LES OUTILS

11.1 OUTILS DE CONSTRUCTION

Nous avons constaté dans le chapitre 10 que la construction du prototype peut être assistée par ordinateur. Nous allons présenter ici les outils que nous avons défini et réalisé pour effectivement assister cette construction.

11.1.1 TRAITEMENT DE COLA

Les traitements du langage de spécifications COLA s'effectuent en trois étapes (FIG.III-11.1). La première est une étape de vérification, de documentation et de construction d'une représentation interne. A partir de cette représentation interne d'autres traitements interviennent.

Dans la seconde étape, le but est de mieux cerner les spécifications en les analysant. L'utilisation des techniques d'analyse de données s'est avéré un outil très intéressant. Ainsi on peut soit mieux comprendre les spécifications, soit contribuer à leur amélioration.

Dans la troisième enfin, on réalise à l'aide d'outils particuliers la mise en oeuvre de l'application spécifiée. Ces outils travaillent le plus souvent de façon plus approfondie sur les différentes parties de ces spécifications. C'est ainsi que l'on traite:

- les dialogues homme-machine au niveau du processus CED,
- les dialogues homme-machine au niveau tâche,
- la mise en place des tâches,
- la mise en place des données.

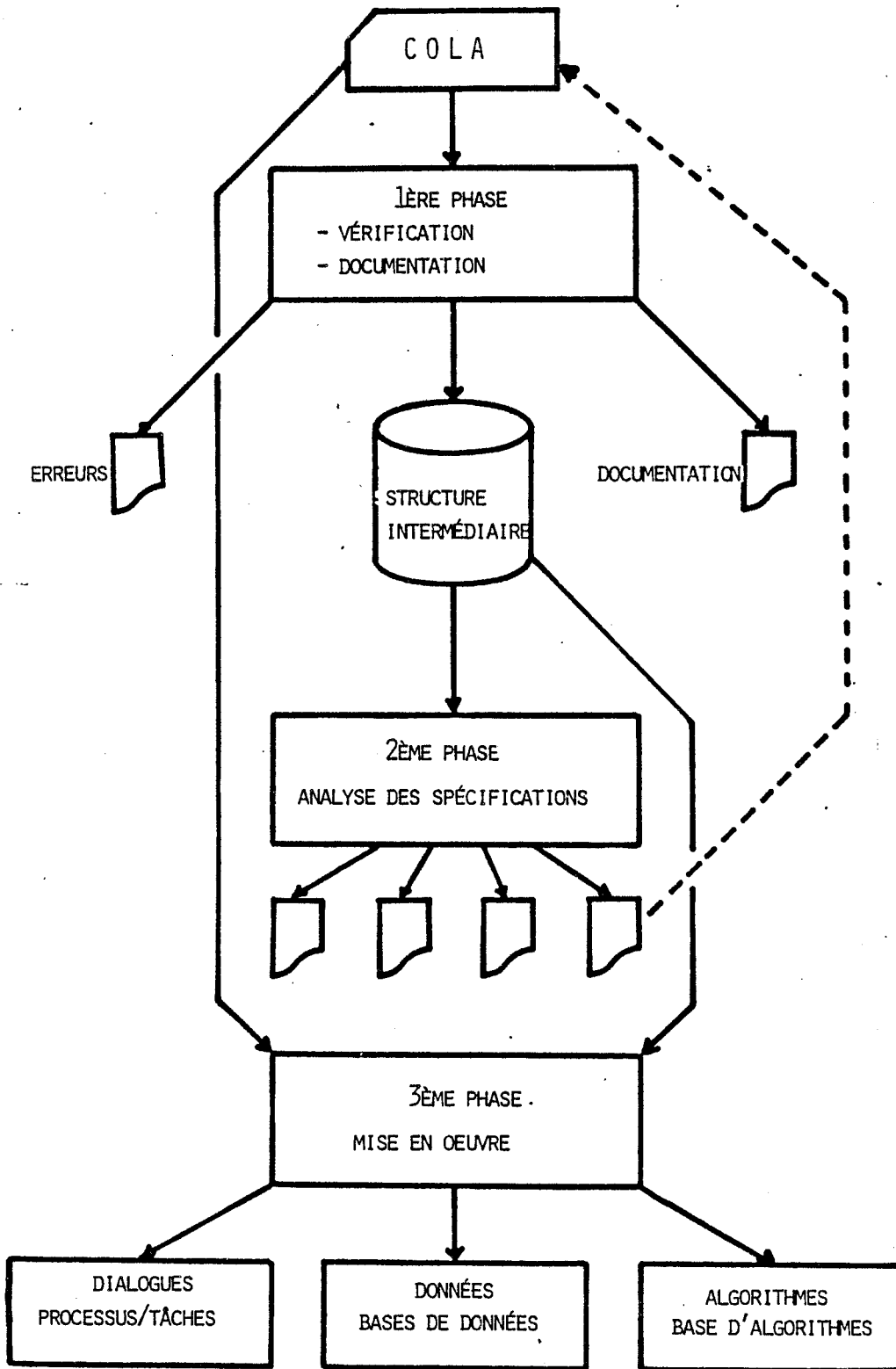


FIGURE III - 11.1 : SCHEMA GLOBAL DE TRAITEMENT DE COLA

En ce qui concerne la première phase du traitement de COLA, elle s'effectue selon l'organisation désormais classique des

compilateurs. Nous nous appuyons sur la démarche et les outils proposés initialement par M.GRIFFITHS et M.PELTIER (Gri.69), constamment entretenus et améliorés par M.DELAUNAY (DeG.80). Il s'agit d'un compilateur dirigé par la syntaxe travaillant de façon descendante. Nous avons opté pour une solution d'interprétation de tables décrivant la grammaire.

Le schéma de production et d'utilisation du traitement de la première étape est représenté sur la figure III-11.2. Ce fonctionnement a été intégré au sein de la MACAO, ce qui a permis d'utiliser les facilités offertes par le système (notamment le module de communication). Nous n'entrerons pas ici dans les détails de réalisation. Montrons plutôt quels sont les résultats de cette première phase (FIG.III-11.3).

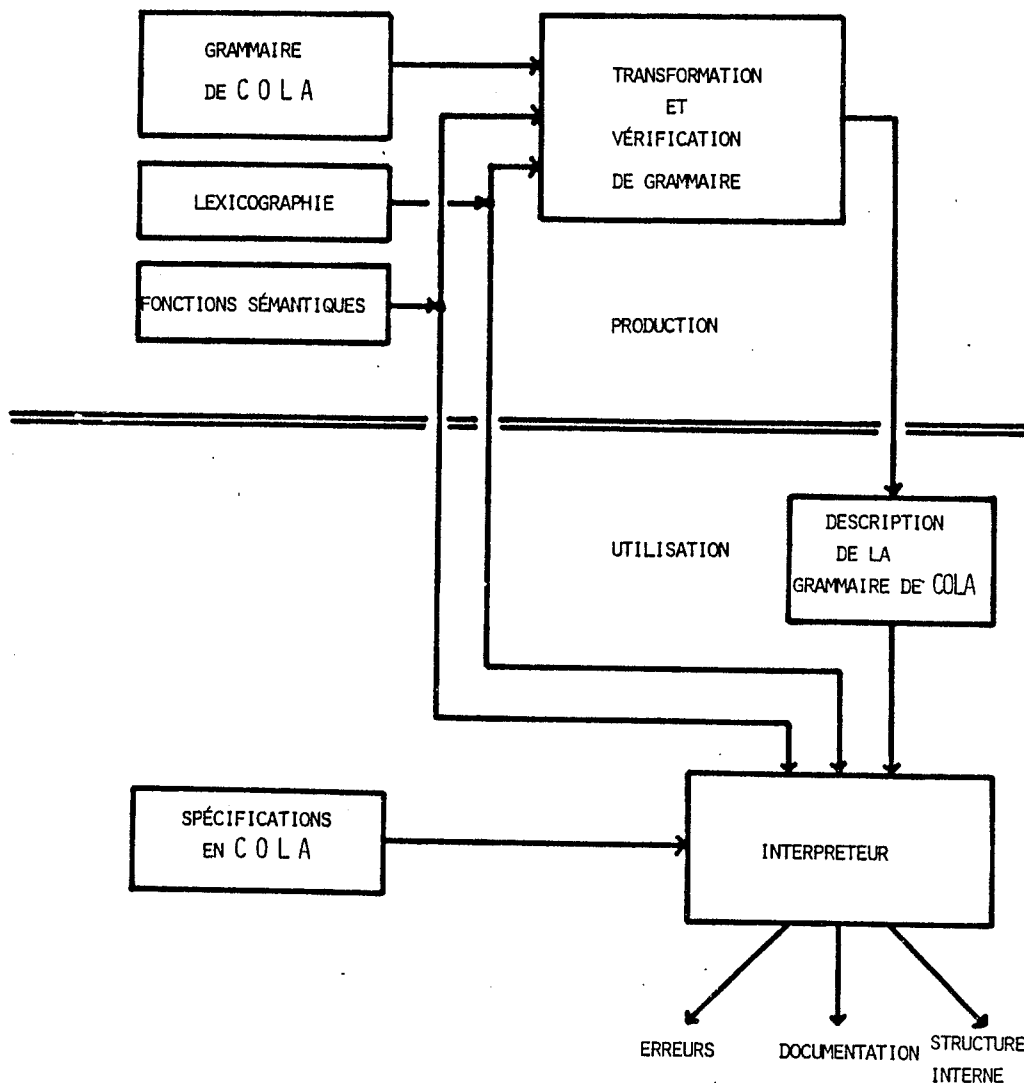


FIGURE III - 11.2 : SCHEMA D'ORGANISATION D'ANALYSE DE COLA

annonce : definition des proprietes

LES OUTILS *****

DÉFINITION DES PROPRIÉTÉS AVEC 204
 TOUTES LES CARACTÉRISTIQUES
 ASSOCIÉES

```

* propriete propre *
-----
* ident. * type *
*propriete * valeur *
-----
* surface = numeriqu *
* avec ----- *
*      *typ.id*      *identificateurs* *
*      ----- *
*      (lectur) ( lect+surf ) *
*      (ecritu) ( ecr+surf ) *
*      (etats) ( tout ) *
*      (fonct.) ( architect ) *
*      ----- *
* contour = tdf *
* avec ----- *
*      *typ.id*      *identificateurs* *
*      ----- *
*      (lectur) ( lectctou , surf+cto , nb+cote+ ) *
*      (ecritu) ( ecr+ctou ) *
*      (etats) ( allocati ) *
*      (fonct.) ( architect ) *
*      ----- *
* schematp = tdf *
* avec ----- *
*      *typ.id*      *identificateurs* *
*      ----- *
*      (lectur) ( lectsch , surf+sch , nb+uct+sc ) *
*      (ecritu) ( ecr+sch ) *
*      (etats) ( tout ) *
*      ----- *

```

```

* app+cto? *
* cout ? *
* nb+piec? *
* nb+appa? *
-----

```

PROPRIÉTÉS UTILISÉES DANS LA SPÉCIFICATION
 MAIS NON DÉFINIES

```

* propriete relation *
-----
* ident. * type *
*propriete * valeur *
-----
* fonc = entier *
* avec ----- *
*      *typ.id*      *identificateurs* *
*      ----- *
*      (lectur) ( lect+fonc ) *
*      (ecritu) ( ecr+fonc ) *
*      (etats) ( prog+arc , prog+arc , allocati ) *
*      (fonct.) ( architect ) *
*      ----- *
* resultan = entier *
* avec ----- *
*      *typ.id*      *identificateurs* *
*      ----- *
*      (lectur) ( lect+res , degre+re ) *
*      (ecritu) ( ecr+res ) *
*      (etats) ( allocati ) *
*      (fonct.) ( architect ) *
*      ----- *

```

```

* releau? *
* rel+hau? *
-----

```

RELATIONS UTILISÉES DANS LA SPÉCIFICATION
 MAIS NON DÉFINIES

FIGURE III - 11.3 : DOCUMENTATION D'UNE SPÉCIFICATION EN COLA (suite.1)

```

enonce : definition des objets
*****

*****
* objets conqus *
-----
* ident. * type * ident. *
* objet *propriete* propriete * valeur initiale
-----
* piece = (propre) surface
*         contour
*         (relati) fonc
*         resultat
* avec -----
* etyp.id* identificateurs
* (lectur) ( lsurf+pi , lcour+pi , lect+fon? , lect+resu )
* (ecritu) ( esurf+pi , ectour+p , ecr+fon? , ecr+resu )
-----
* appart = (propre) surface
*         apparto?
* avec -----
* etyp.id* identificateurs
* (dessus) ( appart )
* (dessous) ( appart , piece )
-----

*****
* objets a concevoir*
-----
* ident. * type * ident. *
* objet *propriete* propriete * valeur initiale
-----
* bat = (propre) cout ?
*         nb+piec?
*         nb+appa?
*         (relati) reiteau?
*         rel+hau?
* avec -----
* etyp.id* identificateurs
* (lectur) ( lcout , lnb+piec , lnb+appa )
-----

enonce : dependance des proprietes
*****

*****
* propriete propre *
-----
* ident * identificateurs de propriete *formule de dependance *
*dependance* *
-----
* depsc * surface , contour => fdepdc *
-----

```

DÉFINITION DES OBJETS CONÇUS ET
DES OBJETS À CONCEVOIR À PARTIR
DES CONSTITUANTS (PROPRIÉTÉS ET
RELATIONS)

EXPRESSION DE LA DEPENDANCE
DES PROPRIÉTÉS

FIGURE III - 11.3 : DOCUMENTATION D'UNE SPECIFICATION EN COLA (suite.2)

```

enonce : definition des correspondances
*****

*****
* synthetique *
-----
* ident de * ident de * ident d' * formule de * SURFACE(NIVEAU APPART)=
*correspon- * propriete * objet * correspondance * SOMME DES SURFACES(NIVEAU
*dance * * * * PIÈCE)
-----
* corrsaur ( surface de piece ) *
* ( surface de appart ) *
-----

enonce : definition des transformations
*****

*****
* etat *
-----
*transfor- * identificateurs *formule de *
*ation * etat * etat * propriete * propriete *transformations*
-----
* trespa * allocati => prog+arc * contour dev contour? par prog+cha *
* * * * nge+rept *
* * * * spatiale *
-----

enonce : definition des traitements
*****

*****
* traitement production *
-----
* ident. * ident. * ident. *
*traitement* dialogue *algorithm.* description du contenu
-----
* spa * * psa * allocation spatiale avec contour de piec *
* * * * es et une seule relation a satisfaire *
* avec -----
* *typ.id* identificateurs
* -----
* (lectur) ( lctour+p , lec+res? )
* (ecritu) ( ecr+sch? )
* (etats) ( allocati )
* (fonct..) ( architec )
* -----

```

FIGURE III - 11.3 : DOCUMENTATION D'UNE SPECIFICATION EN COLA (suite)

11.1.2 ANALYSE DES SPECIFICATIONS

Ayant obtenu des spécifications correctes (mais pas forcément complètes), c'est à dire ayant pu être saisies correctement par la première étape du traitement de COLA, il s'est avéré utile d'effectuer un ensemble d'analyses sur ces spécifications.

Il s'agit tout d'abord d'un ensemble de statistiques sur l'utilisation de divers éléments spécifiés au sein du système CAO décrit. C'est ainsi qu'il semble important de connaître l'utilisation:

- des atomes d'information, où (dans quelles étapes) ils sont produits et où ils sont consommés, éventuellement faire ressortir des lacunes (atomes non produits, non consommés),
- des tâches, où sont elles utilisées, combien fois et par qui,
- des utilisateurs, leurs champ d'action et leurs prérogatives (accès aux données, possibilités de se servir des tâches),
- etc...

Dans un deuxième temps, il est utile de fournir des analyses plus approfondies concernant notamment les relations entre les quatre parties fondamentales des spécifications qui sont: les utilisateurs, le processus de conception, les atomes d'information et les traitements. Ainsi en les étudiant deux à deux:

- utilisateurs - processus,
- utilisateurs - atomes d'information,
- utilisateurs - traitements,
- processus - atomes d'information,
- processus - traitements,
- atomes d'information - traitements,

nous pouvons mieux apprécier leur liens ainsi que leur comportement vis à vis des composantes qui ne sont pas prises en compte dans l'analyse.

La présence dans les spécifications de ces quatre composantes n'est pas indispensable dès le départ. Bien au contraire, il est intéressant de pouvoir dégager les composantes manquantes à partir des informations disponibles. Par exemple, à partir de la relation atomes d'information - traitements, il est possible de suggérer l'organisation du processus de conception, de même qu'à partir de la relation utilisateurs - traitements. C'est la synthèse de ces informations et la vérification de leur

conformité avec la réalité pratique du domaine d'application qui est à faire lors de la spécification du système CAO.

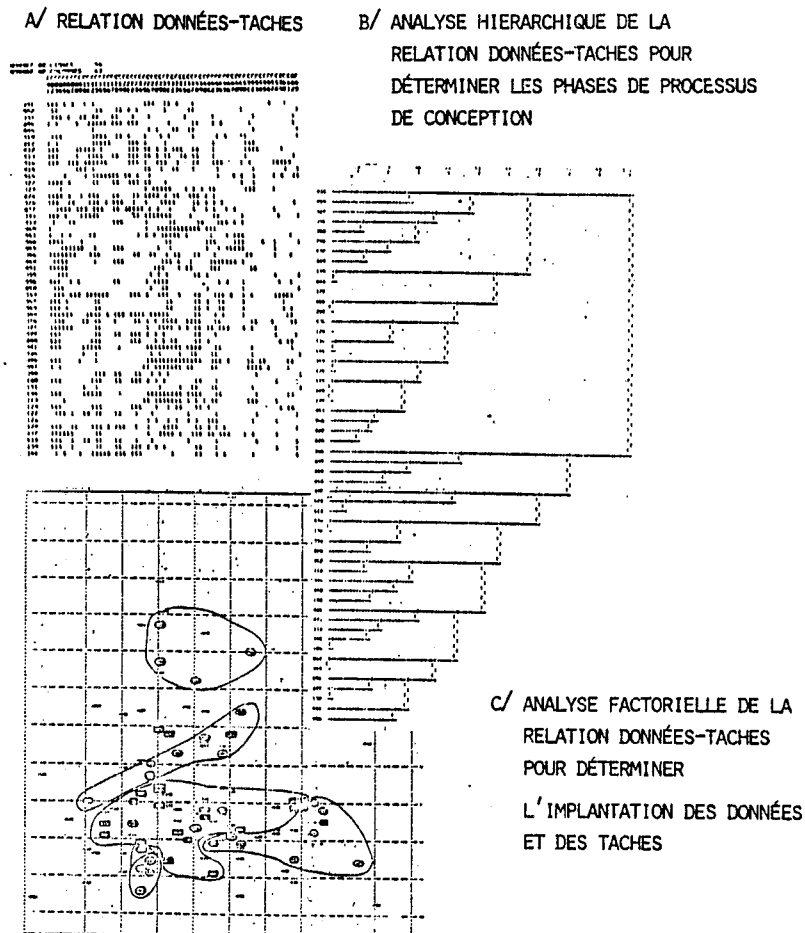


FIGURE III - 11.4 : EXEMPLES D'ANALYSE DES SPECIFICATIONS

11.1.3 TRAITEMENT DES DIALOGUES

Le premier des traitements de la troisième phase concerne la mise en place des dialogues. Comme nous avons pu le constater, dans un système CAO les dialogues apparaissent fondamentalement en deux niveaux. Au niveau du processus de conception, où ils sont gérés par le moniteur de conception, et au niveau d'une tâche du processus où ils s'effectuent de façon interne à la tâche.

Nous avons pu établir qu'un mécanisme unique peut, dans certains cas (voir choix organiques), gérer les deux niveaux de dialogues grâce au moniteur de dialogues. C'est de sa mise en place et de son fonctionnement que nous discutons ici.

11.1.3a MISE EN PLACE DES DIALOGUES AU NIVEAU TACHE

Les dialogues d'une tâche se trouvent décrits dans le sous-énoncé DIALOGUES de l'énoncé DEFINITION DE TRAITEMENT. La mise en place de dialogue consiste donc essentiellement en la traduction des spécifications sous une forme exploitable par le moniteur de dialogues. Le moniteur de dialogues dont nous avons déjà analysé la structure dans le chapitre 6, est principalement un interpréteur de dialogues. Sa structure obéit au schéma classique, c'est à dire une table des dialogues autorisés, un interpréteur associé et un ensemble de fonctions sémantiques correspondant aux actions associées.

La mise en place s'effectue par la traduction d'une spécification externe en une forme exploitable. Comme nous avons voulu rendre utilisable cette méthode de mise en oeuvre des dialogues indépendamment des autres aspects de COLA, nous avons créé une première organisation (FIG.III-11.5).

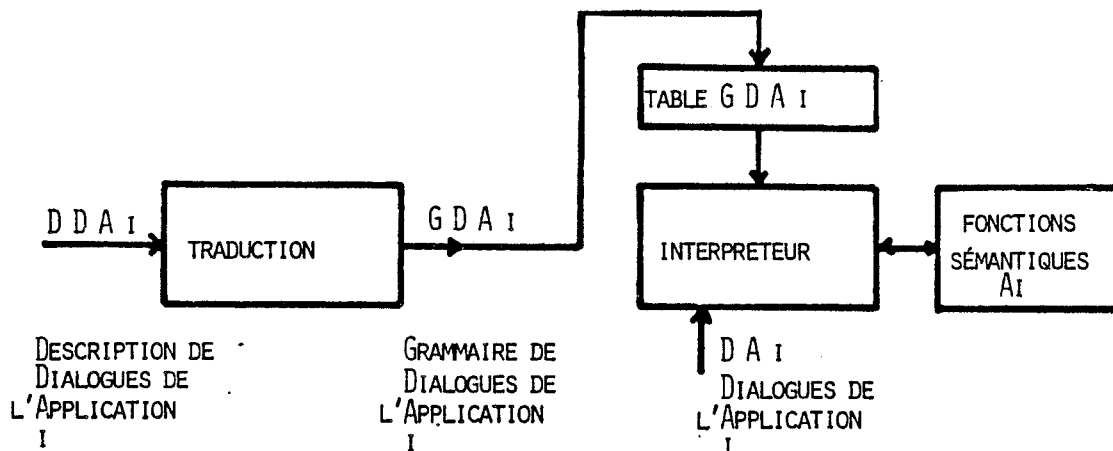


FIGURE III - 11.5 : PREMIERE VERSION DU TRAITEMENT DES DIALOGUES

Vu la spécificité du langage d'expression des dialogues et de la structure syntaxique des dialogues autorisés, nous avons opté pour les traitements spécifiques.

Pour la mise en oeuvre, nous avons réalisé un traducteur particulier des spécifications de dialogues et un interpréteur spécialisé qui génère une structure de données interprétée ensuite par l'interpréteur spécialisé des dialogues.

Finalement, au vue de la multiplication des interpréteurs spécialisés, nous avons choisi dans un deuxième temps, d'utiliser un interpréteur général. Ceci a quelque peu compliqué l'organisation (FIG.III-11.6), car nous avons été amené à

travailler dans la phase de mise en oeuvre en deux étapes car nous avons systématisé la production du moniteur de dialogues. En outre cette structure nous a permis de réaliser assez facilement les différents modes de dialogues: textuel ou graphique, tutorial ou libre.

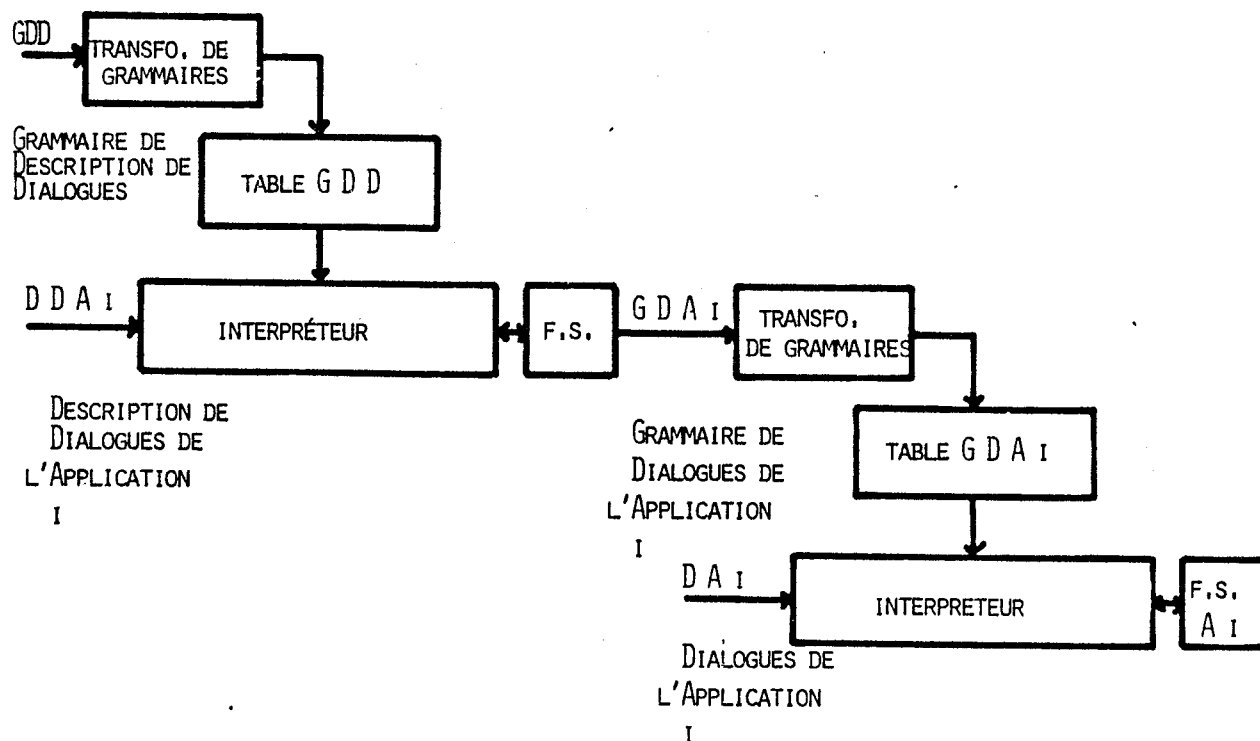


FIGURE III - 11.6 : DEUXIEME VERSION DU TRAITEMENT DES DIALOGUES

11.1.3b MISE EN PLACE DES DIALOGUES AU NIVEAU DU PROCESSUS DE CONCEPTION

Au niveau du processus de conception, les dialogues ne sont pas sous une forme directement exploitable par le traducteur de mise en oeuvre des dialogues. Pour cela il faut donc faire un pré-traitement qui à partir de la description du processus de conception et des tâches activables dans chacun des états, produit la spécification des dialogues sous forme LDC.

La suite des opérations s'effectue de la même façon que pour les dialogues internes à la tâche.

Par contre si on ne peut pas utiliser cette méthode de gestion des dialogues, notamment pour des raisons organiques, il est possible de les gérer directement par le moniteur de tâches. C'est le cas du moniteur de conception implicite.

11.1.4 TRAITEMENT DES TACHES

La mise en place des tâches est un autre traitement que l'on effectue à partir des spécifications en COLA. Ce traitement a déjà été décrit de façon assez précise dans le chapitre 6.

11.1.5 TRAITEMENT DES DONNEES

La mise en place des données dans le système CAO signifie de façon plus précise, la mise en place des structures des données et des modes d'accès à des données.

Il s'agit donc d'interpréter les énoncés concernés de COLA et de générer des ordres de spécialisation, par exemple, de la Base de Données SOMINE. De plus, l'analyse de l'énoncé TRAITEMENT doit permettre la construction du mode d'accès spécifique, selon le schéma présenté dans le chapitre 5.

11.2 LES OUTILS LOGIQUES

Les outils logiques sont les outils qui participent au bon fonctionnement de la machine abstraite MACAO, quelque soit l'application que le système CAO supporte. Nous avons présenté dans les chapitres précédents des outils tels que le SGBD, le SGBA, le moniteur de communication et le moniteur de dialogues, nous allons introduire l'outil assurant la cohérence.

11.2.1 COHERENCE

Un des problèmes fondamentaux dans la conception est le maintien de la cohérence d'un ensemble d'informations sur l'objet en cours de conception (objet à concevoir, objet conçu) tout au long du processus de conception. Il est important qu'en CAO le système CAO puisse gérer cette cohérence.

Pour cela il faut satisfaire deux conditions:

- travailler, dans la mesure du possible, avec des informations non redondantes, c'est à dire choisir des atomes d'information qui ne se superposent pas, mais qui s'assemblent pour constituer des ensembles plus importants,
- exprimer clairement les types de lien de cohérence et systématiser ainsi leur gestion.

Dans COLA nous avons défini un ensemble de types de liens qui sont:

- dépendance locale des propriétés du même objet,
- correspondance globale entre propriétés des objets différents,
- transformation des propriétés lors du cheminement dans le processus de conception,
- accessibilité conceptuelle,
- accessibilité organisationnelle.

Nous montrons maintenant comment ces informations sont utilisées au sein du système CAO:

- les deux derniers types de cohérence ne changent pas pendant le déroulement du projet,
- la transformation est activée lors du changement de phase ou d'étape et constitue une opération de modification, elle peut donc entraîner l'évaluation de la cohérence,
- la dépendance et la correspondance sont des informations élémentaires sur la vie des propriétés, elles doivent donc être réévaluées après chaque opération de modification et de transformation.

Pour cela il faut contrôler l'évolution de la structure de l'objet. Nous avons construit un outil d'évaluation qui gère la vie de l'objet et assure sa cohérence. On fournit d'abord à cet outil, l'ensemble des liens de cohérence tels qu'ils apparaissent dans les spécifications en COLA. Puis au fur et à mesure de l'évolution de l'objet on construit une structure exacte de l'objet à un instant donné.

Lorsqu'une opération de modification ou de transformation remet en cause la structure actuelle de l'objet, l'outil réévalue l'ensemble des liens sémantiques entre propriétés. Pour cela il faut d'abord indiquer quelles sont les informations qui sont invalidées par l'opération, puis les recalculer.

Dans la réévaluation, il faut tenir compte de la nature des liens (synthétisé, hérité) et évaluer d'abord les liens globaux (correspondances), puis les liens locaux (dépendances). Pour assurer que le processus d'évaluation converge, il faut vérifier au préalable la cohérence et la complétude des règles de cohérence exprimées en COLA.

11.3 LES OUTILS GENERAUX (outils concrets et abstraits)

Les outils généraux sont des outils qui participent au sein du système CAO, à la réalisation d'une application précise. Leur caractère général leur permet d'intervenir dans des applications variées. On distinguera:

- outils concrets,
- outils abstraits.

Les outils concrets sont les outils dont la sémantique est précise mais dont le champ d'application est suffisamment grand pour se retrouver dans de nombreuses applications. C'est le cas, par exemple, des outils graphiques dans lesquels la géométrie bi- ou tri-dimensionnelle constitue une sémantique précise avec un champ d'application très grand.

Les outils abstraits sont des outils qui ne prennent pas en compte la sémantique de l'application et travaillent de façon plus structurelle. C'est le cas, par exemple, des outils d'analyse de données ou encore de tout outil de calcul.

Nous allons présenter rapidement quelques outils généraux types.

11.3.1 OUTILS GRAPHIQUES

Les outils graphiques ont pour rôle soit de produire des dessins, soit de construire puis produire ces dessins. C'est une informatisation du travail de dessinateur. Lors de la construction on s'appuie sur les éléments classiques du dessin (droite, cercle, tangente,...) ou sur des dessins de base prédéfinis.

Leur généralité est aussi grande que celle du dessin qu'ils permettent de réaliser.

11.3.2 OUTILS GRAPHIQUES DE MODELISATION

En CAO l'objectif n'est pas de manipuler des entités graphiques et de produire des dessins. Le but est de construire l'objet et pour cela il est nécessaire de manipuler des entités ayant une signification vis à vis de l'objet et éventuellement de sa réalisation. Il faut donc pouvoir manipuler l'ensemble des caractéristiques graphiques ou non graphiques, dans le but d'élaborer une représentation virtuelle de l'objet. Cette représentation doit prendre en compte tous les aspects de l'objet (sa structure, ses caractéristiques physiques, mécaniques, ... et graphiques). A partir de cette représentation on peut alors entreprendre différents traitements et entre autres, la production de représentations graphiques.

A titre d'exemple nous décrivons ici rapidement les possibilités du programme EGIDD, que nous avons développé.

EGIDD (Editeur Graphique Interactif en Deux Dimensions) est un logiciel permettant la construction de "scènes" en deux dimensions. Une scène est un ensemble composé à partir d'objets préalablement définis. Ces objets peuvent être soit des objets de base, soit des objets composés.

Le champ d'application d'EGIDD est celui de la schématique (électrique, logique, électronique,...), de la symbolique (réseau de feux rouges,...) et de l'utilisation d'éléments prédéfinis (plans basés sur les systèmes constructifs en architecture, ...).

EGIDD se compose de 6 modules principaux:

- un module appelé "DEFINITION" qui permet la définition des objets de base. Lors de cette définition on précise la représentation graphique de l'objet, et on indique les paramètres graphiques et non graphiques qui caractérisent l'objet.
- un module appelé "CONSTRUCTION" qui permet la construction d'une scène à partir d'un catalogue. Pour un objet du catalogue on crée une occurrence qui est déterminée par les valeurs de tous les paramètres.
- un module appelé "CONNEXION" qui permet la construction semi-automatique de connexions entre objets de la scène.
- un module appelé "HABILLAGE" qui associe à la scène des informations graphiques et textuelles correspondant à la finition du dessin (cartouche, commentaires,...).
- un module appelé "GESTION DE BIBLIOTHEQUES" qui assure les transferts d'informations (catalogues, scène, habillage) entre la mémoire principale et la mémoire secondaire.
- un module appelé "INTERFACE PROGRAMME D'APPLICATION" qui assure la connexion avec les programmes de l'utilisateur voulant exploiter certaines informations contenues dans la scène (calcul de devis,...).

Ce programme peut prendre place au sein du système SIGMA-CAO et peut donc bénéficier de trois versions d'interface pour la gestion des bibliothèques (catalogue, scène, habillage): fichiers séquentiels, fichiers à accès direct et connexion avec un système de gestion de base de données.

La figure III-11.7 présente un schéma type obtenu avec EGIDD.

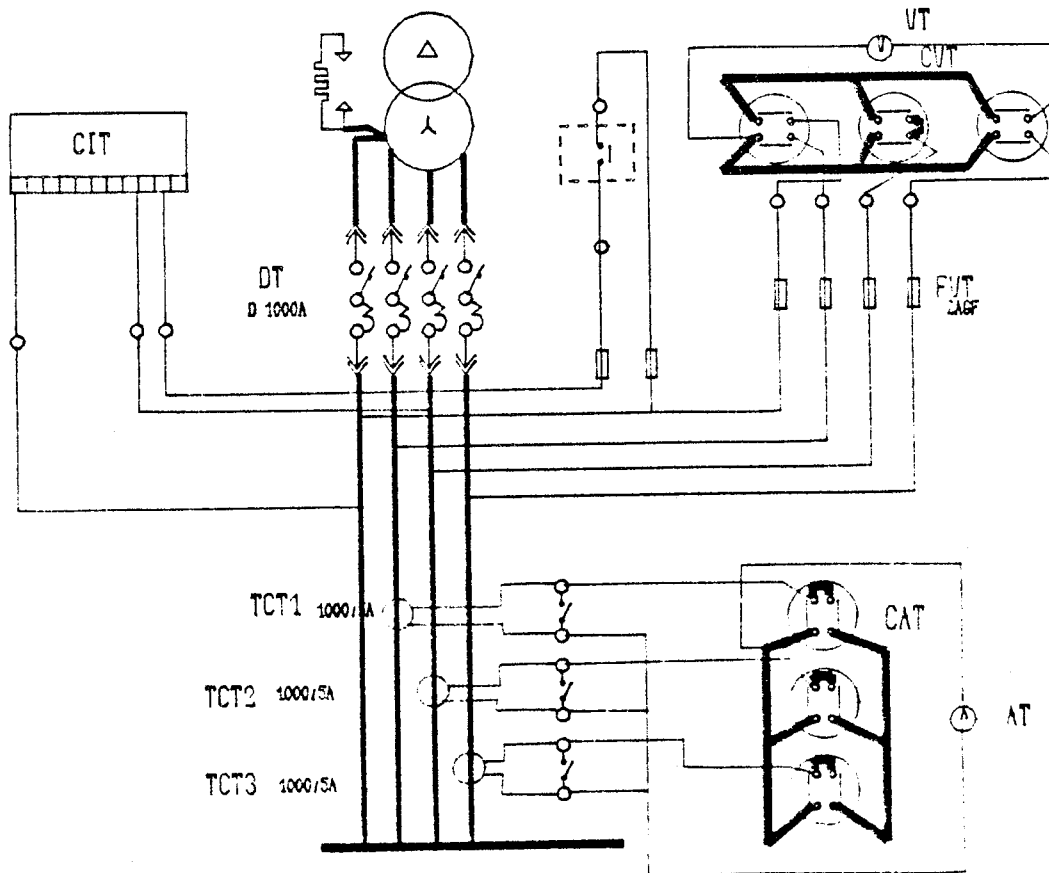


FIGURE III - 11.7 : APPLICATION D'EGIDD AU SCHEMA ELECTRIQUE

11.3.3 OUTILS NON GRAPHIQUES

Outre des outils d'analyse de données dont nous avons déjà présenté les avantages y compris dans la conception d'un système CAO, nous voulons évoquer ici l'intérêt d'un outil interactif d'évaluation de liens mathématiques entre un ensemble de variables.

Il s'agit en fait de mettre à la portée des utilisateurs, sous une forme agréable à manipuler, le module utilisé dans la gestion de la cohérence des informations.

C'est un système Questions - Réponses dans lequel l'utilisateur définit d'abord l'ensemble de variables dont il veut se servir, et un ensemble de liens exprimant les relations entre ces variables. Le système construit une représentation que l'utilisateur peut interroger en indiquant les valeurs qu'il donne aux variables et l'outil calcule toutes les valeurs de toutes les variables possibles.

SIGMA-EVAL A VOTRE SERVICE

"ic"

- LC LISTE DE COMMANDES
- .NV NOMS DES VARIABLES
- NC NOMS DES CONSTANTES
- TL TABLEAU DE LIENS
- LR LISTE DE RELATIONS
- NI NOMS DES INCONNUES
- VV VALEURS DES VARIABLES
- AV AFFECTATION DE VALEURS AUX VARIABLES
- .RI RENDRE INCONNUES DES VARIABLES
- EV EVALUATION DU NOUVEAU SYSTEME
- FI FIN D'UTILISATION
- TR TRACE D'EXECUTION
- .NT NON TRACE D'EXECUTION
- CE CONTINUER EVALUATION
- .LV L'AVANT
- NB NON L'AVANT

LISTE DE COMMANDES DE
SIGMA-EVAL

"nv"

- 1 FXTT
- 2 FXTTIA
- 3 FXTTIO
- 4 FXTTIC
- 5 FXTTIF
- 6 SLOTT
- 7 SLOTTI
- 8 SLOTTO
- 9 SLOTTIC
- 10 SLOTTIF
- 11 NDLXTT
- 12 NDLXTTIA
- 13 NDLXTTIO
- 14 NDLXTTIC
- 15 NDLXTTIF
- 16 RRSFTT
- 17 RRSFTTIA
- 18 RRSFTTIO
- 19 RRSFTTIC

DÉFINITION D'UN EXEMPLE:
CHOIX DE NOMS DES VARIABLES
UTILISÉES

"nc"

- 1 FSI 210
- 2 FSI 210
- 3 FSAFF
- 4 SSAFE
- 5 RRSFTT
- 6 RRSFTTIA
- 7 RRSFTTIO
- 8 IFL

NOMS DES CONSTANTES UTILISÉES

FIGURE III - 11.8: EXEMPLE D'UTILISATION DU PROGRAMME D'EVALUATION EVAL

"1"

```

1      1 PXTT = PXTTIX * IPC ;
      2 PXTT = FXM2HO * SHOTT ;
      3 PXTT = PXM2HB * SHBT ;
      4 PXTT = FXAFM * NEAFTT ;
      5 FXAFM = SHEAFM * PXM2HB ;
      6 SHOTT = SHBMHB * NBNXHB ;
      7 SHBT = SHBAPM * NEAFTT ;
      8 NEAFTT = APSHNB * NBNXHB ;
      9 NBFCTT = NEAFTT * RRPCAM ;
     10 RRSFTT = SHOTT / SHBT ;
     11 RRSFTT = SHONHB / SHBMHB ;
     12 SHONHB = SHEAFM * APSHNB ;
     13 NBNXHB = NBAFTT / APSHNB ;
     14 PXTT = SHOTT * FSM2HO * IFD ;
     15 PXTT = SHBT * FSM2HB * IFD ;
     16 PXTTIX = SHOTT * FSM2HO ;
     17 PXTTIX = SHBT * FSM2HB ;
     18 PXTTIX = NEAFTT * FSAPM ;
     19 SHBT = NEAFTT * SSAPM ;
     20 SHOTT = SHBT / RRSFTT ;
     21 SHONHB = SHBMHB * RRSFTT ;
     22 SHBMHB = APSHNB * SSAPM ;
     23 NBFCTT = NEAFTT * RSFCAM ;
     24 FXAFM = SHEAFM * FSM2HB * IFD ;
     25 FXM2HO = FSM2HO * IFD ;
     26 PXM2HB = FSM2HB * IFD ;
     27 FXAFM = FSAPM * IFD ;
     28 NEAFTT = SHOTT * RRSFTT / SSAPM ;
     29 FCAPM = ((SHEAFM-14.4)/15.8) ;
  
```

LISTE DES RELATIONS:
 DÉFINITION DU SYSTÈME DE
 RELATIONS ENTRE LES VARIABLES
 ET LES CONSTANTES

"2"

TABLEAU LIENS

```

000000001111111111
1234567890123456789
.01|
.02| |
.03| | |
.04| | | |
.05| | | | |
.06| | | | | |
.07| | | | | | |
.08| | | | | | | |
.09| | | | | | | | |
.10| | | | | | | | |
.11| | | | | | | | | |
.12| | | | | | | | | |
.13| | | | | | | | | |
.14| | | | | | | | | |
.15| | | | | | | | | |
.16| | | | | | | | | |
.17| | | | | | | | | |
.18| | | | | | | | | |
.19| | | | | | | | | |
.20| | | | | | | | | |
.21| | | | | | | | | |
.22| | | | | | | | | |
.23| | | | | | | | | |
.24| | | | | | | | | |
.25| | | | | | | | | |
.26| | | | | | | | | |
.27| | | | | | | | | |
.28| | | | | | | | | |
.29| | | | | | | | | |
  
```

TABLEAU DE LIENS:
 VISION STRUCTURELLE DU SYSTÈME;
 VARIABLES UTILISÉES DANS CHAQUE
 RELATION

FIGURE III - 11.8: EXEMPLE D'UTILISATION DU PROGRAMME D'EVALUATION EVAL (suite.1)

```

"ev"

AFFECTATION DE VARIABLES: "10" VALEUR
POUR TERMINER TAPER "FIN"
"pextt" 1500000

OK
"fin"

"ev"

NOUVEAU SYSTEME

VALEURS DES VARIABLES:
.PATT      1499999.
.PATT1A    500000.0
.PW2HO     720.0000
.PW2HB     1200.000
.PWAPM     72000.00
.SI.CTT    2083.333
.SI.HET    1250.000
.SI.HAPM   00.00000
.HEAPTT    20.83333
.HEPCCT    00.41604
.HEPCFT    1.000007
.HEPCAM    2.899999
.PCAM1     2.888075

"hi"

1          8  SHOTB
           9  SHOTB
           11  HAPAT
           12  HAPAT
           13  HAPAT
           17  HAPAT

"fi"
    
```

EXEMPLE D'ÉVALUATION DU SYSTÈME

AFFECTATION DE VALEURS AUX VARIABLES CHOISIES
UNE SEULE VALEUR A ÉTÉ FIXÉE

EVALUATION DU SYSTÈME:
LE SYSTÈME A PU ÊTRE PARTIELLEMENT
ÉVALUÉ, VOICI LE RESULTAT

NOMS DES INCONNUES:
LISTE DES VARIABLES QUI N'ONT PAS
PU ÊTRE VALUÉES

FIN D'UTILISATION

FIGURE III - 11.8: EXEMPLE D'UTILISATION DU PROGRAMME D'EVALUATION EVAL (suite.2)

Le champ d'application de cet outil est important car il s'agit d'un outil abstrait qui peut être spécialisé dans chaque domaine d'application.

11.4 OUTILS TECHNOLOGIQUES

Les outils technologiques sont les outils directement liés à une application particulière ou de façon plus précise à une technologie déterminée. Leur champ d'activité est évidemment très réduit, par contre leur finesse de traitement peut être très grande. De nombreux exemples seront explicités dans les annexes de ce document.

CHAPITRE 12

DIFFERENTES REALISATIONS DE LA MACAO

RESUME: Ce chapitre présente trois différentes réalisations de la machine abstraite pour la CAO. Pour chacune d'elles, on précise son objectif, son contexte, les choix principaux qui ont été faits et les principes de réalisation. La première réalisation, le système MINI-SIGMA, est un système minimum fonctionnant sur une grosse machine. Le système APPLI-SIGMA, système plus complet, fonctionne sur un mini-ordinateur. Le système VRAI-SIGMA fonctionne sur un gros ordinateur: c'est la version la plus étoffée mais ne correspondant pas nécessairement à ce que l'on peut faire de mieux en la matière.

12. DIFFERENTES REALISATIONS DE LA MACAO

Nous avons entrepris trois réalisations différentes de la machine abstraite pour la CAO (MACAO) dans des contextes distincts. Dans ces réalisations nous avons adopté des choix différents pour la mise en oeuvre des diverses parties de la machine abstraite tout en respectant la définition fonctionnelle globale et unique.

Nous allons présenter successivement ces diverses réalisations. Il s'agit tout d'abord du système MINI-SIGMA, puis d'APPLI-SIGMA et enfin de VRAI-SIGMA.

12.1 MINI-SIGMA

12.1.1 OBJECTIF

L'objectif du système MINI-SIGMA est de constituer un support minimum pour des applications CAO. Ce système a été plus particulièrement utilisé dans la réalisation du système SIGMA-ARCHI, le système intégré pour la CAO en architecture (voir chapitre 14).

12.1.2 CONTEXTE

Le contexte de réalisation (FIG.III-2.1) est une grosse machine (IBM 360/67), fonctionnant en temps partagé, permettant une utilisation conversationnelle et s'appuyant sur la notion de machine virtuelle (CP/CMS).

Le poste de travail est un poste de travail graphique avec un terminal à balayage cavalier, une tablette de digitalisation et un clavier de touches de fonctions (IBM 2250).

12.1.3 CHOIX

Les choix principaux que nous avons déterminés sont schématisés sur la figure III-12.2. Fonctionnellement ils concernent:

- la BASE DE DONNEES: une gestion basée sur des fichiers à accès séquentiel,

- la BASE D'ALGORITHMES: le bibliothécaire CMS,
- le MONITEUR DE CONCEPTION: un moniteur exécutable basé sur EXEC-CMS et utilisant l'allocation statique des tâches dans un environnement paginé,
- le MONITEUR DE DIALOGUES: un interpréteur spécialisé,
- le MONITEUR DE COMMUNICATION: un interpréteur standard.

Dans la réalisation, nous avons choisi pour des raisons historiques, de travailler en ALGOLW et pour la communication graphique en GRIGRI.

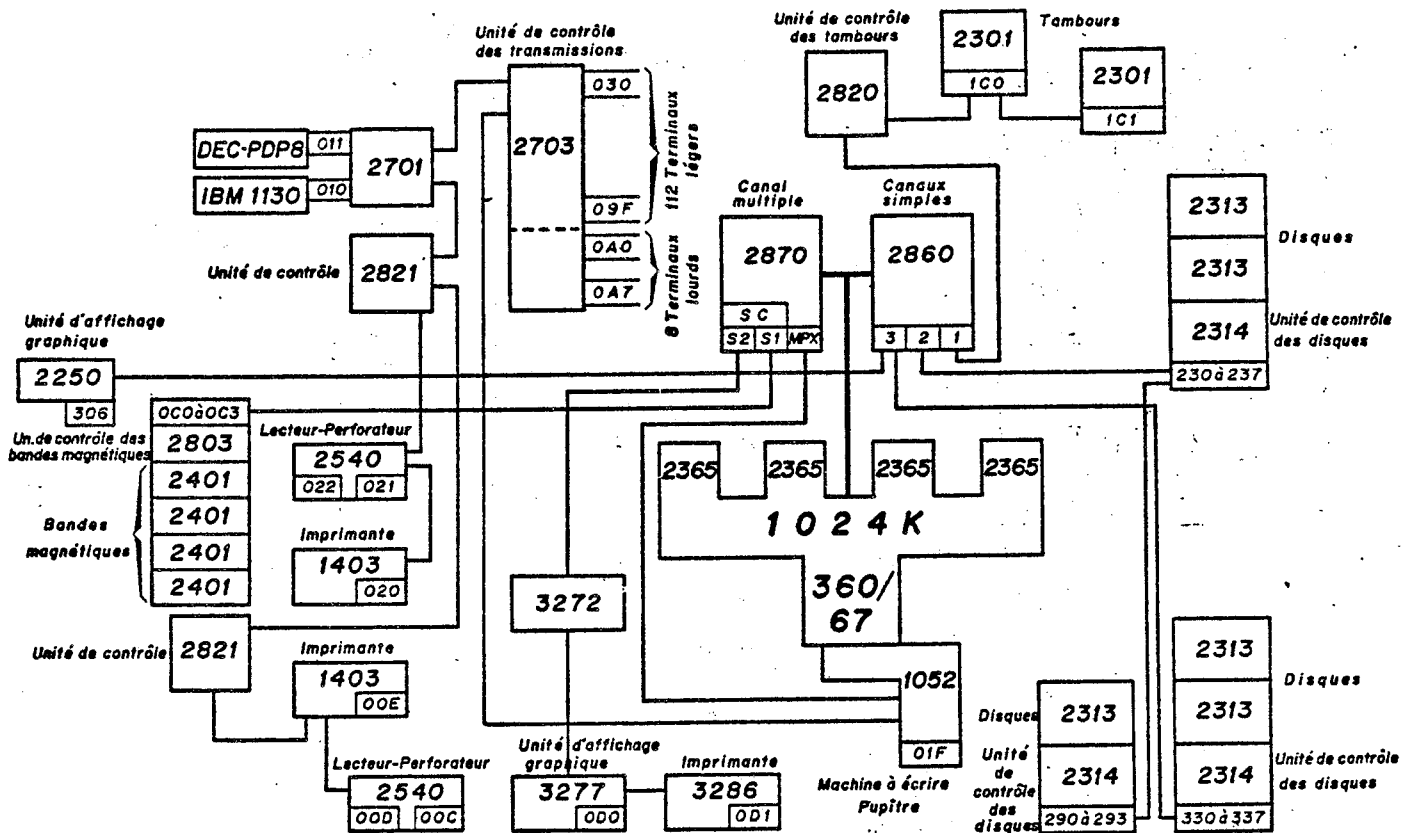


FIGURE III - 12.1 : CONFIGURATION MATERIELLE DU SYSTEME MINI-SIGMA - IBM 360/67 sous CP/CMS

| CONTEXTE | IBM 360/67 CP-CMS |
|---------------------------|--|
| Base de Données | Fichiers à Accès Séquentiel |
| Base d'algorithmes | Bibliothécaire CMS |
| Moniteur de Conception | EXEC-CMS : Allocation Statique Paginée |
| Moniteur de Dialogues | Interpréteur de Dialogues Spécialisé |
| Moniteur de Communication | Interpréteur Unique |
| Langage de Programmation | FORTRAN & ALGOL W |

FIGURE III - 12.2 : PRINCIPAUX CHOIX DE LA REALISATION DE MINI-SIGMA A PARTIR DE MACAO

12.1.4 REALISATION

Les choix que nous avons faits dans la mise en oeuvre du système MINI-SIGMA, ont été guidés par le souci d'aboutir rapidement à une version expérimentale. Ceci essentiellement afin de vérifier la faisabilité et démontrer l'intérêt d'un-tel système.

C'est pourquoi les possibilités de la description de la BDP ont été réduites, par contre l'ensemble des traitements pouvant s'appliquer sur cette base a été maintenu.

Le schéma d'organisation du système est représenté sur la figure III-12.3.

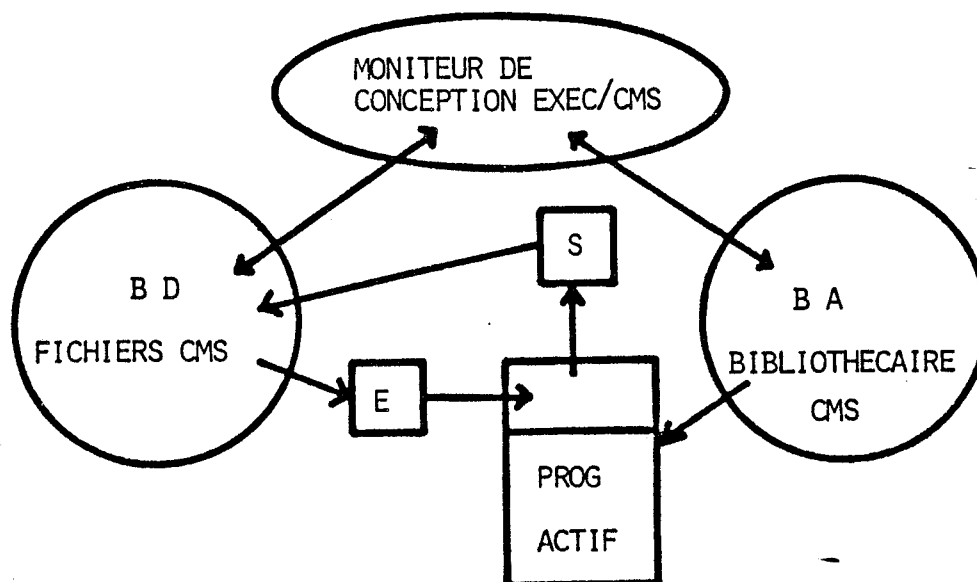


FIGURE III - 12.3 : ORGANISATION DU SYSTEME MINI-SIGMA

Les avantages de cette organisation résident surtout dans la disponibilité de fonctions puissantes facilitant considérablement la mise en oeuvre. De plus, cette organisation facilite la réalisation d'une approche algébrique du processus de conception (description implicite).

Les inconvénients sont notamment:

- la nécessité de concevoir des tâches assez importantes pour minimiser le travail d'activation et plus particulièrement d'accès aux données, car il n'existe pas de zone de stockage des données en mémoire centrale lors du passage d'une tâche à une autre,
- l'accès aux données de la BDP s'effectue au début de la tâche, le retour à la fin,
- les données doivent être organisées en blocs suffisamment importants,
- le lien avec le système d'exploitation est très étroit, car on s'appuie directement sur ses fonctions.

Le choix du langage ALGOLW pour la programmation du système s'est avéré assez intéressant et cela surtout parce que nous avons pu utiliser une version modulaire d'ALGOLW. Cette version même si elle était expérimentale et assez limitée en possibilités, nous a permis de bénéficier de la notion de module très agréable à manipuler.

La limitation de la version disponible résidait surtout dans le fait qu'une seule zone de données pouvait devenir globale et donc partageable par plusieurs modules. Ceci a limité

considérablement la modularité des données par contre la modularité des traitements a été complète.

La possibilité d'appeler à partir d'ALGOLW des programmes écrits en FORTRAN a aussi été très appréciée, notamment pour la communication graphique et les modes d'accès.

Regardons maintenant de façon plus précise la réalisation des différentes parties du système.

12.1.4a Moniteur de communication

Le moniteur de communication travaille comme cela est décrit dans le paragraphe 4 du chapitre 9. Sa réalisation en ALGOLW n'a soulevé aucun problème majeur.

12.1.4b Moniteur de modèle de conception

Le moniteur de conception a été réalisé en EXEC-CMS et MACRO-EDIT. Il comprend une routine d'activation, un ensemble de commandes système, un ensemble ouvert des commandes d'application et un ensemble de routines de service.

La routine d'activation dirige le moniteur, c'est elle qui saisit les demandes d'activation des commandes et les active.

Les commandes système permettent surtout:

- d'expliquer le fonctionnement de la version du système (liste des commandes disponibles et leur signification),
- de connaître l'état de la BDP, globalement ou de façon sélective,
- de terminer une session avec ou sans sauvegarde du travail effectué.

Les commandes d'application effectuent chacune un travail précis au sein de l'application. Leur structure est toujours la même:

- saisie des paramètres,
- vérification des paramètres obligatoires,
- transformation: production automatique d'une représentation nécessaire pour la tâche à partir d'une autre description,
- préfixe: saisie de paramètres optionnels et constitution du fichier particulier (voir gestion des données),
- appel: chargement et activation des algorithmes,
- postfixe: rangement des résultats dans la BDP.

L'exécution d'une commande d'application s'appuie sur les routines de service, qui ont pour rôle de:

- saisir les paramètres de la commande,
- vérifier l'existence des données nécessaires,
- guider l'utilisateur dans le système,
- gérer les accès aux données,
- vérifier la concordance entre le poste de travail actuel et celui demandé par la tâche,
- gérer l'historique d'utilisation du système.

Pour activer un algorithme nous avons choisi une organisation décrite plus loin. Grâce à cette organisation, nous avons réalisé un moniteur de conception basé sur un processus de conception implicite (algébrique), c'est à dire qu'il n'existe pas de description globale et complète du processus mais chaque commande d'application contient ses conditions d'activation. Ainsi chaque commande est activable si et seulement si ses conditions d'activation sont satisfaites.

12.1.4c Base de données

Pour réaliser la BDP nous avons opté pour une utilisation systématique des fichiers séquentiels. La BDP est constituée d'un ensemble de blocs. Chaque bloc décrit le problème d'un certain point de vue. Remarquons que ces blocs sont d'une taille relativement importante. A chaque type de bloc on associe un mode d'accès capable d'effectuer les transferts du bloc entre les mémoires principale et secondaire.

Chaque bloc constitue un fichier séquentiel CMS et contient comme premier enregistrement le descripteur précisant le type du bloc. Les blocs sont manipulés par le gestionnaire de la BDP qui s'appuie sur les fonctions de manipulation des fichiers CMS. Ce gestionnaire reçoit de la part du moniteur de conception la demande de fournir les blocs. Il constitue un fichier standard d'entrée contenant les blocs demandés. Ce fichier est exploité par la tâche qui a été activée. A la fin du traitement, la tâche fournit un fichier standard de données à sauvegarder, qui est traité par le gestionnaire pour être intégré dans la base. Ceci se traduit par l'éclatement de ce fichier en fichiers élémentaires contenant chacun un bloc d'information.

L'utilisateur a donc connaissance de l'existence des blocs d'information mais pas de leur organisation. Ces blocs ont chacun une sémantique précise, l'utilisateur peut donc les connaître et les manipuler à l'aide des algorithmes du système et par l'intermédiaire du langage de commande.

Le fichier standard d'entrée est constitué par le travail solidaire du moniteur de conception et du gestionnaire. Au sein d'une tâche, un mode d'accès spécifique assure le transfert des données dans la zone du programme.

Ce mode d'accès spécifique travaille à trois niveaux: il même, lit l'enregistrement contenant le type de bloc et l'analyse, puis il appelle, le mode d'accès logique connaissant la structure logique du bloc. Le mode d'accès logique fait appel au mode d'accès physique pour la manipulation effective d'informations. Le mode d'accès physique est donc seul à connaître l'organisation physique des données. Ceci assure l'indépendance des données et de leur représentation.

En sortie le mode d'accès spécifique, à l'aide des modes d'accès logiques et physiques, produit le fichier standard de sortie contenant toutes les données qui ont été modifiées par la tâche.

12.1.4d Base d'algorithmes

Pour la base d'algorithmes nous avons choisi d'utiliser le bibliothécaire CMS. Il gère sous forme de fichier aussi bien des fichiers de données que des programmes. Une tâche est stockée de la façon suivante:

- un fichier de type EXEC contient la séquence d'activation de la tâche,
- d'autres fichiers de type TEXT ou MODULE contiennent les algorithmes utilisés par la tâche.

L'activation de la tâche s'effectue selon l'un des deux schémas suivants:

- si la tâche a été réalisée selon les règles d'écriture SIGMA, l'activation s'effectue d'après le schéma standard (FIG.III-12.4),
- si la tâche a été réalisée par ailleurs, il s'agit alors seulement de l'intégrer dans le système et d'utiliser un schéma d'activation particulier (FIG.III-12.5).

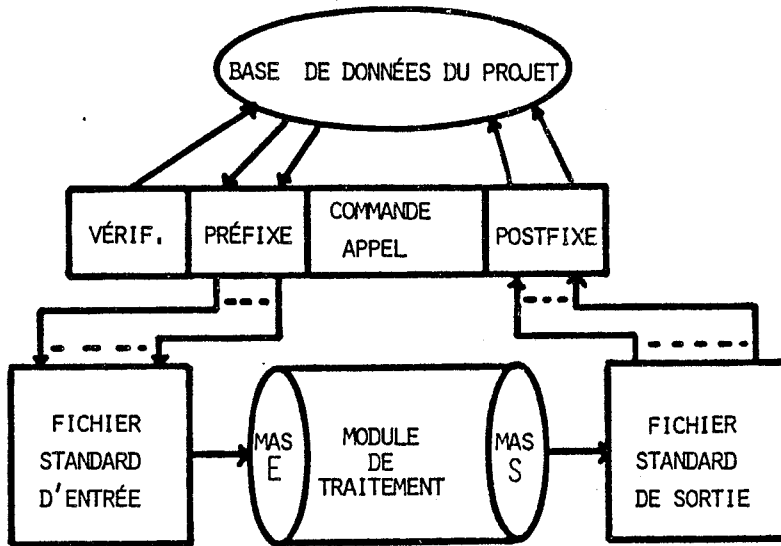


FIGURE III - 12.4 : SCHEMA D'ACTIVATION STANDARD

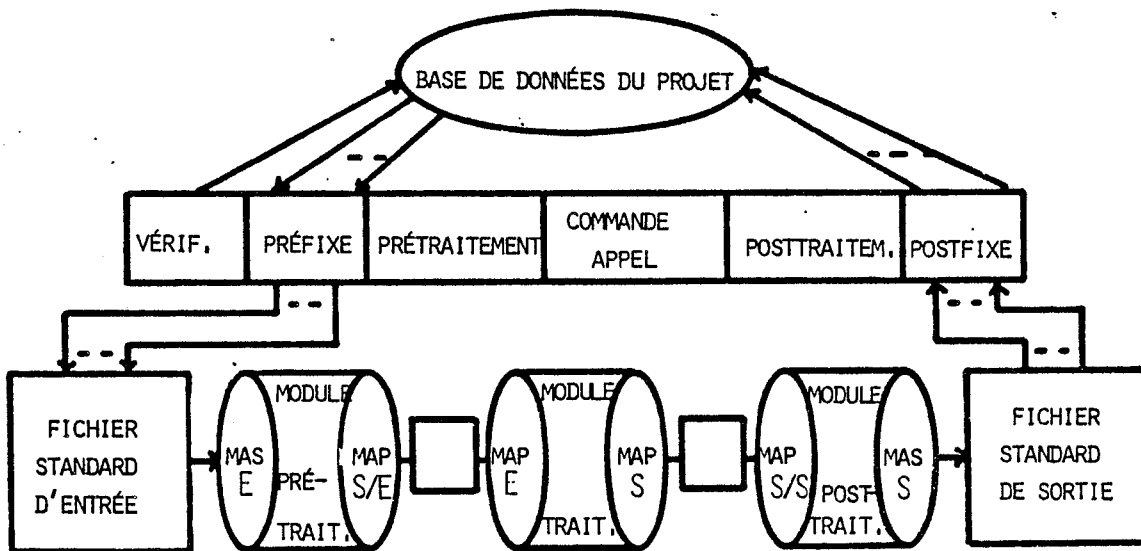


FIGURE III - 12.5 : SCHEMA D'ACTIVATION PARTICULIER

12.1.4e Moniteur de dialogues

Pour le moniteur de dialogues nous avons réalisé une version basée sur l'interprétation de tables. C'est un interpréteur spécialisé construit spécifiquement pour le type de langages de commandes que nous avons défini.

12.2 APPLI-SIGMA

12.2.1 OBJECTIF

L'objectif du système APPLI-SIGMA est de servir de support à des applications CAO dans un contexte de mini-ordinateurs. Il a d'ailleurs été employé dans la réalisation de la maquette du système pour la CAO des équipements électroniques (voir chapitre 15).

12.2.2 CONTEXTE

Le contexte de réalisation (FIG.III-12.6) est un mini-ordinateur (SOLAR16-40), fonctionnant en temps partagé (sous MPES), permettant donc l'utilisation conversationnelle. Une autre alternative basée sur un autre mini-ordinateur (PDP 11-55), fonctionnant en temps réel (sous R2X11M) a aussi été étudiée, ainsi que la possibilité de répartir l'application sur plusieurs machines en s'appuyant sur un support rudimentaire de répartition: le terminal intelligent.

Le poste de travail est un poste graphique avec un terminal à tube mémoire et une tablette de digitalisation (TEKTRONIX 4015, CYBERGRAPHE)

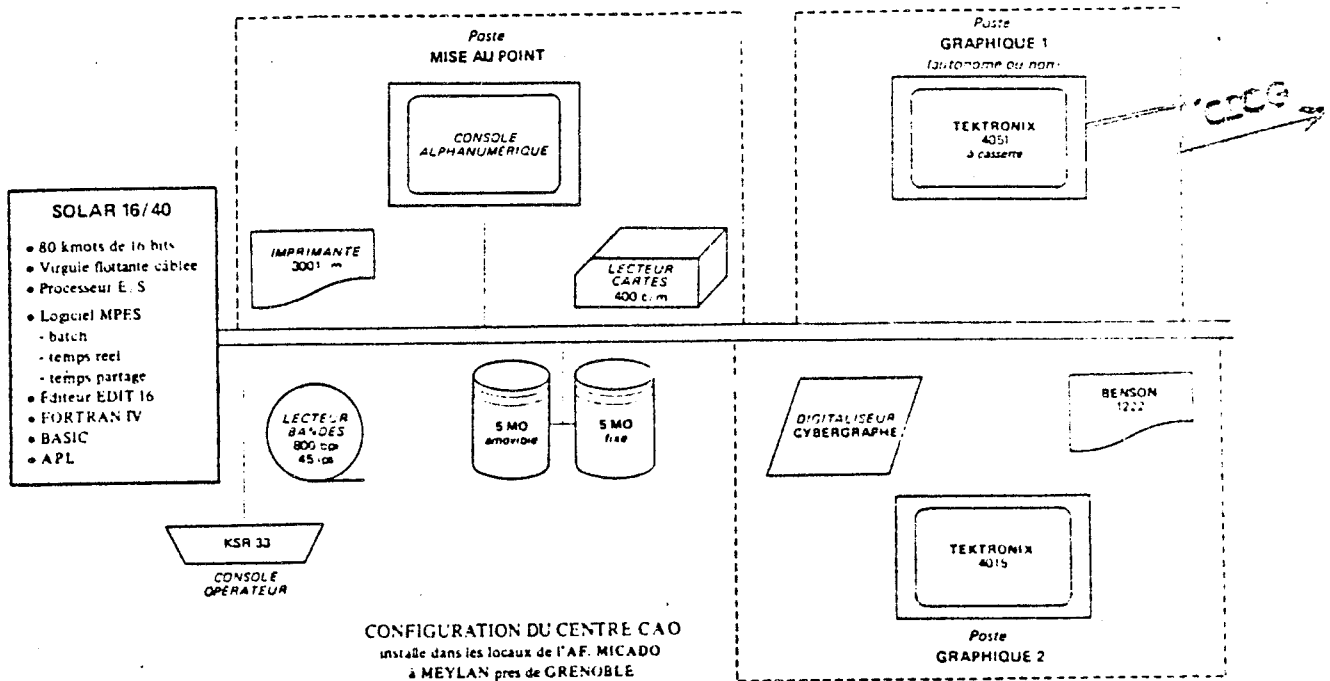


FIGURE III - 12.6 : CONFIGURATION MATERIELLE DU SYSTEME APPLI-SIGMA - SOLAR 16-40 sous MPES

12.2.3 CHOIX

Les choix que nous avons effectués sont schématisés sur la figure III-12.7. Fonctionnellement il s'agit principalement de:

- la BASE DE DONNEES: un SGBD de type réseau (SOMINE),
- la BASE D'ALGORITHMES: bibliothécaire MPES,
- le MONITEUR DE CONCEPTION: TSF-MPES avec allocation statique et recouvrement,
- le MONITEUR DE DIALOGUES: interpréteur général,
- le MONITEUR DE COMMUNICATION: interpréteur unique.

Dans la réalisation effective nous avons choisi pour des questions de portabilité, de programmer en FORTRAN et pour la communication graphique, en GRIGRI.

| CONTEXTE | SOLAR 16/40 MPES |
|---------------------------|---|
| Base de Données | SGBD "réseau" SOMINE |
| Base d'algorithmes | Bibliothécaire MPES |
| Moniteur de Conception | TSF-MPES, Allocation Statique avec Recouvrement |
| Moniteur de Dialogues | Interpréteur de Dialogues Général |
| Moniteur de Communication | Interpréteur Unique |
| Langage de Programmation | FORTRAN |

FIGURE III - 12.7 : PRINCIPAUX CHOIX DE LA REALISATION D'APPLI-SIGMA A PARTIR DE MACAO

12.2.4 REALISATION

Dans la réalisation du système APPLI-SIGMA nous avons bénéficié de l'expérience de la réalisation et de l'utilisation du système MINI-SIGMA. Le changement du langage de programmation (ALGOLW n'est pas supporté par ces mini-ordinateurs) nous a obligé à reprogrammer l'ensemble. Nous en avons profité pour apporter quelques modifications dans les parties où nous avons adopté la même réalisation que dans MINI-SIGMA.

Nous avons mis en oeuvre toutes les parties du système APPLI-SIGMA. Malheureusement il s'est avéré que l'intégration de ces parties en un système complet n'a pas été possible dans le contexte du temps partagé sous MPES, qui limite des partitions mémoire à 64K. Même en exploitant au maximum toutes les possibilités de recouvrement nous n'avons pu intégrer dans une telle zone mémoire en même temps le moniteur de conception, la

gestion de la base de données, la communication graphique et la tâche qui les utilise.

Pour résoudre ce problème dans un environnement de mini-ordinateur à mot de 16 bits, la solution consiste à travailler sous un système temps réel permettant l'activation simultanée de plusieurs tâches et assurant la communication entre ces tâches. C'est ce que nous avons étudié sans le réaliser, dans le contexte d'un PDP 11-55 fonctionnant sous R2X11M. Dans ce cas le système est organisé en 3 tâches système et un ensemble de tâches d'application. Ces trois tâches sont:

- le moniteur de conception,
- le moniteur de la base de données,
- le moniteur de la communication graphique.

Dans le cadre du projet APPLI-SIGMA nous avons aussi expérimenté l'utilisation d'un terminal intelligent comme moyen de répartition rudimentaire. En effet nous avons utilisé le terminal programmable TEKTRONIX 4051 pouvant, d'une part servir de terminal classique et disposant, d'autre part d'une cassette sur laquelle il est capable d'enregistrer des informations émises par le calculateur et éventuellement les reverser dans un autre calculateur pour un autre type de traitement. Nous avons ainsi expérimenté la possibilité de transférer et traiter des données d'une part au CICG et d'autre part au service IBM à LYON.

Nous avons pu assurer un premier niveau de répartition ne nécessitant aucun support logiciel dans aucun des ordinateurs concernés.

Regardons maintenant de façon plus précise la réalisation des différentes parties du système. Il s'agit d'une version intermédiaire entre MINI-SIGMA et VRAI-SIGMA, nous l'examinerons donc succinctement.

12.2.4a Moniteur de communication

C'est une version standard que nous avons reprogrammée en FORTRAN.

12.2.4b Moniteur de modèle de conception

Il s'agit du même principe de fonctionnement que pour MINI-SIGMA, mais plus rudimentaire car les possibilités offertes par le système d'exploitation sont moindres.

12.2.4c Base de données

Nous avons adapté le système SOMINE. Il s'agit d'un SGBD de type réseau que nous utilisons de la manière indiquée dans le chapitre 5, c'est à dire avec une structure de données prédéfinie et extensible.

12.2.4d Base d'algorithmes

L'organisation et l'enchaînement des algorithmes sont semblable à ceux utilisés dans MINI-SIGMA.

12.2.4e Moniteur de dialogues

Après avoir constaté que le nombre d'interpréteurs spécialisés était très important, nous avons opté pour un regroupement. Nous avons donc unifié les interpréteurs en en prenant un général. Cela nous a conduit à revoir l'organisation globale, par contre nous avons pu supporter les trois versions souhaitées (mode libre, mode tutorial, graphique ou textuel).

12.3 VRAI-SIGMA

12.3.1 OBJECTIF

L'objectif du système VRAI-SIGMA est de concrétiser tous les aspects de notre étude dans le contexte du matériel actuel du CIGG. Ce système doit d'une part servir de support pour des applications variées, et d'autre part permettre le développement des applications en servant de support aux processus de spécifications, de construction de prototype et à la mise en oeuvre définitive. C'est sur ce système que les nouvelles recherches sont entreprises.

Présentement il sert de support au système intégré pour des ensembles électroniques et logiques.

12.3.2 CONTEXTE

Le contexte de réalisation (FIG.III-12.8) est un grand ordinateur (HB68), fonctionnant en temps partagé (sous MULTICS).

Le poste de travail est actuellement réduit à un terminal graphique doté d'un tube mémoire de petite taille (TEKTRONIX 4010), mais devrait dans un avenir proche être augmenté d'un terminal à balayage cavalier couleur (CONCEPT 60) avec une tablette de digitalisation.

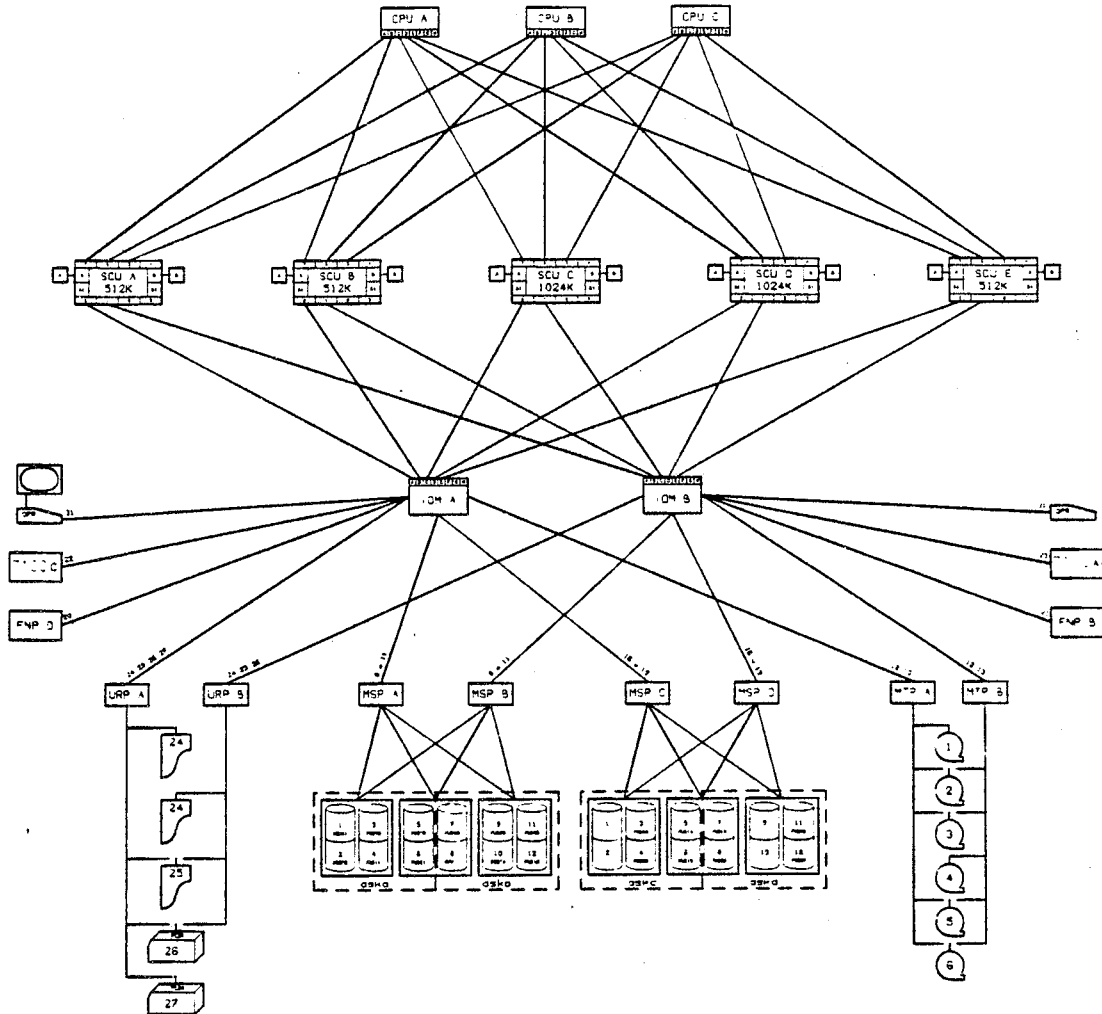


FIGURE III - 12.8 : CONFIGURATION MATERIELLE DU SYSTEME VRAI-SIGMA - 4P68 DPS 3 sous MULTICS

12.3.3 CHOIX

Les choix principaux que nous avons fait sont schématisés sur la figure III-12.9. Fonctionnellement il s'agit de:

- la BASE DE DONNEES: un SGBD de type réseau, modifié pour pouvoir servir de BDP,
- la BASE D'ALGORITHMES: le bibliothécaire MULTICS,
- le MONITEUR DE CONCEPTION: en FORTRAN avec l'allocation dynamique paginée,
- le MONITEUR DE DIALOGUES: l'interpréteur général,
- le MONITEUR DE COMMUNICATION: l'interpréteur unique.

Du point de vue réalisation, nous avons gardé comme langage de mise en œuvre FORTRAN, principalement en raison de sa portabilité et pour les mêmes considérations GRIGRI, pour la communication graphique.

| CONTEXTE | HB 68 MULTICS |
|---------------------------|--|
| Base de Données | SGBD "réseau" SOMINE |
| Base d'algorithmes | Bibliothécaire MULTICS |
| Moniteur de Conception | FORTRAN : Allocation Dynamique paginée |
| Moniteur de Dialogues | Interpréteur de Dialogues Général |
| Moniteur de Communication | Interpréteur Unique |
| Langage de Programmation | FORTRAN |

FIGURE III - 12.9 : PRINCIPAUX CHOIX DE LA REALISATION DE VRAI-SIGMA A PARTIR DE MACAO

12.3.4 REALISATION

Dans la réalisation du système VRAI-SIGMA nous nous sommes retrouvés dans le contexte d'un grand ordinateur avec un système d'exploitation puissant et complet. Ceci nous a permis de réaliser une version complète et d'exploiter certaines particularités offertes.

12.3.4a Moniteur de communication

L'interpréteur standard de APPLI-SIGMA a été gardé.

12.3.4b Moniteur de modèle de conception

Les possibilités d'allocation dynamique et d'empilement des niveaux offertes par le système d'exploitation nous permettent de

réaliser le système complet en FORTRAN. Dans ce cas le moniteur de conception est réalisé par le moniteur de dialogues qui a été spécialisé dans ce sens. Il s'agit donc d'un moniteur de conception explicite.

12.3.4c Base de données

La BDP est assumée par SOMINE, un SGBD de type réseau, selon les principes exposés dans le chapitre 5.

12.3.4d Base d'algorithmes

La base d'algorithmes est le bibliothécaire MULTICS. Le schéma d'activation est exprimé dans le moniteur de conception et s'appuie sur les mêmes principes qu'en MINI-SIGMA et APPLI-SIGMA.

12.3.4e Moniteur de dialogues

Le moniteur de dialogues travaille par interprétation et s'appuie sur un interpréteur général. La solution d'APPLI-SIGMA a été reconduite.

12.4 SUPPORT DE LA DOCUMENTATION

Le problème auquel nous nous sommes intéressé tout particulièrement est celui de la documentation de la réalisation des systèmes que nous avons développés. Ceci a été d'autant plus important que nous avons dû abandonner un support de réalisation dans lequel les notions de module, d'exportabilité, de sélection d'accès étaient gérées (ALGOLW modulaire), au profit d'un support portable mais n'ayant aucun outil moderne facilitant et systématisant la mise en oeuvre.

Nous nous sommes donc efforcés de documenter systématiquement nos réalisations en nous appuyant sur des imprimés, représentés sur la figure III-12s.10, que nous avons définis. Ils permettent de bâtir une véritable structure modulaire des données et des traitements, et servir ainsi de support à la réalisation, la mise au point et la maintenance du système.

| | | |
|---|----------|--|
| FICHE TECHNIQUE PROCEDURE | | |
| NOM: INTERNE, EXPORTABLE MODULE: | | |
| APPEL: | | |
| PARAMETRES: ENTREES: | SORTIES: | |
| PROCEDURES APPELEES: | | |
| BLOCS DE DONNEES UTILISES: VARIABLE MODIFIEE <input type="checkbox"/> , LUE ___: | | |
| FONCTION DE LA PROCEDURE | | |

| | | |
|---------------------------------|--|--|
| FICHE TECHNIQUE BLOC DE DONNEES | | |
| NOM: | | |
| STRUCTURE: | | |
| UTILISE DANS LES MODULES: | | |
| CONTENU: | | |

| | | |
|----------------------------|----------------|--------------------|
| FICHE TECHNIQUE MODULE | | |
| NOM: | TAILLE: | PLACE M.: LANGAGE: |
| PROCEDURES - FONCTIONS | | |
| IMPORÉES: | DEPUIS MODULE: | |
| EXPORTABLES: | VERS: | |
| INTERNES: | | |
| BLOCS DE DONNÉES UTILISÉS: | | |
| OBJET: | | |

| | | |
|-----------------------------|----------|----------|
| FICHE TECHNIQUE SYSTEME | | |
| NOM: | LANGAGE: | MACHINE: |
| LISTE DES MODULES: | | |
| LISTE DES BLOCS DE DONNÉES: | | |
| OBJET: | | |

FIGURE III - 12.10 : IMPRIMES POUR LA DOCUMENTATION DES REALISATIONS

CHAPITRE 13

IMPLANTATION DE MACAO SUR DIFFERENTS SITES

RESUME: Ce chapitre étudie le processus de transport de la machine abstraite dans différents contextes. On présente d'abord les éléments de choix de la configuration en s'appuyant sur les spécifications fonctionnelles, les renseignements issus de la réalisation prototype et les informations quantitatives fournies par les futurs utilisateurs. Puis on s'intéresse au système CAO en tant que système distribué. On présente la typologie de la répartition de la MACAO et des utilisateurs. On étudie la répartition fonctionnelle et la répartition physique et on montre différents niveaux de répartition et la façon de les prendre en compte. Ensuite on explicite les trois points d'appui de la machine abstraite: appui matériel, appui logiciel de base et appui logiciels évolués. Le langage de spécifications organiques permet d'exprimer les choix de ces points d'appui. On termine par la présentation du processus de mise en oeuvre de la machine abstraite dans le contexte choisi.

13. IMPLANTATION DE MACAO SUR DIFFERENTS SITES

13.1 OBJECTIFS

Après avoir obtenu une définition complète et satisfaisante du fonctionnement, nous avons à notre disposition une description fonctionnelle de l'application: la liste des algorithmes utilisés, les données manipulées, les dialogues associés et l'organisation générale du processus de conception.

La phase de mise en oeuvre effective a pour but de produire une réalisation particulière correspondant aux caractéristiques spécifiques de l'application. La description fonctionnelle est indépendante par rapport à une structure de matériel donnée car il est important en CAO de pouvoir suivre les utilisateurs sur leur lieu de travail. Il est impensable de leur imposer un type de matériel; le système doit s'adapter à des contextes variés aussi bien du point de vue des matériels que des logiciels.

Les raisons fondamentales en sont:

- 1/ la vision de plus en plus globale du processus CONCEPTION - ETUDE - DEVELOPPEMENT;
- 2/ l'utilisation de plus en plus répandue des machines de petite taille (mini et micro), souvent orientées vers des utilisations particulières mais interconnectées pour permettre cette vision globale.

Il s'agit d'entreprendre une étude détaillée du comportement du système. Cette étude doit mettre en évidence les caractéristiques des composants du système, c'est à dire les performances du matériel et des logiciels pour un bon fonctionnement de l'application. Ainsi ce travail doit fournir des indications précises sur la nature et le volume d'informations stockées et traitées, les tailles de mémoire principale et secondaire, la puissance de calcul, le type du poste de travail, les logiciels système nécessaires. Le but de cette étude est aussi de dégager sur quelle structure concrète (de matériel et de logiciels) cette application peut fonctionner de façon satisfaisante, et éventuellement de manière encore plus précise de répondre à la question souvent posée: est-il possible que l'application utilise le matériel déjà disponible dans l'entreprise?

13.2 ELEMENTS DU CHOIX DE LA CONFIGURATION

Pour dégager les caractéristiques de la configuration souhaitée nous pouvons nous appuyer sur trois types d'information:

- les spécifications fonctionnelles,
- les renseignements issus de la réalisation prototype,
- les informations supplémentaires à fournir par les futurs utilisateurs.

Essayons de préciser le contenu de chacun.

13.2.1 On peut déduire des spécifications fonctionnelles

- la complexité du processus de conception (nombre d'étapes et de phases, leur enchaînement, etc..),
- la complexité de l'équipe exécutant le projet (hétérogénéité des personnes concernées),
- la variété des traitements,
- la complexité et la diversité des données à gérer.

A partir de ces informations une première conclusion peut être formulée quant à la nature de la configuration souhaitée.

13.2.2 On peut déduire de la réalisation prototype

- la place mémoire nécessaire pour les traitements,
- l'espace secondaire pour les données,
- la puissance souhaitable du processeur (temps de réponse, coût, ..),
- l'organisation du poste de travail.

Toutefois, il ne faut pas oublier que la réalisation prototype a été effectuée dans un but de démonstration de faisabilité, sans aucun souci d'optimisation. Il s'agit donc d'une approximation grossière des caractéristiques de mise en oeuvre.

13.2.3 Supplément d'informations nécessaire

Les spécifications fonctionnelles et la réalisation prototype ne fournissent pas des informations complètes sur l'exploitation que l'utilisateur veut faire du système. Il manque notamment des informations quantitatives sur la charge du système en exploitation et sur les délais à respecter. Pour

dimensionner et organiser la réalisation finale, il faut disposer de ces informations. Il faut donc préciser en particulier:

- le nombre de projets à réaliser en même temps,
- le délai de réalisation,
- le nombre de personnes travaillant sur un projet,
- le volume d'information d'un projet,
-

Des renseignements fonctionnels, des indications sur la réalisation du prototype et des informations quantitatives interviennent dans le choix de l'organisation et de la nature de tous les composants (matériels et logiciels) les mieux adaptés au contexte particulier d'utilisation.

13.3 SYSTEME CAO COMME UN SYSTEME DISTRIBUE

L'étude du comportement prend une signification toute particulière quand on doit déterminer la distribution du système CAO. Par exemple:

- on a besoin d'interroger à distance une base de données unique en France, par exemple de composants électroniques,
- on utilise un programme de simulation qui est soit trop gros pour la configuration, soit non transportable.
- ou plus généralement, on a besoin d'accéder à une ressource particulière (donnée, programme, périphérique) non disponible sur la configuration mère, ou bien à un processeur spécialisé utile pour l'application (machine à tester par exemple) ou pour le système CAO (processeur fichier pour la gestion des données).

Mais en fait, cette répartition est de plus en plus en accord avec l'évolution du processus CED (conception - étude - développement) vers une plus grande intégration. Il s'agit de gérer la continuité du processus et d'assurer sa cohérence. Une organisation mono-machine semble inadaptée à ce problème. C'est pourquoi on s'oriente de plus en plus vers des organisations distribuées.

Avant de montrer comment nous pouvons prendre en compte cette organisation du système en tant que structure distribuée nous devons préciser quelques principes concernant la répartition.

Notre but est de pouvoir prendre en compte le type de répartition approprié selon la nature et la demande de l'application et le contexte de mise en oeuvre choisi. Comme les vrais utilisateurs, nous faisons abstraction du mode de fonctionnement du système de communication. Il nous suffit de savoir qu'il en existe un, sans connaître s'il est simple ou

double (comme par exemple machine de communication, machine de transport de CYCLADES).

Nous n'avons pas non plus d'a priori, ni sur la nature du réseau (homogène ou hétérogène), ni sur son type (général ou spécialisé). C'est à l'application de les choisir, au même titre que d'autres choix de supports d'implantation. Il nous faut être en mesure de prendre en considération les réseaux de n'importe quel type et nature, c'est à dire les primitives que l'on nous fournit pour exploiter ces réseaux.

Il faut notamment prendre en compte les niveaux suivants:

- distribution des accès
- distribution des données
- distribution des traitements
- distribution des utilisateurs

Il semble intéressant d'étudier le comportement de la machine abstraite dans le contexte propre à chacun de ces types de distribution.

La DISTRIBUTION DES ACCES se situe uniquement entre le moniteur de communication et le poste de travail: soit le poste de travail est distant (FIG. III - 13.1a), soit une partie du poste de travail est éloignée des autres parties (FIG. III - 13.1b).

La DISTRIBUTION DES DONNEES est localisée dans la Base de Données (FIG. III - 13.1c).

La DISTRIBUTION DES TRAITEMENTS intervient seulement dans la Base d'Algorithmes (FIG. III - 13.1d).

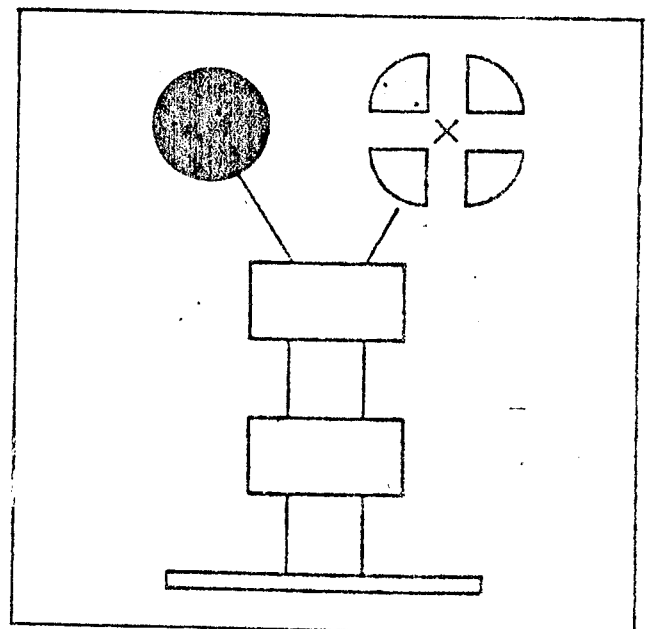
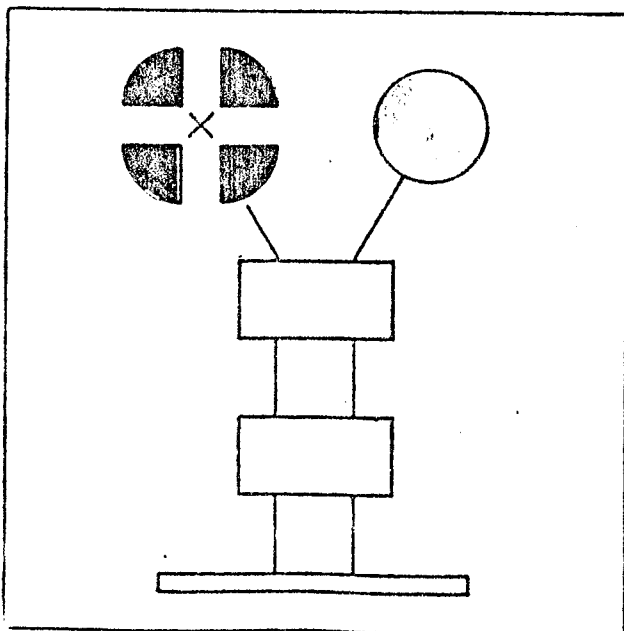
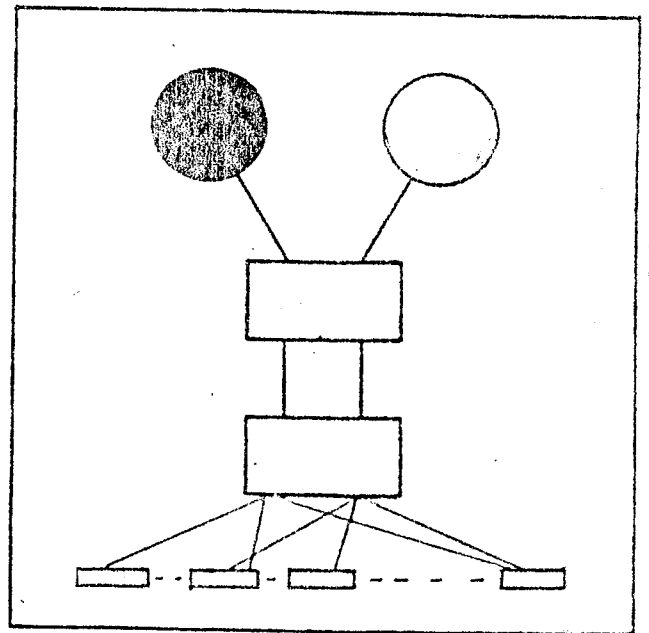
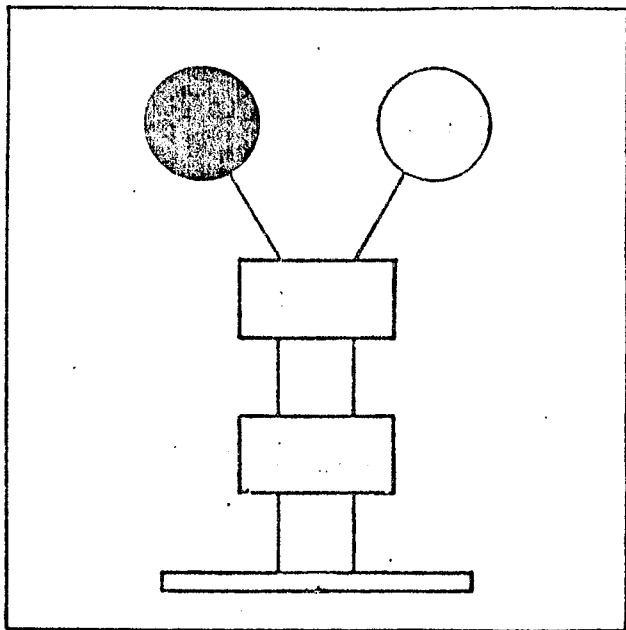


FIGURE III - 13.1 : TYPOLOGIE DE LA REPARTITION DE LA MACAO

La DISTRIBUTION DES UTILISATEURS dépend beaucoup de leur façon de travailler. S'ils travaillent sur des projets différents, nous pouvons considérer qu'ils sont entièrement indépendants et ils ont donc chacun un système complètement indépendant (FIG. III - 13.2a). Toutefois, s'ils travaillent sur le même projet, il existe différentes possibilités de partage. Ils peuvent en effet partager le moniteur de communication (FIG. III -13.2b), le moniteur de conception (FIG. III - 13.2c), ou uniquement des données ou des traitements.

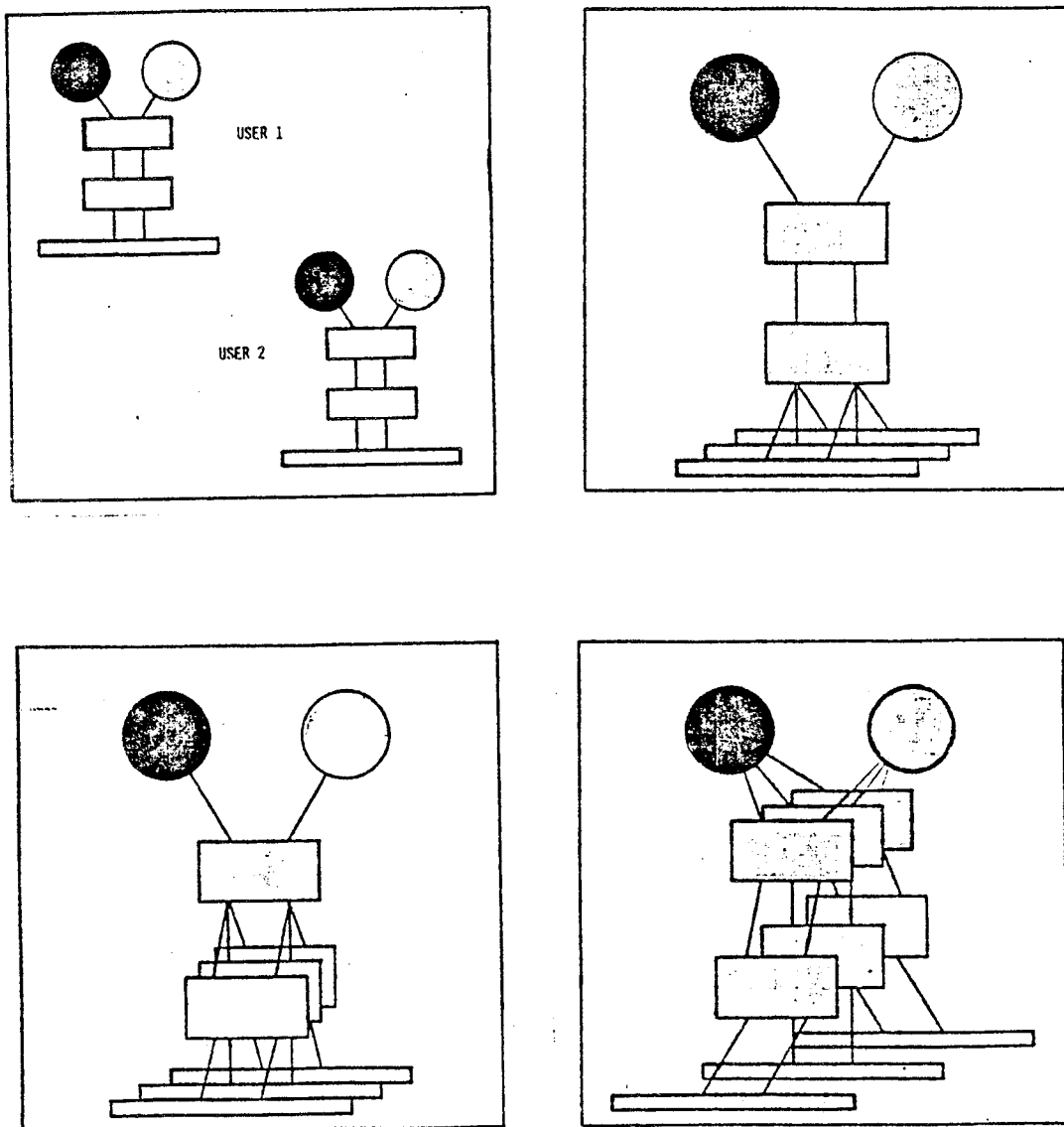


FIGURE III - 13.2 : TYPOLOGIE DE LA REPARTITION DES UTILISATEURS

13.3.1 REPARTITION FONCTIONNELLE

Généralement, dans un système on trouve tous les niveaux de distribution et le choix d'une bonne organisation est d'autant plus difficile. L'élaboration de cette organisation doit tenir compte d'une part des fonctions de l'application et d'autre part des choix que l'on a préférés. On commence par une étude du comportement fonctionnel en vue de déterminer une proposition de répartition, car la description fonctionnelle joue un rôle primordial dans le choix de distribution (utilisateurs concernés, données nécessaires, algorithmes utilisés, ressources demandées).

Les spécifications exprimées en COLA sont très utiles dans la recherche de solutions. En analysant ces spécifications nous déterminons les liens dans les situations suivantes:

- données et algorithmes (FIG. III - 11.4a,c): fait apparaître les regroupements des données par rapport aux algorithmes qui les utilisent (consomment ou produisent), ainsi on obtient des indications sur l'implantation des données et des algorithmes,
- algorithmes et utilisateurs: fait apparaître les regroupements des algorithmes par rapport aux utilisateurs qui peuvent s'en servir,
- utilisateurs et phases de conception: montre la relation entre les utilisateurs et les phases de conception,
- algorithmes et phases de conception (FIG. III - 11.4b): montre la relation entre les algorithmes et les phases de conception dans lesquelles ils sont utilisés,
- ...

En analysant ces différents liens il est possible de déterminer une distribution fonctionnellement satisfaisante, qui se traduise par l'organisation globale des données et des algorithmes par rapport aux utilisateurs et aux phases de conception.

Par exemple dans notre étude de la CAO des équipements électroniques (annexe 2), nous avons obtenu les résultats suivants: l'analyse du processus CED a permis d'identifier 17 programmes et 42 atomes d'information. Il s'agit pour des programmes de:

ACQU = acquisition des données
TRDO = dépouillement des données
ANAF = analyse fonctionnelle
ANAD = analyse dynamique
STIM = description de stimuli
DESO = description des objets
SIMF = simulation, évaluation fonctionnelle

PART = partitionnement
BTHR = bilan thermique
CIMP = implantation circuit imprimé
IMEC = implantation mécanique détaillée
CABL = étude cablage
PLOG = production logiciel
MAPL = mise au point logiciel
FIAB = étude de fiabilité
SECU = étude de sécurité
MDIS = maintenabilité et disponibilité

Les atomes d'informations sont les suivants:

CAPT = capteur
ACTU = actuateur
ENRG = enrégistrement
DATA = données dépouillées
FONC = fonctions à assurer
NOMF = nomenclature graphe fonctionnel
GRAF = graphe fonctionnel
NOMD = nomenclature graphe dynamique
GRAD = graphe dynamique
CONF = conditions de fonctionnement
BIBT = standards, méthodes
STMU = pattern de stimuli
BIBO = bibliothèque partie d'objet
CONR = contraintes de réalisation
OBJV = objet virtuel
RSMF = résultats simulation fonctionnelle
BIBR = bibliothèque de normes et de règles de réalisation
OPGM = objet partitionné grande masse
BITH = bibliothèque thermique
RSTH = résultats simulation thermique
TRAC = tracé du circuit
PERC = liste de perçage
OMEC = objet sous angle mécanique
PCAB = plan de cablage
CONL = conditions de réalisation de logiciel
LANP = langage de programmation
BIBL = bibliothèque logiciel
PROG = programme réalisé
BMAP = bibliothèque de logiciels de mise au point
RMPL = résultats de mise au point logiciel
BIBF = bibliothèque de fiabilité
MDEF = modes de défaillances
CONE = conditions d'exploitation
RECH = rechanges
LAI = lancements
RFIA = résultats fiabilité
POMA = politique de maintenance
RSEC = résultats d'étude sécurité
ETST = étude de testabilité
TREP = temps de réparation

anaspe
verifications

| nom de l'atome | produit | consomme |
|----------------|---------|----------|
| CAPT | 0 | 1 |
| ACTU | 1 | 0 |
| ENRG | 1 | 1 |
| DATA | 1 | 0 |
| FONC | 0 | 4 |
| NOHF | 0 | 1 |
| GRAF | 1 | 6 |
| NGMD | 0 | 1 |
| GRAD | 1 | 4 |
| CONF | 0 | 1 |
| BIBT | 0 | 1 |
| STNU | 1 | 2 |
| BIBO | 0 | 2 |
| CONR | 0 | 6 |
| OBJV | 1 | 6 |
| RSNF | 1 | 0 |
| BIBR | 0 | 3 |
| OBGH | 1 | 4 |
| BITH | 0 | 1 |
| RSTH | 1 | 0 |
| TRAC | 1 | 1 |
| PERC | 1 | 1 |
| OMEC | 1 | 1 |
| PCAD | 1 | 0 |
| CONL | 0 | 1 |
| LAMP | 0 | 1 |
| BIBL | 0 | 1 |
| PROG | 1 | 1 |
| BMAP | 0 | 1 |
| RMPL | 1 | 0 |
| BIBF | 0 | 1 |
| MDEF | 0 | 2 |
| COME | 0 | 3 |
| RECH | 1 | 0 |
| LAI | 1 | 0 |
| RFIA | 1 | 2 |
| POMA | 0 | 2 |
| RSEC | 1 | 0 |
| ETST | 0 | 1 |
| TREP | 0 | 1 |
| LOTR | 1 | 0 |
| REHD | 1 | 0 |

FIGURE III - 13.4 : ANALYSE D'UTILISATION DES ATOMES D'INFORMATION

Les atomes qui ne sont pas consommés font partie du dossier de CED. Les atomes qui sont à la fois produits et consommés constituent des informations intermédiaires du processus.

Une analyse plus détaillée (analyse factorielle) fait apparaître les regroupements des programmes et des atomes. Les résultats obtenus aident d'une part à dégager le processus CED souhaitable (si ce n'est pas déjà fait), et d'autre part à

dégager la distribution de ce processus. Dans notre étude nous avons distingué cinq parties:

- étude et définition du cahier des charges,
- analyse fonctionnelle,
- réalisation logiciel,
- réalisation matériel,
- analyse opérationnelle.

La distribution produite correspond dans ce cas précis au découpage en phases du processus CED. Elle détermine surtout l'implantation souhaitable des programmes et des atomes d'information.

On voit sur le tableau de résultats (FIG.III-13.5) que très peu de données sont utilisées dans diverses parties du processus. Le transfert est donc relativement réduit.

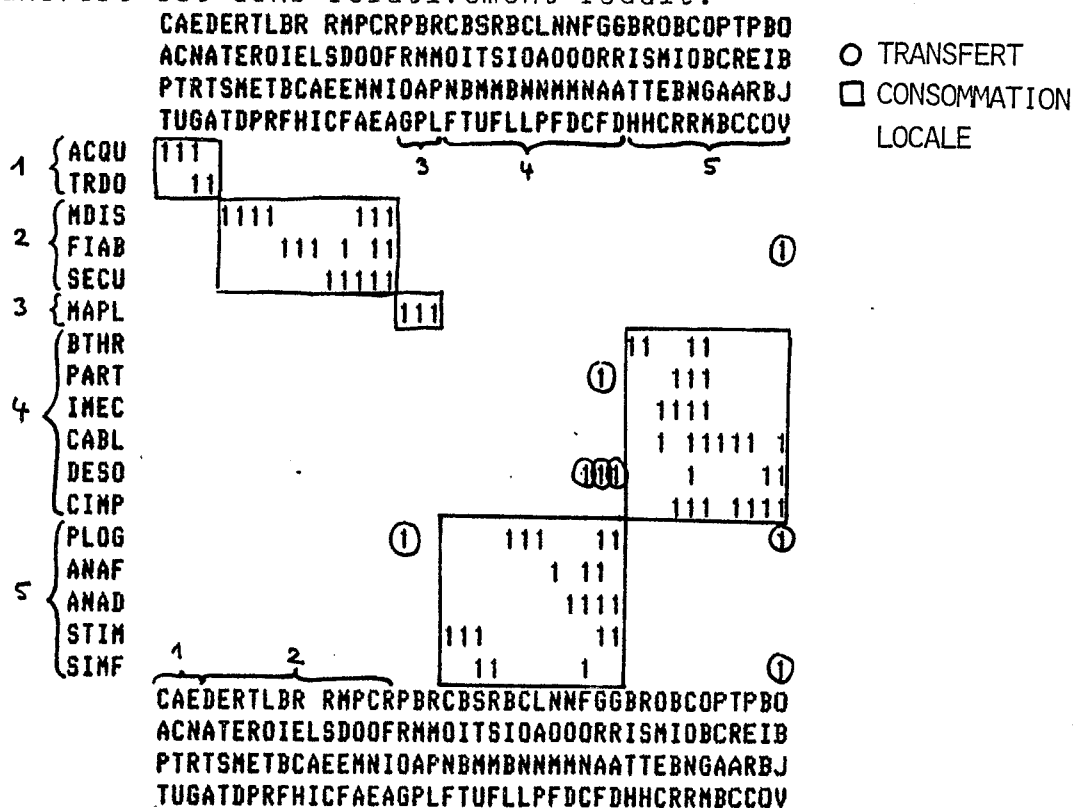


FIGURE III - 13.5 : RESULTATS DE L'ANALYSE DE LA RELATION PROGRAMMES - ATOMES D'INFORMATION

Pour cette étude les analyses des utilisateurs par rapport aux programmes, ou des programmes par rapport aux phases de conception, etc.. confirment les résultats de l'analyse de la relation programmes/atomes. Ceci semble logique au vu de la diversité des programmes, des données et des compétence des utilisateurs dans diverses phases du processus CED.

Dans d'autres applications utilisant des programmes plus polyvalents, des données plus générales et des utilisateurs moins spécialisés, ces études complémentaires permettent de mieux cerner le problème et de définir une distribution intéressante.

A la suite de ces analyses nous sommes en mesure de présenter une organisation de la distribution fonctionnelle de l'application (FIG.III-13.6). Dans celle-ci on remarquera surtout l'implantation des programmes, l'implantation des atomes de la BDT et des atomes de la BDP, ainsi que les transferts de ces atomes.

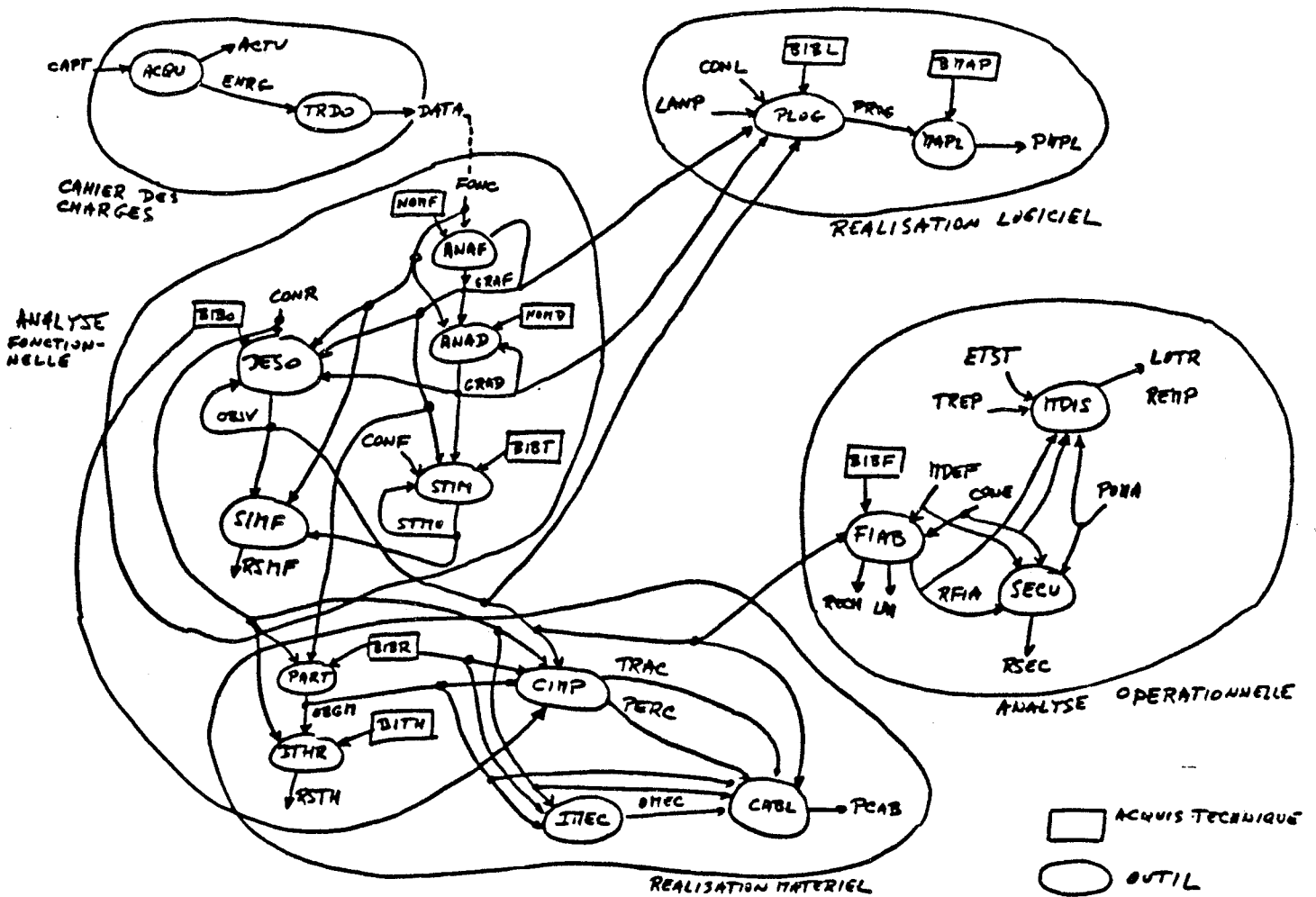


FIGURE III - 13.6 : PROPOSITION DE LA REPARTITION FONCTIONNELLE DE L'APPLICATION

13.3.2 REPARTITION PHYSIQUE

Dans la phase d'étude des supports d'implantation, il s'agit d'affiner cette répartition en vue d'optimiser les conditions d'exploitation. Pour cela il faut prendre en compte des données opérationnelles. Dans le cas le moins limitatif (création de toutes pièces du support matériel et logiciel de ce processus CED), il s'agit d'estimer les conditions d'exploitation de chacun des ordinateurs à mettre en place, et de trouver la structure et le type de leurs interconnexions.

Par exemple dans notre étude sur la CAO des équipements électroniques, il s'agissait de déterminer la puissance de chaque ordinateur, la capacité de sa mémoire (principale et secondaire) et la configuration du poste de travail.

Dans ces cas plus restrictifs (utilisation de matériel ou de logiciel existants), il faut refaire l'analyse de la distribution en faisant apparaître des paramètres supplémentaires caractérisant le contexte.

De plus on peut optimiser le fonctionnement: par exemple la duplication de certaines fonctions importantes minimise les transferts des données.

Au moment du choix de mise en oeuvre trois questions fondamentales se posent:

- choix du niveau de distribution:
nous faisons la distinction entre deux niveaux de distribution, soit à l'extérieur des fonctions de base (base d'algorithmes, base de données, moniteur de conception, moniteur de communication, poste de travail), soit à l'intérieur (donc cachée pour les niveaux plus élevés):
 - + dans le premier cas, le problème se situe dans le choix de la répartition de ces fonctions. Il s'agit surtout d'acheminer les informations entre ces différentes fonctions.
 - + dans le deuxième, la répartition au sein de la base d'algorithmes, de la base de données, ..., s'effectue à l'aide d'outils spécifiques internes à chaque fonction de base.
- choix du niveau de transparence de la distribution: les constatations précédentes sont directement liées à la question fondamentale sur la transparence de la répartition pour l'utilisateur et le système CAO. Sans vouloir répondre à cette question, car elle est trop dépendante de l'application et du contexte d'implémentation, il est

important de constater que l'expérience montre que 80 % des informations utilisées dans un poste de travail resteront toujours locales, et 20 % seulement, seront utilisées ailleurs. De plus la CAO utilise souvent des machines variées très spécialisées, n'ayant pas forcément des systèmes de gestion de bases de données, il est donc difficile de disposer, dans ce contexte, d'un SGBD réparti. Il paraît donc utile que le système CAO soit en mesure de connaître sa répartition. La solution se situe, d'après nous, dans une organisation mixte. Par exemple le moniteur de conception peut être réparti (surtout s'il est implicite); il connaît la localisation des données et des algorithmes et peut donc lancer des opérations précises (transfert des données, exécution locale, exécution distante,...) sur des SGBD et des SGBA autonomes, dupliqués sur chaque site.

- choix du type de contrôle d'exécution: il nous paraît intéressant de garder un contrôle centralisé (maître - esclaves), dans le cas d'un système mono-utilisateur. A partir du moment où on est en présence d'un système multi-utilisateurs, avec distribution des utilisateurs un contrôle réparti s'impose. La synchronisation est alors implicite et s'effectue par les données.

13.3.3 MISE EN OEUVRE DE LA REPARTITION

Pour réaliser un système réparti chaque constructeur propose actuellement des outils de répartition propres à son matériel. Nous n'avons pas voulu nous lier à des primitives précises (et du même coup à un système de répartition précis), mais travailler d'abord de façon plus conceptuelle et prévoir un processus d'adaptation à une réalisation particulière. Ce processus d'adaptation d'un mécanisme abstrait à un mécanisme concret est de même nature pour les autres fonctions de base (activation, chargement, ..) que nous étudions plus loin.

En analysant les outils proposés nous les avons classés en cinq niveaux de complexité croissante et nous avons étudié comment les intégrer dans la production d'un système CAO réparti:

- la répartition des accès, que l'on appelle aussi le fonctionnement en TERMINAL VIRTUEL, est la plus simple à prendre en compte. Pour l'établir, il suffit d'insérer dans le programme d'initialisation de la session, la séquence d'établissement d'accès au processeur distant, et à la fin de la session, de rompre l'accès.
- le partage de ressources: un ensemble compatible d'outils dans chaque ordinateur du système échange des messages dans un format commun et selon un ensemble de règles définies liées aux conventions d'accès aux ressources souvent

- périphériques.
- le transfert de fichiers: un ensemble d'outils permettant l'échange de messages et ayant accès à la gestion des fichiers dans chaque point du réseau.
 - l'accès aux fichiers à distance: un ensemble d'outils permettant d'accéder indifféremment à des fichiers locaux ou distants.
 - l'application distribuée: un ensemble d'outils qui permet de ne pas prendre en compte la localisation des données et des traitements lors de la conception de l'application (la transparence de la répartition). Celle-ci peut d'ailleurs évoluer dans le temps. Les outils de base sont: la LOCALISATION des données et des programmes, le TRANSFERT DE FICHIER d'un point du réseau à un autre, et la possibilité de COMMUNICATION ENTRE TACHES qui se déroulent en différents points du réseau. Ces outils de base permettent de gérer la répartition dynamique des données et des traitements. On peut donc faire appel à ces outils directement dans les séquences d'activation des différentes tâches du processus de conception ou prendre en compte des outils de plus haut niveau, comme par exemple SIGOR (Dan.78), construits à partir de ces outils de base.

13.4 TYPOLOGIE DES MATERIELS ET DES LOGICIELS

Nous avons introduit la notion de machine abstraite afin de disposer d'une description fonctionnelle stable et indépendante du contexte particulier du matériel et des logiciels. La machine abstraite doit pouvoir être réalisée indépendamment des applications que l'on bâtit dessus, mais en tenant compte de la configuration spécifique matériel-logiciel.

L'ensemble des liens entre la machine abstraite et sa réalisation, définit des POINTS D'APPUI établissant de façon précise les choix que l'on, doit effectuer pour obtenir une réalisation concrète de la machine abstraite (FIG. III - 13.7).

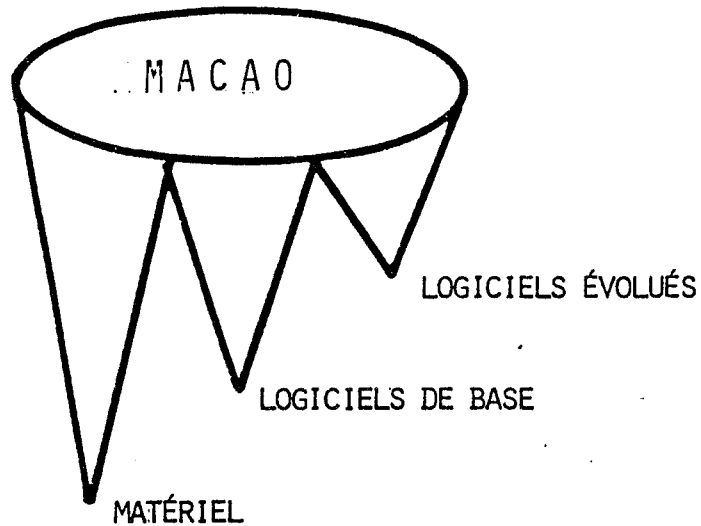


FIGURE III - 13.7 : POINTS D'APPUI DE LA MACAO

Il s'agit donc de déterminer de manière précise l'ensemble des points d'appui et les choix que l'on peut effectuer pour chacun. Enfin d'éviter une variété trop importante il est indispensable de minimiser le nombre de ces points d'appui et de bien les définir. Toutefois ces points d'appui ne peuvent pas être tous du même niveau car certains doivent préciser des choix de base (mémoire principale paginée ou pas), et d'autres des choix plus macroscopiques (choix de logiciel pour la réalisation de la BDP),.

C'est pourquoi nous avons mis en évidence trois niveaux de point d'appui concernant respectivement le matériel, les logiciels de base et les logiciels évolués.

Nous pouvons maintenant lister les points d'appui que nous avons déterminé pour chacun des niveaux:

A/ POINTS D'APPUI MATERIEL:

il s'agit de préciser les caractéristiques principales du matériel sur lequel le système CAO fonctionnera:

- mémoire: taille limitée ou non, longueur des mots,...
- chargement: dynamique, à la demande, recouvrement, pagination, segmentation,..
- langage de programmation: pour tâches, pour enchaînement,..
- composition du poste de travail,

- etc..

B/ POINTS D'APPUI LOGICIELS DE BASE:

il s'agit de répertorier tous les logiciels de base sur lesquels le système CAO s'appuie et notamment:

- logiciel d'exploitation,
- logiciel de distribution,
- logiciel graphique,
- etc...

C/ POINTS D'APPUI LOGICIELS EVOLUES

il s'agit de préciser les logiciels évolués que le système CAO utilisera et notamment ceux qui réalisent les fonctions de base de la machine abstraite MACAO:

- la base de données,
- la base d'algorithmes,
- le moniteur de conception,
- ...

13.5 LANGAGE DE SPECIFICATIONS ORGANIQUES

Le langage de spécifications organiques a pour rôle d'exprimer les choix concernant la mise en oeuvre particulière et définitive du système CAO spécifié fonctionnellement (FIG.III-13.8). Pour cela il offre un ensemble d'énoncés permettant de formuler tous les choix.

Cet ensemble d'énoncés est constitué principalement de trois grandes classes d'énoncés:

- les énoncés décrivant le contexte matériel et logiciel choisi,
- les énoncés décrivant les choix de réalisation de la MACAO dans ce contexte,
- les énoncés décrivant, s'il y a lieu, la distribution de l'application, c'est à dire la projection de l'application sur différents matériels.

L.S.O.

LANGAGE DE SPECIFICATIONS ORGANIQUES

- ▶ **CONTEXTE MATERIEL - LOGICIEL**
 - ◇ MATERIELS
 - ◇ LOGICIELS DE BASE
 - ◇ LOGICIELS EVOLUES

- ▶ **REALISATION DE LA MACAO**
 - ◇ BASE DE DONNEES ET BASE D'ALGORITHMES
 - ◇ MONITEUR DE CONCEPTION ET DE TACHES
 - ◇ MONITEUR DE DIALOGUES
 - ◇ MONITEUR DE COMMUNICATIONS

- ▶ **CHOIX DE PROJECTION DE L'APPLICATION**
 - ◇ REPARTITION DU PROCESSUS DE CONCEPTION
 - ◇ IMPLANTATION DES DONNEES
 - ◇ IMPLANTATION DES TACHES

FIGURE III - 13.3 : LANGAGE DE SPECIFICATION DE MISE EN OEUVRE

La structure de tous les énoncés du langage LSO est identique pour en faciliter l'apprentissage. C'est essentiellement un ensemble de chapitres et dans chacun, une séquence de couples: type de choix - choix. Si l'ensemble des types de choix est figé, au moins pour une version du système car il faut modifier la grammaire des énoncés du LSO, la liste des choix est ouverte. Elle peut être augmentée dynamiquement si on a pris en compte dans le processus d'adaptation le traitement de ce nouveau choix. Par exemple si l'on permet de prendre en compte l'ordinateur de type MITRA, il faut être en mesure de produire par exemple des commandes conformes au langage du commande du MITRA.

Dans les énoncés CONTEXTE, on décrit le contexte particulier du déroulement de l'application. Principalement on décrit le matériel et sa distribution éventuelle, les logiciels de base et les logiciels évolués, leurs caractéristiques importantes, ainsi que leurs liens de dépendances verticales.

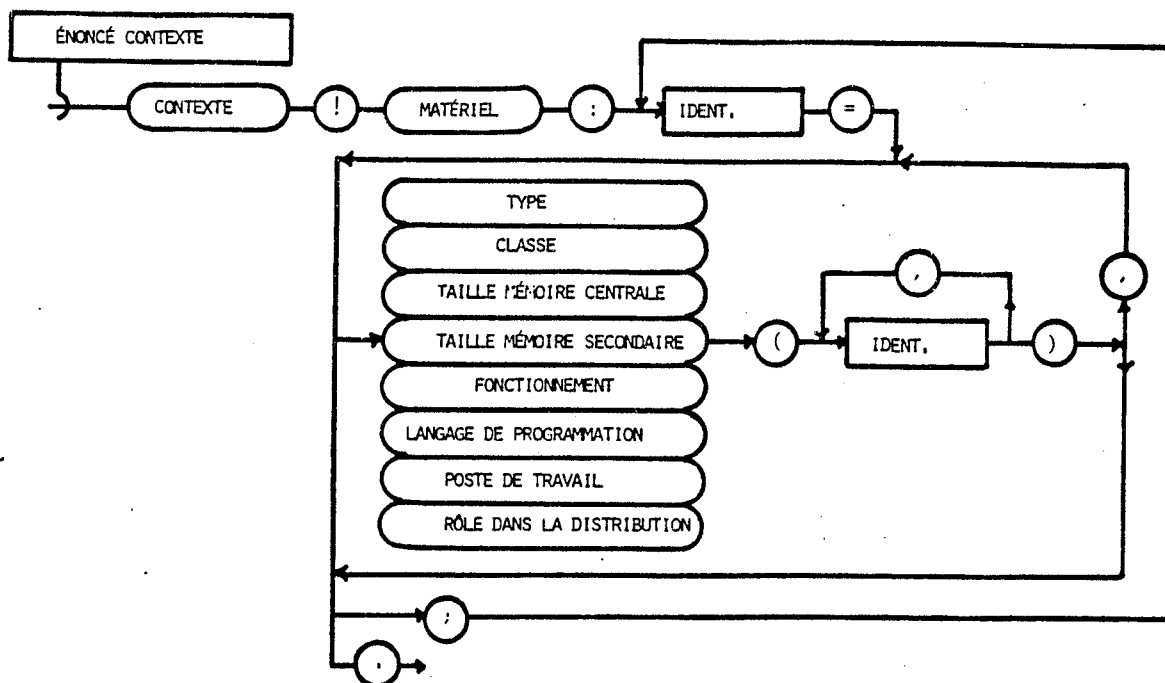


FIGURE III - 13.9 : ENONCE CONTEXTE

Dans les énoncés de réalisation de la MACAO, on précise pour chaque matériel quelles sont les fonctions de la MACAO qu'il réalise et comment.

Il peut évidemment contenir une réalisation complète et autonome de la Machine Absraite, mais il peut aussi, dans le contexte réparti, contribuer de façon partielle à la réalisation de certaines fonctions.

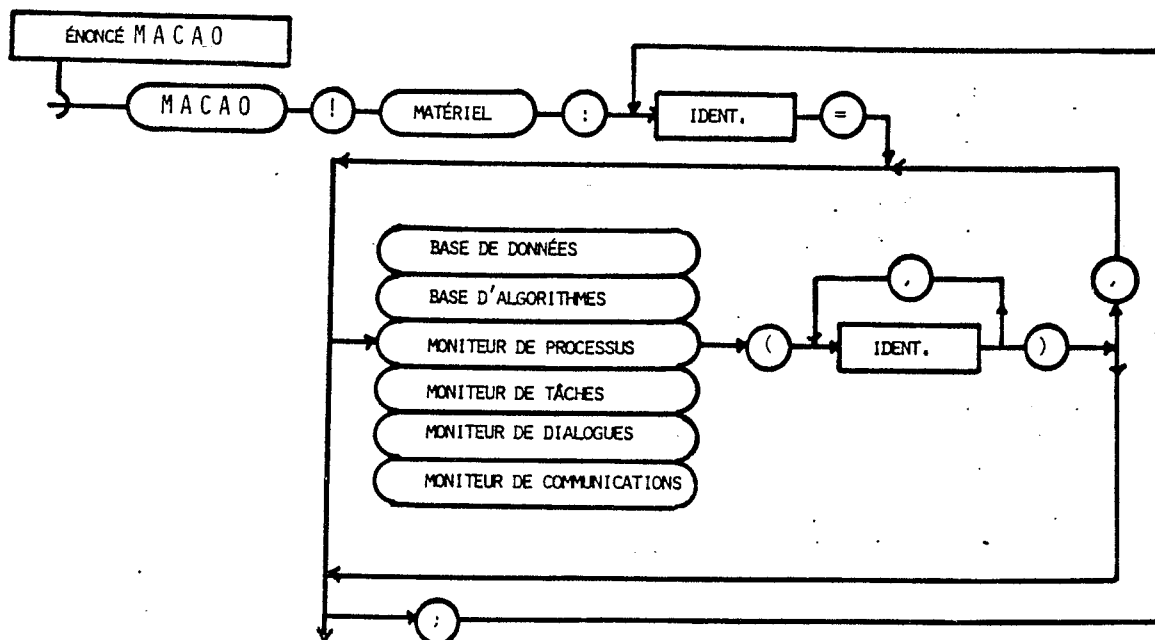


FIGURE III - 13.10 : ENONCE MACAO

Dans les énoncés de distribution, on précise pour chaque matériel son comportement fonctionnel, c'est à dire les phases du processus de conception dans lesquelles il est utilisé, les utilisateurs qui peuvent s'en servir, les atomes d'information qui y sont implantés, ainsi que les tâches qui peuvent s'y dérouler.

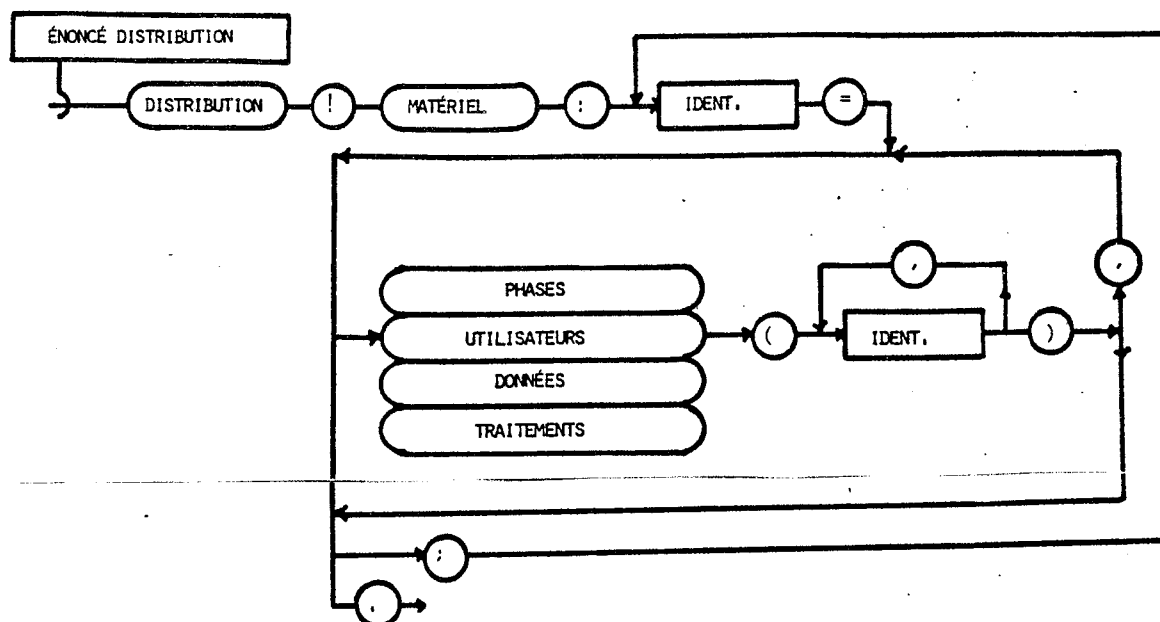


FIGURE III - 13.11 : ENONCE DISTRIBUTION

Si nous comparons LSO à PMS (BeN.71) nous constatons que PMS se limite à la description de la structure (Processor - Memory - Switch) tandis que dans LSO nous décrivons les trois niveaux de choix organiques (matériel, logiciel de base, logiciel évolué).

13.6 METHODE DE MISE EN OEUVRE

La production d'une mise en oeuvre particulière sur une structure cible (matériel et logiciel) peut être accomplie de façon automatique en utilisant la description fonctionnelle fondée sur la machine abstraite. La FIG. III - 13.12 explicite cette méthode. Il s'agit donc d'établir la connexion entre la machine abstraite et la réalisation la plus appropriée pour l'application étudiée.

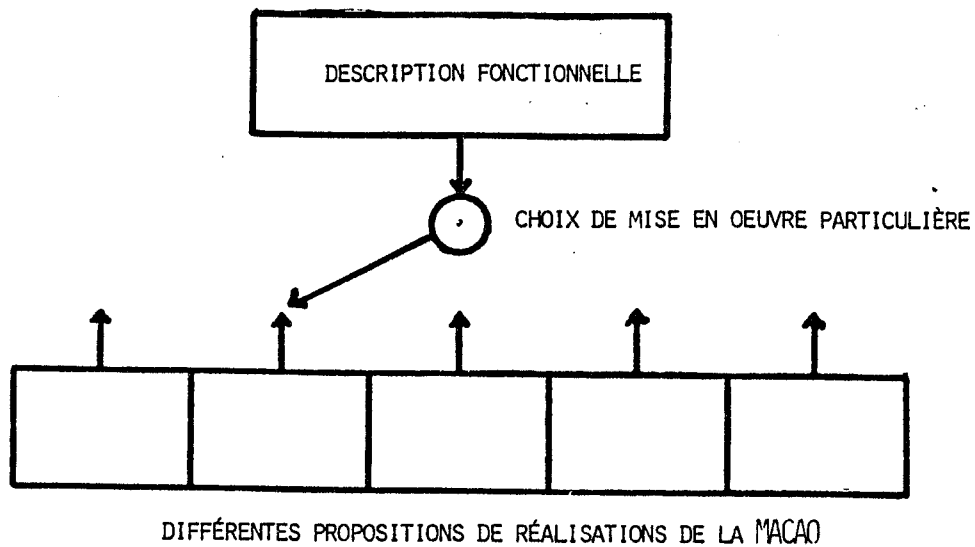


FIGURE III - 13.12 : METHODE DE MISE EN OEUVRE

Les choix de réalisation guident le générateur lors de la prise en compte des modules appropriés de la bibliothèque et de la production d'une version finale du système CAO.

Le processus de mise en oeuvre est schématisé sur la FIGURE III - 13.13. Pour l'illustrer, nous pouvons prendre comme exemple la mise en place du moniteur de tâches.

Nous disposons de la spécification fonctionnelle (donc abstraite et indépendante de toute contrainte de réalisation) de l'ensemble des tâches du système CAO (FIG.III-13.14). Si nous voulons mettre en place ces tâches dans le contexte EXEC-CMS, le processus d'adaptation va d'abord chercher dans la bibliothèque des modules les modules écrits en EXEC-CMS et réalisant les fonctions de base du moniteur de tâches. Puis il va générer des appels à ces modules de base (FIG.III-13.15).

Si on veut travailler dans le contexte FORTRAN, il rend accessible la bibliothèque des modules réalisant les fonctions de base en FORTRAN et génère les appels en FORTRAN.

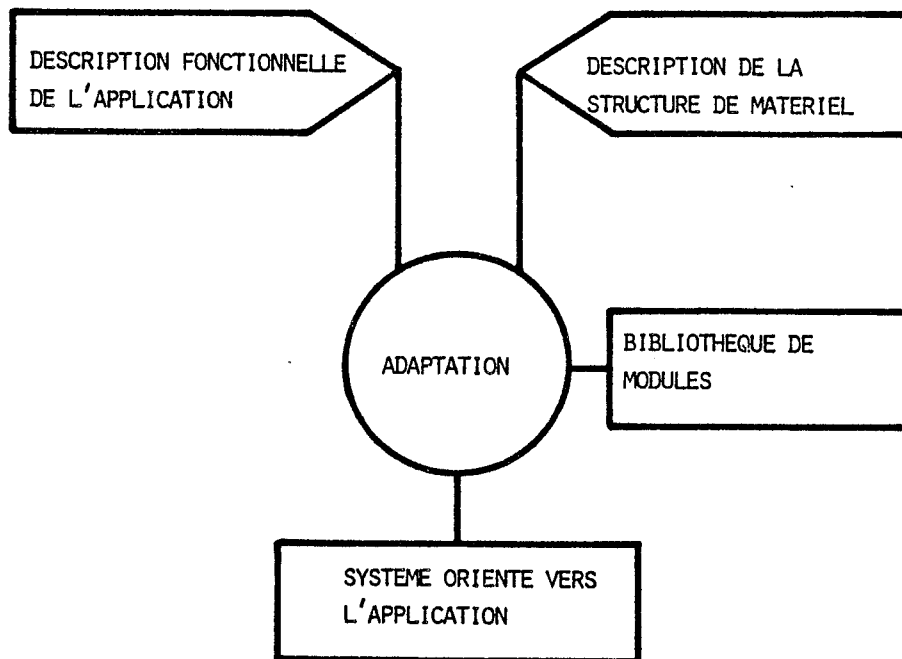


FIGURE III - 13.13: PROCESSUS DE MISE EN OEUVRE

C = SPE " SPECIFICATIONS DU PROGRAMME ARCHITECTURAL "

E : N,R,* ,P* ; S : R* ,P* ,

RES : T2250 ;

VT : N,R,P ;

PE : PARMs(R,*),PARMS(P,*);

A : ER(7),LOAD(TRMATVG),START;

PS : ECLAT ,

C = URBA " ALLOCATION SPATIALE : PROGRAMME URBALLOC "

E : N,R,P ; S : T ,

RES : T2250 ;

VT : N,R,P ;

PE : ER(7),PARMS(R,1),PARMS(T,1) ;

A : LOAD(MAPA),START(IN,FILE,FT01F001,D);

PS : ECLAT ,

FIGURE III - 13.14 : SPECIFICATION FONCTIONNELLE DE TACHES

```

OFFLINE READ C.SPE EXEC
*** C.SPE SPECIFICATIONS DU PROGRAMME ARCHITECTURAL
*****
* E: N R * P *
* S: R * P *
EXEC S,POSTE T2250 &2
EXEC S,VERIF VT &1 N R P
PARMS R *
PARMS P * S,POSTE, S,VERIF, ECLAT - FONCTIONS DE SERVICE
ER 7 DE MISE EN OEUVRE EN EXEC
LOAD TRMATVNG
START PARMS, ER, LOAD START - FONCTIONS DE SERVICE
EXEC ECLAT DE MISE EN OEUVRE DU SYSTEME
D'EXPLOITATION

OFFLINE READ C.URBA EXEC
*** C.URBA ALLOCATION SPATIALE : PROGRAMME URBALLOC
*****
* E: N R P
* S: T
EXEC S,POSTE T2250 &2
EXEC S,VERIF VT &1 N R P
ER 7
PARMS R 1
PARMS T 1
LOAD MAPA
START IN FILE FT01F001 D
EXEC ECLAT
    
```

FIGURE III - 13.15 : APPELS DE TACHES ADAPTES AU CONTEXTE CHOISI (ICI EXEC-CMS)

En adoptant la technique de programmation fondée sur les abstractions et la modularité, nous avons élaboré une gamme de modules interchangeable réalisant les fonctions de la machine abstraite dans des contextes structurels (matériel et logiciel) différents. C'est ainsi que nous sommes en mesure de supporter les configurations suivantes pour les principales parties de la MACAO:

Pour la Base de Données, on peut soit:

- faire appel à un système de gestion de base de données existant et accessible directement par les différents programmes (solution idéale pour une grosse installation),
- utiliser un système de gestion de base de données, mais pas à l'intérieur des programmes, ce qui est souvent le cas pour les SGBD existants, et assurer le lien avec les programmes par fichier intermédiaires.
- assurer la gestion des données en utilisant des fichiers à accès direct (utilisable pour les applications ayant une structure de données monotone et sur de petites configurations),
- utiliser uniquement des fichiers à accès séquentiel.

Pour la base d'algorithmes, on peut soit:

- utiliser le système de gestion de mémoire virtuelle: l'ensemble du système est chargé une fois, puis c'est le mécanisme de gestion de mémoire qui se charge d'effectuer dynamiquement les transferts,
- utiliser le mécanisme de recouvrement du système d'exploitation,
- avoir un superviseur qui gère dynamiquement les chargements des programmes stockés dans une base, en s'appuyant sur les primitives du système d'exploitation,
- avoir recours à l'édition de lien dynamique (MULTICS)
- faire appel à la segmentation

Le moniteur de conception peut être réalisé soit comme:

- un programme directement exécutable
- un interpréteur particulier,
- un interpréteur général dirigé par des données (tables)

En structurant convenablement les interfaces entre ces différentes parties, nous avons rendu possible l'indépendance des choix.

Nous pouvons donner à titre d'exemple trois configurations sur lesquelles nous avons travaillé (FIG. III - 13.16).

Il faut remarquer que ces modules de la MACAO, s'ils sont équivalents fonctionnellement ne le sont pas forcément du point de vue des performances ou de la souplesse d'utilisation. C'est une constatation importante car le choix approprié des composants du système CAO constitue une des conditions majeures du succès du système CAO, tout comme la capacité intrinsèque des algorithmes de traiter efficacement le problème posé.

Dans tout ce travail nous nous sommes limités à l'aspect organisationnel du système CAO. Nous sommes toujours restés au niveau macroscopique et nous n'avons pas étudié par exemple la spécification et la mise en oeuvre des algorithmes de façon interne et microscopique. Nous l'avons fait volontairement car la conception détaillée d'algorithmes fait l'objet de recherches importantes en génie logiciel et les algorithmes utilisés en CAO peuvent être développés par les méthodes proposées.

Nous avons étudié la méthode de transport par modules interchangeables ce qui nous oblige à élaborer les modules équivalents sur le nouveau site. Pour les algorithmes la méthode

| | | | | |
|-------------|----------------------------------|---|--|--|
| CONTEXTE | MATERIEL | IBM 360/67 | SOLAR 16-40 | HB 68 |
| | LOGICIELS DE BASE | CP/CMS GRIGRI | MPES GRIGRI | MULTICS GRIGRI |
| | LOGICIELS EVOLUES | ----- | SOMINE | SOMINE |
| REALISATION | BASE DE DONNEES | FICHIERS SEQUENTIELS | SGBD "RESEAU" SOMINE | SGBD "RESEAU" SOMINE |
| | BASE D'ALGORITHMES | BIBLIOTHECAIRE CMS | BIBLIOTHECAIRE MPES | BIBLIOTHECAIRE MULTICS |
| | ACCES AUX DONNEES | STATIQUE | STATIQUE | STATIQUE |
| | MONITEUR DE CONCEPTION | IMPLICITE | IMPLICITE | IMPLICITE |
| | MONITEUR DE TACHES | GEN-MT _S :EXEC-CMS | GEN-MT _S : TSF-MPES | GEN-MT _S : FORTRAN |
| | EDITION DE LIENS ET CHARGEMENT | PREALABLE STATIQUE-PAGINAT | PREALABLE RECouvreMENT | DYNAMIQUE PAGINATION |
| | MONITEUR DE DIALOGUES | ITERPRETEUR PARTICULIER IP-MD _S | INTERPRETEUR GENERAL IG-MD _S | INTERPRETEUR RENERAL IG-MD _S |
| | MONITEUR DE COMMUNICATION | INTERPRETEUR UNIQUE I-COM _S | INTERPRETEUR UNIQUE I-COM _S | INTERPRETEUR UNIQUE I-COM _S |
| | LANGAGE DE PROGRAMMAT. DE TACHES | FORTRAN+ALGOLW | FORTRAN | FORTRAN |

FIGURE III - 13.16 : PRINCIPALES CONFIGURATIONS

descendante semble être plus appropriée; pour pouvoir utiliser le même algorithme aussi bien dans un environnement FORTRAN (sur un mini-ordinateur) que dans un environnement BASIC (sur un micro-ordinateur) il semble souhaitable de disposer d'une représentation sous forme de schéma de programme afin de produire automatiquement des versions BASIC, FORTRAN, PASCAL

CONCLUSION

CONCLUSION

Comme l'ont démontré les applications étudiées, les systèmes intégrés pour la CAO ne sont pas une vision utopique des chercheurs ou des informaticiens en général, mais correspondent bien à un besoin et à une réalité industrielle.

L'intérêt de l'intégration conception-fabrication dû à la contrainte de la diffusion d'informations et au fait qu'aucune étape précise, conception ou fabrication, ne nécessite ni ne justifie à elle seule l'utilisation de l'ordinateur. L'ordinateur intervient donc comme un outil polyvalent facilitant l'organisation et la communication d'informations.

Il est donc important d'exploiter toutes les possibilités de la CAO:

- remplacement de tâches répétitives par des tâches automatiques,
- collaboration entre l'homme et la machine lors des tâches créatives,
- intégration dans un même système de l'ensemble de ces outils,
- gestion de la cohérence au sein du projet.

Si pour les deux premiers points il est possible de trouver actuellement des solutions opérationnelles sur le marché, il n'en est pas de même pour les deux autres. Nous avons même pu constater que de très bons outils autonomes ne peuvent pas s'insérer dans un système intégré. Ce qui pose de sérieux problèmes.

Il semble donc légitime d'étudier une méthodologie de construction de systèmes intégrés (conception et mise en oeuvre). Les problèmes que l'on rencontre sont très nombreux et variés, mais peuvent être néanmoins classés d'une façon non exclusive, en différentes catégories:

- les problèmes humains, qui sont dus, d'une part à la crainte de l'ordinateur (il nous prendra notre travail, il sait tout faire, etc...), et d'autre part, au refus des individus d'être dérangés dans leurs habitudes (cela marche, alors pourquoi changer ?).

- les problèmes fonctionnels, liés aux problèmes humains, s'amplifient à cause du refus de changement des structures organisationnelles et éventuellement, décisionnelles.
- les problèmes d'innovation, qui sont souvent très importants, car il paraît intéressant d'associer au changement d'organisation, un changement dans la manière de travailler des individus. Souvent, le succès global de l'opération est intimement lié au degré d'innovation apporté. Par exemple, informatiser le dessin, ne signifie pas forcément que le dessinateur fera exactement les mêmes opérations qu'avant, car il s'avère souvent délicat et inutile de garder la même démarche avec un outil différent.
- les problèmes matériels, qui peuvent être trop contraignants avec une configuration figée, ou bien au contraire pas assez avec une liberté absolue quant au choix du matériel.
- les problèmes d'interaction homme/machine, car dans le système CAO, on doit établir une véritable collaboration entre l'homme et la machine.

Pour mener à bien un tel travail, il est indispensable d'avoir d'une part, une connaissance approfondie de ce que la CAO peut apporter, et d'autre part, une connaissance détaillée du milieu d'intégration, contrainte satisfaite très difficilement et uniquement par un travail collectif. Cette tâche est d'autant plus délicate, que pour des raisons diverses, on a beaucoup et souvent trop de degrés de liberté, ce qui devient un couteau à double tranchant: "définissez-nous l'organisation, la structure, le travail, le matériel nécessaire, et faites vite, mais attention si cela ne marche pas".

Ceci n'est pas spécifique à la CAO. Le même problème se pose pour tout système complexe situé dans un environnement complexe, les systèmes d'information par exemple. C'est pourquoi on a vu naître des méthodes de conduite de projets informatiques, d'abord empiriques, puis de plus en plus formelles. Mais ceci a conduit notamment en informatique de gestion, à quelques ratés remarquables. Lorsqu'on en analyse les raisons on constate que les concepteurs ont confondu les méthodes de construction et de mise en oeuvre. Mais le problème fondamental est que ces systèmes ne reflètent pas forcément la réalité du terrain mais le comportement d'autres systèmes du même type. Pour aboutir à des résultats plus convaincants, il faut partir de la pratique et proposer des systèmes et des méthodes de façon plus globale et descendante.

En CAO le même problème a été rencontré et les systèmes généraux pour la CAO, caractérisent et concrétisent cette démarche; une démarche plus conceptuelle est actuellement encore

pratiquement inexistante. Nous pensons avoir contribué à sa promotion.

Ce manque n'est pas uniquement dû aux concepteurs de la CAO. Il faut se rappeler que dans les années 1970-1975, le génie logiciel n'existait pas encore, en tout cas pas en France. On rencontrait seulement dans des domaines d'application variés quelques tentatives de mise en place d'un génie logiciel spécifique à ces domaines. C'était un travail suffisamment complexe pour qu'il occupe les constructeurs des systèmes spécialisés pendant très longtemps.

Actuellement ces méthodes sont devenues opérationnelles et de plus, le génie logiciel a été reconnu comme une discipline importante, autonome et dans une certaine mesure indépendante des applications. Il est donc possible d'effectuer des études moins spécifiques, plus orientées vers la conception des applications.

Nous avons commencé nos travaux bien avant l'apparition du génie logiciel et nous avons travaillé à tous les niveaux de la démarche d'étude, de spécification et de mise en oeuvre des applications CAO. C'est pourquoi nous avons modélisé le cadre de l'action, c'est à dire ce qui caractérise toute application CAO. Ceci a conduit à la définition du modèle conceptuel et nous avons défini le langage conceptuel COLA, permettant d'exprimer les spécifications d'une application CAO.

Ensuite, nous avons modélisé le système support sur lequel les applications CAO peuvent fonctionner. Pour des raisons de généralité et d'indépendance vis-à-vis des configurations particulières de matériel et de logiciel, nous avons d'abord défini un modèle fonctionnel, appelé Machine Abstraite pour la CAO (MACAO). Puis nous avons étudié différentes réalisations concrètes de cette machine dans des contextes spécifiques.

Ayant déterminé ces deux repères fondamentaux nous avons déterminé la méthode permettant de construire des systèmes CAO spécifiques. Cette méthode permet d'exprimer les caractéristiques de l'étude, d'obtenir un système prototype par des techniques de transformation des spécifications et d'aboutir au choix de la configuration finale et de la mise en oeuvre de la version définitive. Cette méthode a été complétée d'une réflexion sur son utilisation dans le milieu industriel.

Les points fondamentaux en sont:

- LA MACHINE ABSTRAITE: machine fonctionnelle facilitant la portabilité et la répartition,
- LE MODELE CONCEPTUEL: définissant le cadre de toute application CAO,
- LE LANGAGE CONCEPTUEL COLA: permettant de décrire les spécifications d'une application de façon précise et non ambiguë,
- LES OUTILS D'AIDE à la construction de prototype,
- LA GESTION DU POSTE DE TRAVAIL,
- LA BASE DE DONNEES DU PROJET et sa gestion,
- LA MODELISATION du processus de conception.

Nous pensons avoir fait un travail original mais il est loin d'être terminé puisque si nous avons bâti une méthode et démontré la faisabilité des outils proposés, nous ne prétendons pas avoir assuré leur opérationnalité ou leur efficacité; une réalisation prototype de la démarche a été construite et quelques applications menées à bien, mais il faut maintenant bâtir une version opérationnelle et l'appliquer à de nombreux cas réels. Ceci permettra de dégager les retombées méthodologiques de ces études, d'affiner éventuellement la méthode et de mener d'autres recherches plus détaillées au sein du cadre défini par ce travail. Ce n'est qu'à partir de ce stade qu'une réalisation d'un système industriel de construction de système CAO pourra être envisagée.

Mais pour traiter de telles applications, il est important de former des utilisateurs de CAO. Actuellement ceci n'est pas encore suffisamment entrepris, comme nous avons pu le constater en étudiant les programmes d'enseignement existants. L'enseignement de la CAO en est encore à ses premiers balbutiements. Pourtant nous pensons que l'enseignement peut constituer un premier banc d'essai des outils ou des systèmes développés, et peut donc servir à valoriser les recherches comme nous avons pu le vérifier avec le système SIGMA-ARCHI (DZZ.77, DZZ.79).

Mais dans l'enseignement de la CAO il ne suffit pas de présenter uniquement des programmes, comme c'est pratiquement toujours le cas actuellement. Il faut aussi et surtout, enseigner les fondements de la CAO et les méthodes d'investigation permettant d'introduire dans les meilleures conditions, la CAO dans le domaine concerné.

Conscient de l'importance de l'enseignement de la CAO, nous avons réfléchi à son contenu et à ses formes afin de proposer une méthode d'élaboration de programmes d'enseignement (Dav.79, Dav.81).

En plus des méthodes de construction, il reste en CAO un certain nombre de points précis sur lesquels des études fondamentales devraient être entreprises, comme:

- le problème de partage des connaissances entre l'homme et la machine,
- la communication homme-machine,
- la gestion des données en CAO.

Il nous semble que notre expérience pourrait être intéressante pour d'autres applications, la méthode et les outils que nous avons développés peuvent se généraliser.

Nous pourrions également travailler sur l'automatisation plus grande de la production des programmes afin de diminuer la programmation notamment celle des programmes spécifiques, soit par la production automatique de ces programmes, soit par le remplacement de ceux-ci par des programmes plus généraux fondés sur l'intelligence artificielle.

Toutefois il nous semble intéressant de poursuivre encore les recherches dans le domaine de la CAO actuellement en pleine expansion, et qui manque encore de travaux fondamentaux ainsi que d'une expérience d'utilisation suffisante des réalisations pratiques.

ANNEXES
APPLICATIONS

RESUME: Ce travail n'aurait pas pû être réalisé sans un lien direct et étroit avec des applications concrètes. On a donc mené de front l'étude présentée avec un certain nombre d'applications: la CAO des systèmes logiques, la CAO en architecture, la CAO des mécanismes électro-mécaniques, des équipements électroniques, des chaussures et des meubles. Dans ces annexes on présente deux applications en axant chaque présentation sur un aspect important du problème traité. La CAO en architecture met en évidence l'intérêt des systèmes intégrés et montre le lien généralité-spécialité des systèmes. La CAO des équipements électroniques, tout en confirmant l'intérêt de l'approche intégrée, étudie surtout le problème de l'introduction de la CAO en milieu industriel.

ANNEXE 1
CAO EN ARCHITECTURE

RESUME: présentation des principaux résultats des travaux sur la CAO en architecture. Ceux-ci ont été orientés vers la justification de l'approche intégrée de la CAO.

A1. CAO EN ARCHITECTURE

A1.1 CADRE ET OBJECTIFS DE L'ETUDE

Dans cette annexe nous présentons les travaux que nous avons menés en Conception Architecturale Assistée par Ordinateur (C.A.A.O.).

Si dans d'autres applications nous avons travaillé en collaboration avec des entreprises, ce qui donnait à nos recherches un caractère opérationnel, dans cette étude nous avons travaillé de façon différente. En effet l'intérêt de la CAO pour les architectes n'est pas encore évident et le milieu n'y est pas encore sensible. Les raisons en sont diverses et parmi celles-ci la taille réduite des agences (souvent un seul architecte avec un dessinateur et une secrétaire), le coût élevé du matériel informatique et surtout la façon de travailler des architectes (travail très créatif, intuitif où des considérations esthétiques, sociologiques et psychologiques priment).

Aussi nous avons donné à nos travaux un caractère plus prospectif, et pour pouvoir mener ces travaux nous avons constitué au sein de l'équipe de recherche un groupe pluridisciplinaire travaillant sur le sujet.

Le caractère exploratoire de cette étude nous a conduit à montrer surtout les principes sans espérer aboutir à des outils opérationnels rapidement.

Pendant les dix dernières années, les tentatives d'utilisation de l'ordinateur dans le processus de conception architecturale ont été assez nombreuses. Nous pouvons distinguer deux approches. Dans la première, on a recherché au sein du processus de conception les étapes dans lesquelles l'ordinateur peut intervenir, et on a développé des outils spécifiques pour assister ces diverses tâches. Les étapes les mieux étudiées ont été l'analyse de données et l'allocation spatiale.

Dans la deuxième approche, le processus de conception est considéré comme un tout, et on a essayé d'assurer la gestion des données et leur évolution pendant le processus de conception. L'intégration d'outils d'aide à la conception constitue une évolution naturelle de cette approche.

Nous avons orienté nos recherches plus particulièrement sur l'utilisation d'une approche intégrée de la CAO. Celle-ci c'est concrétisée par la définition d'un système pour la CAAO, le système SIGMA-ARCHI, que nous décrirons.

Ce système est un support général pour la CAAO qui peut être, d'une part enrichi par de nouvelles opérations d'intérêt général, d'autre part adapté à une application spécifique par adjonction d'opérations spécialisées. Nous allons illustrer ces deux aspects, le premier par la présentation de la prise en compte d'un système constructif ouvert, et le second par la construction d'un sous-système orienté vers le logement: SIGMA - LOGEMENT. Nous voulons surtout montrer la hiérarchie des systèmes et comment on peut les construire à l'aide de notre démarche.

A1.2 JUSTIFICATION DE L'APPROCHE INTEGREE

Au lieu de présenter un éventail complet de justifications de l'approche intégrée, étudions plutôt un exemple.

Nous avons voulu savoir pourquoi les programmes thermiques, d'éclairage, de chauffage, de structure et de métré par exemple sont si peu utilisés. Ces programmes existent et sont maintenant opérationnels. En outre la situation actuelle impose de plus en plus de prendre en considération les problèmes d'énergie dès le début du projet.

A l'occasion de la réalisation d'un programme de calcul thermique pour un bureau d'études techniques, nous avons fait le bilan d'utilisation de ce programme. Pour un bâtiment de quatre étages avec quatre appartements par étage, il faut deux jours pour saisir les plans et les transcrire sur les bordereaux. La perforation demande un jour; le passage en machine peut être estimé à 100 F et il faut une demi-journée pour analyser les résultats et donner le verdict.

La saisie et la perforation sont effectuées par le personnel très peu qualifié. Ces deux étapes sont non seulement non créatives, mais sont en plus une source d'erreurs de transcription qui font qu'il faut souvent plusieurs passages en ordinateur pour les éliminer.

On constate que le coût est assez considérable et surtout que le rapport travail total / travail utile est très faible. Il semble donc important d'éliminer les étapes non créatives qui sont ici le codage et la perforation. Ceci est possible si la phase précédente, c'est à dire l'établissement des plans est déjà informatisée et peut donc fournir des plans codés.

L'idée d'intégration est donc de permettre l'enchaînement des programmes et de n'entrer des informations qu'au moment créatif. Une fois en machine, ces informations sont disponibles pour tous les traitements que l'on veut effectuer.

A1.3 OBJECTIFS DU SYSTEME SIGMA - ARCHI

Le système SIGMA-ARCHI est un système intégré pour la Conception Architecturale Assistée par Ordinateur, développé par notre groupe pluridisciplinaire composé essentiellement de F.Z'GRAGGEN et F.ZANELLI - architectes, P.LESPINASSE et moi-même - informaticiens, V.RIVERO - ingénieur civil et R.VILLAMAYOR - experte en recherche opérationnelle (DaR.74b, DLR.76a, DLR.76b, DRZ.75, Equ.76a, Equ.76b)).

Dans la conception du système nous avons tenu compte des recherches en méthodologie de la conception architecturale effectuées au cours de ces dix dernières années (Alexander, Negroponte, Eastman, Rittel,...).

Le système constitue un support permanent des informations, et assure le dialogue homme-machine à partir d'un poste de travail graphique (terminal graphique, tablette à numériser, table traçante, ...). Le système gère les données architecturales et permet de les manipuler pour obtenir un objet architectural complet.

A1.4 DIFFERENTS NIVEAUX D'INTEGRATION

L'étude de ce système nous a conduit à réaliser progressivement différentes versions qui correspondent chacune à un niveau d'intégration.

La première qui s'attaque à l'aspect le plus visible, les programmes et leur enchaînement. Nous avons inventorié un ensemble d'outils de conception couvrant un grand domaine d'activités de conception:

- ACQUISITION DE DONNEES: à l'aide d'interaction graphique, l'utilisateur définit les unités de conception du projet, leurs propriétés et leurs relations mutuelles,
- MANIPULATION DE DONNEES: différentes opérations sur les propriétés et les relations permettant de mieux percevoir le projet (composition, combinaison, ...),
- ANALYSE DE DONNEES: opérations de hiérarchisation pour décomposer le projet en sous-projets, analyser des proximités, etc...

- GENERATION DE FORMES: à partir des données de surface et de proximité, générer des configurations,
- ALLOCATION SPATIALE: en utilisant diverses techniques d'allocation (automatiques ou interactives) et selon les différents types de contraintes et la complexité des objets, produire les schémas de plan,
- EVALUATION DES CONFIGURATIONS: vérification graphique de la satisfaction des contraintes. L'utilisateur choisit les configurations et les relations qu'il souhaite évaluer,
- LANGAGE DE DESCRIPTION de configuration en 2 ou 3 dimensions,
- VISUALISATIONS de plans et différentes vues d'objets en 3 dimensions (perspectives),
- CREATION, MODIFICATION ET FINITION DE PLANS, utilisation de catalogues d'équipements,
- EVALUATIONS TECHNIQUES: métré, calculs thermiques, etc..

L'utilisation de programmes spécifiques correspondant chacun à une tâche ponctuelle du processus de conception, nécessite une préparation considérable des données, des périodes longues et fastidieuses d'interprétation des résultats ainsi que la préparation de nouvelles données pour le programme suivant. La manipulation importante de données par l'utilisateur contribue à rendre l'emploi des programmes plus difficile, tout en étant une source d'erreurs fréquentes, pouvant même annuler les avantages offerts par certains programmes.

La première solution consiste à remplacer la transcription manuelle des données par une transcription automatique par des programmes d'interface. Evidemment ces programmes ne peuvent pas s'enchaîner n'importe comment, il faut donc étudier le processus de conception au sein duquel ils doivent être utilisés.

Mais en architecture on s'appuie essentiellement sur un processus de conception implicite et non formalisé, le rendre explicite n'est pas facile, car les méthodes utilisées varient d'un architecte à l'autre et souvent d'un projet à un autre. Si on arrive à l'explicitier il est alors possible de créer des chaînes méthodologiques des programmes.

C'est ce que nous avons réalisé dans la première version du système SIGMA-ARCHI (FIG.A-A1.1), où nous avons organisé un réseau de traitements reliés par des interfaces. L'inconvénient majeur de cette solution est la rigidité des chaînes pré-définies, la complexité et le coût des programmes d'interface et la

nécessité d'écrire de nouveaux programmes d'interface chaque fois que l'on veut insérer un nouveau traitement.

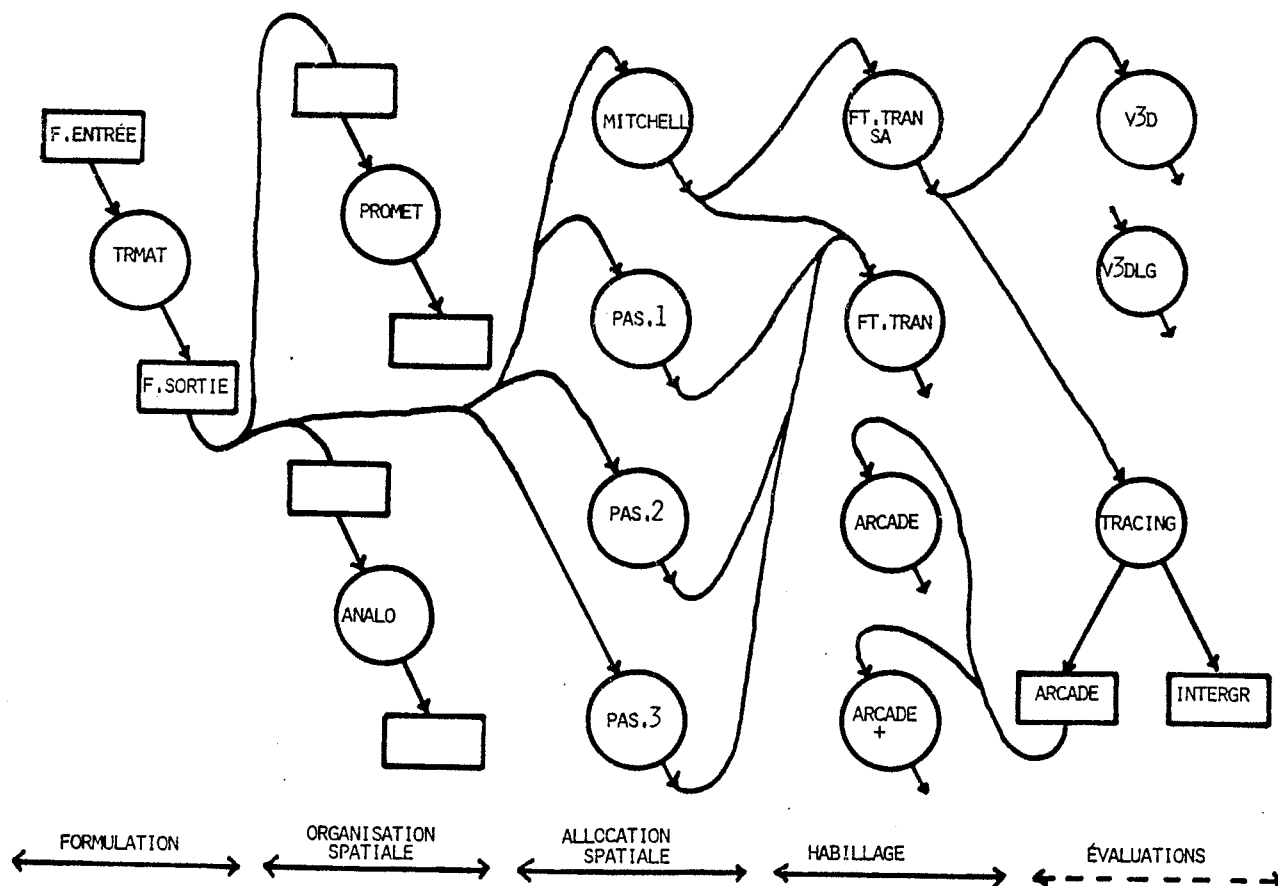


FIGURE A - A1.1 : ORGANISATION DE SIGMA-ARCHI - PREMIERE VERSION - INTEGRATION DES TRAITEMENTS

La seconde solution que j'ai étudié avec V.RIVERO (Riv.77), tend à rendre l'organisation plus homogène en éliminant les programmes de transcription. L'objectif de l'étude est de permettre l'intégration des différents programmes assurant une circulation "naturelle" des données sans manipulations superflues.

Dans la première solution nous avons insisté sur l'intégration des traitements, dans la seconde nous voulons nous appuyer sur les données, c'est à dire réaliser l'intégration par les données.

Pour cela il faut répertorier toutes les données manipulées par le système et dégager leur progression au sein du système (au sein du processus de conception). Dans ce cas les traitements servent à faire évoluer ces données.

Pour les données, on constitue une base de données intégrée, capable de recevoir toutes les données nécessaires à l'ensemble des programmes. Ces données sont stockées directement dans les formats demandés, sans transcriptions importantes.

Pour les programmes, on procède à l'éclatement de leur structure en séparant les entrées/sorties du traitement proprement dit. Les entrées/sorties classiques sont remplacées par des accès à la base.

De cette façon, on obtient une organisation naturelle des données car les données de chaque programme sont bien définies et stockées.

Néanmoins, cette organisation pose un problème important. Beaucoup de données stockées sont liées fonctionnellement et la modification d'une donnée demande une mise à jour des données associées. Maintenir l'intégrité des données est très compliqué et coûteux.

La solution consiste en une base de données ne contenant que des informations non redondantes; les informations redondantes sont produites à partir des données stockées chaque fois que c'est nécessaire. Dans ce cas, les données fonctionnellement liées sont réduites à des parties indépendantes, stockées de façon indépendante. Quand un programme particulier demande des données organisées d'une façon particulière, les données sont calculées à partir des données stockées, fournies aux programmes, puis réintégrées dans la base sous la forme non redondante. Ainsi le problème de consistance disparaît. De plus, on est en présence d'une base de données plus compacte et plus maniable. Il en découle une possibilité d'enchaînement à la demande, et une extensibilité plus grande.

Les principes de cette seconde solution étant précisés, nous pouvons présenter les trois aspects de la démarche architecturale assistée par ordinateur.

AI.5 PROCESSUS DE CONCEPTION ARCHITECTURALE

L'introduction de la CAAO dans le domaine du bâti ne se fait pas sans difficultés, l'obstacle principal au développement des outils CAO pour l'architecte est le manque de connaissance sur le but de son activité et sur son activité elle-même.

Les travaux sur la méthodologie architecturale sont très récents et font encore l'objet de grandes controverses. Ils ont permis d'éclaircir quelque peu ce qui jusqu'alors était considéré comme un processus de créativité, de nature fondamentalement intuitive ayant un caractère irrationnel. Ceci a permis aussi d'identifier des phases, voire des tâches bien définies qui

pourraient, dans certains cas, être automatisées et soumises à l'ordinateur.

En classant et nommant ces phases, on peut, malgré un manque de consensus, arriver au modèle général suivant:

- formulation
- organisation spatiale
- modularisation
- allocation spatiale
- instrumentation
- traitements documentaires

La FORMULATION est l'énoncé du futur contenu socio-culturel du projet, en ce qui concerne les activités, l'ambiance, les us et coutumes. C'est une phase de spécification du projet qui produit comme résultat un "PROGRAMME ARCHITECTURAL".

L'ORGANISATION SPATIALE permet de caractériser les "espaces" qui contiendront les activités en fonction des propriétés (surface, éclairage, isolations acoustique et thermique etc.) requises pour le déroulement normal de l'activité. Par activité on entend tout processus qui est ou qui peut être réalisé dans un lieu précis, indépendamment de tout autre processus (manger, se reposer, lire, etc.). Le résultat de la phase d'organisation spatiale est donc une liste d'espaces qualifiés par leurs propriétés et par leurs relations avec les autres espaces.

La MODULARISATION est une phase optionnelle, nécessaire lorsque pour des raisons économiques, on veut utiliser des composants industriels standardisés ou/et réduire le nombre de types de composants utilisés dans un projet spécifique.

L'ALLOCATION SPATIALE génère une configuration et un arrangement des espaces satisfaisant aux relations définies lors de la phase d'organisation spatiale.

L'INSTRUMENTATION détermine l'habillage matériel des enveloppes résultant de la phase précédente.

Le TRAITEMENT DOCUMENTAIRE génère tous les documents nécessaires à la production de l'objet conçu: devis, plans, etc..

Du point de vue des informations traitées, ce modèle du processus de CONCEPTION ARCHITECTURALE peut être schématisé par la figure A - A1.2. L'association des activités spécifiées par le "programme architectural" et des valeurs des propriétés précisant les conditions d'ambiance permettant leur exécution, détermine l'"espace de réalisation" de chaque activité. Ces activités, se réalisant dans un univers commun, sont en relation. La formalisation d'un problème de conception architecturale exige la détermination d'un ensemble de relations, exprimant toutes les

interactions possibles entre les espaces de réalisation des différentes activités.

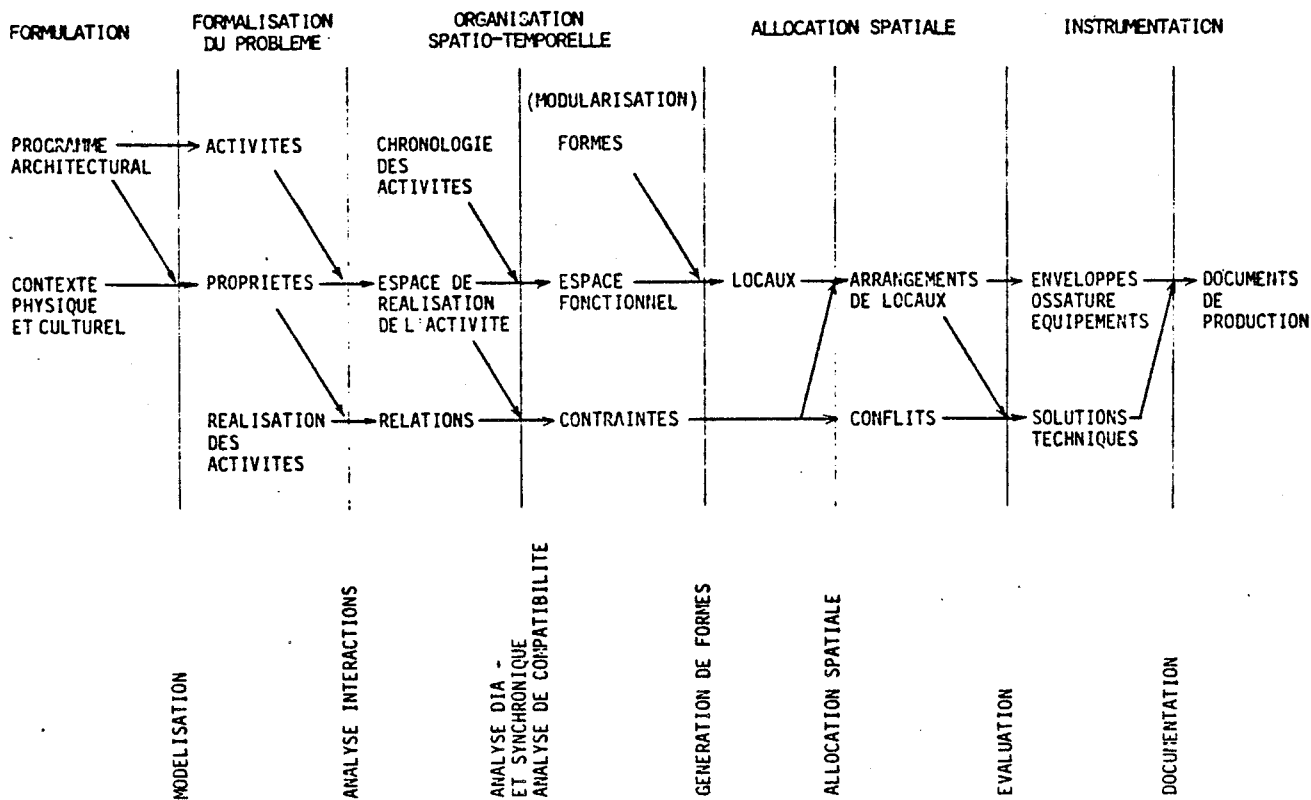


FIGURE A - A1.2 : UN PROCESSUS DE CONCEPTION ARCHITECTURALE (Equ.76b, Riv.77)

La réalisation et les espaces de réalisation d'une ou plusieurs activités - y compris elle-même - peuvent être conflictuels. Il faut donc formuler des contraintes éliminant ces conflits: c'est l'ensemble de contraintes qui devra être satisfait pour toute configuration valable. L'analyse de l'évolution temporelle des activités et de la compatibilité (équivalence) de leur espaces de réalisation permet de regrouper certaines activités compatibles en "espaces fonctionnels".

Les critères d'ambiance de chaque espace fonctionnel sont déterminés par l'union des critères d'ambiance de chacune des activités qu'il englobe.

De ce fait, on peut déduire certaines propriétés de l'espace fonctionnel à partir des propriétés correspondantes des activités.

Les valeurs des propriétés géométriques, topologiques (voire sémiologiques) de l'espace fonctionnel (surface, volume, rapports dimensionnels, convexité, etc) permettent de délimiter l'ensemble de formes adaptables à un espace fonctionnel. Si un processus de modularisation a eu lieu, alors seulement certaines familles de formes sont permises et l'ensemble de formes possibles est encore réduit.

Un espace fonctionnel muni d'un ensemble de formes constitue un local. La définition de l'ensemble des locaux correspondant à un problème de conception architecturale, marque l'aboutissement d'un "processus morphogénétique atomique" dans lequel une unité de conception, définie au départ en des termes flous en tant qu'ensemble d'activités, est progressivement caractérisée - conçue, affinée, imagée - jusqu'à l'obtention d'un local, atome de forme, qu'il faudra placer par rapport aux autres, et matérialiser pour obtenir la configuration finale, solution du problème.

A1.6 INFORMATIONS GEREES PAR LE SYSTEME

A partir du modèle du processus de conception, nous déduisons trois types principaux d'informations permettant de formaliser un problème architectural :

- les objets du problème: unités de conception
- les relations entre ces objets
- les formes de ces objets et leurs arrangements

Les informations concernant les unités de conception sont de deux types :

- une description documentaire sous forme de texte, contenant les spécifications de l'unité et exprimant tous les aspects non formalisables. Elle n'est exploitable que pour la production de documents destinés à l'information du groupe de concepteurs.
- une modélisation de l'unité de conception à l'aide des propriétés pertinentes: surface, volume, conditions requises en ce qui concerne le bruit, l'éclairage, le chauffage, etc. Ces valeurs, sauf redéfinition du problème, restent invariables durant la conception. Elles sont soit données par l'utilisateur, soit calculées par des outils du système.

L'ensemble de RELATIONS permet d'exprimer les liens fonctionnels et les notions de proximité, de compatibilité, etc.. entre les différentes unités de conception. Pour des raisons opérationnelles nous avons partitionné les relations en deux classes :

- relations binaires, représentées par un graphe ou matrice booléenne,
- relations renseignées, représentées par un graphe renseigné sur les arcs ou par une matrice.

Cette partition est imposée par la différence entre les opérations pouvant être réalisées avec les éléments de l'une ou l'autre classe.

Les informations de FORME ont été aussi partitionnées en deux classes:

- Les FORMES BRUTES: schémas de plans représentés sur plusieurs supports, tels que les tableaux de domaines simples et les tableaux de domaines à dimensions variables,
- Les FORMES FINIES: structures hiérarchiques ou d'adjacences, support pour la génération de documents de production: plan, perspectives, élévations, coupes, façades.

La structure du système SIGMA-ARCHI dans sa deuxième version est représentée sur la figure A-A1.3.

L'ensemble des informations contenues dans le système à un moment donné reflète l'état d'avancement du projet à cet instant. L'évolution HISTORIQUE DU PROJET est contenue dans un réseau où chaque noeud représente soit une information élémentaire, soit une commande pour l'exécution d'un opérateur SIGMA. Le système enregistre chronologiquement chaque commande de l'utilisateur et l'ensemble des paramètres impliqués. Cet "arbre généalogique" du projet constitue une "trace", permettant à l'utilisateur d'arrêter une séance de travail à n'importe quel moment, et de vérifier ce qu'il a fait pour réorienter la stratégie de conception. L'architecte peut ainsi contrôler tout le fil de sa démarche et connaître le déroulement de ses actions.

A1.7 LES TRAITEMENTS DISPONIBLES

L'apparence externe du système du point de vue utilisateur, est celle d'un réseau d'outils couvrant une vaste gamme d'activités de conception, soit:

- Définition du projet
- Traitements de propriétés et de relations
- Traitements de formes brutes
- Traitements de formes finies
- Traitements d'appréciations
- Traitements documentaires

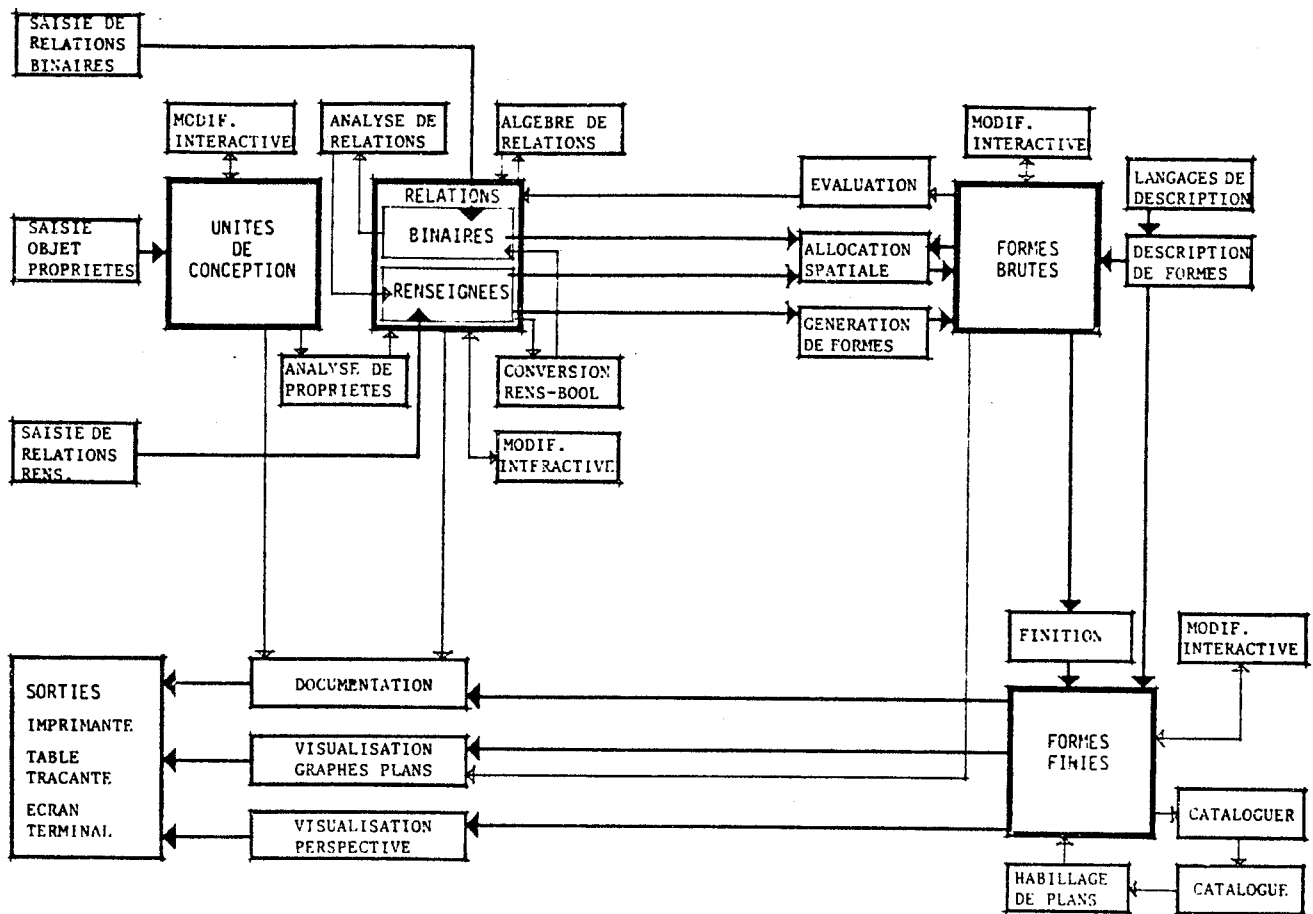


FIGURE A - A1.3 : ORGANISATION DE SIGMA-ARCHI - DEUXIEME VERSION - INTEGRATION DES DONNEES (Equ.76b, Riv.77)

A1.7.1 DEFINITION DU PROJET

A partir de l'ensemble d'activités constituant le programme architectural, le concepteur formalise son problème en spécifiant le nombre d'unités de conception qui interviennent. Il précise de plus leur nom respectif et le nom et type de valeur des propriétés nécessaires. Ces données permettent la génération du "squelette" de la base de données du projet.

A1.7.2 TRAITEMENTS DES PROPRIETES ET DES RELATIONS

Une fois le projet initialisé, l'utilisateur peut fournir d'une part les valeurs des propriétés caractérisant chaque activité, d'autre part les relations entre unités de conception. Les outils de SAISIE DE DONNEES permettent d'évaluer la validité de l'ensemble des valeurs, en fonction de la sémantique associée au type de propriété ou de relation correspondant. Toute valeur

non conforme est immédiatement refusée. L'utilisateur doit alors la corriger interactivement. Des formats d'entrée sont générés automatiquement en tenant compte, du nombre d'unités de conception du problème et du type des relations et des propriétés à saisir. Les moyens interactifs graphiques à disposition, simplifient ainsi au maximum la tâche de l'utilisateur tout en réduisant les possibilités d'erreur.

Pour mieux comprendre les différentes implications des éléments qu'il manipule, l'architecte utilise des outils d'ANALYSE DE DONNEES concernant les propriétés et les relations.

- L'analyse de compatibilité de propriétés permet de formuler des contraintes en décelant les unités de conception conflictuelles, c'est une mise en relation des U.C.
- L'analyse des relations permet une prise en compte globale d'un ensemble non structuré de contraintes, et effectue une hiérarchisation ou une décomposition du problème en sous-problèmes.

Ces traitements effectuent une synthèse de l'ensemble des données, donc d'affiner la définition du problème.

AI.7.3 TRAITEMENTS DE FORMES BRUTES

Une fois que l'architecte a maîtrisé les données et formalisé son problème, il peut procéder au traitement des formes brutes. Pour ceci trois types d'outils existent: les GENERATEURS ou les ALLOCATEURS de formes, les LANGAGES DE DESCRIPTION.

Ces outils génèrent des configurations à partir de la surface des éléments et des relations de proximité, (forces liant des éléments) ou encore à partir des différents degrés d'adjacence: exigé, souhaitable, possible, indifférent, impossible, éloignement maximum.

Les ALLOCATEURS, à la différence des premiers, cherchent à placer des éléments de formes connues en satisfaisant des contraintes définies par un ensemble de relations: adjacence, distance, orientation, accès visuel, etc, pour lesquels ils possèdent des heuristiques de résolutions de conflicts.

Les LANGAGES DE DESCRIPTION permettent également la saisie des configurations. Ce sont des langages algorithmiques contenant des primitives de production d'objets simples, tels que polygones réguliers, prismes, polyèdres réguliers, etc.. Ils contiennent aussi les opérations pour manipuler ces objets, et les instructions de contrôle de l'enchaînement des actions et la prise de décisions.

Le concepteur peut ainsi produire des formes brutes pour chaque élément du problème, un arrangement spatial de ces éléments, et finalement une enveloppe extérieure.

A1.7.4 TRAITEMENTS DE FORMES FINIES

SIGMA dispose d'un support d'amélioration des formes jusqu'à la génération de documents destinés à la réalisation: plans, élévations, coupes, façades, perspectives.

Les opérations de FINITION DE PLANS permettent de traiter les formes brutes, tableaux de caractères représentant une juxtaposition de locaux, par des procédés syntaxiques pour enrichir la description en introduisant des éléments de cloison, des ouvertures, des équipements adaptés à chaque type de local. Le résultat, forme finie, comprend une structure de données plus complexe, permettant de produire des graphismes évolués et de passer à des traitements plus sophistiqués.

A partir de là, il faut, pour aboutir à un prototype réalisable et compact, effectuer un certain nombre de MODIFICATIONS INTERACTIVES: améliorer les formes, préciser les solutions techniques non imposées, éventuellement faire intervenir d'autres équipements. Pour ceci, l'architecte peut reprendre des formes finies ou s'en inspirer, pour en construire de nouvelles à l'aide des outils interactifs. Ces outils permettent un traitement métrique exact, ainsi qu'un travail hiérarchisé dans lequel différents niveaux de description sont permis.

Par ailleurs pour HABILLER CES PLANS, l'architecte dispose de catalogues pouvant contenir des éléments de structure, des équipements du mobilier, etc..

Pour contrôler globalement son travail, il a la possibilité immédiate de produire des images en trois dimensions des formes qu'il a obtenues; par exemple en utilisant la hauteur des locaux pour construire des prismes à partir des plans de base. Il suffit de spécifier le point de vue pour obtenir une série de perspectives par l'intermédiaire desquelles l'architecte peut tourner autour de son objet, passer au-dessus ou pénétrer à l'intérieur.

A1.7.5 TRAITEMENTS D'EVALUATION

Jusqu'à présent nous avons décrit le processus qui permet, à partir de données non structurées au départ, de parvenir à un objet architectural fini. Tout au long du déroulement du projet, SIGMA-ARCHI contrôle le processus de conception, notamment en ce

qui concerne le degré de satisfaction des résultats obtenus et la cohérence de la démarche.

Une fonction d'évaluation globale des configuration permet à l'architecte de connaître le niveau de performance de ses résultats (APPRECIATION DES RESULTATS). Cette fonction détermine le degré de satisfaction par la configuration obtenue, de l'ensemble des contraintes, posées en tant qu'exigences.

Ceci permet la détection des critères non suffisamment satisfaits, et choisit entre des solutions "naturelles" (déplacement interactif d'une pièce à l'intérieur de la configuration) ou "artificielles" (moyens techniques) pour y remédier.

Il n'est pas toujours suffisant de se contenter de l'analyse du résultat pour résoudre une carence ou un blocage dans le processus du projet, il devient parfois nécessaire d'analyser la démarche. SIGMA permet d'EVALUER LA COHERENCE de chaque action de la séquence par rapport à l'état d'avancement du projet résultant des actions précédentes.

A1.7.6 TRAITEMENTS DOCUMENTAIRES

Conséquence du travail pluridisciplinaire, le projet architectural est très rarement le fait d'un seul individu et intervient le problème de la circulation d'une information cohérente entre les différents membres de l'équipe participant à la conception. L'utilisateur peut commander la production de documents sur un support permanent. Ces imprimés peuvent être des listes alphanumériques d'unités de conception, de propriétés ou de relations; des dessins automatiques sur calques, de tableaux, graphes, arbres, plans, coupes, axonométries, perspectives, à toutes échelles et tous niveaux de détail. Ces documents de travail peuvent être ainsi communiqués aux intéressés.

A1.8 UTILISATION DU SYSTEME SIGMA-ARCHI

Le champ d'application est une des caractéristiques fondamentales d'un système CAO. S'il s'applique à un domaine assez large d'applications on parle d'un système général, s'il concerne seulement un domaine très restreint on parle d'un système spécialisé.

Le système SIGMA-ARCHI est un système général pour la conception architecturale. Il permet la manipulation des unités de conception, de leurs propriétés et leurs relations, ainsi que des formes brutes et des formes finies. Les outils proposés sont soit des outils abstraits (ne prenant pas en compte la sémantique des données), soit des outils concrets (prenant en compte la

sémantique). Dans le deuxième cas, cette sémantique est valable dans tout le domaine d'application (par exemple la relation de proximité nécessaire pour l'allocation spatiale). Il semble important de préciser comment on s'en sert.

Comme tout système CAO, ce système est défini par un modèle abstrait qui exprime:

- le processus de conception supporté,
- les objets manipulés,
- les traitements disponibles.

Le traitement d'un projet nécessite de définir le modèle concret du champ d'application auquel on veut s'intéresser pour prendre en compte la sémantique spécifique du projet. Ceci se fait par la définition des types des propriétés et des relations que l'on veut considérer. Ainsi on bâtit progressivement un modèle spécialisé adapté à ce champ. Une fois ce modèle en place on peut traiter le projet précis.

Si la spécialisation des données peut s'effectuer dynamiquement, ceci est beaucoup plus difficile pour le processus et les traitements. Une solution est de définir un processus de conception assez général pour être valable dans différents cas et d'utiliser des traitements concrets ou abstraits plus généraux en effectuant le retour sémantique manuellement. Si ceci n'est pas suffisant il faut pouvoir modifier le champ d'action du système général et ceci soit en largeur soit en profondeur.

Horizontalement cela signifie: assurer l'extension du système aux nouveaux aspects pour l'ensemble des applications traitées.

Verticalement cela signifie: restreindre le champ d'application du système en intégrant des outils plus spécifiques exploitant beaucoup plus la sémantique propre de ce nouveau champ d'application.

Le système SIGMA-ARCHI est un support général de la CAO pouvant être enrichi par de nouveaux outils d'intérêt général, et particularisé par l'adjonction d'outils spécifiques. Nous avons mené ces deux types d'opération.

Dans le premier cas F.Z'GRAGGEN a étudié comment étendre les possibilités du système en permettant de prendre en compte les systèmes industrialisés ouverts (Z'G.80) et R.VILLAMAYOR en introduisant un nouveau générateur d'allocation spatiale (Vil.80).

Le deuxième cas, traité par F.ZANELLI, concerne la construction d'un système plus particulièrement orienté vers le logement: le sous-système SIGMA-LOGEMENT (Zan.79). Il s'agit

d'un système spécialisé dans la conception de logements. Ce système procède en trois étapes (FIG.A-A1.4):

- l'élaboration du schéma directeur à l'aide d'un programme conversationnel capable de traiter la problématique du champ d'application logement (dormir, manger, se reposer< ...) et génération de plans d'appartements et de cellules d'habitat,
- la composition de tranches d'immeubles à partir des cellules en tenant compte du système constructif utilisé, du programme architectural et des règles d'arrangement,
- l'élaboration du plan masse à partir des tranches d'immeubles, du programme urbanistique et des contraintes du site.

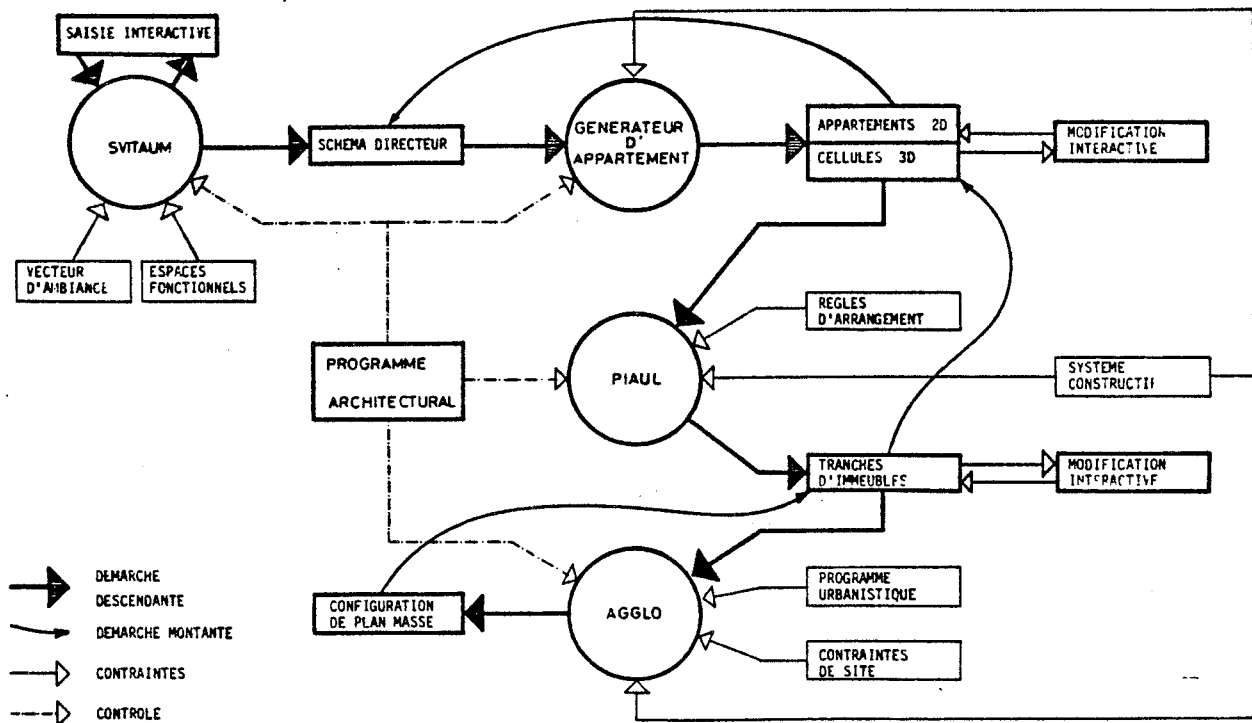


FIGURE A - A1.4 : ORGANISATION DE SIGMA-LOGEMENT (Zan.79)

Ce sous-système contient donc en plus des outils standards du système SIGMA-ARCHI, des outils spécialisés (SVITAUM, GENAP, PIAUL, AGGLO). Sa structure d'utilisation (FIG.A-A1.5) est un ensemble de trois environnements correspondant aux trois étapes

de l'élaboration du logement, plus l'environnement SIGMA-ARCHI sous-jacent aux autres environnements.

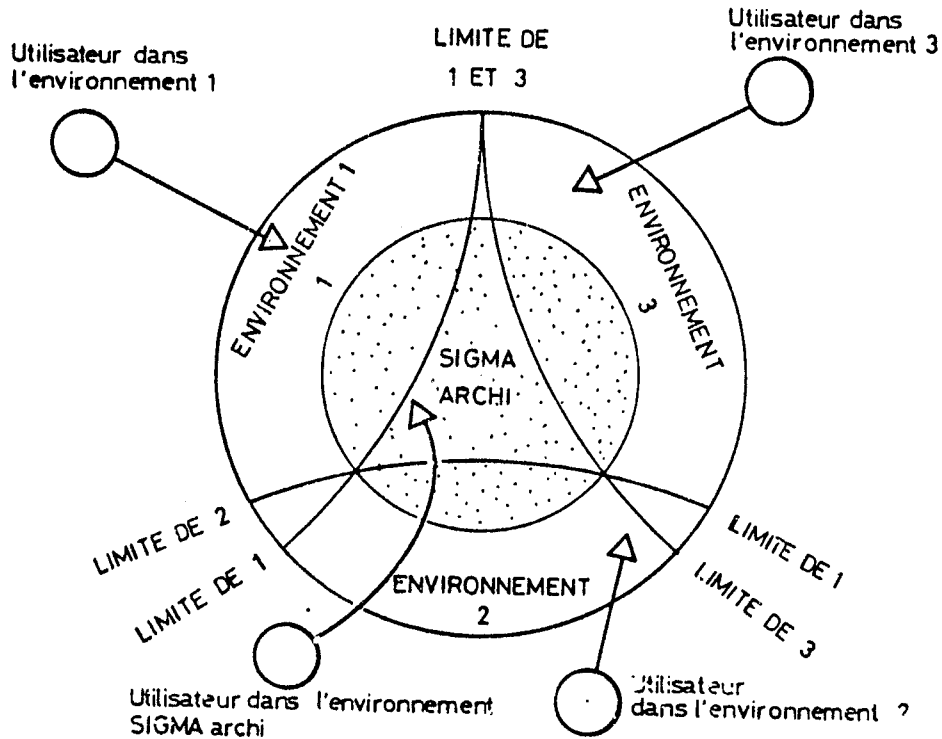


FIGURE A - A1.5 : STRUCTURE D'UTILISATION DE SIGMA-LOGEMENT (Zan.79)

A1.9 MISE EN OEUVRE DU SYSTEME SIGMA-ARCHI DANS LE CONTEXTE DE SIGMA-CAO

Le système SIGMA-ARCHI a été construit à l'aide de la méthodologie SIGMA-CAO. Pour cela une version de la MACAO a été réalisée. Il s'agit de MINI-SIGMA présentée dans le chapitre 12.1. La BDP est organisée en quatre parties principales correspondant aux:

- unités de conception et leurs propriétés
- relations entre les unités de conception
- description de formes brutes
- description de formes finies du projet

Les traitements qui constituent le système SIGMA-ARCHI sont spécifiés sur la figure A-A1.6. Leur traduction, directement

exécutable sous forme d'EXEC-CMS, a permis d'obtenir un moniteur de conception implicite.

```

C = V2D " VISU EN 2 DIM "
E : N,T,R * ; S : .
RES : STGR ;
VT : N,T,R ;
PE : PARMS(T,1),PARMS(P,* ) ;
M : SPEV3D(2) ;
A : ER(7),LOAD(ZT3),START ;
PS : ECLAT .

C = URBA " ALLOCATION SPATIALE : PROGRAMME URBALLOC "
E : N,R,P ; S : T .
RES : T2250 ;
VT : N,R,P ;
PE : FR(7),PARMS(R,1),PARMS(T,1) ;
A : LOAD(MAPA),START(IN,FILE,FTOIF001,D) ;
PS : FCLAT .

C = SPE " SPECIFICATIONS DU PROGRAMME ARCHITECTURAL "
E : N,R*,P* ; S : R*,P* .
RES : T2250 ;
VT : N,R,P ;
PE : PARMS(R,* ),PARMS(P,* ) ;
A : ER(7),LOAD(TRMATVVG),START ;
PS : ECLAT .

C = PAS3 " GENERATION SPATIALE : PROGRAMME PAS3 "
E : N,R,P ; S : T .
RES : T2250 ;
VT : N,R,P ;
PE : FR(7),PARMS(R,1),PARMS(P,1) ;
A : LOAD(PAS3),START ;
PS : ECLAT .

C = PAS1 " GENERATION SPATIALE : PROGRAMME PAS1 "
E : N,R,P ; S : T .
RES : T2250 ;
VT : N,R,P ;
PE : FR(7),PARMS(R,1),PARMS(P,1) ;
M : EFF ;
A : LOAD(PAS1),START ;
PS : FCLAT .

C = IGR " MANIPULATIONS GRAPHIQUES : INTERGRAPH "
E : ; S : .
RES : T2250 ;
VNT : IGR ;
VT : N,T ;
TR : IGR ;
PE : COMBINE ;
A : LOAD(INTERGRA),START ;
PS : FCLAT .

C = V3D " VISUALISATION EN 3 DIMENSIONS "
E : ; S : .
RES : T2250 ;
VNT : V3D ;
VT : N,T,P ;
TR : V3D ;
PE : COMBINE(SYSIN) ;
M : GLOBAL(PL1),LOADMDD(V3D),START .

C = V3DC " VISUALISATION EN 3 DIM. AVEC COULEURS "

```

FIGURE A - A1.6 : SPECIFICATION DES TRAITEMENTS DE SIGMA-ARCHI


```

E : ; S : .
RFS : T2250 ;
VNT : V30C ;
VT : N,T,P ;
TR : V30C ;
PE : COMBINE (DATA) ;
M : EXEC(VT$113D,FILE).
C = ARC " ARCADE "
E : ; S : .
RES : T2250 ;
VT : STRUC ;
M : LOADMOD(EX),START(IZENTER) .
C = ARCP " ARCADEP "
E : ; S : .
RFS : T2250 ;
VNT : ARCP ;
VT : N,T ;
TR : ARCP ;
PE : ER(7),COMBINE ;
A : LOAD(ARCADEP),START ; PS : ECLAT .
C = PRO " ANALYSE DE DONNEES : PG PROMET "
E : ; S : .
RFS : T2250 ;
VNT : PRO ;
VT : N,R ;
TR : PRO ;
PE : COMBINE,ER(7) ;
A : LOAD(PROMET),START ;
M : IMPR .
C = ANA " ANALYSE DE DONNEES : PG ANALO "
E : ; S : .
RFS : T2250 ;
VNT : ANA ;
VT : N,P ;
TR : ANA ;
PE : COMBINE,ER(7) ;
A : LOAD(ANALO),START ;
M : IMPR .
C = DEF " DEFINITION DU PROJET ARCHITECTURAL "
E : ; S : N .
RES : T2250 ;
M : NRUC .
C = MIT " ALLOCATION SPATIALE : PG MITCHELL "
E : ; S : T .
RFS : T2250 ;
VNT : MIT ;
VT : N ;
TR : MIT123 ;
PE : COMBINE,ER(7),ER(8) ;
A : LOAD(MIT),START ;
M : MIT4 $

```

FIGURE A A1.6 : SPECIFICATION DES TRAITEMENTS DE SIGMA-ARCHI (suite)

Pour l'activation des outils nous n'avons pas choisi une approche méthodologique (liée à une méthode de conception), mais nous avons préféré une approche procédurale (activation de l'outil lorsque tous les paramètres d'entrée sont disponibles). Cette approche plus générale autorise une plus grande liberté dans le choix de la méthode de travail.

A1.9 CONCLUSION

Faute de demandes opérationnelles des agences d'architectes, nous avons surtout cherché à explorer les possibilités d'utilisation d'une approche intégrée. Un effort d'analyse et de modélisation a été mené dans les trois aspects fondamentaux: processus de conception, données utilisées et traitements souhaités.

Deux types d'intégration (intégration par les traitements et intégration par les données) ont été étudiés, mis en oeuvre et comparés. La méthode de construction de systèmes CAO a été testée, ainsi que la hiérarchie de spécialisation de ces systèmes.

Ceci constitue un ensemble de résultats fondamentaux qui contrebalance le manque de résultats opérationnels.

ANNEXE 2

CAO DES EQUIPEMENTS ELECTRONIQUES

RESUME: présentation d'un cas concret de l'étude sur l'introduction de la CAO dans une entreprise (équipements électroniques). C'est un domaine de conception très complexe, disposant déjà de nombreux outils autonomes, où il n'est plus nécessaire de justifier la notion de système intégré. C'est donc la méthode d'introduction de la CAO en milieu industriel qui est intéressante ici.

A2. CAO DES EQUIPEMENTS ELECTRONIQUES

A2.1 CADRE ET BUT DE L'ETUDE

Rappelons d'abord le cadre dans lequel cette étude a été menée. Il s'agit d'une entreprise d'environ 4500 personnes dont la division "aéronautique et systèmes" comporte un grand nombre d'ingénieurs. Les produits réalisés correspondent en général à des séries moyennes (de quelques dizaines à quelques milliers). Ils comportent à la fois des circuits électroniques, logiques et analogiques et des ensembles mécaniques de précision. Il s'agit par exemple des équipements de navigation pour des avions ou des hélicoptères (instrumentation de bord, centrales anémométriques, capteurs, ...), des équipements téléphoniques (publiphone), ou des équipements de distribution de titres de transports. La part du prix des études dans le coût des produits est élevée. Le temps de réponse aux appels d'offre est crucial: c'est lui qui permet d'emporter les marchés. La fiabilité des produits doit être très grande. Ces quelques caractéristiques définissent bien les points-clés du processus. On voit qu'un effort doit être fait pour permettre

- a/ de faire très rapidement des pré-études en vue de satisfaire un cahier des charges,
- b/ d'augmenter la fiabilité des produits,
- c/ de réduire leur coût de développement.

Le plus souvent on trouve des programmes variés travaillant de façon autonome et réalisant diverses tâches ponctuelles du processus de conception. Des capacités de mémorisation sont aussi utilisées par exemple pour stocker les éléments entrant dans la réalisation des nouveaux objets. Mais la conception assistée par ordinateur prend toute sa signification à partir du moment où on s'intéresse au processus de conception de façon globale, c'est à dire en la considérant comme un tout en introduisant la notion de système. En plus des capacités de calcul et de mémorisation de l'ordinateur, on s'appuie surtout sur ses possibilités organisationnelles. Il semble finalement que ce soit cette démarche organisationnelle qui soit la plus importante en C.A.O.; et l'on constate souvent que la seule étude du processus de conception en vue d'en dégager une organisation permettant une approche C.A.O., fournit des résultats importants quant à la bonne organisation de l'entreprise.

A2.1.1 RAPPEL DE LA DEMARCHE METHODOLOGIQUE

Pour mener à bien le travail d'informatisation du processus Conception - Etude - Développement, il faut procéder de façon systématique. Pour cela nous avons bâti une méthode définissant complètement les phases par lesquelles il est indispensable de passer, pour avoir des chances raisonnables d'aboutir effectivement au terme de l'étude à un système satisfaisant, remplissant toutes les fonctions possibles et fonctionnant dans de bonnes conditions économiques.

Nous voulons discuter ici le déroulement d'une opération concrète, s'appuyant sur la méthode que nous avons présenté dans le chapitre 3.

A2.1.2 CALENDRIER DES OPERATIONS

L'idée de l'étude est déjà assez ancienne car nous pouvons la situer vers les années 1976, 1977. En faite la grande partie du travail a été effectuée entre l'été 1978 et le printemps 1980. C'est en effet dans cette période que les étapes un à six ont été entreprises.

Cette période correspond au déroulement du contrat de prédéveloppement dont cette étude a bénéficié. Depuis le printemps 1980 CROUZET a poursuivi cette étude en abordant de façon interne les autres étapes du processus d'introduction de la CFAO dans l'entreprise.

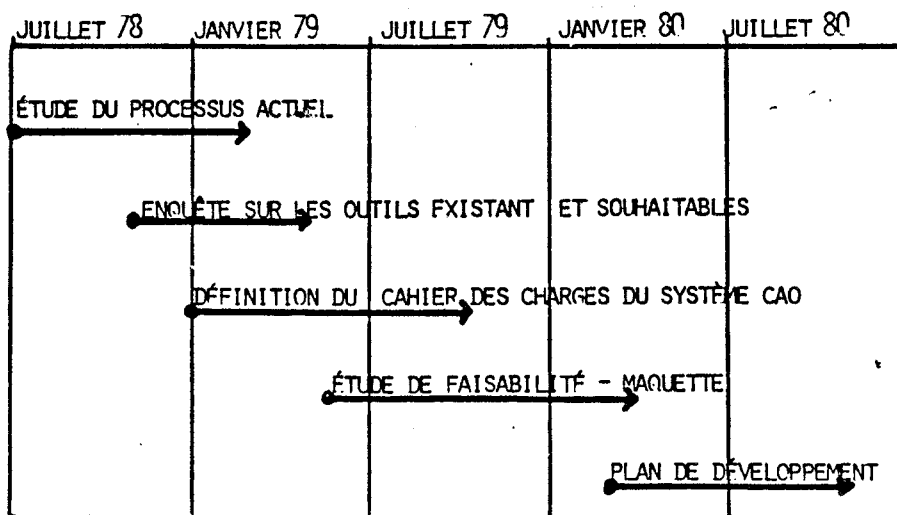


FIGURE A2.1: CALENDRIER DES OPERATIONS

A2.1.3 STRUCTURE DE L'EQUIPE CHARGEE DE L'ETUDE

Pour mener cette étude une équipe a été constituée. Elle a été organisée à deux niveaux conformément à la structure préconisée.

Le groupe opérationnel a été initialement composé de:

- un représentant de la direction,
- un représentant des "études avancées",
- un représentant de la production,
- un représentant du contrôle de qualité et de la fiabilité,
- un expert en CAO, conseiller extérieur à l'entreprise.

A ce groupe se sont progressivement joints un autre représentant des "études avancées" plus particulièrement chargé de l'aspect informatique de l'application et un technicien-informaticien, chargé de la réalisation de la maquette du système.

Le second groupe, ayant un rôle consultatif, a été composé de:

- des responsables des Départements correspondant aux différents secteurs d'activités concernés (Spatial, Aeronautique, Téléphonie, Transports),
- des responsables des Services internes à l'entreprise.

Des réunions ont eu lieu entre ces deux groupes chaque fois quand cela semblait souhaitable. Il s'agissait soit de compte-rendu de l'état d'avancement de l'étude soit au moment de passer à une autre étape.

A2.2 DEROULEMENT DE L'ETUDE

A2.2.1 ETUDE DU PROCESSUS ACTUEL DE CONCEPTION

La fabrication des équipements électroniques s'effectue jusqu'à présent toujours à la demande et pour le compte d'un client donné (même si actuellement on voit apparaître la notion de produit "sur étagère", c'est à dire du produit conçu à priori). C'est pourquoi le processus de conception contient deux étapes très différentes appelées respectivement, l'avant-projet et le projet.

Lors de l'avant-projet, il s'agit de répondre à un appel d'offre d'un client en fournissant un document réponse. Ce document doit contenir toutes les informations importantes concernant aussi bien l'aspect fonctionnel (réalisation de fonctions demandées, fonction en plus, etc...), que l'aspect économique (coût, délais, etc...). Ce document, qui a évidemment

une importance cruciale pour l'entreprise car il conditionne l'obtention de la commande et détermine par l'estimation financière sa rentabilité, s'élabore souvent dans des conditions très difficiles, notamment en ce qui concerne les délais de réponse (une semaine, souvent même deux jours). Le problème de disponibilité du personnel qualifié souvent surchargé, est alors fondamental.

Après l'analyse de l'appel d'offre intervient le processus de l'avant-projet. On rédige alors le cahier des charges interne à l'entreprise dans lequel on définit, aussi bien l'aspect technique que l'aspect commercial de la proposition, avec des informations internes telles que l'analyse de la concurrence, la politique industrielle, le prix de revient, le délai, le niveau de risque, etc... A partir de ce document, la direction décide ou non de s'engager dans la bataille. La figure A2.2 schématise cette partie de l'avant projet.

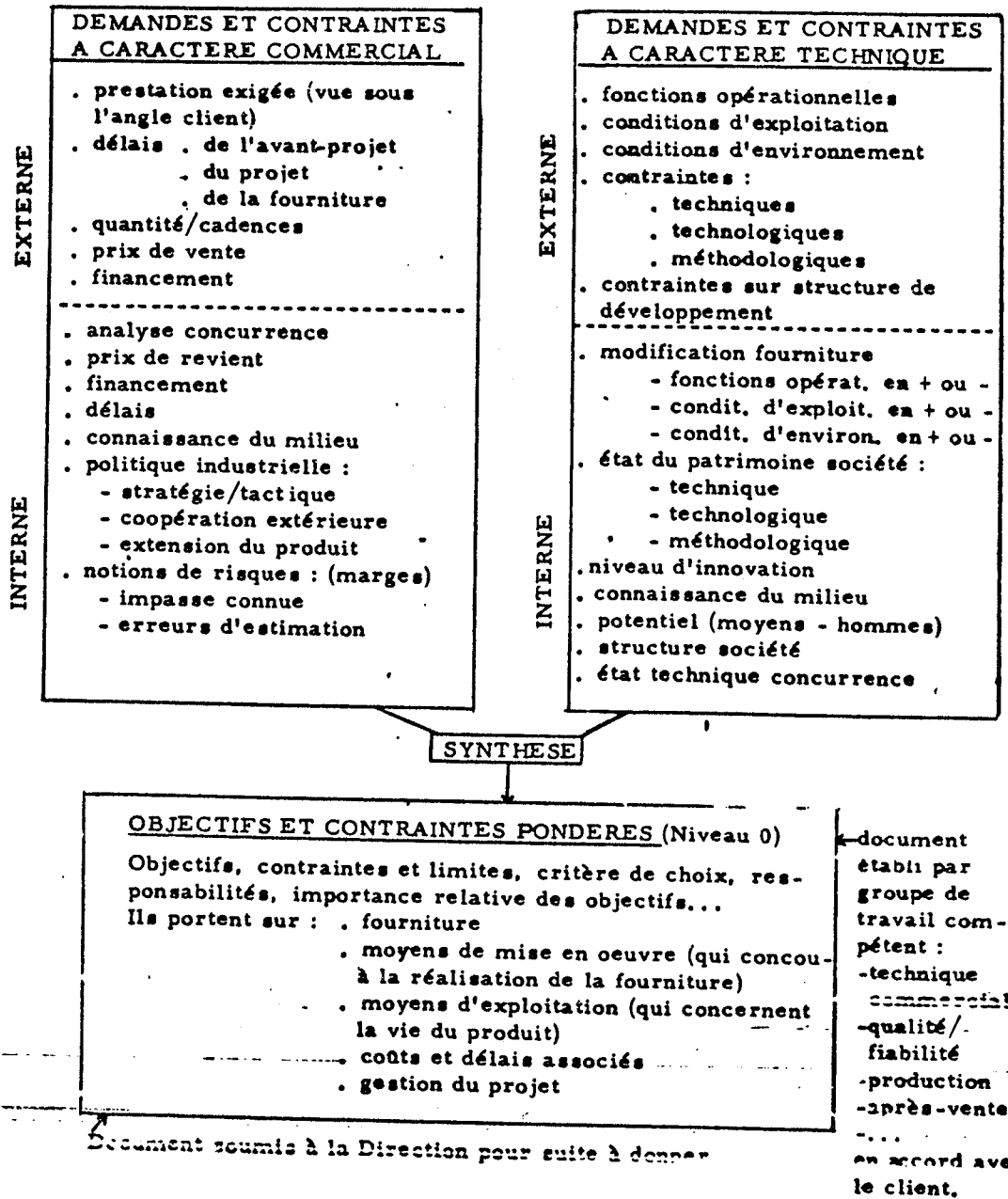


FIGURE A2.2: DESCRIPTION DETAILLEE DE LA PHASE "DEFINITION DE LA PRESTATION" DE L'AVANT PROJET (CZI.79)

Les trois phases suivantes sont :

- 1/ ETUDE GENERALE ayant pour rôle d'élaborer un schéma de réalisation possible compte tenu des ressources, des projets déjà engagés et de la faisabilité technique,
- 2/ ETUDES TECHNIQUES DETAILLEES dont le but est d'assurer de façon plus précise, la faisabilité des points clés ou les innovations fondamentales de la proposition,
- 3/ SYNTHESE des études techniques devant assurer l'homogénéité de la proposition, fournir le plan de développement et élaborer les deux volets du dossier de proposition: proposition client et dossier interne.

Ces deux documents sont ensuite discutés et s'ils sont acceptés la proposition au client peut être faite. La figure A2.3 schématise le déroulement de l'avant projet.

Le temps qui sépare l'avant-projet du projet est très variable et peut facilement atteindre un an. Le projet s'effectue à la commande du client, le marché est donc conclu, même si quelquefois on peut anticiper sur la commande.

Il est très possible que le cahier des charges définitif fourni par le client, ne corresponde pas beaucoup au cahier des charges initial. Ceci est aussi vrai pour le dossier de proposition, car l'évolution économique et technologique peut être très rapide.

Le projet doit donc commencer de façon similaire à l'avant-projet pour reactualiser la proposition au client et le dossier interne. Pendant le projet il faut avoir à l'esprit que les trois grandes phases de la vie d'un produit sont: sa conception, sa réalisation et son exploitation. Il faut donc prendre en compte les contraintes spécifiques à chacune de ces phases.

La démarche de conception est une démarche fonctionnelle, depuis la fonction vers la réalisation. Après avoir déterminé l'organisation du projet, c'est à dire son planning, son allocation de ressources, on entreprend la conception du produit.

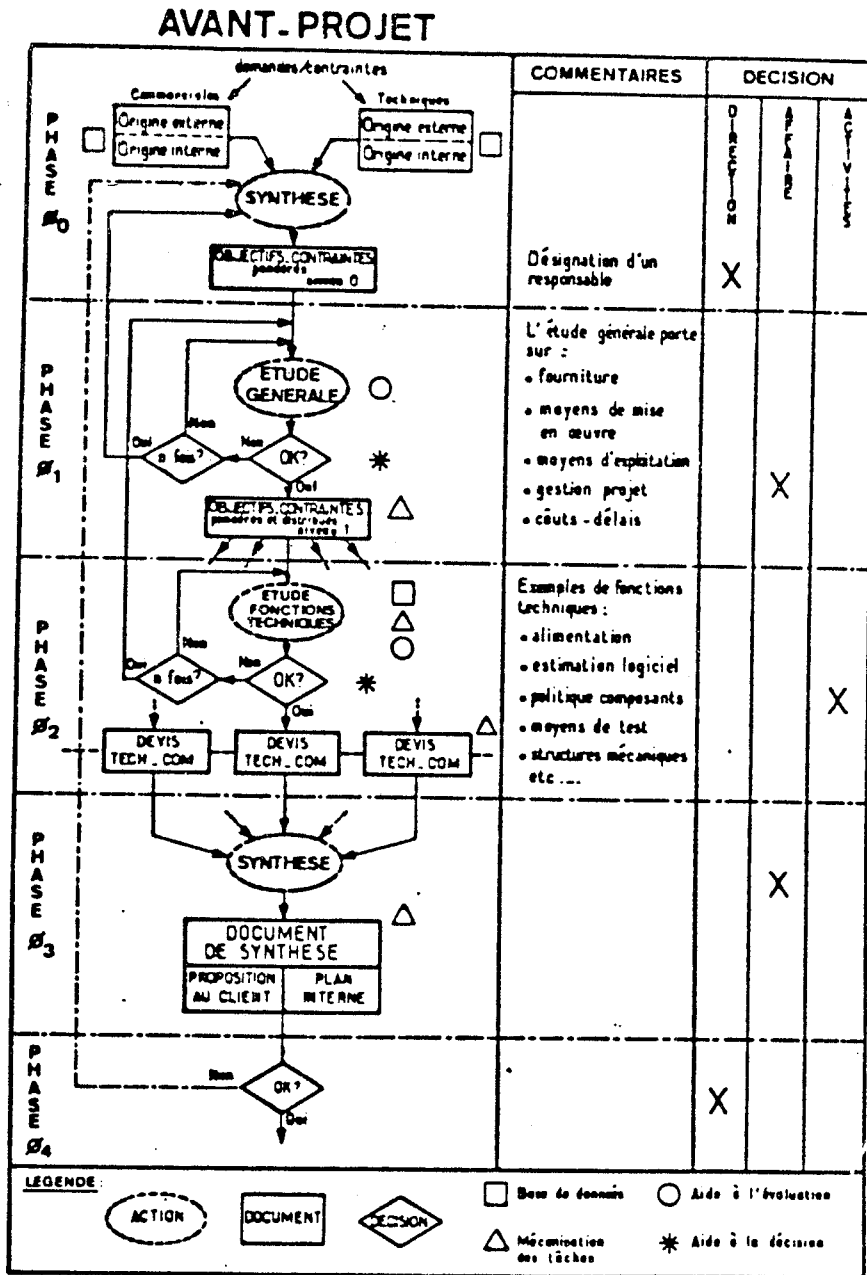


FIGURE A2.3: DEROULEMENT DE L'AVANT-PROJET (CZT.79)

Les différentes étapes en sont:

- 1/ ANALYSE GENERALE DU SYSTEME: le système à concevoir est une boîte noire comportant des entrées et des sorties spécifiées par le cahier des charges. Le premier travail consiste à décomposer cette boîte en FONCTIONS reliées par des LIENS. Ces fonctions correspondent à des expressions mathématiques paramétrées. Ces fonctions et ces paramètres sont extraits d'une bibliothèque. Il est probable que la décomposition de la boîte noire en fonctions conduise à plusieurs solutions: dans ce cas le concepteur évalue les différentes solutions en affectant des valeurs aux paramètres.
- 2/ ANALYSE DETAILLEE: parvenu à cette phase, le concepteur connaît les fonctions qu'il veut maintenant réaliser. En électronique, la réalisation peut s'effectuer soit à l'aide du matériel, soit à l'aide du logiciel (le choix dépend des critères contenus dans le cahier des charges). Le concepteur doit effectuer de nouvelles évaluations en tenant compte des expériences déjà réalisées par l'entreprise. Il doit essentiellement respecter les contraintes suivantes : temps de réponse, volume mémoire occupé, coûts (étude - achat - production - consommation), fiabilité, maintenabilité, etc... Ces évaluations s'effectuent soit sur une maquette virtuelle représentée en ordinateur et calculée, soit sur un prototype effectivement réalisé. Il est clair que la première solution est beaucoup moins coûteuse et permet d'entreprendre l'étude plus rapidement et de comparer des solutions variées.
- 3/ CONCEPTION DETAILLEE: elle est menée de front dans les trois volets de l'étude, (aspects mécanique, matériel et logiciel). En mécanique il s'agit de concevoir l'enveloppe pouvant contenir l'électronique et satisfaisant aux contraintes du cahier des charges (poids, encombrement, résistance aux conditions d'utilisation telles que températures, vibrations, etc...). Le matériel concerne la conception d'une carte électronique. Pour cela le concepteur doit choisir une famille de composants, puis un composant dans la famille, en tenant compte des contraintes telles que le coût, le délai d'approvisionnement, les indications du cahier des charges sur les composants préférentiels, etc... Sur le schéma d'implantation logique il doit pouvoir effectuer des évaluations comme le bilan de consommation, le coût, l'encombrement physique, la fiabilité, la testabilité etc...
- 4/ CONCEPTION DE PRODUIT: la première opération est une mise au propre au bureau de dessin du schéma qui devient alors le schéma de référence. Vient ensuite l'implantation physique du schéma sur les cartes, avec l'implantation des éléments

dans des boîtiers physiques normalisés, l'implantation des boîtiers sur la carte et le tracé de connexions.

Puis intervient la réalisation du circuit: le typon de photogravure, la liste de perçage, la liste d'insertion automatique des composants.

L'assemblage doit tenir compte de l'implantation des cartes dans le boîtier, l'agencement des composants des alimentations, la vérification du montage/démontage, etc...

- 5/ INDUSTRIALISATION: c'est le bureau de méthodes qui assure l'industrialisation et l'organisation de la fabrication. Il doit assurer que le produit est effectivement réalisable tout en garantissant la production d'outils spécifiques pour la mise au point, et l'ensemble de documentations: le dossier de fabrication permettant la réalisation du produit, le dossier d'exploitation destiné au service après-vente pour la maintenance ainsi que le dossier permettant l'utilisation du produit. Le service Qualité/Fiabilité contrôle la fabrication et établit le dossier de contrôle assurant le bon fonctionnement du produit.

A2.2.2 LES POSSIBILITES OFFERTES PAR LA CAO

Les outils informatiques dans le contexte qui nous intéresse sont déjà assez courants. Une enquête effectuée auprès des différents services de l'entreprise a fait ressortir, d'une part des outils existants et d'autre part des besoins à satisfaire. Pour cela nous avons distribué dans l'ensemble des services de CROUZET des questionnaires dont le modèle est reproduit sur la figure A2.4.

Le dépouillement de cette enquête a été effectuée à l'aide d'un autre imprimé reproduit sur la figure A2.5. Le caractère confidentiel de cette enquête nous empêche de publier ici des résultats explicites. Les principaux renseignements sont les suivants:

- un bon accueil général de l'enquête,
- des réponses très précises en ce qui concerne des outils existants,
- des réponses beaucoup plus limitées, aussi bien en nombre qu'en variété, en ce qui concerne des besoins exprimés à satisfaire.

Nous avons pu constater que les réponses précises ont été formulées pour décrire, la plupart du temps, des programmes que l'on peut appeler technologiques, c'est à dire directement utilisables dans une étape précise du processus et produisant des résultats directement exploitables.

FICHE

BESOINS EXPRIMES

N°

ASPECT OPERATIONNEL UNIQUEMENT

-DESCRIPTION SOMMAIRE DES OBJECTIFS DU PROGRAMME

-METHODE MATHEMATIQUE EMPLOYEE (BASES THEORIQUES), REFERENCES SCIENTIFIQUES

-DESCRIPTION DES ENTREES ET DES SORTIES DU POINT DE VUE DE L'USAGE

-DOMAINE D'APPLICATION

DESCRIPTION DETAILLEE DU FONCTIONNEMENT

FIGURE A2.4a: MODELE DU QUESTIONNAIRE DE L'ENQUETE - BESOINS EXPRIMES

FICHE

MOYENS EXISTANTS

N°

NOM DU PROGRAMME:

--

ASPECT OPERATIONNEL

- DESCRIPTION SOMMAIRE DES OBJECTIFS DU PROGRAMME

 - DESCRIPTION DES ENTREES ET DES SORTIES DU POINT DE VUE DE L'USAGER

 - DOMAINE D'APPLICATION

 - DOCUMENTATION SUR LE PRODUIT
-

ASPECT INFORMATIQUE

- LANGAGE DE PROGRAMMATION
 - NOMBRE D'INSTRUCTIONS DU PROGRAMME SOURCE
 - NOMBRE NECESSAIRE DE FICHIERS DE DONNEES (TYPE DE SUPPORT ET TAILLE)
 - TAILLE DE MEMOIRE CENTRALE MINIMUM (16 OU 32 BITS)
 - UTILISATION CONVERSATIONNELLE OU DIFFEREE ?
 - TYPES DE MACHINES SUR LESQUELLES LE PROGRAMME FONCTIONNE
 - PERIPHERIQUES INDISPENSABLES
-

ASPECT ECONOMIQUE

- AUTEURS (ADRESSE)
- DISPONIBILITE
- CONDITIONS D'EXPLOITATION ET DE CESSION DU PROGRAMME

FIGURE A2.4b: MODELE DU QUESTIONNAIRE DE L'ENQUETE - MOYENS EXISTANTS

FICHE DEPOUILLEMENT DE RESULTATS

N°

NOM DU PORGRAMME

N° FICHE "BESOINS EXPRIMES"

N° FICHE "MOYENS EXISTANTS"

CLASSE:

BASE

GENERAL

SPECIFIQUE

REMARQUES:

PRIORITE SI EXISTE:

COUT ENVISAGEABLE:

PROIRITE SI A FAIRE:

COUT ENVISAGEABLE:

MODALITES D'ACQUISITION

FIGURE A2.5: SUPPORT DE DEPOUILLEMENT DE L'ENQUETE

Parmi ces outils nous pouvons citer:

- + Recherche de solutions
 - description de stimuli
 - description de l'objet à tester
 - simulation et évaluation
- + Identification
 - acquisition de données (mesure et enregistrement d'information sur le site en vue d'exploitation)
 - traitement de données (exploitation des informations enregistrées)
- + Analyse fonctionnelle
 - graphes fonctionnels (étude des relations fonctionnelles entre "éléments" en vue d'identification et caractérisation de tâches)
 - graphes dynamiques (prise en compte de problème de temps, synchronisation)
- + Implantation
 - partitionnement (implantation grandes masses)
 - calculs de structure (étude de vibrations)
 - bilan thermique (dissipation thermique)
 - implantation circuit imprimé
 - implantation mécanique
 - câblage
- + Industrialisation
 - codification des composants
 - élaboration des gammes d'usinage, de montage, de câblage, de mise au point
 - élaboration de programme de test pour des testeurs généraux ou spécialisés
 - gestion d'un fichier technique pour les sous-traitants
 - gestion des standards (composants électroniques, pièces mécaniques)
- + Analyse opérationnelle
 - fiabilité
 - testabilité
 - miantenabilité
 - disponibilité
- + Outils de fabrication et de contrôle
 - pour des machines à commande numérique du centre d'usinage
 - aide à la fabrication de circuits imprimés
 - aide à l'insertion de composants
 - aides au câblage
 - aides au réglage

- test des composants
- test des sous-ensembles

On peut constater qu'il s'agit soit d'outils permettant de réaliser de façon automatique des tâches non créatives, soit des outils interactifs associant l'homme et la machine dans la résolution d'une tâche créative.

Des souhaits ont été formulés d'exploiter les possibilités de mémorisation de l'ordinateur pour assurer la gestion automatique de divers bibliothèques ou catalogues.

- bibliothèque de fonctions
- bibliothèque de schémas
- bibliothèque de composants
- bibliothèque de pièces mécaniques

Si l'enquête a fait apparaître en éventail d'outils très complet, nous avons pu constater que la plupart du temps aucun lien n'a été prévu pour passer d'un outil à un autre. Les limitations de cette vision des choses sont apparues très rapidement surtout dans un contexte de conception complexe, nécessitant l'intervention de plusieurs personnes possédant des compétences variées. Le temps passé à transcrire les données pour pouvoir utiliser un autre programme est considérable.

C'est la raison d'une autre approche, fondée sur une vision globale du processus de conception.

Malheureusement, cette vision de la CAO n'est pas encore très établie dans l'industrie. Nous avons pu nous rendre compte de cela quand nous avons voulu compléter l'enquête interne par une enquête plus générale sur les outils CAO disponibles sur le marché français ou international.

Nous avons, en effet, rencontré beaucoup de produits logiciels ou produits matériels et logiciels constituant un tout monolithique et effectuant un travail précis et bien délimité sans aucune possibilité d'intégration dans un ensemble plus important et plus complet.

A2.2.3 PROCESSUS CONCEPTION-ETUDE-DEVELOPPEMENT ADAPTE A LA CFAO

Pour mettre en place l'ensemble des possibilités qu'offre la CAO actuellement il est nécessaire d'adapter le processus C.E.D. (Conception - Etude - Développement) à cette nouvelle façon de voir le projet.

Le nouveau processus doit être bâti à partir du

- processus courant,
- l'analyse des outils CAO existants aussi bien à l'intérieur qu'à l'extérieur de l'entreprise,
- les demandes des utilisateurs concernant de nouveaux outils,
- les possibilités offertes par l'approche intégrée de la CAO.

La conception déborde très largement du cadre traditionnel du bureau d'étude. Il faut considérer quatre étapes fondamentales dans la vie d'un produit:

Dans la PREMIERE ETAPE le produit prend vie sous forme conceptuelle. Conçu sous forme de boîte noire, de relations, de fonctions mathématiques ou logiques, il répond déjà aux spécifications finales, ou tout au moins, les intègre. Il pourra être testé sous forme de maquette dont la définition sera généralement très macroscopique et la vérification se fera à partir des principes retenus.

La SECONDE ETAPE conduit à la transformation de l'objet conceptuel en un objet virtuel, c'est à dire à un objet entièrement défini, généralement sur le papier. La conception consiste à sauvegarder les principes et fonctions retenus en tant que solutions liées aux aspects physiques de la future réalisation: choix de composants, organigramme logiciel, fiabilité, masse...

Les maquettes auront le degré de finesse de la représentation physique finale permettant d'effectuer des tests et des évaluations plus précises, puisque l'objectif est de supprimer des maquettes physiques.

La TROISIEME ETAPE réalise l'objet réel. La réalisation s'effectue par opérations manuelles, semi-automatiques ou automatiques.

On peut citer les machines d'usinage à commande numérique, les testeurs automatiques, les machines à insérer les composants. Ces "machines" sont programmables et les programmes d'exécution doivent être générés à partir du dossier du produit automatiquement pour certaines, en mode assisté pour d'autres.

La QUATRIEME ETAPE concerne la vie du produit chez l'utilisateur. Les aspects maintenance, qualité, fiabilité, ont une importance aujourd'hui décisive et doivent être pris en considération dès la phase conceptuelle.

Le processus de conception est modélisé par un algorithme réentrant. En effet, les étapes ainsi définies ne sont pas séquentielles. Il est fréquemment nécessaire, en phase de conception de l'objet conceptuel, de parcourir une branche de conception de l'objet virtuel et même parfois de l'objet réel.

Cette nécessité répond à la réalité d'une conception qui n'est jamais totalement descendante.

Pour dégager ce nouveau processus de conception il faut étudier quatre aspects fondamentaux qui y interviennent:

- LE PROCESSUS: le premier aspect important de l'application, qui nécessite un intérêt tout particulier, est l'étude de la structure du processus CED selon lequel l'application doit se dérouler.

Pour le dégager il faut s'inspirer du processus actuel et lui apporter les modifications que l'introduction de la CAO et d'autres changements de structure provoquent. On définit donc les différents états (ou environnements) du processus ainsi que le cheminement qui les relie.

De façon accessoire il est possible de noter tous les autres aspects de l'application, mais le processus CED doit rester le fil conducteur de l'étude.

- LES DONNEES: le deuxième aspect important de l'application est celui des données intervenant dans l'application. Les données sont ici prises au sens large, il s'agit aussi bien des informations que des produits.

Ce qui nous intéresse c'est d'inventorier toutes les données de l'application, afin de les classer pour dégager les atomes d'informations et les types de structures qui les lient. Il s'agit de définir le comportement structural et sémantique de ces données ainsi que leur évolution pendant le déroulement du processus CED.

Cette démarche par les données fournit, non seulement un inventaire des données mais aussi une certaine vision du processus CED. Ces résultats doivent donc être confrontés avec ceux obtenus par la démarche basée sur le processus.

- LES TÂCHES: le troisième aspect de l'application est celui des tâches que l'on entreprend dans l'application. Il s'agit de formuler la liste des traitements que l'on effectue dans l'application. Quand on dresse cette liste, il est important de spécifier pour chaque tâche l'ensemble des informations sur lesquelles elle travaille. De cette façon il est possible de retrouver d'une manière indirecte, les données sur lesquelles l'application s'appuie, ainsi que le processus CED selon lequel ces tâches s'enchaînent. On reçoit ainsi une autre vision de l'application.

- LES UTILISATEURS: le dernier aspect dont il faut tenir compte pour les applications dont le processus CED fait intervenir des personnes variées, de compétences différenciées, est celui des utilisateurs. Il est important de suivre aussi ce fil conducteur. En effet, souvent dans ce cas, c'est le seul qui permette de reconstruire à partir des visions fragmentaires de chaque participant, une vision globale de l'application.

Chaque participant doit donc fournir sa vision du processus CED, c'est à dire son champ d'intervention, les données qu'il manipule et les tâches qu'il doit entreprendre. A partir de ces informations de tous les participants, on obtient une spécification globale de l'application.

On peut constater que c'est la concordance de ces quatre aspects de l'application qui assure la cohérence des spécifications et fournit un nouveau processus CED adapté à la CFAO.

La synthèse de ces différents aspects peut être effectuée à l'aide des schémas suivants:

- spécification d'une tâche (FIG.A2.6)
- classification de tâches (FIG. II-7.2)
- définition du processus CED (FIG.A2.7)
- définition des liens entre le processus de conception, les opérations à effectuer, les outils informatiques, etc... (FIG. A2.8-14)

Finalement nous avons pu constater que le système CAO peut être restreint à la phase "études - développements" car d'une part, celle-ci est relativement indépendante de la phase "fabrication", et d'autre part, le coût des études est assez élevé pour le justifier. En effet, les séries étant assez courtes, les ateliers de mécanique se restructurent souvent pour de nouvelles productions (ou pour de simples prototypes) et l'incorporation dans le nouveau produit d'éléments existants, lorsqu'elle a lieu s'effectue, simplement au niveau de la conception sans recherche dans une base de données (en fabrication). L'électronique quant à elle, nécessite presque toujours le lancement de nouveaux sous-ensembles, mais sans changement d'outillage (donc à l'intérieur des technologies déjà maîtrisées par l'entreprise).

Réciproquement, les capacités de fabrication assez polyvalentes ne sont pas étudiées suffisamment en détail pour en déduire le type de produit qui permettrait de mieux les utiliser, et d'agir ainsi rétroactivement sur la conception.

Ce dernier point qui pourrait être un préalable dans d'autres domaines, sera seulement une éventuelle retombée future de l'usage du système CAO: sur l'organigramme ci-joint du processus, les flèches entre la partie pré-étude - étude - développement et la partie industrialisation - fabrication - exploitation sont à sens unique et descendantes.

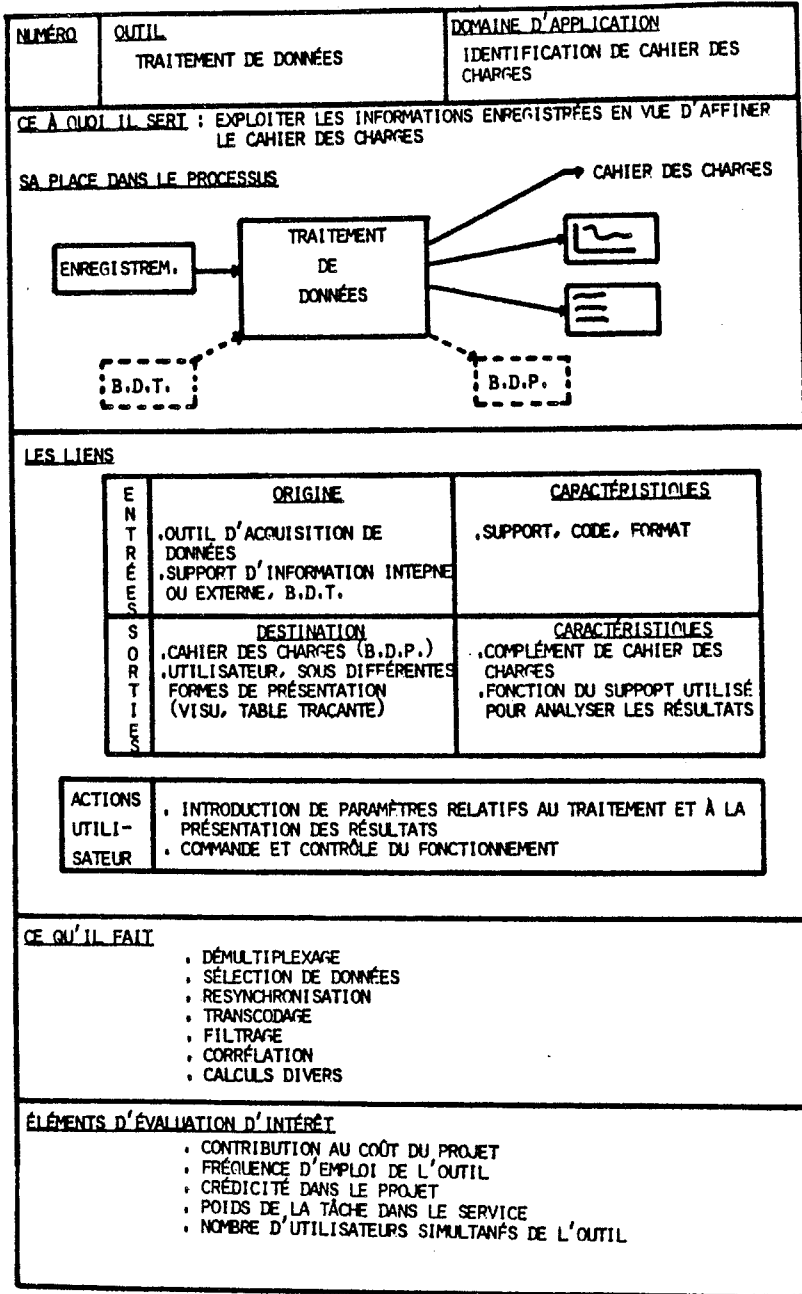


FIGURE A2.6: SPECIFICATION D'UNE TACHE

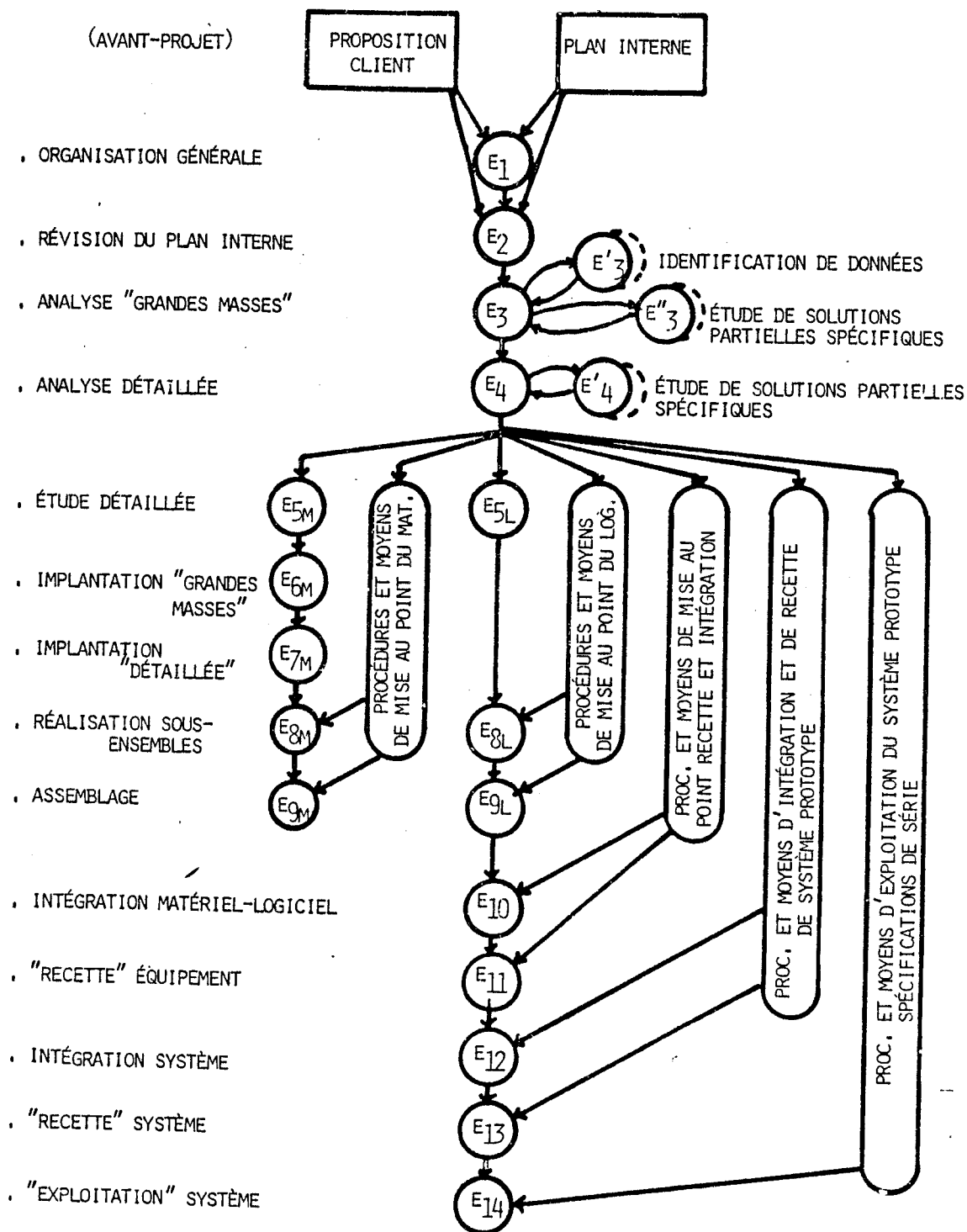


FIGURE A2.7: DEFINITION DU PROCESSUS CED

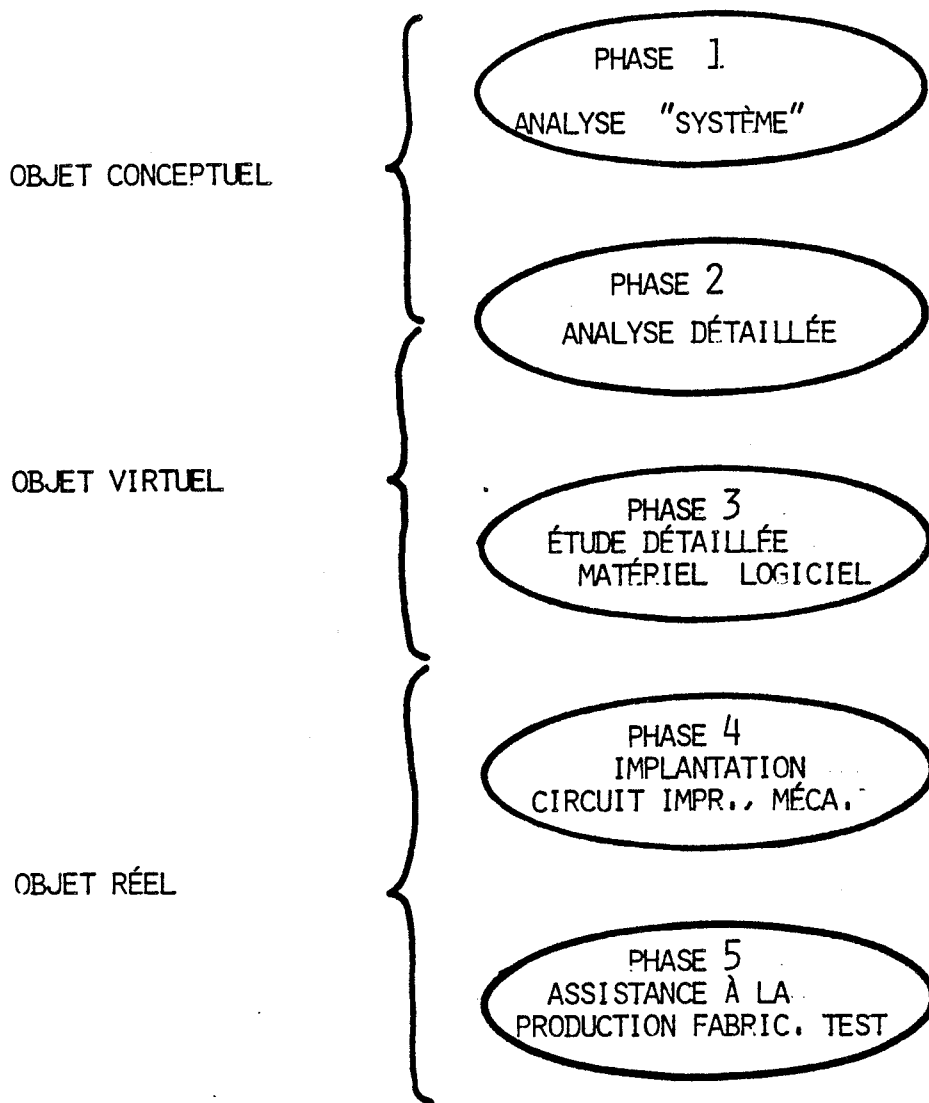


FIGURE A2.8: PROCESSUS DE CONCEPTION "EQUIPEMENT ELECTRONIQUE"

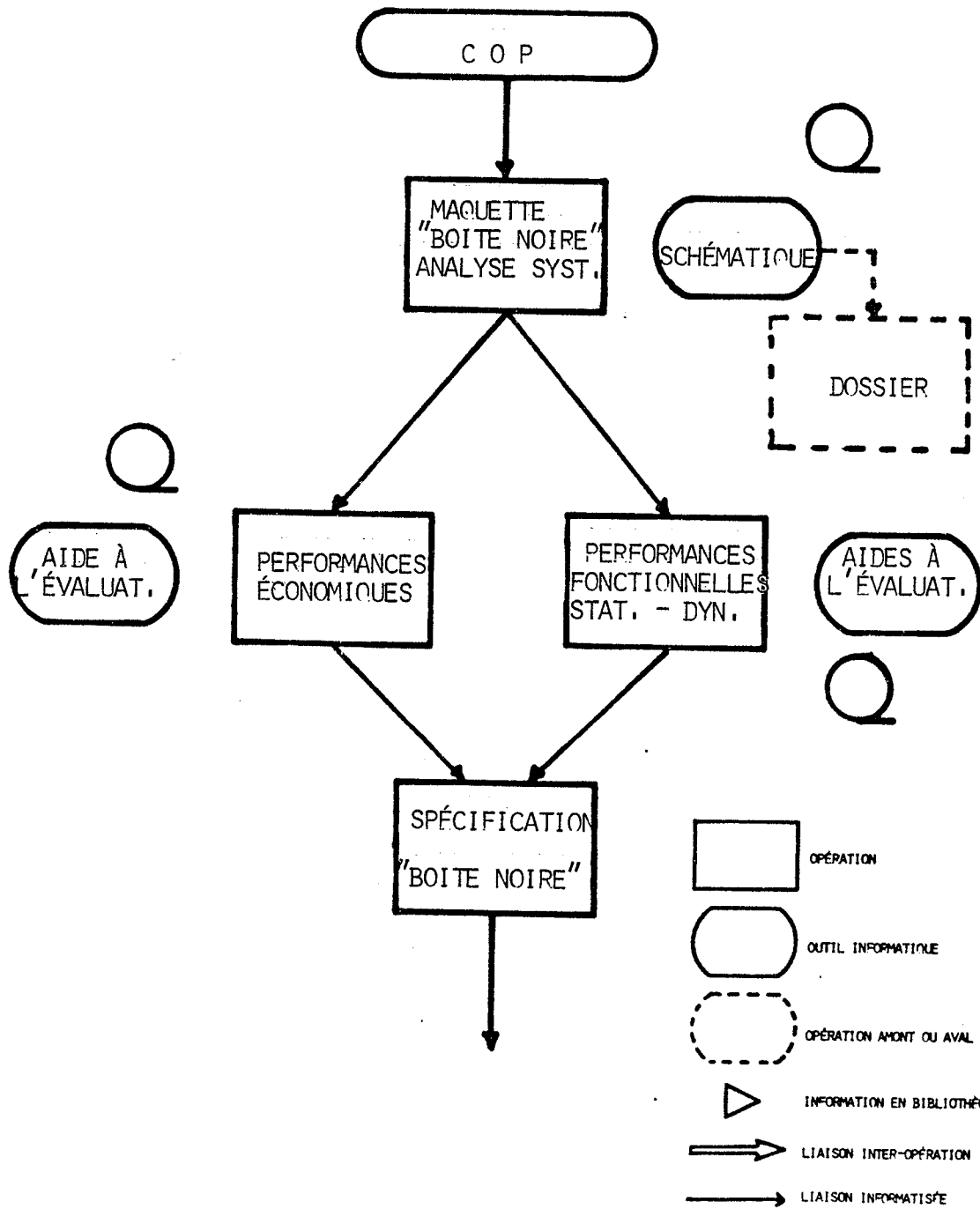


FIGURE A2.9: PHASE 1 - ANALYSE SYSTEME

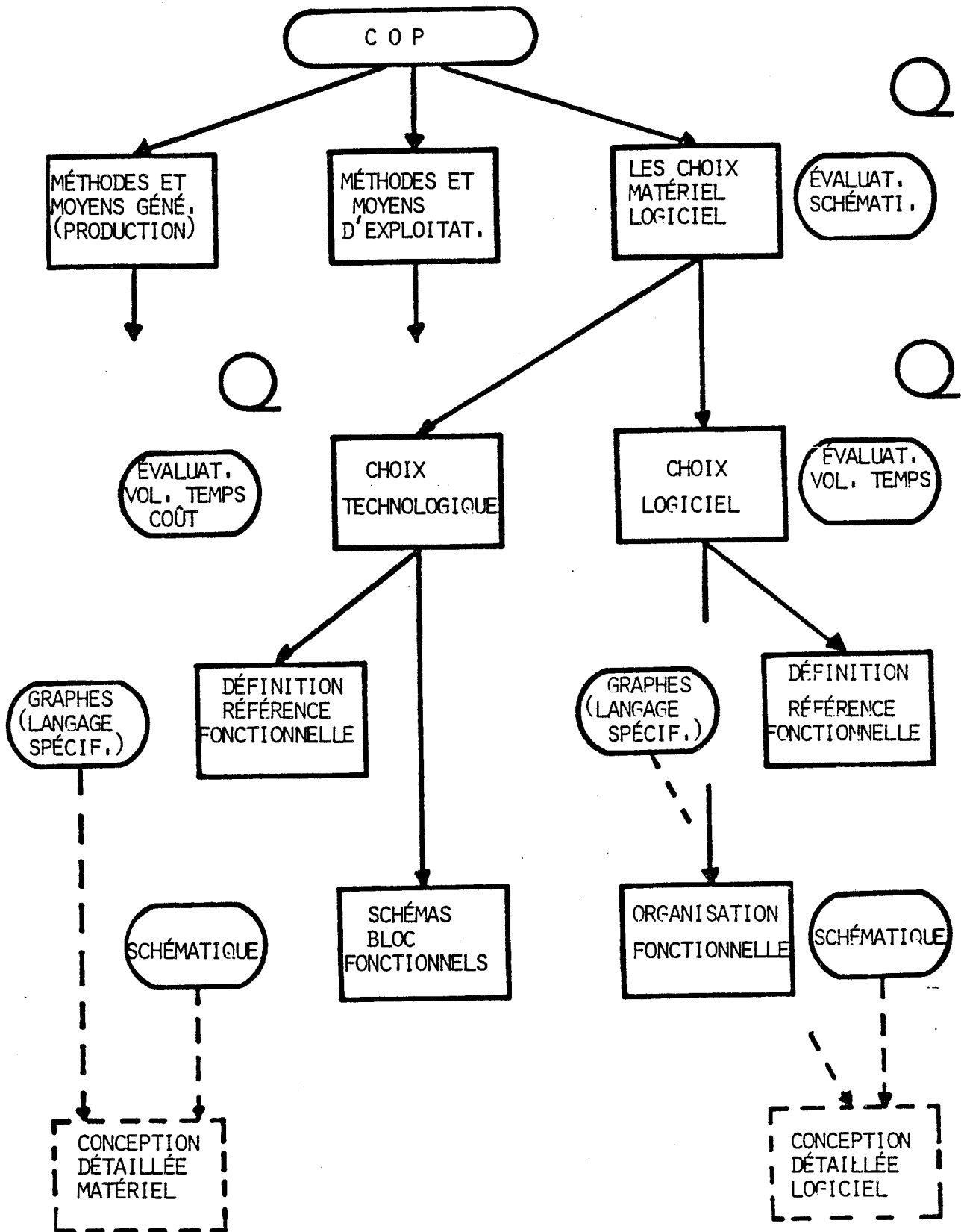


FIGURE A2.10: PHASE 2 - ANALYSE DÉTAILLÉE

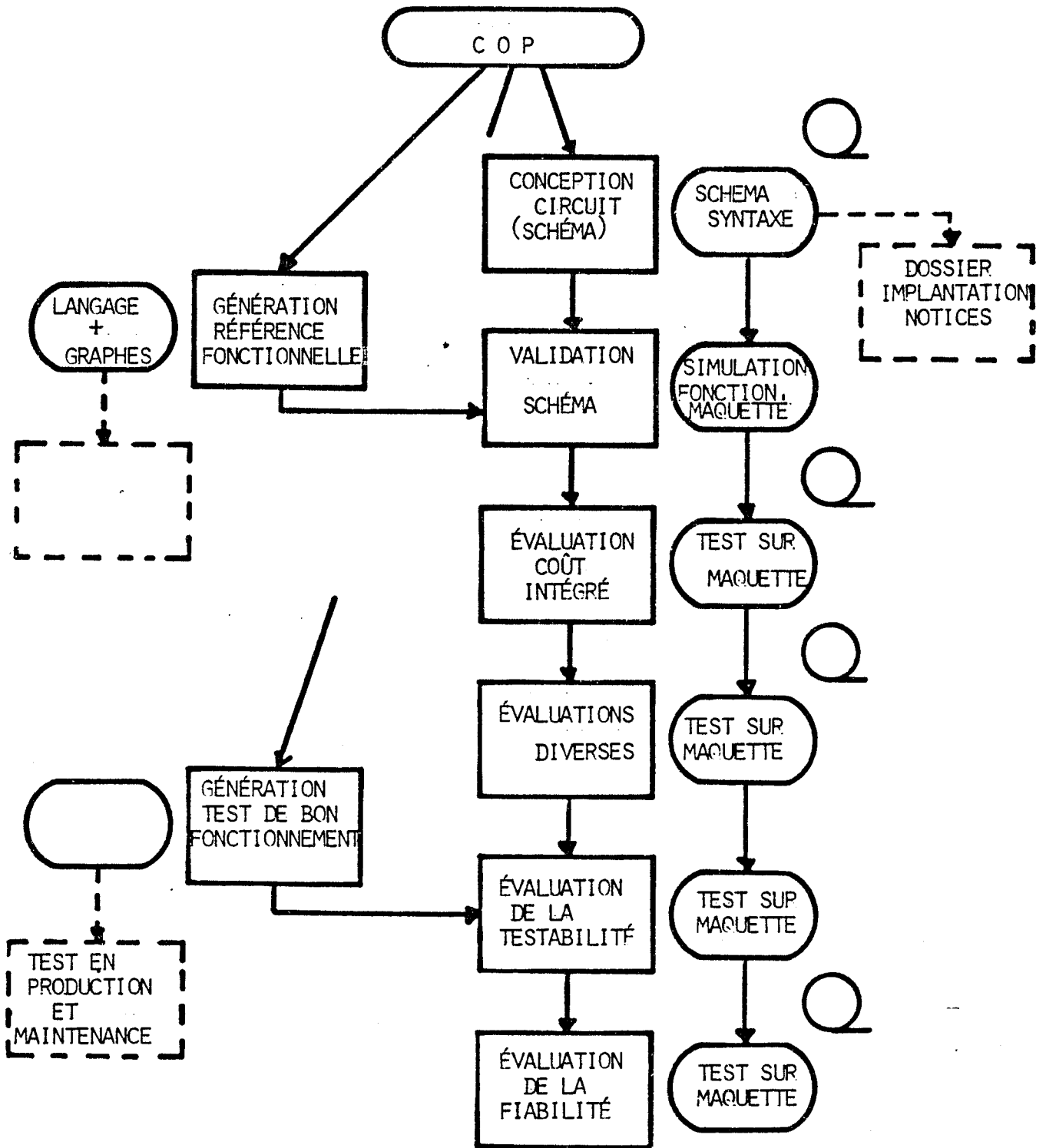


FIGURE A2.11: PHASE 3 - ETUDE DETAILLEE (MATERIEL)

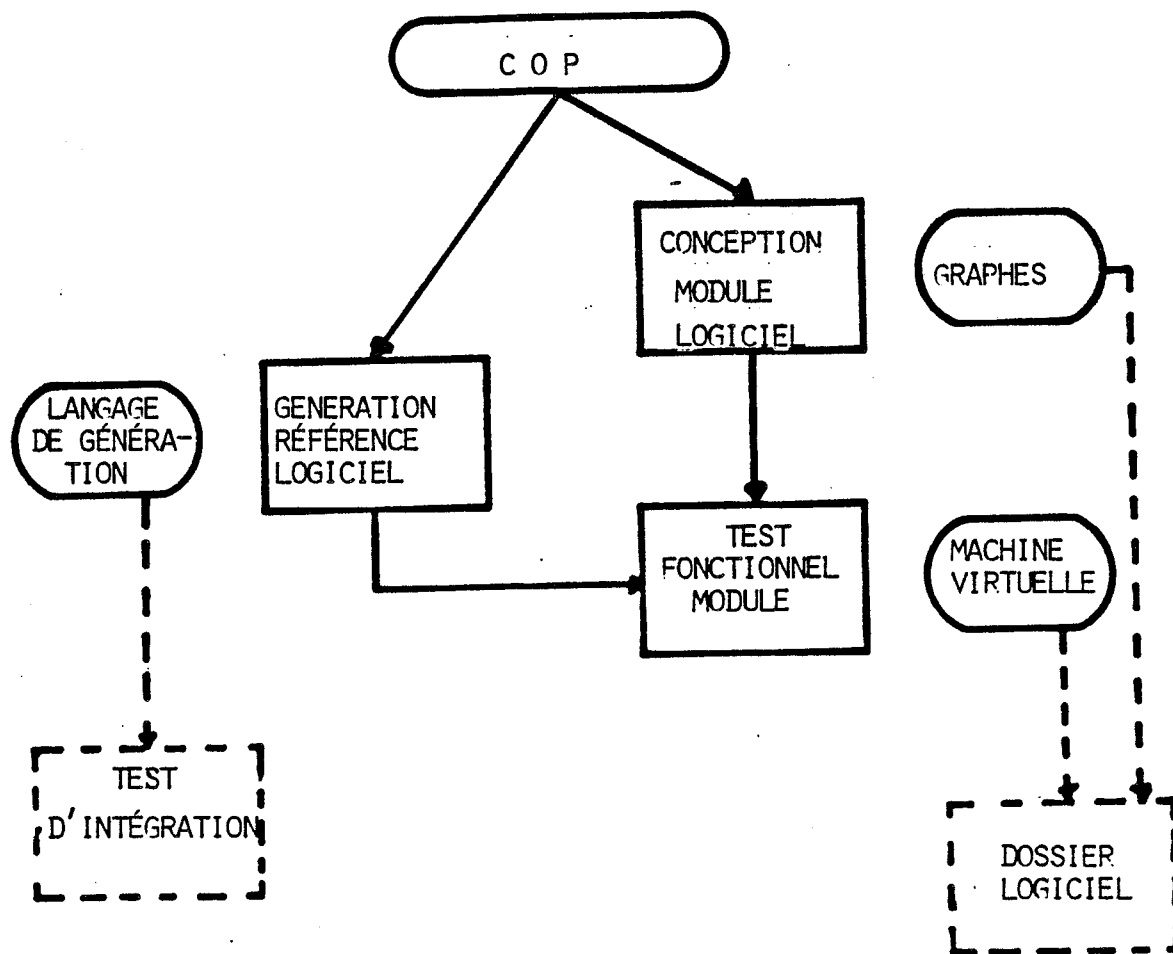


FIGURE A2.12: PHASE 3 - ETUDE DETAILLEE (LOGICIEL)

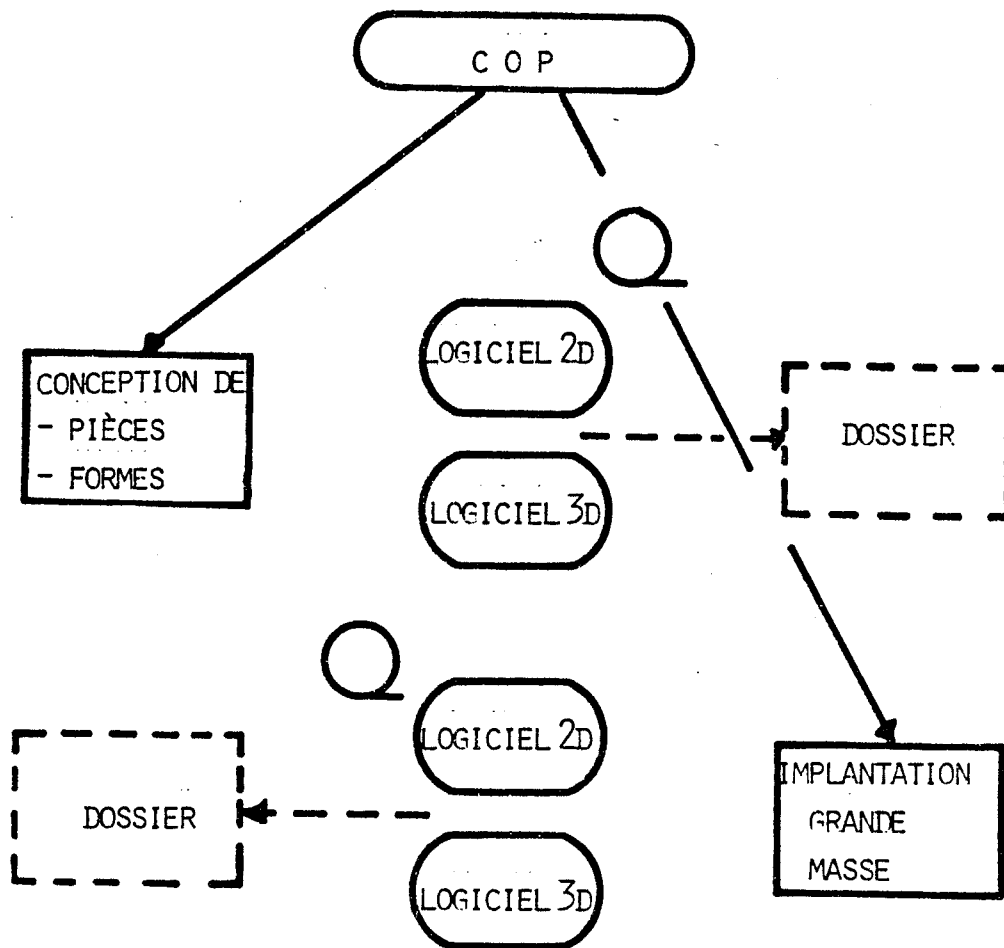


FIGURE A2.13: PHASE 4 - IMPLANTATION, CONCEPTION MECANIQUE

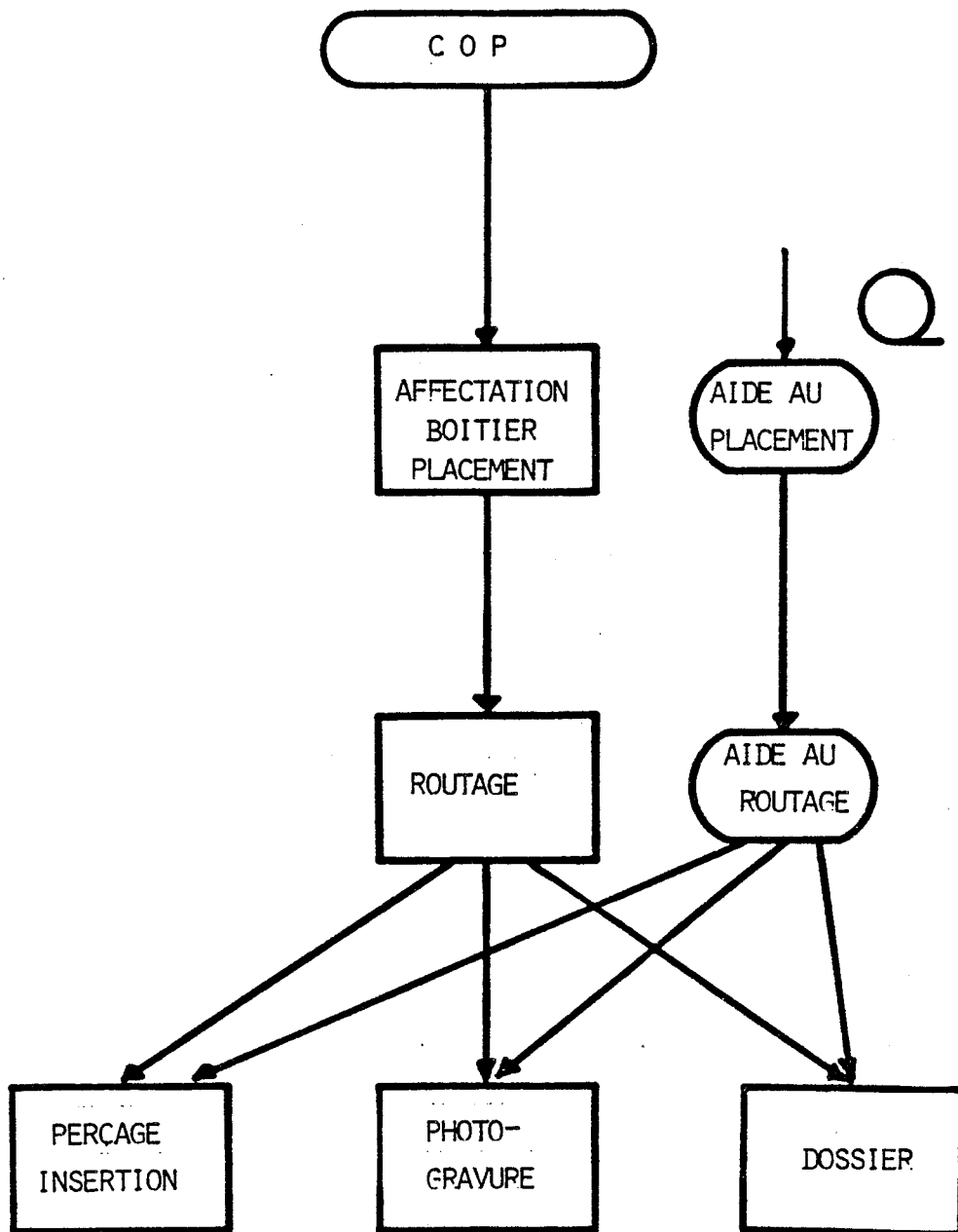


FIGURE A2.14: PHASE 4 - IMPLANTATION CIRCUIT IMPRIME

A2.2.4 ETABLISSEMENT DU CAHIER DES CHARGES FONCTIONNEL DU SYSTEME CAO

Le cahier des charges fonctionnel du système CAO contient deux parties. La première concerne le système support et précise le fonctionnement souhaité du système. Elle correspond pour notre étude à ce que nous avons présenté dans les chapitre 4-9.

La seconde concerne l'application elle même pour laquelle on doit énoncer à l'aide de COLA ses spécifications.

A2.2.5 ETUDE DE FAISABILITE - MAQUETTE

Nous avons pu constater qu'il n'est pas simple de produire directement des spécifications détaillées du système CAO complet. Nous avons donc préféré procéder en deux temps: d'abord nous avons spécifié une partie du processus puis nous avons réalisé le système correspondant. Ce premier travail a pour but de démontrer la faisabilité du projet et de permettre un affinement du cahier des charges. Car souvent pour les néophytes une heure devant le poste de travail remplace des jours d'explication et surtout, ouvre de nouveaux horizons quant aux possibilités offertes par la CAO.

Cette maquette doit représenter deux aspects du système CAO, d'une part le système CAO support dont nous avons donné le cahier des charges dans le chapitre 9 et d'autre part l'application proprement dite montrant, au moins partiellement, le comportement du système dans le domaine d'application choisi. Nous allons donc présenter successivement ces deux aspects.

A2.2.5.1 MAQUETTE DU SYSTEME SUPPORT

Dans le cadre de cette étude de faisabilité nous avons réalisé une version du système support qui est une réalisation particulière de MACAO. Il s'agit du système APPLI-SIGMA décrit dans le chapitre 12.2.

A2.2.5.2 MAQUETTE DE L'APPLICATION

Nous avons choisi de réaliser une maquette traitant la partie conception de matériel et plus spécialement le passage de la fonction vers la réalisation.

Le fonctionnement de la maquette est le suivant (FIG.IV-A2.15):

A partir d'une bibliothèque de fonctions que l'utilisateur peut définir, il construit le schéma fonctionnel de son application en procédant à l'interconnexion de ces schémas de base et au choix des paramètres associés.

Après avoir défini son application il peut tester son comportement à l'aide d'un simulateur logique. Pour cela le système se charge de transformer cette description schématique en une forme simulable.

Par ailleurs l'utilisateur doit définir les stimuli de cette simulation par l'intermédiaire d'un programme interactif graphique qui lui est fourni. Il peut donc effectuer ce travail de façon très naturelle et agréable afin d'activer cette simulation.

Un programme de visualisation de comportements permet d'étudier les résultats de la simulation sous différents angles, tels que le comportement instantané des connexions ou le comportement temporel (chronogrammes).

Les résultats de la simulation peuvent servir de stimuli pour d'autres simulations. De cette façon il est possible de vérifier le comportement du schéma d'une manière progressive.

Un autre aspect important est le passage du schéma à la réalisation. Pour cela il est indispensable de choisir l'ensemble de composants que l'on veut utiliser, et d'effectuer l'allocation des fonctions dans ces composants.

Ceci n'est pas un travail facile car cette allocation a un impact important, aussi bien sur le comportement opérationnel du circuit (fiabilité) que sur sa réalisation (tracé de connexions). En conséquence on procède souvent de façon itérative en couplant cette opération soit à l'étude de fiabilité, soit à celle de définition du circuit imprimé, soit aux deux à la fois.

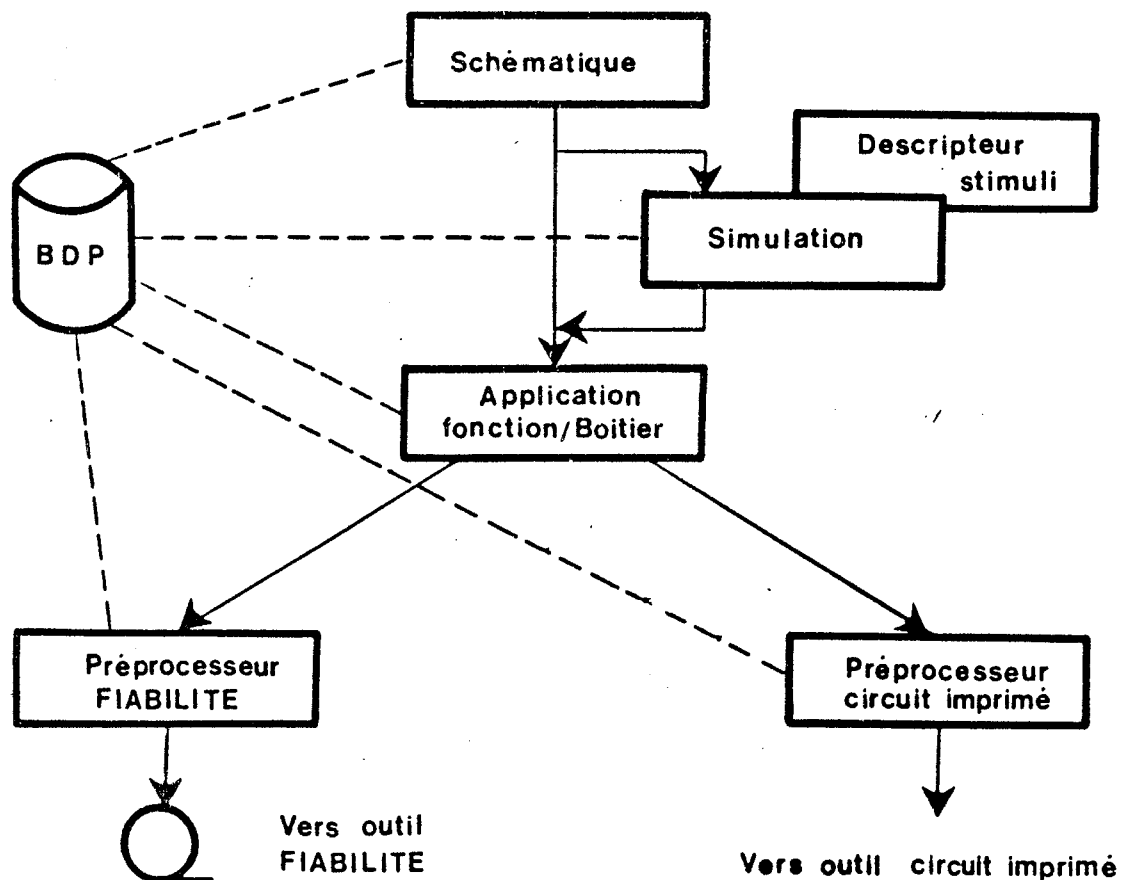


FIGURE A2.15 : STRUCTURE DE LA MAQUETTE DE L'APPLICATION

Dans cette maquette nous avons envisagé toutes ces possibilités car le système assure de façon automatique la liaison entre les programmes de calcul de fiabilité, l'outil de conception et la réalisation de circuits imprimés.

Avant de présenter plus en détail les différents programmes, nous remarquons que cette maquette a permis de traiter et de tester un aspect important du processus de conception. Elle a été réalisée sur un mini-ordinateur, sauf la simulation qui s'effectue sur un ordinateur plus gros et distant. Elle a été toutefois limitée car nous ne disposons pas de dispositions pour son élaboration que de six hommes-mois.

Dans la réalisation de la maquette d'application, l'effort a surtout porté sur l'élaboration de deux programmes, un destiné à la manipulation des schémas, et l'autre à la définition des stimuli, à l'établissement d'une bibliothèque d'objets à utiliser et à la définition de trois programmes d'interface.

A2.2.5.2a Manipulation de schémas

Le but du programme EGIDD, élaboré au laboratoire IMAG, est de permettre la création et la manipulation des schémas électroniques. Pour cela le programme permet, d'une part la définition des objets à stocker dans le catalogue, et d'autre part la construction d'un schéma à l'aide de ces objets.

Lors de la création des objets on définit pour chaque objet:

- son graphisme,
- ses points d'attache (ses liens avec d'autres objets),
- des commentaires (des textes associés aux objets),
- des paramètres que l'utilisateur peut préciser lors de l'utilisation de l'objet.

Quelques objets d'un tel catalogue sont représentés sur la figure IV-A2.16.

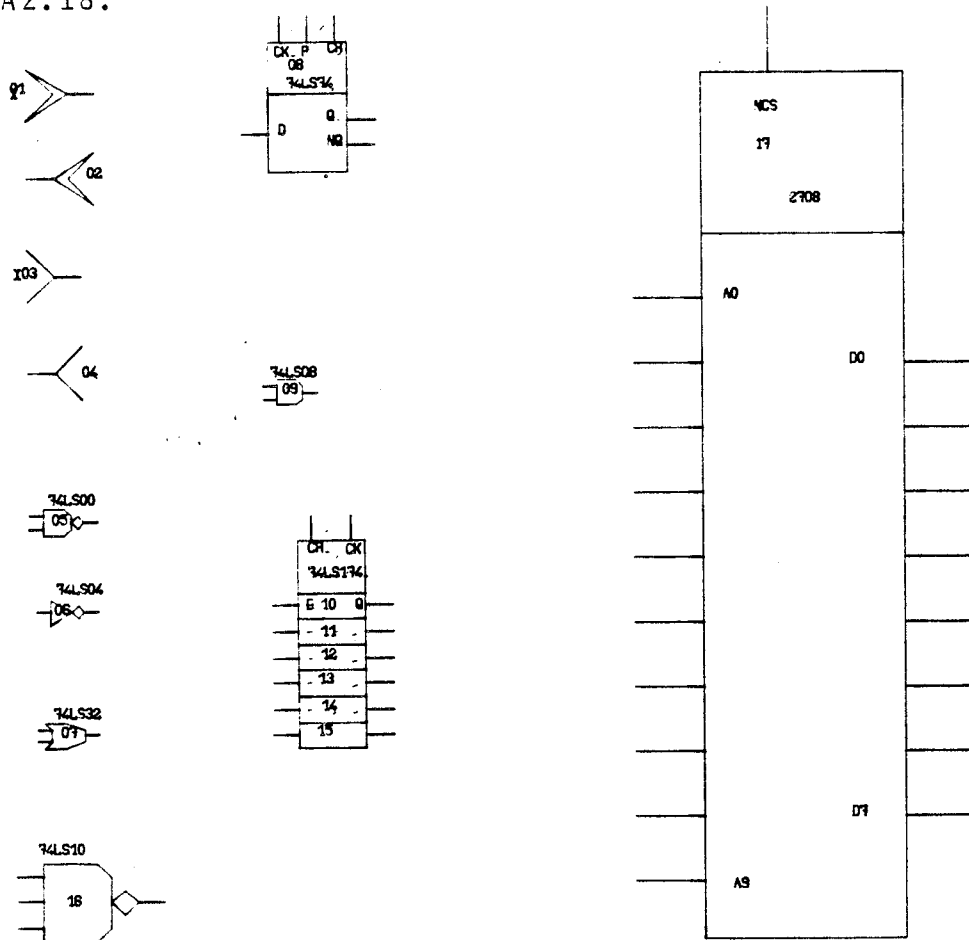


FIGURE IV-A2.16: VISUALISATION DES OBJETS DU CATALOGUE

Les objets catalogués peuvent être utilisés dans la construction d'un schéma. Pour cela un ensemble d'opérations est

proposé afin de permettre une construction agréable et efficace. Les principales opérations sont les suivantes:

- recherche dans le catalogue,
- choix de l'objet à placer,
- placement de l'objet (translation, rotation, changement d'échelle),
- valuation de l'objet (choix de valeurs des paramètres),
- établissement de liens avec d'autres objets (tracé manuel des connexions),
- manipulation du schéma construit (déplacement des objets avec leurs connexions, rotation, alignement,...).

Les figures IV-A2.17-20 montrent ces possibilités.

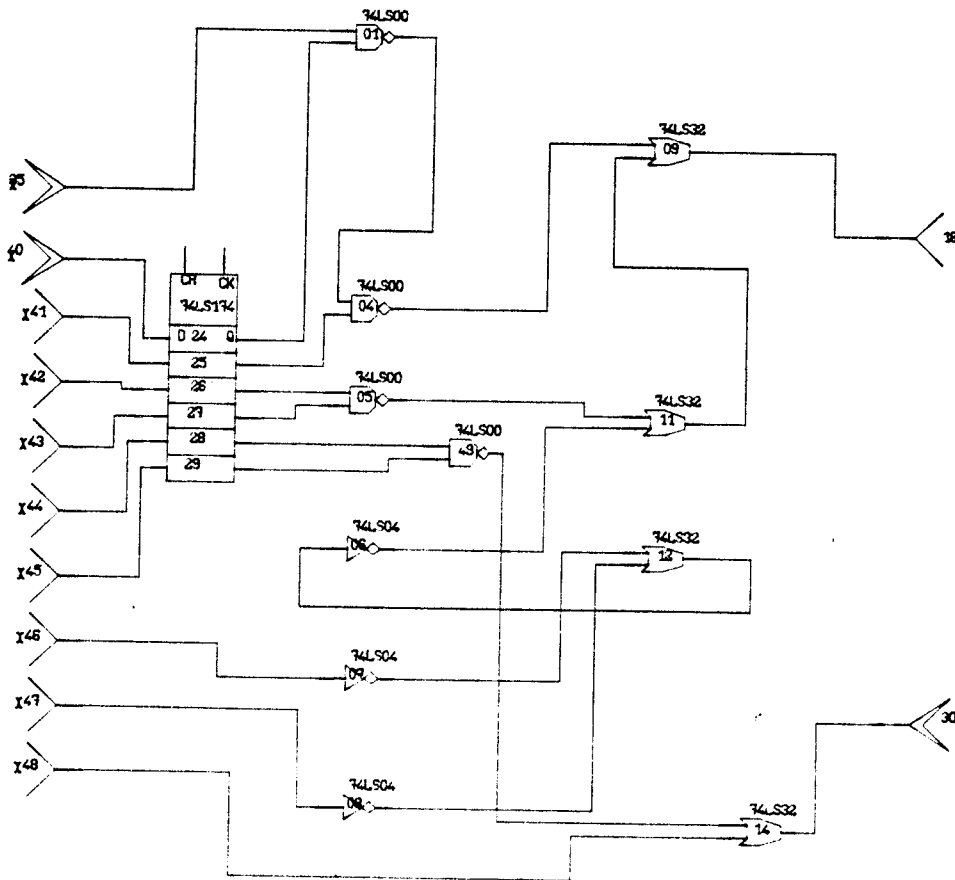


FIGURE A2.17 : PLACEMENT DES OBJETS, PUIS TRACE MANUEL DES CONNEXIONS

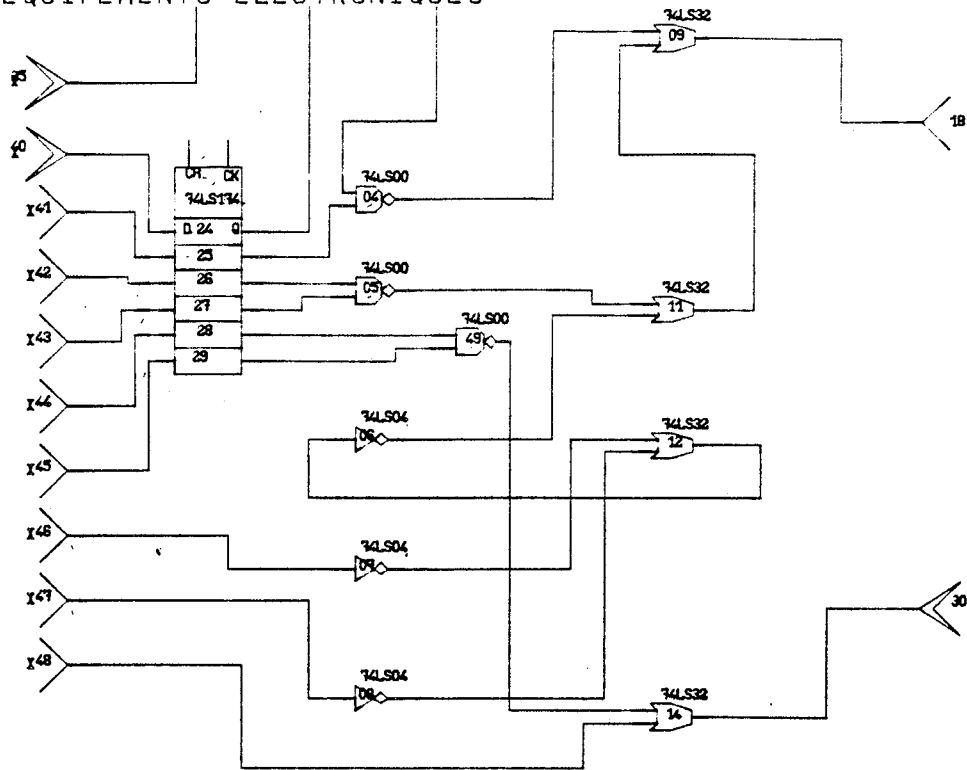


FIGURE A2.18 : TRANSLATION HORIZONTALE DE L'OBJET 14

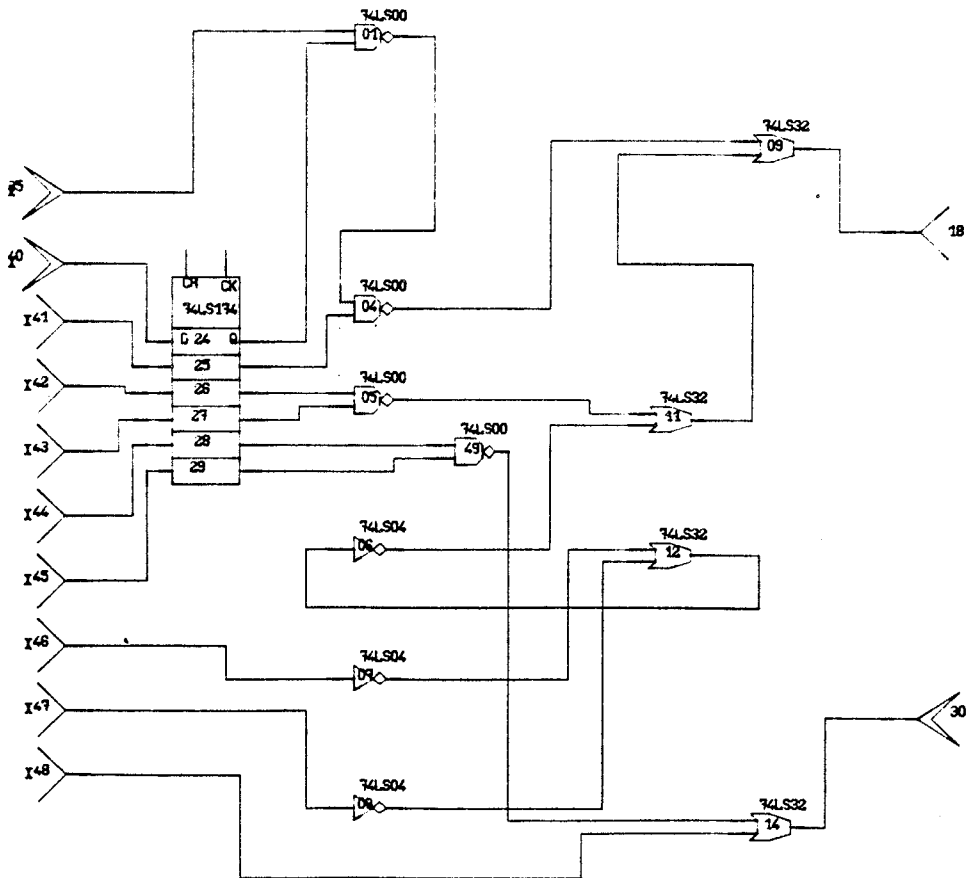


FIGURE A2.19 : ALIGNEMENT VERTICAL DE L'OBJET 9 PAR RAPPORT A L'OBJET 14

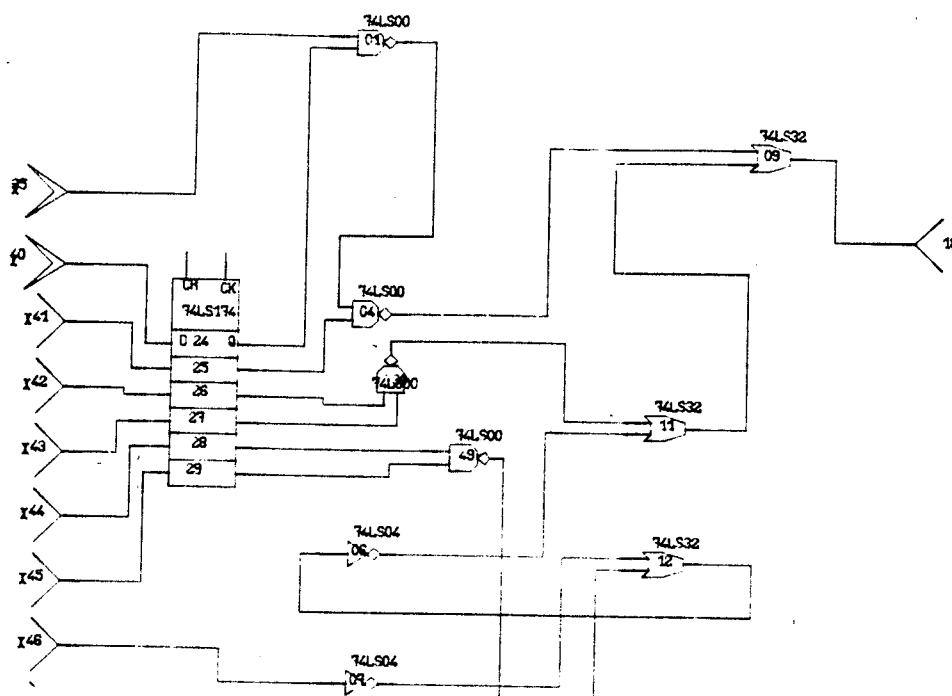


FIGURE A2.20 : ROTATION DE L'OBJET 5

A2.2.5.2b DESCRIPTION DE STIMULI

Le programme de définition de stimuli nous permettra d'une part de montrer, comment on peut créer et manipuler graphiquement des stimuli et d'autre part, comment il s'appuie sur le support APPLI-SIGMA.

Le but du programme est de faciliter le travail de création de stimuli dont on a besoin pour tester l'objet virtuel en simulation.

Il s'est avéré intéressant de permettre la construction de ces stimuli de façon graphique. Nous avons donc étudié quels sont les modes de création qu'il est souhaitable d'autoriser. Nous avons déterminé cinq :

- construction graphique complète (pour toute la durée de la simulation) (FIG.IV-A2.21a),
- construction graphique par période (on précise la durée de la période et le programme complète en répétant la séquence pour obtenir une définition complète) (FIG.IV-A2.21b),
- construction par contexte (on indique que l'horloge ou le signal se comporte comme un autre déjà défini) (FIG.V.9a),
- construction numérique parallèle (on précise combien de signaux participent à la définition et le programme bâtit, par exemple à partir d'une définition hexadécimale, le graphique des quatre signaux (FIG.IV-A2.21c),

- construction numérique en série (on indique la durée de la série et le nombre est décomposé en une suite d'impulsions).

Il est évidemment possible de mélanger ces différents modes de construction (série, par période ou parallèle par période, ...). L'intérêt de ces différents modes de création s'est confirmé à l'usage. A l'aide de ce programme l'utilisateur peut visualiser les signaux, et éventuellement les dessiner sur la table traçante.

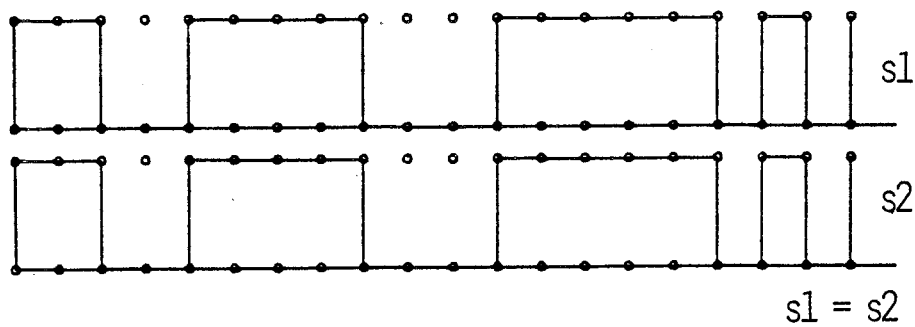


FIGURE 19.22a : CONSTRUCTION GRAPHIQUE COMPLETE ET PAR CONTEXTE

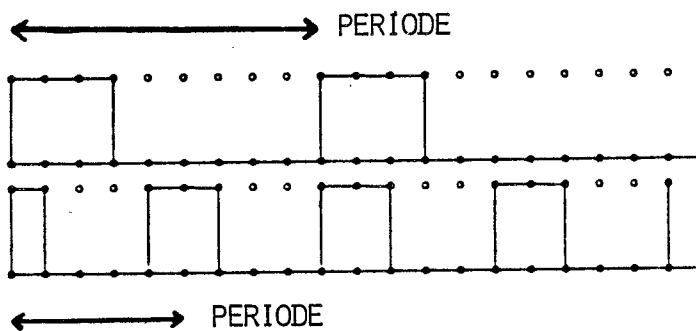


FIGURE A2.21b : CONSTRUCTION GRAPHIQUE PAR PERIODE

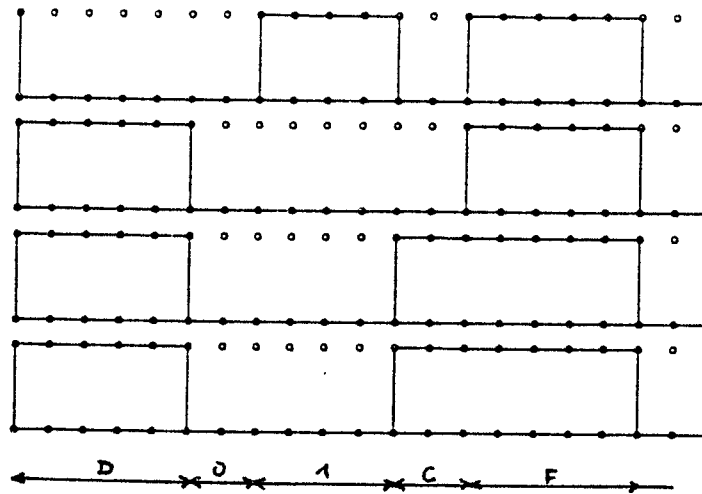


FIGURE A2.21c : CONSTRUCTION NUMERIQUE PARALLELE - HEXADECIMALE

Du point de vue réalisation ce programme s'appuie sur le système APPLI-SIGMA, plus particulièrement sur le moniteur de communication et le moniteur de dialogues. Les dialogues du programme ont été spécifiés à l'aide de LDC (FIG.IV-A2.22) et fournis au moniteur de dialogue. Le programme est, dans ce contexte, constitué uniquement d'un ensemble de fonctions sémantiques, que l'on a caractérisé par leurs numéros dans la clause EXEC de la spécification des dialogues en LDC.

```

appli      'creation des histogrammes' debut=exec(1);
histogra = fils(creerchr,visualiser,supprimer,fin) repeter sauf pour
           (fin),exec(1),alt;
creerchr=fils(norlojes,signaux),alt,exec(2);
norlojes=fils(complete,periode,contexte),alt,exec(3);
signaux=fils(complete,periode,contexte,numerique),alt,exec(4);
complete=aide('duree globale de la simulation' 0),
           param(1 entier 1),
           exec(5),fils(releve);
periode=aide('duree de la periode, nombre de periodes' 0),
           param(2 entier 1),
           exec(5),fils(releve);
releve=aide('relever les pas du chronogramme' 0),exec(7);
contexte=aide('facteur d espacement' 0),
           param(1 entier 1),exec(3);
numerique=fils(seriecon,serieper,paralcom,paralper),
           alt,aide('duree globale de la simulation ou periode'
                   1),
           exec(3),
           param(1 entier 1,1 entier 2 =base 15);
seriecon=exec(11);
serieper=exec(11),aide('nombre de periodes' 0),
           param(1 entier 1);
paralcom=exec(12);
paralper=exec(13), aide('nombre de periodes' 0),
           param(1 entier 1);
visualiser=exec(14),fils(benson,suite),alt;
benson = exec(17),comp;
suite=comp;
supprimer=exec(15);
fin = exec(15)
finappli
eof

```

FILS - CHOIX DES ACTIONS (CONTENU DU MENU)
 EXEC - NUMERO DE L'ALGORITHME ASSOCIE
 PARAM - NATURE DU PARAMETRE A SAISIR
 AIDE - MESSAGE EXPLICATIF

FIGURE A2.22 : SPECIFICATION DES DIALOGUES EN LDC

A2.2.5.2c Etablissement de la bibliothèque

Pour pouvoir travailler avec des éléments (fonctions, boîtiers, ...), il faut fournir leurs caractéristiques (nomenclature, symbolisme graphique, caractéristiques de fiabilité, ...). C'est un travail qui est surtout long et fastidieux, car dans une entreprise il faut saisir des milliers d'éléments (composants de différentes familles, de différents fabricants, ...).

Sans entrer dans les détails nous voulons rappeler seulement ce qu'il faut fournir pour pouvoir utiliser ces éléments dans la maquette que nous avons réalisée.

Dans tout système CAO on a besoin de désigner de façon claire et sans ambiguïté tous les éléments manipulés. Il s'agit d'un choix de nomenclature qui conditionne l'ensemble des travaux.

Pour la manipulation de schémas nous avons besoin de connaître les symbolismes des différentes représentations des éléments, et pour cela il faut définir les conventions de la définition. Il s'agit par exemple de définir pour des fonctions: des entrées, des sorties, des entrées de service, des sorties de service et la désignation de la fonction.

Pour la simulation il faut décrire le comportement fonctionnel des fonctions élémentaires ainsi que des fonctions complexes. Pour ces fonctions complexes on peut soit procéder par assemblage des fonctions élémentaires soit les modéliser directement. Nous avons ainsi constitué toute une bibliothèque de descriptions en CASSANDRE de ces fonctions.

Pour le calcul de la fiabilité on doit passer des sous-ensembles technologiques aux sous-ensembles fonctionnels. Il s'agit d'une autre modélisation à l'aide d'un ensemble d'éléments de modélisation de la fiabilité. Il faut donc disposer de cet ensemble et pouvoir de façon interactive construire un schéma de fiabilité pour un schéma technologique donné. En plus un ensemble d'informations de fiabilité doit être fourni pour tous les sous-ensembles technologiques. La grande partie de ces informations ne dépend pas du produit particulier mais des éléments utilisés; elle peut donc être stockée une fois pour toute dans la bibliothèque.

Pour l'implantation d'une carte de circuit on a besoin de la liste des interconnexions et de l'ensemble des informations sur les éléments utilisés (nomenclature, bornage sur la carte, ...). Ces informations doivent aussi être stockées dans une bibliothèque.

A2.2.5.2.d Programmes d'interface

Quand on veut ou doit utiliser, dans un système CAO intégré, des programmes technologiques existants conçus et réalisés pour fonctionner de façon autonome, ce qui est très souvent le cas, on doit entreprendre l'intégration de ces programmes dans le système.

Ceci s'effectue généralement à l'aide des programmes d'interface dont le principe a été discuté dans le chapitre 7 (FIG.II-7.6). Comme nous l'avons expliqué il s'agit d'enfermer le programme technologique par deux traitements: le pré-traitement et le post-traitement.

Le pré-traitement a pour rôle de prendre dans le système les informations nécessaires pour le programme, de les transformer et de les mettre sous la forme souhaitée par le programme technologique.

Le post-traitement a pour rôle de prendre les informations générées par le programme technologique, de les mettre sous la forme standard et de les retourner au système CAO.

Dans la maquette nous avons étudié l'écriture d'interfaces pour la SIMULATION, la FIABILITE et l'IMPLANTATION DE CIRCUITS.

Pour la simulation il s'agit de fournir à CASSANDRE la description du schéma, décrite en langage de description CASSANDRE. Il est relativement simple de générer cette description à partir des informations dont nous disposons après la création graphique du schéma, car il s'agit de décrire uniquement l'interrelation des éléments, pour lesquels, la description en CASSANDRE du fonctionnement existe en bibliothèque. En plus il faut fournir l'ensemble des commandes de la simulation et les exprimer à l'aide du langage de commande du système CASSANDRE. Ici il s'agit principalement de transformer la description des stimuli sous la forme souhaitée.

Après la simulation on obtient des résultats sous la forme CASSANDRE. Ce sont notamment des valeurs que prennent des signaux à différentes étapes de la simulation, en correspondance avec les stimuli.

Le post-traitement doit donc retraduire ces résultats, pour pouvoir les utiliser dans le système, pour l'étude des chronogrammes obtenus et pour analyser la dynamique du schéma.

Pour la fiabilité et l'implantation de circuit, il faut réaliser uniquement des pré-traitements; car les informations produites par ces programmes technologiques ne sont pas utilisés dans le système dans l'état actuel.

La maquette ayant fourni des nouveaux éléments d'appréciation il est possible maintenant de poursuivre la mise en place du système CAO en effectuant les étapes restant encore à entreprendre.

A2.2.6 REDACTION DU CAHIER DES CHARGES DETAILLE

Lors de la rédaction du cahier des charges détaillé, il s'agit de produire les spécifications fonctionnelles du système CAO. En fait on affine les spécifications produites à la phase 4 en tenant compte des renseignements obtenus lors de l'étude de faisabilité et on les rédige sous la même forme.

Ces spécifications concernent d'une part le système support et d'autre part l'application elle-même. Le cahier des charges du système CAO support a été présenté dans les chapitres 4-9. Le cahier des charges de l'application se trouve dans le document "Cahier des charges d'un système CAO" publié par CROUZET (CZI.89).

A2.2.7 ETABLISSEMENT DU CAHIER DES CHARGES ORGANIQUE

Après avoir déterminé comment doit fonctionner le système CAO, il faut maintenant définir la façon de bâtir le système, c'est à dire effectuer les choix de matériel et de logiciel et les inévitables compromis (à l'intérieur des choix techniques, entre le technique et l'économique).

Pour cela il faut disposer non seulement d'informations qualitatives mais aussi quantitatives afin de déterminer les besoins (en puissance de calcul, en taille mémoire, en taux d'utilisation des périphériques, etc...) de chaque étape du processus, puis globalement en tenant compte de la charge du processus et donc envisager de dédoubler certains postes de travail.

A partir de ces informations on doit bâtir une stratégie concernant le choix de matériel (aussi bien de ou des ordinateurs que des périphériques) et de logiciels (réutilisation, élaboration, achat, location, service bureau, etc...).

Différentes solutions doivent être envisagées, comparées et proposées.

Nous ne pouvons présenter les résultats de cette étape et de celles qui vont suivre et ce pour deux raisons essentielles:

- la première est opérationnelle. En effet cette étape n'a pas été abordée dans le cadre de l'étude que nous avons menée mais de façon interne par CROUZET,

- la seconde est stratégique. Les choix de matériel, d'organisation et de planification sont des atouts stratégiques et, comme toute stratégie, elle ne peut être dévoilée à n'importe quel moment.

Les autres étapes du processus d'introduction de la CFAO en milieu industriel (plan d'investissement et plan de mise en oeuvre, mise en oeuvre) n'ont pas été abordées dans le cadre de cette étude.

A2.3 CONCLUSIONS

Nous pouvons conclure en insistant d'une part sur les problèmes psychologiques qu'une telle étude peut provoquer et d'autre part sur les résultats obtenus.

A2.3.1 PROBLEMES PSYCHOLOGIQUES

Il est évident qu'une telle opération ne se passe pas toujours sans problèmes. Il faut tenir compte et essayer d'éviter à tout prix certains problèmes graves, notamment les problèmes psychologiques. Nous n'en citerons que deux:

- l'intervention de personnes extérieures à l'entreprise peut souvent être mal ressentie. C'est pourquoi il est important que celles-ci soient intégrées au sein de l'équipe opérationnelle, et que le contact avec les autres personnes de l'entreprise s'effectue par l'intermédiaire des personnes permanentes ou tout au moins en leur présence,
- la crainte du personnel, à tous les niveaux, de perdre des responsabilités par la banalisation des compétences (n'importe qui pourra utiliser le système et donc faire leur travail). Pour pallier cela il faut nécessairement définir les prérogatives de chacun et préserver une structure dans laquelle la compétence et la responsabilité de chacun deviennent ou restent bien définies.

A2.3.2 RAPPEL DES RESULTATS

L'intérêt de la CFAO intégrée semble indéniable. Si on veut aboutir à des résultats satisfaisants il faut lors de l'introduction de la CFAO en milieu industriel étudier tous les aspects de la conception et de la fabrication. Ce n'est qu'une approche globale et réfléchie qui peut fournir des résultats tangibles. Nous pouvons récapituler l'étude que nous avons menée à l'aide des schémas suivants (Fig.IV-A2.23):

- processus actuel
- processus nouveau
- outils (généraux ou spécifiques)
- données (générales ou spécifiques à un projet)

Ils schématisent ce par quoi il faut commencer toute étude d'introduction de la CFAO dans le milieu industriel. Un autre aspect n'est pas représenté sur ce schéma. Il s'agit du support informatique que nous avons étudié dans les chapitres précédents. Il constitue la face cachée de la CAO mais conditionne en grande partie la réussite de l'application.

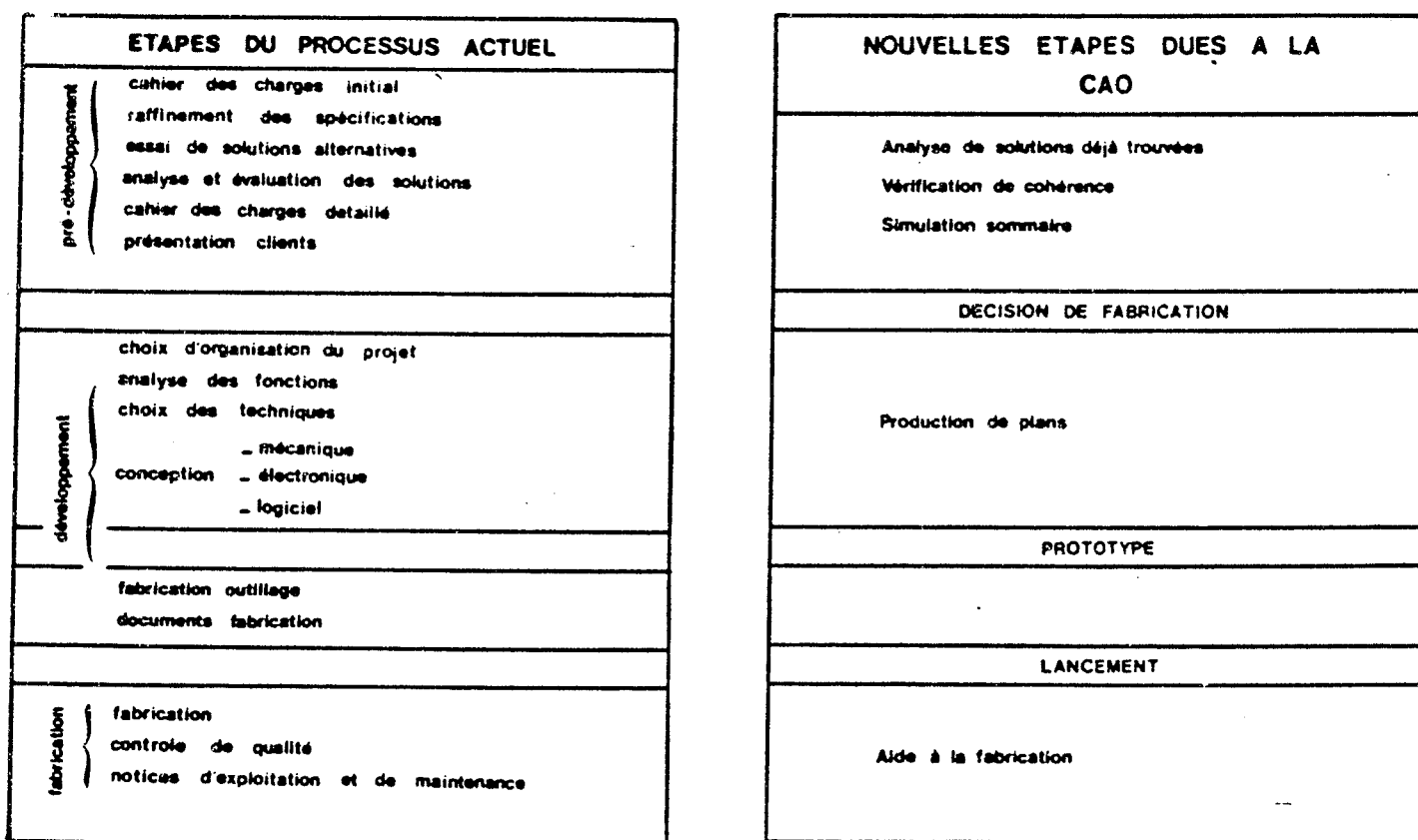


FIGURE A2.23a : PROCESSUS ACTUEL ET PROCESSUS AVEC CFAO

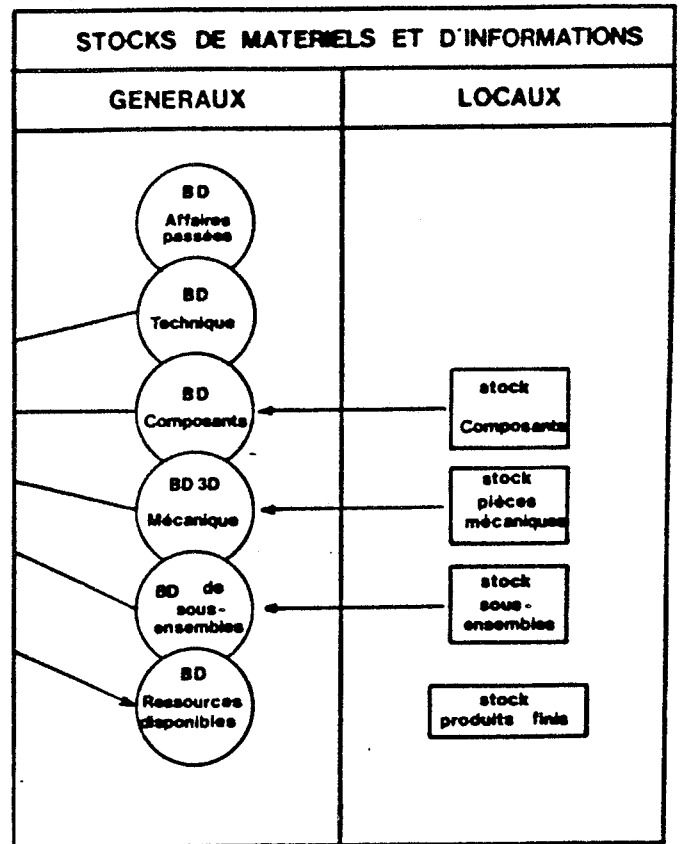
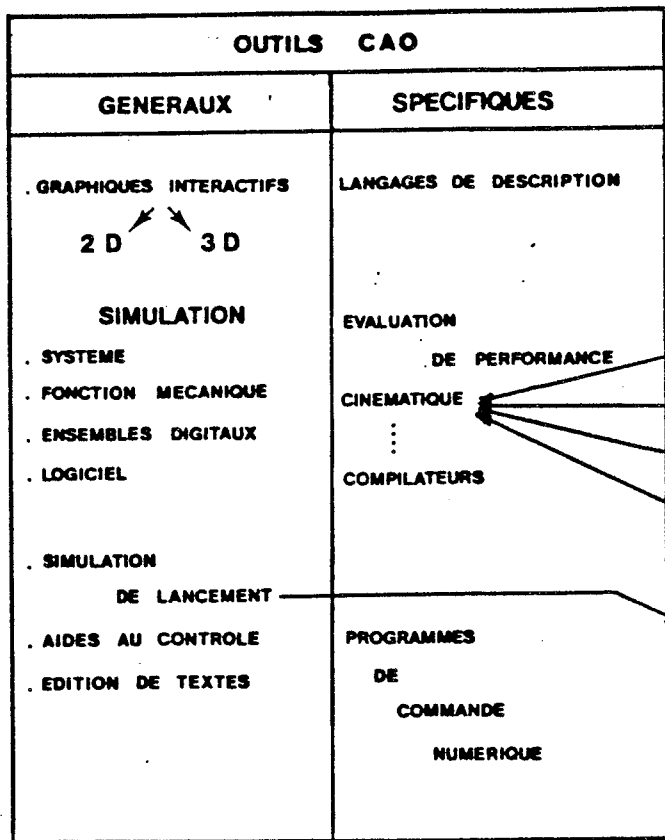


FIGURE A2.23b : OUTILS CAO ET DONNEES MANIPULEES PAR LE SYSTEME CAO

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- Abr.78 J.R.ABRIAL
Présentation du langage Z
- ACH.75 A. de CHELMINSKI
Le projet MACSI-1
Thèse Docteur-Ingénieur, Grenoble, Juin 1975
- ACL.78 S.A.ABBAS, A.COULTAS and B.S.LEE (Ed.)
CAD - ED : International Conference on COMPUTER AIDED
DESIGN EDUCATION
IPC Science and technology Press, 1978
- ACM.74 ACM Computing Surveys
Special issue: Programming
Vol.6, No.4, Dec.1974
- ADD.76 M.ADIBA, C.DELOBEL
Les modèles relationnels de bases de données
IRIA - SEFI, avril 1976
- Alf.77 M.W.ALFORD
A Requirements Engineering Methodology for Real-Time
Processing Requirements
IEEE Trans. on Software Engineering, Vol. SE-3, no.1,
Jan. 1977
- All.76 J.ALLAN (ed.)
Proceedings of the IFIP W.G. 5.2 Working Conference on
CAD Systems
AUSTIN, Texas, fev. 1976
- All.78 J.J.ALLAN
CAD Education to the Year 2000
in: S.A.ABBAS, A.COULTAS and B.S.LEE (Ed.) CAD-ED, IPC
Press, 1978
- ArH.76 B.S.ARMITAGE and P.A.V HALL
Conceptual schema for CAD
The British Ship Research Association, Wallsend, Tyne
and Wear, England, 1976

- ArW.76 M.ARKINSON and N.WISEMAN
Data management requirements for large scale design and production
Rainbow Memo 128, Cambridge University Computer Laboratory, august 1976
- Ath.78 R.J.ATHAY
Object models for Computer Aided Design: an overview
SIGGRAPH'78
- BaB.75 P.G.BAZILLOU, G.E.BENCI
Présentation et analyse de S.G.B.D. commercialisés en France
IRIA, S.T.I., Groupe "applications", Octobre 1975
- BaE.74 P.L.BAUER, J. EICKEL
Compiler Construction
Springer Verlag, 1974
- BaH.75 M.J.BASTARACHE, E.A.HERSHEY
Problem Statement Analyser- Command Descriptions
ISDOS Working Paper no.91, July 1975
- BaI.73 R.BALZER
Automatic Programming
Institute Technical Memo, USC-ISI, 1973
- Bar.80 J.L.BARRON
Dialogue Organization and Structure for Interactive Information Systems
Technical Report CSRG-108, University of TORONTO, january 1980
- Bat.75 W.F.BATES
Technical Information Processing Systems
J. of Professional Activities, ASCE, Vol.101, No.EI2, april 1975
- BBD.77 T.E.BELL, D.C.SIXLER, M.E.DYER
An Extendable Approach to Computer Aided Software Requirements Engineering
IEEE Trans. on Software Engineering, Vol. SE-3, Jan. 1977
- BBD.80 M.BARBACCI, D.BORRIONE, D.DIETMEYER, R.PILOTY, P.SKELLY
CONLAN - A formal Construction Method for Hardware DEscription Languages
NCC 80 AFIPS Conference Proceedings, Volume 49
- BDF.75 Y.BRESSY, B.DAVID, Y.FANTINO, J.MERMET
A hardware compiler for Interactive Realization of Logical Systemes described in CASSANDRE

Workshop on Computer Hardware Description Languages and Their Applications, New-York, Sept. 1975

- BDL.76 D.BORRIONE, B.DAVID, C.LE FAOU, G.VITRY
Projet de groupe d'études coordonnées (GRECO) d'Analyse de systèmes en Rhône-Alpes: Une approche pour l'élaboration de systèmes d'aide à la modélisation et à la simulation
E.N.S.I.M.A.G., Mars 1976
- BEH.79 A.BAER, C.EASTMAN, M.HENRION
Geometric Modelling: a Survey
Computer Aided Design, Vol.11, No.5, sept 1979
- BeN.71 G.BELL, A.NEWELL
Computer Structures: Readings and Examples
McGRAW-HILL, 1971
- Ber.78 G.BERGER-SABBATEL
Etude fonctionnelle d'un processeur de bases de données hiérarchiques
Thèse de 3ème cycle, Grenoble, juin 1978
- BeR.79 G.BENCI, C.ROLLAND
LES BASES DE DONNEES: d'une conception canonique à une réalisation extensible
Editions S.C.M. 1979
- Ber.79 O.P.BERTRAND
Le développement d'applications interactives
IBM France, Scientific Center, mai 1979
- BGL.71 BOGO, GUYOT, LUX, MERMET, PAYAN
CASSANDRE and the Computer-Aided Logical Systems Design
Proceedings of the IFIP Congress, 1971
- BGW.78 R.BALZER, N.GOLDMAN, D.WILE
Informality in Program Specifications
IEEE Trans. on Software Engineering, Vol. SE-4, no.2, March 1978
- BLR.74 G.BLAIN, A.LABARTHE, J.C.RAULT, M.SCIARDIS, P.ZAMANSKY
BAL: an Aid to Scientific Programming Based upon a "Bank of Algorithms"
Proc. IFIP Congress 1974
- Bo..77 K.30
Text Analyzer and Text Generator for Interactive Command Languages in CAD
in: J.J.ALLAN (Ed.), CAD Systems, North Holland, 1977

- BoG.79 D.BORRIONE, J.F.GRABOWIECKI
Informal Introduction to LASSO: A Language for
Asynchronous Systems Specification and Simulation
Proceedings Euro IFIP 79, London, sept. 1979
- Bor.75 D.BORRIONE
LASCAR: A Language for Simulation of Computer
Architecture
3rd International Symposium in Computer Hardware
Description Languages and their Applications, New York,
sept. 1975
- Bou.77 F.BOUILLE
Un modèle universel de banque de données, simultanément
partageable, portable et répartie
Thèse de doctorat d'Etat es sciences, Paris, avril 1977
- Boy.79 J.BOYSE
Interference Detection Among Solids and Surfaces
CACM, Vol.22, No.1, Jan.1979
- Bra.75 I.C.BRAID
The SYNTHESIS of Solids Bounded by Many Faces
CACM, Vol.18, No.4, april 1975
- Bra.78 I.C.BRAID
On Storing and Changing Shape Information
SIGGRAPH'78
- Bra.78 I.C.BRAID
New Directions in Geometric Modelling
Proceedings of Geometric Modelling Project Meeting,
CAM-I, March 1978
- Bru.78 J.M.BRUN
EUCLID
Rapport du LIMSI, 1978
- BRZ.75 B.DAVID, V.RIVERO, F.Z'GRAGGEN
Apports de l'Informatique à l'Architecture et à
l'Urbanisme
Séminaire à l'Ecole d'Architecture, Université de
Genève, 5 juin 1975
- BuD.79 A.P.BUCHMANN and A.G.DALE
Evaluation criteria for logical database design
methodologies
CAD, Vol.11, No.3, may 1979
- CAC.77 Selected Papers from the Conference on Data:
Abstraction, Definition and Structure
CACM, Vol.20, no.6, June 1977

- CAM.78 Proceedings on Data Base Seminar
CAM-I
Bournemouth, march 1978
- Cap.78 M.CAPLAIN
Langage de spécifications
Thèse d'état, Grenoble, Dec. 1978
- CEC.78 The Commission of the European Communities
Symposium on Computer Aided Design of Digital
Electronic Circuits and Systems
Bruxelles, nov. 1978
- CER.74 CERT-DERI
Journées sur les systèmes généraux pour la CAO
CERT-DERI, Toulouse, Dec. 1974
- Chu.75 Y.CHU
A Methodology for Software Engineering
IEEE Trans. on Software Engineering, Vol.SE-1, No.3,
september 1975
- Chu.78 J.C.CHUPIN
Répartition d'applications et de bases de données sur
un réseau général d'ordinateurs
Thèse d'Etat, Grenoble Octobre 1978
- Col.78 R.P.A.COLLINSON
The Diagram System
Computer Aided Design, Vol.10, No.1, jan. 1978
- CZT.79 CROUZET
Procédé de C.A.O. utilisable dans le processus
étude-développement-production de produits et
équipements à dominante dégtale
Rapport d'avancement, CROUZET, 1979
- CZT.80 CROUZET
Cahier des charges d'un système CAO
Rapport CROUZET, 1980
- Dal.77 E.B.DALY
Management of Software development
IEEE Trans. on Software Engineering, May 1977
- Dan.78 Ng.X.DANG
Système et langage portable pour le traitement des
applications réparties
Thèse de 3ème cycle, Grenoble, Mars 1978

- DaR.74a B.DAVID, V.RIVERO
An Approach to Computer Aided Architectural Design
CAD'74: International Conference and Exhibition on
Computers in Engineering and Building Design, London,
september 1974
- DaR.74b B.DAVID, V.RIVERO
An Approach to Computer-Aided Design
Proceedings of the 9th Yugoslav International Symposium
on Information Processing, INFORMATICA 1974, Bled
(Yougoslavie), Octobre 1974
- DaS.77 Ng.X.DANG, G.SERGEANT
Etude d'une machine réseau logique SIGOR (Système
d'Interprétation Général Orienté Réseau)
Rapport de Recherche IMAG no 78, juin 1977
- Dav.73 B.DAVID
Aspect graphique d'un système conversationnel pour la
Conception Architecturale Assistée par Ordinateur
Journées Graphiques, AFCET-IRIA, Paris, Dec.1973
- Dav.74 B.DAVID
Aspect graphique d'un système conversationnel pour la
Conception Architecturale Assistée par Ordinateur
Automatisme, No.4, avril 1974
- Dav.75 B.DAVID
Pour une généralisation des systèmes C.A.O.: Approche
et applications
Thèse de doctorat de spécialité (3e cycle), USMG-INPG,
Grenoble, Octobre 1975
- DaV.77 C.G.DAVIS and C.R.VICK
The Software Development System
IEEE Trans. on Software Engineering, Vol.SE-3, No.1,
january 1977
- Dav.77 B.DAVID
An Approche for Elaboration of Computer Aided Design
Systems: SIGMA-CAD System
5th International Congress on Data Processing in
Europe, Vienna, March 1977
- DaV.78 B.DAVID, G.VITRY
SIGMA-CAD: Some New Concepts in Design of General
Purpose CAD Systems
15th Design Automation Conference, Las Vegas, June 1978
- Dav.79 B.DAVID
Elaboration of a CAD Education Curriculum

Computer application in industry International Symposium and Exhibition, Grenoble, june 1979

- Dav.80a B.DAVID
Systèmes intégrés d'aide à l'architecte
Conférence-débat organisé par la Chambre de Commerce de NANTES sur le thème "Conception Assistée par Ordinateur en architecture et
- Dav.80b B.DAVID
An Integrated CAD System for Architecture
17th Design Automation Conference, Minneapolis, juin 1980
- Dav.80c B.DAVID
Definition and implementation of an integrated CAD-CAM system for digital equipments
3rd International Symposium on Large Engineering Systems, Memorial University of Newfoundland, Canada, juillet 1980
- Dav.80d B.DAVID
Spécification et mise en oeuvre d'un système intégré pour la CAO: Méthodologie et exemple
MICAD 80: Première conférence européenne sur la CAO dans les Moyennes et petites industries, Paris, Sept 1980
- Dav.80e B.T.DAVID
Man-Machine Communication in the Specification, Implementation and Use of an Integrated CAD/CAM System
IFIP W.G.5.2-W.G.5.3 Working Conference on Man-Machine Communication in CAD/CAM, Tokyo, octobre 1980
- Dav.81 B.DAVID
Computer-Aided Education: an Overview
WCCE 81, 3e conférence mondiale: informatique et enseignement
- Dav.81a B.DAVID
Specification and implementation of an integrated CAD system: Methodology and Examples
in J.MERMET (Ed.): CAD in Medium sized and Small Industries, North-Holland, 1981
- DaV.82 B.DAVID, G.VITRY
C.A.D.System as a Distributed Computing System
soumis à PROLAMAT'82
- DaV.XX B.DAVID, G.VITRY
La base de données pour la CAO: recherche d'un modèle

et de sa mise en oeuvre
en préparation

- DaZ.79 B.DAVID, F.Z'GRAGGEN
Le système SIGMA-ARCHI appliqué à l'industrialisation ouverte
Congrès international sur l'Emploi des Ordinateurs dans l'Architecture, la Construction et la Planification Urbaine, PARC 79, Mai 1979
- dCh.75 A. de CHELMINSKI
Le projet MACSI-1
Thèse de docteur-ingénieur, juin 1975
- DDH.72 O.J.DAHL, E.W.DIJKSTRA, C.A.R.HOARE
Structured programming
Academic Press, 1972
- DeG.80 M.DELAUNAY, J.F.GRABOWIECKI
Note Technique d'Utilisation du Transformateur de Grammaire L1(1)
Rapport de recherche no.234, IMAG, 1980
- DeK.76 F.De REMER, H.H.KRON
Programming in the Large Versus Programming in the Small
IEEE Trans. on Software Engineering, Vol. SE-2, no.2, June 1976
- Del.79 M.DELAUNAY
PORTISH
Bulletin GROSPLAN no.9, CARGESSE, mai 1979
- Der.74 J.C.DERNIAME
Le projet CIVA: un système de programmation modulaire
Thèse de Doctorat d'Etat, Nancy, Janvier 1974
- DGM.80 B.DAVID, Y.GARDAN, J.MERMET
C.A.D.in small and medium sized industries
8th World Computer Congress (IFIP Congress 80), Tokyo, octobre 1980
- DLR.76a B.DAVID, P.LESPINASSE, V.RIVERO, R.VILLAMAYOR,
F.ZANELLI, F.Z'GRAGGEN
SIGMA-ARCHI: un système interactif graphique pour méthodes d'aide à la conception architecturale
Congrès AFCET, Paris, Nov.1976
- DLR.76b B.DAVID, P.LESPINASSE, V.RIVERO, R.VILLAMAYOR,
F.ZANELLI, F.Z'GRAGGEN
Projet de groupe d'études coordonnées (GRECO) d'Analyse de systèmes en Rhône-Alpes: Equipe Conception

Architecturale Assistée par Ordinateur
E.N.S.I.M.A.G., Mars 1976

- DRV.75 B.DAVID, V.RIVERO, R.VILLAMAYOR, F.Z'GRAGGEN
AGGLO: Automatique Générative de Groupements de Logements
Proceedings of the 10th Yugoslav International Symposium on Information, Bled (Yougoslavie), Octobre 1975
- DRV.76a B.DAVID, V.RIVERO, R.VILLAMAYOR, F.ZANELLI, F.Z'GRAGGEN
SIGMA-ARCHI for dwelling architectural design
INFORMATICA 76, Bled (Yougoslavie), Octobre 1976
- DRV.76b B.DAVID, V.RIVERO, R.VILLAMAYOR, F.ZANELLI, F.Z'GRAGGEN
Une approche pour l'élaboration d'un système pour la conception architecturale assistée par ordinateur
ASSPA International Symposium, Fribourg (Suisse), Octobre 1976
- DRV.77 B.DAVID, V.RIVERO, R.VILLAMAYOR, F.ZANELLI, F.Z'GRAGGEN
Computer Aided Architectural Design: SIGMA-ARCHI
IFAC Conference on Systems Approche for Development, Cairo, Nov.1977, publié par Pergamon Press, Editor M.A.R.GHONAIMY
- DRZ.75 B.DAVID, V.RIVERO, F.ZANELLI, F.Z'GRAGGEN
Courte présentation des travaux de l'équipe dans le cadre du Workshop: "Informatique et création de forme en urbanisme et en architecture"
38èmes Journées d'Information de l'A.S.S.P.A. (Association Suisse pour l'Automatique) sur le thème: "Aides à la Création", Université de Genève, Sept.1975
- DRZ.75 B.DAVID, V.RIVERO, F.Z'GRAGGEN
Apports de l'informatique à l'architecture et à l'urbanisme
Séminaire d'informatique en architecture et en Urbanisme, Ecole d'architecture, Université de Genève, juin 1975
- Dub.80 D.DUBOIS
Utilisation de la Théorie des Ensembles Flous en CAO
Rapport de Recherche no 199, Laboratoire IMAG, Avril 1980
- DVV.79 B.DAVID, P.VENIER, G.VITRY
Une méthode pour la prise en compte de la répartition dans les systèmes pour la Conception Assistée par Ordinateur

Journées "BIGRE": Méthodes et Outils pour la Conception de systèmes Répartis, Nancy, Janvier 1979

- DZZ.77 B.DAVID, F.ZANELLI, F.Z'GRAGGEN
Teaching CAD in an Architectural School with the SIGMA-ARCHI System
CAD-ED, Computer Aided Design Education International Conference and Exhibition, Middlesbrough, July 1977
- DZZ.79 B.DAVID, F.ZANELLI, F.Z'GRAGGEN
Teaching CAD in an Architectural School with the SIGMA-ARCHI System
Computer & Graphics - An International Journal, Vol.4, pp.193-199, 1979
- EaH.77 C.EASTMAN, M.HENRION
GLIDE: a Language for Design Information Systems
Proceedings 1977 SIGGRAPH conference, ACM, New York, 1977
- Eas.76 C.M.EASTMAN
Databases for physical system design: a survey of US efforts
CAD'76 Proceedings
- Eas.78 C.EASTMAN
The Representation of Design Problems and Maintenance of their Structure
IFIP W.G. 5.2 Working Conference on Artificial Intelligence and Pattern recognition in Computer-aided Design, Grenoble, 1978
- Eas.80a C.EASTMAN
A prototype integrated building model
IBS Reserch Reprt No.3, january, 1980
- Eas.80b C.M.EASTMAN
System facilities for CAD databases
IBS Research Report No.5, april 1980
- EaT.79 C.EASTMAN and R.THORNTON
A report on the GLIDE2 Language Definition, Preliminary Draft - 22 March 1979
Institute of Physical Planning, Carnegie-Mellon University
- ELS.76 C.EASTMAN, J.LIVIDINI, D.STOKER
A Database for Designing Large Physical Systems
Workshop on computer representation of physical systems, Carnegie-Mellon University, August 1976

- Enc.80 J.ENCARNACAO (Ed.)
Computer Aided Design: Modelling, Systems Engineering,
CAD Systems
CREST Advanced Course, Springer-Verlag, 1980
- Equ.75 Equipe CAO de L'ENSIMAG
Système SIGMA-ARCHI: Plaquette et film vidéo
Inova 75, Paris
- Equ.76a Equipe CAO de l'ENSIMAG
Système SIGMA-ARCHI
Film vidéo réalisé par l'Ecole d'Architecture de
l'Université de Genève, Janvier 1976
- Equ.76b Equipe CAO de l'ENSIMAG
Système SIGMA-ARCHI
Plaquette
- PBL.71 E.A.FEIGENBAUM, B.G.BUCHANAN, J.LEDERBERG
On Generality and Problem Solving: A Case Study Using
the DENDRAL Program
Machine Intelligence 6, Edinburgh University Press,
1971
- Fis.79a W.E.FISCHER
The interface between CAD/CAM-software and CODASYL-data
base management systems
EUROGRAPHICS'79, Bologna, october 1979
- Fis.79b W.E.FISCHER
PHIDAS - a database management system for CAD/CAM
application software
CAD, Vol.11, No.3, may 1979
- PLR.79 O.FOUCAUT, D.LAMY, C.ROLLAND
Proposition pour une conception et une réalisation
structurée des traitements d'un SI
Informatique et Gestion No.104, mars 1979
- For.78 J.M.FORESTIER
Machine relationnelle pour système segmenté
Thèse de 3ème cycle, Grenoble, septembre 1978
- For.78 A.R.FORREST
A Unified Approach to Geometric Modelling
SIGGRAPH'78
- GaH.75 J.D.GANNON, J.J.HORNING
Language Design for Programming Reliability
IEEE on Software Engineering, Vol. SE-1, no.2, June
1975

- Gai.76 M.GAILLARD
SOMINE: Gestion des mémoires; Enseignement assistée par ordinateur
Thèse de 3ème cycle, INPG, Ecole des Mines de St.Etienne, jan. 1976
- Ged.75 M.GEDIN
Méthodes de conduite des projets informatiques
Les Editions d'Organisation, Paris, 1975
- GeJ.77 J.S.GERO, W.G.JULIAN
High Level Non-Graphical Interaction In Computer-Aided Design
in: J.J.ALLAN (Ed.), CAD Systems, North Holland, 1977
- GeM.75 C.M.GESCHKE and J.G.MITCHELL
On the Problem of Uniform References to Data Structures
IEEE Trans. on Software engineering, june 1975
- Gir.77 J.P. GIRAUDIN
Le projet M.A.C.S.I.-P. Methode d'aide à la conception des systemes d'informations orientee vers la fabrication de prototypes d'applications informatiques de gestion
Thèse de doctorat de 3ème cycle, Grenoble, juillet 1977
- GKB.76 H.GALLAIRE, P.KALFON, J.le BIHAN, J.SEGUIN, J.M.VILLARD
Bases de données réparties
IRIA - SESORI, Lans en Vercors, mars 1976
- GMS.76 M.GAILLARD, P.MARTY, C.SAYETTAT
Contribution à la conception , la réalisation et l'utilisation du système de bases de données SOMINE
Thèses de 3ème cycle, INPG, Ecole des Mines de Saint-Etienne, janvier 1976
- Gre.77 D.P.GREENBERG
An Interdisciplinary Laboratory for Graphics research and Applications
SIGGRAPH'77
- GrE.79 H.GRABOWSKI and M.EIGNER
Semantic datamodel requirements and realization with a relational datastructure
CAD, Vol.11, No.3, may 1979
- Gri.69 M.GRIFFITHS
Analyse déterministe et compilateurs
Thèse d'Etat, IMAG, 1969
- Gro.76 R.W.GROSSMAN
Some data base applications of constraint expressions

- Technical Report No.156, Laboratory. for computer Science, MIT, february 1976
- HaS.75 H.HASSELMEIER and W.G.SPRUTH (Ed.)
Data Base Systems, Proceedings, 5th informatik symposium
Lecture Notes in Computer Science No.39, Springer-Verlag, september 1975
- Haw.75 D.HAWKES (Ed.)
Models and Systems in Architecture and Building
LUBFS Conference Proceedings Number 2, The Construction Press Ltd, 1975
- HaZ.76 M.HAMILTON, S.ZELDIN
Higher Order Software - A Methodology for Defining Software
IEEE Trans. on Software Engineering, Vol. SE-2, no.1, March 1976
- HNS.74 S.HATVANY, W.M.NEWMAN, M.N.SABIN
World Survey of Computer-Aided Design
IIASA Report, 1974
- Hos.79 E.M. HOSKINS
Descriptive databases in some design manufacturing environments
CAD, Vol.11, No.3, may 1979
- Hub.79 G.U.HUBBARD
Computer-assisted logical database design
CAD, Vol.11, No.3, may 1979
- HuE.76 M.HUSTIN, J.P.ECKLY
Les banques de données geometriques
Convention Informatique, septembre 1976
- HuK.77 D.S.HUMCKE, D.P.KENT
Ergonomics on a Large Interactive Graphics Operation
Computer-Aided Design, Vol.9, no.4, oct. 1977
- IGD.77 Dossier
Méthodes nouvelles de conception des systèmes d'information
Informatique et Gestion , No.85, mars 1977
- IRI.76a IRIA
La recherche en informatique des organisations
colloque INFORSID, CAEN, novembre 1976
- IRI.76b IRIA
Séminaire sur le langage d'analyse et de conception des

systèmes d'information automatisés
Cahier INFORSID No.1, Nancy, mai 1976

- IRI.77 IRIA
Modèles, Méthodes et Outils pour la réalisation des
systèmes d'information automatisés
Séminaire INFORSID, Groupe II, Lyon, mai 1977
- IRI.78 IRIA
La dynamique dans les systèmes d'information
Séminaire INFORSID groupe II, Cergy Pontoise, octobre
1978
- ISD.75 Problem Statement Language (PSL)
User's Manual
ISDOS Project, March 1975
- JoL.78 A.K.JONES, B.H.LISKOV
A Language Extension for Expressing Constraints on Data
Access
C ACM, Vol.21, No.5, may 1978
- Jor.77 Ph.JORRAND
Very High Level Languages: Some Aspects of the
Evolution of Language Design
International Computing Symposium 1977
- JRV.74 R.JACQUART, P.REGNIER, F.R.VALETTE
Système GERMINAL
Rapport CERT-DERI, Août 1974
- KLM.75 S.KRAKOWIAK, M.LUCAS, J.MONTUELLE, J.MOSSIERE
A Modular Approach to the Structured Design of
Operating Systems
Rapport de Recherche no.3, Grenoble, Mai 1975
- KuK.79 T.L.KUNII and H.S.KUNII
Architecture of a virtual graphic database system for
interactive CAD
CAD, Vol.11, No.3, may 1979
- Laf.77 G.LAFUE
Outline of a multi-purpose relational spatial data base
Institute of Physical Planning, Research Report No.69,
October 1977, and Proceedings CAD'78
- Laf.79 G.M.E. LAFUE
Integrating language and database for CAD applications
CAD, Vol.11, No.3, may 1979
- Lam.77 D.LAMY
Les traitements: leurs structures et leur génération

dans le projet REMORA
Thèse de Doctorat de 3ème cycle, Nancy, Mai 1977

- Lan.74 C.A.LANG
Achievements in Computer-Aided Design
Proc. IFIP Congress 1974
- Lat.77 J.C.LATOMBE
Une application de l'intelligence artificielle à la
Conception Assistée par Ordinateur (TROPIC)
Thèse d'Etat, Grenoble, nov. 1977
- Lat.78 J.C.LATOMBE (Ed.)
Artificial Intelligence and Pattern Recognition in CAD
North-Holland, 1978
- LeR.77 C.LE FAOU, J.C.REYNAUD
Functional Simulation of Complex Electronic Circuits
ESSCIRC 77 Conference, Sept. 1977
- LeS.77 K.LEINEMANN, E.G.SCHLECHTENDAHL
The REGENT System for CAD
in: J.J.ALLAN (Ed.): CAD Systems, North-Holland
Publishing Company, 1977
- Lin.78 C.A.LINDEN
Grammars which Describe Large Bodies of Data
Computer-Aided Design, vol.10, no.1, jan. 1978
- Lin.79 R.LINDNER
Aspects of Interactive Computer-Graphics
in Nouvelles tendances de la communication
homme-machine, IRIA, avril 1979
- Lis.72 B.H.LISKOV
A Design Methodology for Reliable Software Systems
FJCC, AFIPS Conf. Proc. Vol.41, 1972
- Lis.76 B.LISKOV
An Introduction to CLU
MIT, Computer Structures Group Memo 136, Fev. 1976
- LiZ.74 B.LISKOV, S.ZILLES
Programming with Abstract Data Types
Proceedings of a Symposium on Very High Level
Languages, SIGPLAN, Vol.9, No.4, April 1974
- LiZ.75 B.H.LISKOV, S.N.ZILLES
Specification Techniques for Data Abstractions
IEEE on Software Engineering, Vol. SE-1, no.1, March
1975

- Lle.78 A.LLEWELYN
CAD study for the commission of the European Communities
1978
- LLM.78 A.LUDUC-LEBALLEUR, M.LUCAS, F.MARTINEZ
Conception et réalisation d'un logiciel graphique interactif indépendant du contexte d'utilisation: le logiciel de base GRIGRI
RAIRO Computer Science, Vol.12, No.2, pp. 147-173, 1978
- LMW.79 P.C.LOCKEMANN, H.C.MAYR, W.H.WEIL, W.H.WOHLLEBER
Data Abstractions for Database Systems
ACM Transactions on Data Systems, Vol.4, No.1, march 1979
- Lon.77 W.J.LONG
A Program Writer
MIT, LCS TR-187
- LoN.79 R.A.LORIE, J.F.NILSSON
An Access Specification Language for a Relational Data Base System
IBM J.RES. DEVELOP., Vol.23, No.3, may 1979
- Low.78 J.R.LOW
Automatic Data Structure Selection: An Example and Overview
C ACM, Vol.21, No.5, may 1978
- Mar.76 P.MARTY
Contribution à la conception, la réalisation et l'utilisation du système de bases de données SOMINE: Structuration des informations
Thèse de 3ème cycle, INPG, Ecole des Mines de St.Etienne, jan. 1976
- Mar.79 J.MARTIN
Design of Man-Computer Dialogues
Prentice-Hall 1979
- Mas.76 P.MASSE
Le Système CASCADE
Thèse de Docteur-Ingénieur, Grenoble, nov.1976
- MaT.79 J.MADDEN and V.T.TAYLOR
Design data requirements in the process industries
CAD, Vol.11, No.3, may 1979
- Me1.78 J.MELESE
Hiérarchie de systèmes et niveaux de langages
Informatique et Gestion No.96, mai 1978

- Mer.73 J.MERMET
Etude méthodologique de la conception assistée par ordinateur des systèmes logiques: CASSANDRE
Thèse d'Etat ès Sciences, Grenoble, mars 1973
- MIC.76 Séminaire MICADO
sur la Méthodologie de la Conception,
St.Pierre de Chartreuse, nov. 1976
- Mic.79 A.F.MICADO
Séminaire sur la formation à la CAO
St.Pierre de Chartreuse, 12-15 nov., 1979
- MiO.78 W.MITCHELL, M.OLIVERSON
Computer Representation of Three-Dimensional Structures
for CAEADS
Construction engineering research laboratory, Technical
Report p-86, february 1978
- MiO.78 M.J.MITCHELL, M.OLIVERSON
Computer Representation of Three-Dimensional Structures
for CAEADS
TRP-86, Construction Engineering Research Laboratory,
1978
- Mis.78 J.MISRA
An Approach to Formal Definitions and Proofs of
Programming Principles
IEEE Trans. on Software Engineering, Vol. SE-4, no.5,
Sept. 1978
- MMS.79 A.C.MASSABO, Y.MOAL, J.STARK
SYSTRID 1
Proceedings EUROGRAPHICS 79, Bologne, Oct.1979
- MoL.76 MORVAN, LUCAS
Images et Ordinateur, Introduction à l'Infographie
interactive
Larousse 1976
- Mos.77 J.MOSSIERE
Méthode pour l'écriture des systèmes d'exploitation
Thèse d'Etat, Grenoble, Sept. 1977
- NCC.78 A Panel Session;
User Experience With New Software Methods
Proceedings on National Computer Conference, 1978
- NeS.73 NEWMAN, SPROULL
Principles of Interactive Computer Graphics
Mc Graw-Hill, 1973

- Par.72a D.L.PARNAS
A Technique for Software Module Specification with
Examples
CACM, Vol.15, no.5, May 1972
- Par.72b D.L.PARNAS
On the Criteria to be used in Decomposing Systems into
Modules
CACM, Vol.15, no.12, Dec. 1972
- Par.77 R.PARENT
A System for Sculpting 3-D Data
SIGGRAPH'77
- Pea.78 J.M.PEAUCELLE
Le modèle objet-événements
Informatique et Gestion No.93
- Pec.75 F.PECCOUD
M.A.C.S.I.: Méthode d'Aide à la Conception des Systèmes
d'Informations
Thèse d'Etat es-sciences, Grenoble juin 1975
- Pel.76 C.PELLEGRINI
Un système interactif d'aide à la conception de
programmes
Thèse, Université de Genève, 1976
- Per.76 W.D.PEREA-VILLACORTA
Méthode d'analyse dans le projet REMORA
Thèse de Doctorat de 3ème cycle, Nancy, Juin 1976
- PTW.74 Proceedings of the International Workshop held in
Namur, Belgium on
Data structure models for information systems
Presses universitaires de Namur, mai 1974
- Pur.80 P.PURCELL
Computer Education in Architecture
CAD, Vol.12, No.5, pp.239-251, sept. 1980
- RAB.79 P.ROMAND, ALEONARD, BENEVISE, LE FRANC, B.DAVID
La C.A.O.intégrée pour la production d'équipements
électroniques
ADEPA - Journées Scientifiques et Techniques de la
Production Automatisée, Nancy, Juin 1979
- RaH.78 C.V.RAMAMOORTHY, G.S.HO
A Design Methodology for User Oriented Computer Systems
National Computer Conference 1978, AFIPS Vol.47

- Ree.78 W.T.REEVES
A Device-Independant General Purpose Graphic System in
a Minicomputer Time-sharing Environment
Technical Report CSRG-93, 1978
- ReE.80 P.L.READ, P.ELIAS
Impact des systèmes CFAO sur la productivité des
petites et moyennes industries
MICAD 80
- RFR.79 C.ROLLAND, O.FOUCAUT, C.RICHARD, O.THIERY
Information System Design and Computer Aided Design
EURO-IFIP 79, North Holland, 1979
- RGI.75 D.T.ROSS, J.B.GOODENOUGH, C.A.IRVINE
Software Engineering: process, principles and goals
COMPUTER, may 1975
- Rie.XX R.F.RIESENFELD
Current trends in Computer-Graphics
Computer and Graphics, Vol.3, pp.115-122
- Riv.73 V.RIVERO
Etude méthodologique de l'aide à la conception dans les
projets architecturaux
Rapport du D.E.A., USMG-ENSIMAG, Grenoble, Sept.1973
- Riv.75 V.RIVERO
Conception Architecturale Assistée par Ordinateur:
Présentation du Système SIGMA-ARCHI
Exposée pour l'Ecole d'été d'Informatique de l'AF CET,
Rabat(Maroc), juillet 1975
- Riv.77 V.RIVERO
Une contribution à la Conception Architecturale
Assistée par Ordinateur: le système SIGMA-ARCHI
Thèse de Docteur-Ingénieur, ENSIMAG, Grenoble, Juin
1977
- RoA.76 D.F.ROGERS, J.A.ADAMS
Mathematical Elements for Computer Graphics
Mc Graw-Hill, 1976
- Rom.77 G.C.ROMAN
An Argument in Favor of Mechanized Software production
IEEE Trans. on Software Engineering, Vol. SE-3, no.6,
Nov. 1977
- Roo.67 D.ROOS
ICES System Design
MIT Press, 1967

- RoS.77 D.T.ROSS, K.E.SCHOMAN
Structured Analysis for Requirements Definition
IEEE Trans. on Software Engineering, Vol. SE-3, no.1,
1977
- Ros.77 D.T.ROSS
Structured Analysis (SA): A Language for Communication
Ideas
IEEE Trans. on Software Engineering, Vol. SE-3, no.1,
1977
- Rou.79 N.ROUSSOPOULOS
Tools for designing conceptual schemata of databases
CAD, Vol.11, No.3, may 1979
- Rut.76 G.RUTH
Automatic Design of Data Processing Systems
3rd ACM Symposium on Principles of Programming
Languages, 1976
- Rut.77 F.R.RUTH
Analysis of Algorithm Implementations
MIT, MAC TR-130
- SAD.76 An Introduction to SADT:
Structured Analysis and Design Technique Software
SoftTech Inc., Waltham, MA, doc no.9022-78R, Nov. 1976
- Say.76 C.SAYETTAT
SOMINE: Les accès aux bases; CAO
Thèse de 3ème cycle, INPG, Ecole des Mines de
St.Etienne, jan. 1976
- Sch.78 J.W.SCHMIDT
On the implementation of relations: a key to efficiency
Technical Report CSRG -89, University of Toronto,
january 1978
- SEC.76 Proceedings of the
Second International Software Engineering Conference
San Francisco, October 1976
- Seg.78 J.SEGUIN
Traitement distribué d'informations réparties dans les
réseaux d'ordinateurs
Thèse d'Etat, Grenoble, Mars 1978
- Sho.76 E.SHORTLIPPE
Computer-Based Medical Consultations: MYCIN
American Elsevier, 1976

- Sim.75 H.A.SIMON
Style in Design
in Spatial Synthesis in Computer-aided Building Design,
C.M.Eastman (ed.), Applied Science Publishers, 1975
- SMC.74 W.P.STEVENS, G.J.MYERS, L.L.CONSTANTINE
Structured Design
IBM System Journal, no.2, 1974
- SmS.77 J.M.SMITH and D.C.SMITH
Data Base Abstractions: Agregation
C ACM, Vol.20, No.6, june 1977
- Spi.74 W.R.SPILLERS (Ed.)
Basic Questions of Design Theory
North-Holland, 1974
- Tar.77 H.TARDIEU
Base de données: Une méthode et un outil d'aide à leur
conception
Informatique et Gestion No.88
- TeH.78 D.TEICHROEW, E.A.HERSHEY
Computer-Aided Structured Documentation and Analysis of
Information Processing System Requirements
SHARE XLVII, Montréal, August 1978
- Ter.78 CH.J.TERMAN
The Specification of Code Generation Algorithms
MIT, LCS TR-199
- TeW.76 D.TEICHROEW, E.WINTERS
Recent Developments in System Analysis and Design
Atlanta Economic Review, Nov.-Dec. 1976
- Thi.76 O.THIERY
L'aide à la conception dans le projet REMORA
Thèse de Doctorat de 3ème cycle, Nancy, Sept, 1976
- TNP.79 H.TARDIEU, D.NANCI, D.PASCOT
Conception d'un système d'information: construction de
la base de données
les Editions d'organisation, 1979
- TRH.73 D.TEICHROEW, W.J.RATAJ, E.A.HERSCHEY
An Introduction to Computer Aided Documentation of User
Requirements for Computer Based Information Processing
Systems
ISDOS Working Paper no.72, 1973

- Tsu.79 M.TSUBAKI
Family generation in integrated engineering systems
CAD, Vol.11, No.3, may 1979
- VCL.75 W.M.VAN CLEEMPUT J.G.LINDERS (Ed.)
Proceedings of the workshop on data bases for
interactive design
held at Waterloo, Canada, september 1975
- Ven.XX P.VENIER
Système CAO comme un système distribué
Thèse de Docteur-Ingénieur, Grenoble, en cours
- Vil.80 R.VILLAMAYOR
Définition formelle d'un générateur de solutions
d'allocation spatiale en deux dimensions. Application à
l'urbanisme.
Thèse de doctorat de spécialité (3e cycle), Grenoble,
juillet 1980
- VIZ.77 R.VILLAMAYOR, F.Z'GRAGGEN
AGGLO: automatique générative de groupement de
logements
GAMSAU Bulletin, Vol.3, no 4, Février 1977
- VlW.73 J.WLIETSTRA, R.F.WIELINGA (Ed.)
Computer-Aided Design
Proc. IFIP, Working Conference on Principles of CAD
- VoR.78 H.VOELCKER, A.REQUICKA and all
The PADL-1.0/2 System for Defining and displaying solid
objects
SIGGRAPH'78
- War.77 E.A.WARMAN
The Possibility for Automatic Production of Command
Languages
in: J.J.ALLAN (Ed.), CAD Systems, North Holland, 1977
- WCE.72 N.E.WISEMAN, C.J.CHENEY, M.ETHERION, J.O.HILES, U.LEMKE
RAINBOW : a Multi-Purpose CAD System
Software Practice and Experience, Vol.2, pp.359-375,
1972
- Whi.74 D.E.WJITNEY, M.K.MILLEY
CADSYS: a New Approach to Computer-Aided Design
IEEE Transactions on Systems, Man and Cybernetic,
Jan.1974
- Wil.74 R.WILLIAMS
On the Application of Relational Data Structures in
Computer Graphics

North-Holland publishing company, Proceedings IFIP Congress 74

- Wir.77 N.WIRTH
MODULA
Software - Practice and Experience, Vol.7, 1977
- WLS.76 Wm.A.WULF, R.L.LONDON, M.SHAW
Abstraction and Verification in ALPHARD
Carnegie-Mellon University, March 1976
- WoM.77 H.K.T.WONG and J.MYLOPOULOS
Two views of data semantics: a survey of data models in
artificial intelligence and database management
INFOR, Vol 15, No.3, october 1977
- Yao.78 S.BING YAO (Ed.)
Proceedings on forth international conference on very
large data bases
West-Berlin, september 1978
- Yas.80 Y.YASKY
Transforming a set of building drawings into a
consistent database
IBS Reserch Report No.2, january 1980
- Z'G.74a F.Z'GRAGGEN
Vers une conception scientifique du logement:
Production systématique de plans masses à l'aide de
l'ordinateur
Dept.d'Architecture, Ecole Polytechnique Fédérale de
Lausanne, 1974
- Z'G.74b F.Z'GRAGGEN
Production de plans-masses de quartiers de logements à
l'aide d'un ordinateur
HABITATION, no 11, Nov 1974
- Z'g.80 F.J.Z'GRAGGEN
SIGMA-Industrialisation Ouverte: La CAO en Architecture
utilisant des systèmes constructifs
Thèse de Doctorat ès Sciences Techniques, Département
d'Architecture, Ecole Polytechnique Fédérale de
LAUSANNE, Septembre 1980
- ZaL.74 P.ZANELLI, J.LEBET
Vers une conception scientifique du logement:
production systématique de groupements à l'aide de
l'ordinateur
Dept.d'Architecture, Ecole Polytechnique Fédérale de
Lausanne, 1974

- Zan.76 F.ZANELLI
La Conception Architecturale Assistée par Ordinateur:
Présentation d'un système
WERK, revue suisse d'Architecture, Nov.1976
- Zan.79 F.ZANELLI
Une contribution à la Conception Architecturale
Assistée par Ordinateur: le système SIGMA-LOGEMENT
Thèse de Doctorat ès Sciences Techniques, Département
d'Architecture, Ecole Polytechnique Fédérale de
LAUSANNE, Mars 1979

INDEX BIBLIOGRAPHIQUE

INDEX BIBLIOGRAPHIQUE

- BASES DE DONNEES: AdD.76, ArH.76, ArW.76, Ath.78, BaB.75, Bar.80, Bat.75, Ber.78, BeR.79, Bou.77, BuD.79, CAM.78, Chu.78, dCh.75, EaH.77, Eas.76, Eas.80a, Eas.80b, EaT.79, ELS.76, Fis.79a, Fis.79b, FLR.79, For.78, Gai.76, GeM.75, Gir.77, GKB.76, GMS.76, GrE.79, Gro.76, HaS.75, Haw.75, Hos.79, Hub.79, HuE.76, IGD.77, IRI.76a, IRI.76b, IRI.77, IRI.78, JoL.78, KuK.79, Laf.77, Laf.79, LMW.79, LoN.79, Low.78, Mar.76, MaT.79, Mel.78, MiO.78, Pea.78, Pec.75, PTW.74, RFR.79, Rou.79, Say.76, Sch.78, Seg.78, SmS.77, Tar.77, Tsu.79, VCL.75, Wil.74, WoM.77, Yao.78, Yas.80,
- SYSTEMES DE BASE: BDL.76, BeN.71, Dav.75, Dav.77, DaV.78, Dav.80d, Dav.80e, DaV.82, DaV.XX, Dub.80, DVV.79, ReE.80, Ven.XX,
- PROCESSUS DE CONCEPTION: MIC.76, Riv.77, Sim.75, Spi.74,
- PROGRAMMATION: ACM.74, Bal.73, CAC.77, DaV.77, DDH.72, DeG.80, DeK.76, Del.79, GaH.75, GeM.75, HaZ.76, JoL.78, Jor.77, Lis.76, LiZ.74, Lon.77, Low.78, Pel.76, RGI.75, Rut.76, SAD.76, SEC.76, SMC.74, WLS.76,
- METHODOLOGIE: Alf.77, BBD.77, BeR.79, BLR.74, CER.74, Chu.75, Dal.77, DaV.77, dCh.75, DDH.72, Der.74, Eas.76, FBL.71, Ged.75, Gir.77, IGD.77, Jor.77, JRV.74, KLM.75, Lam.77, Lat.77, LeS.77, Lis.72, Mas.76, Mis.78, Mos.77, Par.72a, Par.72b, Pec.75, Per.76, RaH.78, RFR.79, RGI.75, Rom.77, Roo.67, Rut.76, Rut.77, SEC.76, Sho.76, Tar.77, Thi.76, TNP.79, VIW.73, WCE.72, Whi.74, Wir.77,
- SPECIFICATIONS: Abr.78, ACh.75, BaH.75, BGW.78, Cap.78, Dav.81a, HaZ.76, ISD.75, LiZ.75, NCC.78, RoS.77, Ros.77, SAD.76, TeH.78, Ter.78, TeW.76, TRH.73, War.77,
- TYPES ABSTRAITS: CAC.77, Jor.77, LiZ.74, RGI.75, Rut.76,
- GRAPHIQUE: BEH.79, Bra.75, Bra.78, Bra.78, Bru.78, Col.78, EaH.77, ELS.76, For.78, Gre.77, HuE.76, HuK.77, KuK.79, Lin.78, Lin.79, LLM.78, MiO.78, MiO.78, MMS.79, MoL.76, NeS.73, Par.77, Ree.78, Rie.XX, RoA.76, Tsu.79, VoR.78, Wil.74,

ENSEIGNEMENT DE LA CAO: ACL.78, All.78, Dav.79, Dav.81,
DZZ.77, DZZ.79, Enc.80, Lle.78, Mic.79, Pur.80,

CAO EN ARCHITECTURE: BRZ.75, DaR.74a, DaR.74b, Dav.73, Dav.74,
Dav.75, Dav.80a, Dav.80b, DaZ.79, DLR.76a, DLR.76b,
DRV.75, DRV.76a, DRV.76b, DRV.77, DRZ.75, DRZ.75,
DZZ.77, DZZ.79, Equ.75, Equ.76a, Equ.76b, Haw.75,
MiO.78, MiO.78, Riv.73, Riv.75, Riv.77, Vil.80, ViZ.77,
Z'G.74a, Z'G.74b, Z'g.80, ZaL.74, Zan.76, Zan.79,

APPLICATION CAO: BDF.75, CEC.78, CZT.79, CZT.80, Dav.80c,
Dav.81a, DGM.80, RAB.79,

INTERACTION HOMME-MACHINE: Ber.79, Bo..77, GeJ.77, Mar.79,
War.77,

LANGAGES DE DESCRIPTION: BBD.80, BGL.71, BoG.79, Bor.75,
LeR.77, Mer.73,

RESEAUX: Chu.78, Dan.78, DaS.77, Seg.78,

PUBLICATIONS DE L'EQUIPE: DRZ.75, Dub.80, Equ.75, Equ.76a,
Equ.76b, Riv.73, Riv.75, Riv.77, Ven.XX, Vil.80,
ViZ.77, Z'G.74a, Z'G.74b, Z'g.80, ZaL.74, Zan.76,
Zan.79,

DAVID: BDF.75, BDL.76, BRZ.75, DaR.74a, DaR.74b, Dav.73,
Dav.74, Dav.75, Dav.77, DaV.78, Dav.79, Dav.80a,
Dav.80b, Dav.80c, Dav.80d, Dav.80e, Dav.81, DaV.82,
DaV.XX, DaZ.79, DGM.80, DLR.76a, DLR.76b, DRV.75,
DRV.76a, DRV.76b, DRV.77, DRZ.75, DRZ.75, DVV.79,
DZZ.77, DZZ.79, RAB.79,

THESES: Dav.75, Riv.77, Ven.XX, Vil.80, Z'g.80, Zan.79,

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 5 de l'arrêté du 16 avril 1974,

VU les rapports de Messieurs

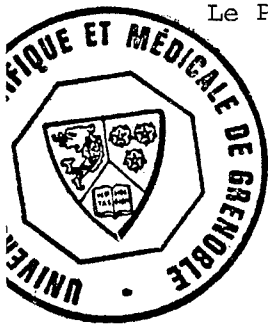
- . Jean MERMET, Maître de Recherche au CNRS Grenoble
- . Jean-Marc BRUN, Maître de Recherche au CNRS, LIMSI Paris
- . Michel LUCAS, Professeur à l'Université de Nantes,

Monsieur Bertrand D A V I D


est autorisé à présenter une thèse en soutenance pour l'obtention du grade de DOCTEUR d'ETAT, discipline "SCIENCES".-

Fait à Grenoble, le 22 Septembre 1981

Le Président de l'U.S.M.-G.



Le Président de l'I.N.P.-G.


D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble