



HAL
open science

Intégration de l'activité de conception dans l'enseignement assisté par ordinateur : le système C.A.S.C.A.D.E

Philippe Massé

► **To cite this version:**

Philippe Massé. Intégration de l'activité de conception dans l'enseignement assisté par ordinateur : le système C.A.S.C.A.D.E. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1976. Français. NNT : . tel-00287074

HAL Id: tel-00287074

<https://theses.hal.science/tel-00287074>

Submitted on 10 Jun 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.

N° C.N.R.S. A.O. 13 027

THÈSE

présentée à

L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

pour obtenir le titre de

DOCTEUR INGÉNIEUR

PAR

PHILIPPE MASSÉ

Ingénieur I.N.P.G.

TITRE

INTEGRATION DE L'ACTIVITE DE CONCEPTION DANS
L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR :
LE SYSTEME C.A.S.C.A.D.E.

Soutenue le 24 septembre 1976 devant la Commission d'Examen

J U R Y

Monsieur Ph. TRAYNARD

Président

Messieurs C. FOULARD

R. JACOUD

J.C. SABONNADIÈRE

G. VEILLON

} Examineurs

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M. Philippe TRAYNARD
Vice-Président : M. Pierre-Jean LAURENT

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BLOCH Daniel	Physique du solide
BONNETAIN Lucien	Chimie Minérale
BONNIER Etienne	Electrochimie et Electrometallurgie
BOUDOURIS Georges	Radioélectricité
BRISSONNEAU Pierre	Physique du solide
BUYLE-BODIN Maurice	Electronique
COUMES André	Radioélectricité
DURAND Francis	Métallurgie
FELICI Noël	Electrostatique
FOULARD Claude	Automatique
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie-Physique
PAUTHENET René	Physique du solide
PERRET René	Servomécanismes
POLOUJADOFF Michel	Electrotechnique
SILBER Robert	Mécanique des Fluides

PROFESSEUR ASSOCIE

M. ROUXEL Roland Automatique

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuel	Electronique
BOUVARD Maurice	Génie Mécanique
COHEN Joseph	Electrotechnique
LACQUME Jean-Louis	Géophysique
LANCIA Roland	Electronique
ROBERT François	Analyse numérique
VEILLON Gérard	Informatique Fondamentale et appliquée
ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM. ANCEAU François	Mathématiques Appliquées
CHARTIER Germain	Electronique
GUYOT Pierre	Chimie Minérale
IVANES Marcel	Electrotechnique
JOUBERT Jean-Claude	Physique du solide
MORET Roger	Electrotechnique Nucléaire
PIERRARD Jean-Marie	Mécanique
SABONNADIÈRE Jean-Claude	Informatique Fondamentale et appliquée
Mme SAUCIER Gabrièle	Informatique Fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M. LANDAU Ioan Automatique

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

MM. FRUCHART Robert

Directeur de Recherche

ANSARA Ibrahim

Maître de Recherche

CARRE René

Maître de Recherche

DRIOLE Jean

Maître de Recherche

MATHIEU Jean-Claude

Maître de Recherche

MUNIER Jacques

Maître de Recherche

Je tiens à exprimer mes remerciements très sincères à :

Monsieur le Professeur TRAYNARD, Président de l'Institut National Polytechnique de Grenoble qui a bien voulu me faire l'honneur de présider le Jury de cette Thèse ;

Monsieur le Professeur JACOUD, Directeur du Laboratoire Ordinateur pour Etudiants de l'Université de Paris VII ;

Monsieur le Professeur FOULARD du Laboratoire d'Automatique de l'Institut National Polytechnique de Grenoble, et

Monsieur le Professeur VEILLON du Laboratoire d'Informatique et de Mathématiques Appliquées de l'Institut Polytechnique de Grenoble,

pour l'intérêt qu'ils ont manifesté envers ce travail en acceptant d'honorer le Jury de leur présence.

Monsieur SABONNADIÈRE, Maître de Conférences à l'Institut National Polytechnique de Grenoble, m'a proposé ce sujet passionnant. Par ses conseils éclairés, il m'a constamment guidé au cours de cette étude sans ménager sa peine. Qu'il soit assuré de ma profonde amitié.

Je présente mes plus vifs remerciements :

- à Monsieur PERRET, Maître Assistant Agrégé à l'Institut National Polytechnique qui m'a amicalement éclairé de ses compétences de chercheur et d'enseignant dans l'étude de l'application de mes travaux à l'enseignement de la conception de Machines Asynchrones,

- aux membres du Laboratoire d'Automatique de Grenoble pour la collaboration très précieuse qui s'est instaurée lors de la définition d'un système d'enseignement selon notre méthodologie appliquée à l'automatique.

Je remercie chaleureusement tous mes camarades et amis du Laboratoire d'Electrotechnique, chercheurs, enseignants, ingénieurs, techniciens et en particulier :

.../...

Messieurs BOLOPION, CAMPMAS et LATOMBE, pour les nombreux travaux que nous avons réalisés ensemble dans le cadre d'une équipe de recherche sympathique et unie.

Monsieur BEVIER, Ingénieur responsable de notre centre de calcul qui m'a apporté un soutien quotidien lors de la réalisation du prototype.

Je tiens à remercier tout spécialement Mademoiselle CAYLA pour la gentillesse et le soin avec lesquels elle a bien voulu réaliser l'impression de ce mémoire.

S O M M A I R E

	Pages
INTRODUCTION.....	1
CHAPITRE I	
1. CAO EAO ECAO.....	5
2. Les buts de l'ECAO.....	8
2.1. Le contenu de l'ECAO les choix fondamentaux.....	9
2.2. Caractéristiques d'une pédagogie associée à l'ECAO.....	11
2.3. Le rôle de l'enseignant.....	11
3. Les contraintes d'un système d'ECAO.....	13
3.1. Les contraintes dues au mode d'utilisation.....	13
3.2. Les contraintes dues à l'élaboration du sys- tème par l'enseignant.....	14
4. L'environnement d'un système d'ECAO.....	16
4.1. La discipline à enseigner.....	16
4.2. L'activité de conception.....	16
4.3. Les fondements méthodologiques.....	17
4.4. La stratégie pédagogique.....	18
4.5. L'environnement matériel.....	19
5. La modélisation.....	19
CHAPITRE II	
1. Le modèle d'un système d'ECAO et de son environ- nement.....	22
2. Du travail de l'étudiant au modèle potentiel des modèles potentiels.....	24
2.1. Le travail de l'étudiant.....	24
2.2. Le modèle potentiel.....	25
2.3. Le modèle potentiel des modèles potentiels.....	26
2.3.1. Entité.....	28
2.3.2. Liaison.....	31
2.3.3. Exemples de modèles potentiels.....	34
2.3.3.1. Cas d'un procédé commandé.....	34
2.3.3.2. Cas de l'identification d'un procédé..	34
2.3.3.3. Exemple de la machine asynchrone.....	37
2.3.3.4. Exemple du transformateur.....	37
2.3.4. Exemple d'un modèle intégrant conception et gestion pédagogique.....	37

	Pages
3. Les ressources de simulation.....	43
4. L'interface entre l'étudiant et le système.....	43
4.1. Le poste de travail.....	43
4.2. Les ressources d'édition.....	44
4.3. Le langage mis à la disposition des étudiants...	44
5. La stratégie pédagogique.....	45
5.1. Décomposition de la stratégie pédagogique.....	46
5.2. Bloc Élémentaire d'Analyse Stratégique.....	49
5.3. Divers types d'enchaînements de BEAS.....	50
5.4. Description de la stratégie pédagogique.....	52
6. Le modèle complet.....	52
CHAPITRE III	
LE SYSTEME CASCADE	
1. Introduction.....	55
2. Les besoins en Informatique de Base.....	56
2.1. Mini-Ordinateur ou ordinateur.....	56
2.1.1. Décentralisation.....	56
2.1.2. Facilité d'accès pour le concepteur.....	57
2.1.3. Souplesse de gestion.....	57
2.1.4. Temps de réponse et pédagogie.....	57
2.1.5. Adaptation des postes de travail.....	58
2.1.6. Capacité mémoire.....	58
2.1.7. Temps de traitement pour les tâches complexes.....	58
2.1.8. Logiciels de base et langages évolués.....	59
2.1.9. Temps partagé ou multi-programmation.....	59
2.2. Les options du système CASCADE.....	60
2.2.1. Portabilité.....	60
2.2.2. Modularité.....	60
2.2.3. Interprétation.....	60
3. Les langages du système CASCADE.....	61
3.1. Terminologie et notations.....	61
4. Le langage MODPHY.....	64
4.1. Syntaxe du langage MODPHY.....	64
4.1.1. Structure des instructions.....	64
4.1.2. Notations.....	65
4.1.3. Les instructions du langage.....	67
4.2. Exemple du moteur asynchrone.....	73
4.3. Exemple d'erreurs détectées par MODPHY.....	77
5. Le langage EDITER.....	83
6. Langage de définition de langage de commande.....	85

	Pages
7. Description de la stratégie pédagogique.....	90
7.1. Syntaxe du langage BLOPRO.....	90
7.1.1. Les variables.....	91
7.1.1.1. Les noms des variables.....	91
7.1.1.2. Les indices des variables.....	93
7.1.1.3. Variable Alphanumérique de Référence..	93
7.1.2. Les expressions arithmétiques.....	93
7.1.3. Les expressions logiques.....	94
7.1.4. Instructions d'affectation simple.....	94
7.1.5. Instruction d'affectation externe.....	95
7.1.6. Instructions de branchement.....	95
7.1.7. Arrêt d'exécution.....	95
7.1.8. Appels de modules de simulation.....	96
7.1.9. Enchaînements de Bloc.....	96
7.1.10. Instruction de Retour de Bloc.....	97
7.1.11. Instructions de Gestion des Ressources d'édition.....	97
7.1.12. Instructions de Gestion du travail de l'étudiant.....	98
7.2. Exemples de stratégies pédagogiques.....	105
7.2.1. Exemple associé à la commande créer.....	105
7.2.2. Exemple associé au calcul du schéma équivalent d'un moteur asynchrone.....	107
7.3. Détection des erreurs et aide à la mise au point.....	116
8. Définition des modules de simulation.....	118
9. Conclusion.....	121
CONCLUSION.....	123
BIBLIOGRAPHIE.....	125

INTRODUCTION

Les premiers essais d'introduction de l'ordinateur au sein du processus pédagogique ont été réalisés le plus souvent à l'initiative d'informaticiens. Derrière l'écran que constitue inévitablement l'utilisation d'une technique nouvelle, le pédagogue découvre dans la plupart de ces systèmes la transposition sur ordinateur des méthodes d'enseignement traditionnelles où le savoir délivré par le maître est véhiculé jusqu'à l'élève par des supports pédagogiques variés. Dans un contexte fortement automatisé, l'utilisation de cette pédagogie descendante s'appuie sur une conception directive de l'apprentissage ("Behaviorisme") et les "systèmes d'enseignement programmés" qui en ont résulté sont orientés vers la transmission d'un savoir fortement structuré préalablement enregistré par l'enseignant.

En réaction contre ce mode d'enseignement directif, un mouvement de recherche, partant d'une conception plus humaniste et personnalisée de l'apprentissage, a dégagé une approche nouvelle qui laisse une part beaucoup plus importante à la créativité, en intégrant l'enseignement dans un processus plus général de formation autodidactique de l'élève.

Dans ce courant de recherche, la réalisation du système ESPACE par J.C. LATOMBE a montré que les techniques informatiques actuelles rendaient possible la modélisation d'objets complexes et la simulation de leur fonctionnement, tout en restant compatible avec les contraintes importantes qu'entraîne un environnement pédagogique rigoureux. L'expérimentation du système ESPACE par A. BOLOPION, a validé l'apport didactique de ce nouveau mode d'enseignement et a ouvert ainsi la voie de l'application de nos options pédagogiques à d'autres disciplines.

Parallèlement, l'implantation de logiciels de base destinés à l'enseignement assisté sur mini-ordinateur, réalisée par M. CAMPMAS, permet au concepteur de s'affranchir des contraintes liées à l'utilisation d'un grand centre de calcul.

Il était alors nécessaire de procurer aux enseignants les moyens de construire leur système d'EA0, sans s'éloigner du domaine pédagogique de leur propre discipline.

Nous avons conduit une étude détaillée de la modélisation d'un système didactique cohérente avec les hypothèses pédagogiques qui ont guidé les travaux de notre équipe. Cette recherche nous a permis de définir des outils d'aide aux professeurs dont la généralité rend désormais réalisable l'extension des principes pédagogiques d'ESPACE à d'autres disciplines, sans que la somme de travail et le temps nécessaire à la construction d'une application ne soit rédhibitoire.

--:--:--:--

Le travail présenté dans les pages suivantes est une approche théorique de la modélisation d'un système didactique (chap. II) dont les options pédagogiques de base sont détaillées au chapitre I.

La réalisation pratique sous la forme d'un système informatique, avec ses composants et ses langages est analysée au chapitre III. Le logiciel correspondant a été développé sur l'ordinateur T 1600 de l'ENSEGP.*

* Ecole Nationale Supérieure d'Electrotechnique et de Génie Physique de Grenoble.

Nous examinerons en conclusion les perspectives de recherche que notre travail peut apporter dans le domaine de l'ECAO et qu'il serait souhaitable de voir développer dans un proche avenir.

CHAPITRE I

=====

LE CONTEXTE PEDAGOGIQUE

I - INTRODUCTION : C A O E A O E C A O

II - LES BUTS DE L ' E C A O

1. Le contenu de l' E C A O
2. Caractéristiques d'une pédagogie associée à l' E C A O
3. l'Enseignant et l' E C A O

III - LES CONTRAINTES DE L ' E C A O

1. Les contraintes dues au mode d'utilisation
2. Les contraintes dues à l'élaboration du système par l'enseignant.

IV - L'ENVIRONNEMENT D'UN SYSTEME D ' E C A O

1. La discipline à enseigner
2. L'activité de conception
3. Les fondements méthodologiques
4. La stratégie pédagogique
5. L'environnement matériel

V - LA MODELISATION

1. INTRODUCTION C A O E A O E C A O

1.1 - La Conception Assistée par Ordinateur a connu un développement important au cours des dix dernières années. Réservé à l'origine à la construction de circuits électroniques, son champ d'application s'est diversifié au point de toucher presque tous les domaines : les arts eux-mêmes n'ont pas échappé à cette généralisation. Cet essor est dû aux récents progrès réalisés dans les disciplines qui constituent les fondements de la CAO [1] : les mathématiques, l'informatique, les sciences de l'ingénieur, le matériel et le logiciel.

- En mathématiques, des outils puissants ont été développés par de nombreux chercheurs, notamment au sein de la théorie des graphes (graphes orientés [2], [3], [4], treillis [5]), et des sous-ensembles flous [6]. L'analyse numérique et l'algorithmique ont fait également de gros progrès : les méthodes fondées sur l'utilisation de différences finies et d'éléments finis permettent actuellement de résoudre des problèmes physiques complexes [7] [8]. Aux algorithmes classiques d'optimisation viennent maintenant s'ajouter les approches heuristiques propres à l'intelligence artificielle.

- La technique du traitement de l'information a suivi un essor tout aussi important : la structure des systèmes informatiques s'est adaptée aux besoins de la conception. Dans ce domaine le système ICES a beaucoup innové [11] en introduisant la notion de ressources informatiques communes et de sous-systèmes indépendants. L'utilisation de banques d'algorithmes par des processus de résolution de problèmes complexes est désormais courante : (Système GERMINAL [13], Système COMRADE [14]). La gestion de banques de données est dorénavant réalisable avec des temps de réponse convenables grâce aux techniques modernes de gestion de la mémoire (pagination, espace virtuel... (Système SOCRATE [15]) et des processus de recherche de l'information (technique de "Hash Coding"...).

- Les sciences de l'ingénieur se sont adaptées au nouvel outil que constitue l'informatique. Des modèles de simulation très détaillés ont été construits dans presque tous les domaines (notamment en électrotechnique où l'on peut remarquer une nouvelle approche de la conception de machines électriques (Système FLUX [16]) et des réseaux de transport d'énergie [17] [18]).

- Le matériel a connu une progression analogue et l'apparition des multiprocesseurs et des microprocesseurs a élargi le potentiel technique dont nous disposons : unités centrales plus rapides, terminaux graphiques interactifs, analyseurs et synthétiseurs de voix... etc. Le logiciel a également connu une grande évolution : les techniques de multiprogrammation et de temps partagé sont aujourd'hui disponibles sur mini-ordinateurs. Des langages évolués tels que ALGOL W, ALGOL 68, LISP... permettent une programmation fine et structurée. Grâce à des interpréteurs ou des compilateurs généraux, le concepteur de système peut définir son propre langage et disposer immédiatement des analyseurs syntaxiques correspondants.

Ce développement de tous les composants de la CAO a été harmonisé par de nombreuses études [19], [20], [21] qui ont réalisé la synthèse des précédents travaux [22] ou exploré de nouvelles voies théoriques et pratiques [23].

1.2 - Dans le même temps, l'Enseignement Assisté par Ordinateur se développait avec dynamisme, en bénéficiant de l'essor de ces mêmes composantes. Aujourd'hui on peut affirmer que l'EAO apparaît dans presque toutes les disciplines. De nombreux systèmes ont dépassé le stade du prototype et sont entrés dans une phase active d'exploitation quelquefois à grande échelle comme les Systèmes PLATO [24] et TICCIT [25] aux U.S.A. et à l'OPE en France [10][12].

Cependant, plus que la taille d'un système, ses options pédagogiques sont déterminantes, car en enseignement beaucoup plus qu'en conception assistée, la psychologie joue un rôle fondamental.

Durant ces dix dernières années on aura pu observer la constitution de deux écoles dont les particularités ne cessent de s'affirmer. Après l'impulsion considérable que lui avait donné SKINNER, l'enseignement programmé a continué à progresser : utilisation d'outils nouveaux comme la cybernétique par FRANK [26] : étude plus détaillée des "modèles d'apprentissage" (LANSKY [27]) modélisation poussée de ces systèmes (QUERE [28]), etc... Mais très rapidement, il est apparu nécessaire de dépasser le stade du simple apprentissage mécanisé de connaissances ponctuelles, pour

accéder à un véritable enseignement [29] qui développe la réflexion, la créativité et la prise de décisions [30]. Des expériences d'EAO non directif ont vu le jour (Système LEGIS [31]) surtout dans des disciplines où la prise de décisions est fondamentale : Sciences de l'ingénieur [32] ou médecine (élaboration de diagnostics [33][34]). Dans chacun de ces cas, l'enseignement est fondé sur l'utilisation de nombreuses simulations et des techniques de résolution de problèmes.

C'est dans le contexte du développement de l'EAO non directif que l'on peut situer les recherches qu'a menées l'équipe "Enseignement et Conception Assisté par Ordinateur", du Laboratoire d'Electrotechnique de l'INPG, depuis 1970. Les travaux réalisés ont tout d'abord cherché à jeter les bases d'un Enseignement Assisté par Ordinateur, fondé sur l'utilisation de la Conception Assistée. Cette recherche a été concrétisée par la constitution du prototype ESPACE ([35] à [38]) qu'une évaluation pédagogique a permis de valider puis de rendre opérationnel par la réalisation d'un système plus élaboré ([39] à [44]). Parallèlement, nous avons étudié la définition d'un logiciel de base, adapté à l'implantation de ce nouvel enseignement sur petit calculateur ([45] [46]). Dans le cadre de ce travail d'équipe, il nous incombait plus particulièrement de développer des outils généraux d'aide à la construction de tels systèmes d'EAO, et de réaliser la synthèse des recherches précédemment conduites dans le domaine de l'électrotechnique pour proposer un modèle général d'un système d'enseignement utilisable dans d'autres disciplines ([47] à [51]).

Nous présenterons dans ce travail une approche globale, à la fois pédagogique et informatique de ce que nous appellerons : "l'Enseignement par la Conception Assisté par Ordinateur".

1.3 - L'Enseignement par la Conception Assistée par Ordinateur : ECAO

Le développement actuel de la CAO justifierait à lui seul l'intégration à courte échéance de cette discipline dans l'enseignement. Les expériences américaine [55] et anglaise [56] nous y incitent largement.

Les cadres dont le monde scientifique aura besoin demain, doivent être formés aux méthodes nouvelles de la conception. Mais la CAO est-elle seulement une technique ou un savoir faire? Non, car son domaine déborde

en fait rapidement le cadre étroitement spécialisé d'une technique moderne. En effet, la science des modèles et des systèmes, les méthodes élaborées d'investigation, d'analyse, de résolution de problèmes constituent le milieu théorique au sein duquel l'utilisateur de la CAO doit travailler.

Les aptitudes requises par cette nouvelle science, prise de décisions, capacité de synthèse et d'analyse selon des critères dynamiques, densité de réflexion intense à un rythme élevé imposé par la machine, nécessité de maintenir une attitude critique vis-à-vis d'une technique dans laquelle les généralisations doivent être proscrites..., sont les principales qualités nécessaires à l'ingénieur ou au chercheur.

Pour le pédagogue, la CAO représente alors un terrain de prédilection qui jusqu'à ces derniers temps est resté pratiquement inexploité. L'Enseignement au moyen de la Conception Assistée par Ordinateur (ECAO) nous offre un continuum pédagogique qui s'étend de l'acquisition de connaissances ponctuelles jusqu'à une véritable épistémologie de la modélisation. La science des modèles, qui répond pleinement à l'accroissement continu des connaissances, trouve dans l'ECAO un domaine concret d'application. Pour l'enseignant, c'est l'un des rares outils qui lui permette de répercuter au niveau des exercices pratiques l'orientation délibérément structuraliste prise par l'enseignement des matières théoriques et notamment des mathématiques, de la physique et de l'économie.

Dans ce chapitre, nous allons analyser un peu plus en détail les caractéristiques de l'ECAO en dégageant tout d'abord les dominantes pédagogiques de cet enseignement, puis en proposant une approche structurale de ce nouveau domaine.

2 - LES BUTS DE L'E C A O

=====

L'Enseignement par la Conception Assisté par Ordinateur élargit le domaine de l'EAO simultanément dans trois directions :

- Le contenu de l'enseignement dispensé,
- Le principe de la méthode pédagogique utilisée,
- Le rôle de l'enseignement au cours de la conception et de la gestion d'un système d'EAO.

2.1 - Le contenu de l'ECAO : les choix fondamentaux

Dans le cadre de l'ECAO, les étudiants ont à réaliser une application pratique de conception assistée dans un domaine particulier. Un premier but de cet enseignement est de permettre l'apprentissage des méthodes générales liées à la CAO et les techniques de conception propres à chaque application concrète proposée à l'étudiant. Pour faciliter une telle approche, nous avons étudié en détail les processus de conception [22] dont nous avons dégagé les principes fondamentaux. Cette analyse nous conduit à proposer à l'enseignant une méthodologie générale de la conception qu'il pourra ensuite adapter à chaque cas particulier à l'aide des outils informatiques mis à sa disposition.

L'apprentissage de la prise de décisions dans un contexte très proche de la réalité est un deuxième aspect de cet enseignement. Ce souci influencera profondément la méthode pédagogique que nous préconisons (§ 2.2.). Il va également imposer l'utilisation de nombreuses simulations pour que l'étudiant puisse évaluer directement les performances de l'objet qu'il aura construit. L'élève exercera ainsi son jugement en conduisant lui-même l'approche d'une discipline à travers une activité de conception.

Par ces deux aspects fondamentaux, la découverte de la méthodologie de conception d'un objet et la pratique de la prise de décisions, l'étudiant parviendra à une nouvelle étape de son apprentissage : (fig I).

- Découverte des relations intrinsèques caractéristiques de l'objet conçu,
- Principes méthodologiques généraux de la conception,
- Découverte des modèles et de leurs domaines de validité,
- Complément et vérification de connaissances ponctuelles pratiques et théoriques,
- Organisation des connaissances au sein d'une discipline,
- Découverte de relations interdisciplinaires concrètes,
- Méthode générale de travail.

Certes, nous semblons ainsi nous attacher beaucoup plus à la méthode qu'à la matière à étudier. Ce choix est délibéré, car il restitue à notre avis, le véritable contenu de l'enseignement d'une matière qu'il serait absurde de réduire à une mosaïque de connaissances atomisées.

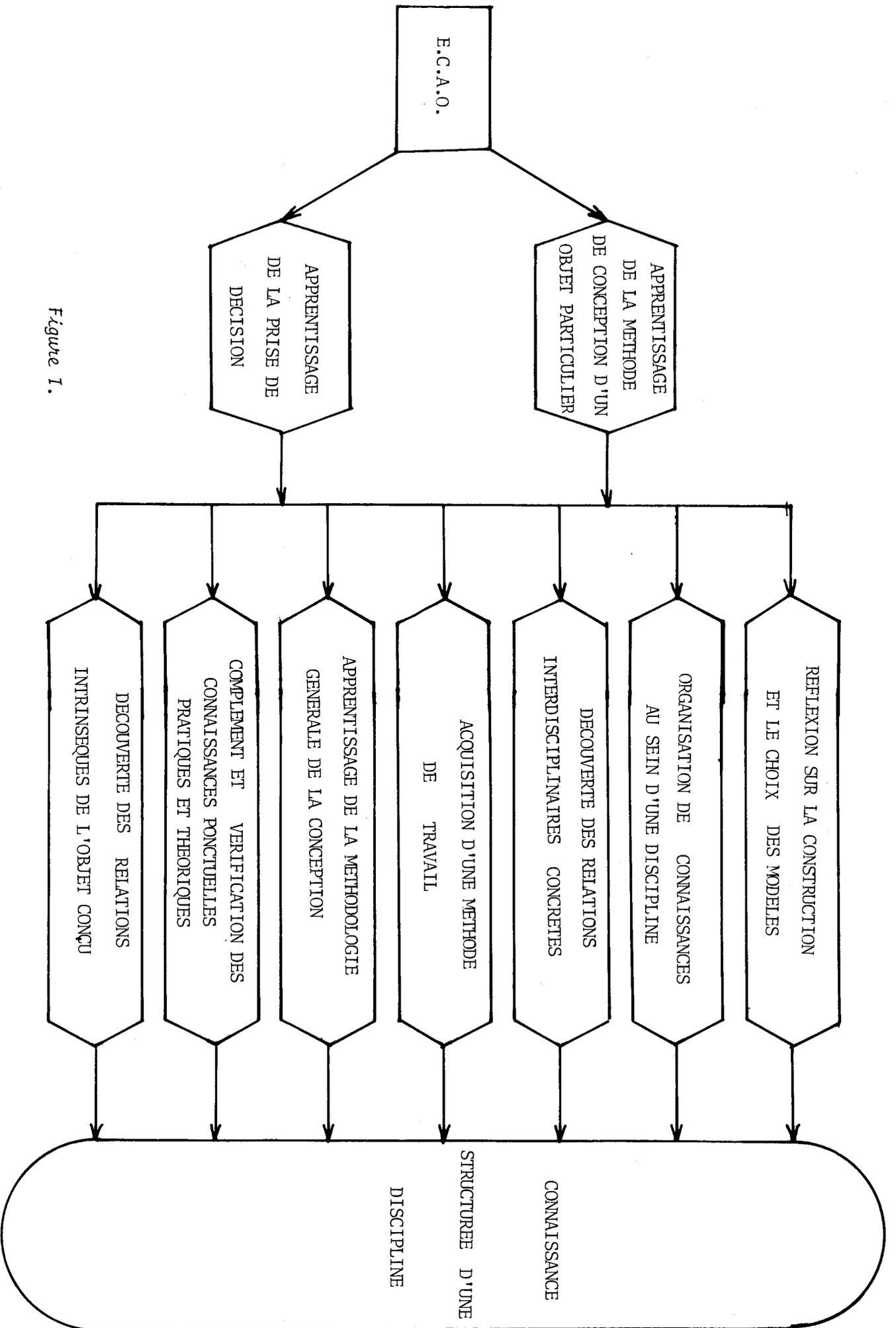


Figure 1.

2.2 - Caractéristiques d'une pédagogie associée à l'ECAO

La définition même de l'ECAO implique une pédagogie qui prône la libre découverte par l'étudiant du contenu méthodologique et cognitif de cet enseignement. Chaque enseignant aura bien entendu la liberté de doser l'importance d'un contrôle plus ou moins actif du système sur le travail de l'étudiant. Mais dans tous les cas, l'ECAO devra prendre la forme d'une activité en libre-service, fortement personnalisée, au cours de laquelle l'étudiant sera le plus souvent le maître d'oeuvre de son propre travail.

La caractéristique pédagogique fondamentale de l'ECAO est d'aborder l'apprentissage de toute une discipline et même d'un ensemble interdisciplinaire, en parallèle avec la Conception Assistée par Ordinateur d'un objet illustrant cette matière. La motivation de l'étudiant sera ainsi considérablement accrue et les révisions ou exercices qu'il réalisera dans la discipline étudiée et dans les matières annexes seront toujours reliées à un contexte concret et pratique. Il sera ainsi amené à "prendre le recul nécessaire par rapport à son acquis théorique pour organiser ses connaissances et découvrir enfin l'utilité de posséder tant d'outils d'analyse" [42].

Certes, le choix de cet objet à concevoir, ce "support pédagogique de la discipline" est primordial, mais les sciences physiques ne manquent pas de telles illustrations qui sont peut-être plus délicates à trouver dans les sciences humaines : Conception d'une machine électrique en électrotechnique, de turbines hydrauliques, de ponts en génie civil, de commandes en automatique, de modèles de simulations en gestion [52], etc...

2.3 - Le rôle de l'enseignant

L'enseignant spécialiste d'une discipline est rarement formé aux techniques de la CAO et de l'EAO. Quelle place lui réserver dans cette nouvelle forme d'enseignement? Certains voient dans l'essor de l'EAO, une tendance au remplacement de l'enseignant par l'ordinateur. En ECAO, le professeur est irremplaçable. En effet, c'est lui qui devra concevoir, mettre au point, et enfin gérer ce système d'ECAO : cette tâche inhabituelle

va demander un gros travail de formation préalable. Or la perspective d'une étude détaillée des techniques de l'EAO et de la CAO risque de rebuter la majorité des volontaires. C'est pourquoi nous avons appliqué à l'ECAO elle-même ses propres principes pédagogiques : la discipline enseignée devenant alors la construction de systèmes d'ECAO, le professeur prenant la place de l'étudiant et le support pédagogique de la discipline étant... le système d'ECAO relatif à la matière dont le professeur est le spécialiste.

Ainsi pendant la conception de son système d'ECAO, l'enseignant pourra déjà se rendre compte par lui-même des contraintes d'utilisation d'un tel enseignement. D'autre part la méthodologie de construction que nous lui proposons lui permettra d'approfondir l'aspect général des méthodes de Conception Assistée par Ordinateur et d'affiner le système d'ECAO particulier qu'il construit.

Cependant, la tâche de l'enseignant ne s'arrête pas à la conception du système d'ECAO, en effet la mise au point d'un tel système est toujours très délicate et ne peut être menée à bien que par un pédagogue qui évaluera le bien fondé de ses hypothèses. De nombreuses modifications sont généralement nécessaires avant d'obtenir un résultat satisfaisant.

Enfin la gestion pédagogique du système construit incombe à l'enseignant. Cette tâche est également très délicate car le principe didactique n'est pas classique [53], l'environnement psychologique est très fortement marqué par une relation homme-machine difficilement maîtrisable. En outre, l'intégration de ce nouveau style d'enseignement au sein d'une structure traditionnelle ne va pas toujours sans quelques problèmes d'interaction avec les autres activités pédagogiques.

Le travail de l'enseignant est donc considérable ; l'une de nos premières préoccupations sera de l'alléger au maximum.

3. LES CONTRAINTES D'UN SYSTEME D'E C A O

=====

Les buts que nous avons assignés à l'ECAO étant assez ambitieux, les contraintes liées à un tel système seront très importantes. Un souci d'efficacité à court terme (obtenir un temps de réponse très rapide) aurait restreint considérablement la portée de notre projet. Mais l'essor des performances du matériel nous permet de négliger cette limitation. Nous avons donc conduit une étude assez poussée sur le plan du logiciel pour concevoir un prototype dont l'efficacité sera améliorée dans un proche avenir.

Nous ne ferons donc pas intervenir dans un premier temps les contraintes technologiques pour nous consacrer uniquement au cahier des charges pédagogiques très serré dans lequel deux phases bien distinctes sont à considérer :

- La phase d'utilisation d'un système d'ECAO par un étudiant ;
- La phase d'élaboration d'un système d'ECAO par un enseignant.

Chacune de ces deux étapes va nous permettre d'approcher de plus près la structure idéale d'un système d'ECAO.

3.1 - Les contraintes dues au mode d'utilisation

Tout système d'ECAO devra respecter les objectifs suivants qui sont les conséquences directes des choix pédagogiques que nous avons pris.

- * Interaction homme-machine très poussée ;
- * Initiative des interactions prise par l'étudiant et non par le système ;
- * Contenu des interactions Machine → Opérateur fortement individualisé (messages et textes adaptés à chaque cas) ;
- * Indépendance du système par rapport au poste de travail ;
- * Possibilité d'accès de l'utilisateur à la structure du système et au détail de son fonctionnement à tous les niveaux, notamment en cours de résolution d'un problème complexe.

3.2 - Les contraintes dues à l'élaboration du système par l'enseignant

Détaillons un peu plus les diverses étapes de cette phase qui peut se décomposer selon le schéma de la figure II.

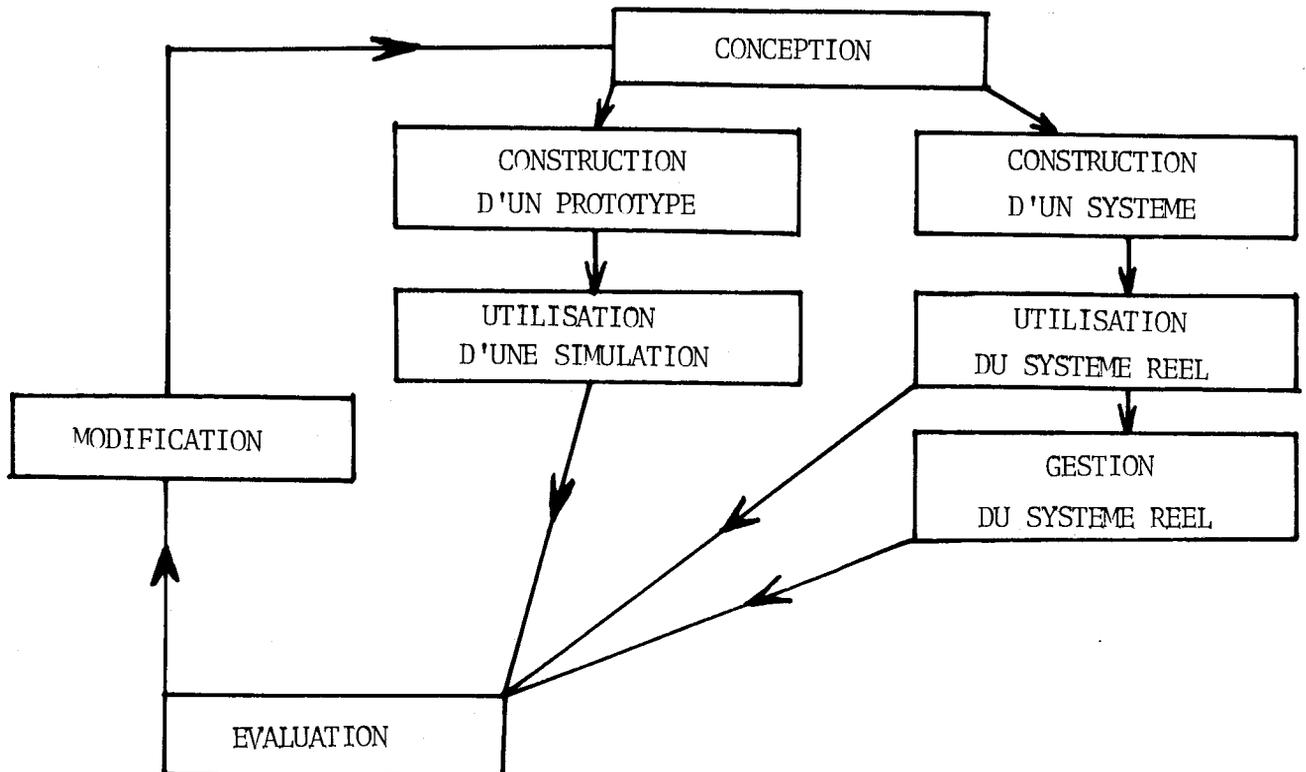


Figure II.

La tâche de construction d'un système réel étant particulièrement lourde, il nous a paru essentiel de fournir au concepteur la possibilité de décrire rapidement une simulation de son système grâce à laquelle il pourra réaliser une évaluation préalable fort utile.

En enseignement il est pratiquement impossible de fournir un cahier des charges définitif. Ses grandes lignes sont souvent issues d'hypothèses pédagogiques, que seule une expérimentation longue et délicate pourra justifier. La phase de modification d'un système est donc souvent la plus importante des étapes de son élaboration. Concevoir un système d'ECAO figé

conduirait à un échec quelles que soient par ailleurs les finesses les plus sophistiquées qu'il pourrait offrir à ses utilisateurs :

UN SYSTEME D'ECAO DOIT ETRE EVOLUTIF.

Cette première contrainte va intervenir dans presque toutes les options informatiques que nous prendrons au cours de la conception de notre système.

Pour qu'un enseignant non informaticien puisse élaborer lui-même la plus grande partie de son système d'ECAO, il faut qu'il soit aidé à deux niveaux :

- Sur le plan de la méthodologie de construction : l'enseignant spécialiste d'une discipline ne connaît généralement pas en détail les techniques de la CAO.

L'ELABORATION D'UN SYSTEME D'ECAO DOIT ETRE ELLE-MEME ASSISTEE.

- Sur le plan des techniques informatiques utilisées : les problèmes informatiques posés par l'ECAO (définition de langage, gestion dynamique de structures de listes complexes, construction de processus de résolution de problèmes), sont généralement inconnus de l'enseignant et posent des problèmes de mise en oeuvre assez fastidieux, c'est pourquoi :

LA RESOLUTION DES PROBLEMES INFORMATIQUES DE BASE DOIT ETRE A LA CHARGE DU SYSTEME GENERATEUR.

Toutes ces contraintes nous conduisent à définir un ensemble de ressources informatiques réunies dans un Système de Construction Assistée de Systèmes d'ECAO : le Système CASCADE qui sera utilisé par l'enseignant suivant un mode très proche de celui de l'étudiant travaillant sur un système d'ECAO. Ce générateur sera donc lui-même soumis à toutes les contraintes dues à une utilisation dans un contexte pédagogique.

Nous allons maintenant étudier de plus près les diverses composantes d'un système d'ECAO et également ses interactions avec son environnement immédiat.

4. L'ENVIRONNEMENT D'UN SYSTEME D'ECAO

=====

Délimiter au sein d'un processus d'enseignement global le domaine qui relève de l'ECAO, est une tâche très délicate que seul un enseignant spécialiste d'une discipline peut mener à bien. Nous pouvons cependant déjà préciser la forme que peuvent prendre en ECAO les parties fondamentales d'un enseignement classique.

4.1. La discipline à enseigner

Remarquons d'emblée qu'une discipline n'est qu'un concept qui permet de représenter globalement un ensemble structuré de connaissances utiles à la résolution de problèmes qui possèdent une finalité commune.

En enseignement, une discipline est généralement découpée en ressources pédagogiques (Cours, Exercices, etc...) qui traitent des divers aspects de la matière étudiée. Mais ce découpage détruit souvent la logique de la structure de la discipline à un point tel, que sa finalité peut parfois ne plus être perçue à travers l'enseignement.

En ECAO, le constructeur d'un système pourra, s'il le désire, décrire un modèle de la discipline qu'il veut enseigner, à l'aide de l'ensemble des diverses ressources pédagogiques dont il peut disposer réunies au sein d'une structure commune.

Ces ressources pourront être les éléments d'un découpage classique tel que : Chapitre de cours, Textes de T.P., Simulations de démonstration, Tests, ou toutes autres formes. Le modèle défini par l'enseignant pourra servir de référence ou simplement de cadre suffisamment large pour que l'étudiant puisse élaborer lui-même l'enchaînement des diverses ressources qu'il consulte. La structure de cet enchaînement traduit alors la démarche suivie par l'étudiant au cours de son apprentissage et sa perception de la discipline étudiée.

4.2 - L'activité de conception

En pédagogie traditionnelle une activité de conception est très rarement directement intégrée au processus d'enseignement. Les "Projets de fin d'étude" ont cette orientation, mais bien souvent, comme leur nom

l'indique, ils interviennent trop tard. D'autre part ces projets, qui sont généralement réalisés dans le cadre industriel, échappent aux pédagogues. Il nous semble souhaitable que l'enseignant introduise des exercices de conception beaucoup plus tôt dans le cours des études et diversifie ses domaines d'applications.

Choisir des sujets de conception associés à une discipline, n'est pas un problème lié au niveau de connaissance de la population étudiante concernée, mais à celui de l'adaptation d'une pédagogie souvent conservatrice à une activité nouvelle.

La CAO industrielle nous offre déjà de nombreux exemples dont la plupart sont facilement adaptables dans un contexte pédagogique. Les applications d'ECAO que nous proposerons, auront pour cadre une école d'ingénieurs qui est de fait un terrain d'expérience très propice. Dans un contexte différent et notamment lorsque la population étudiante a une formation moins complète (dans les classes du secondaire par exemple) l'approche que devra réaliser l'enseignant sera plus complexe : le modèle de la discipline sera en effet très simplifié et l'activité de conception qui l'illustrera sera plus délicate à trouver. Nous sommes toutefois persuadés qu'il y a là matière à une recherche pédagogique originale.

Au sein d'un système d'ECAO, le pédagogue peut également choisir plusieurs exercices de conception qui présenteront ainsi l'avantage de faire découvrir une discipline par des biais différents et complémentaires. L'impact de l'activité de conception sur le mode d'apprentissage de la discipline est très certainement déterminant : il sera donc enrichissant pour le pédagogue d'évaluer les conséquences du choix des sujets de conception sur l'efficacité de l'enseignement dispensé.

4.3 Les fondements méthodologiques

Dans le domaine des sciences exactes et expérimentales qui forment le cadre privilégié de l'application de l'ECAO, c'est la méthode cartésienne qui sous-tend toute investigation et qui constituera donc la base didactique de tout système d'ECAO.

L'étudiant devra pouvoir vérifier la plupart des lois qu'on lui propose, suivant son propre rythme, en exerçant son jugement personnel de telle sorte que la connaissance ainsi acquise parvienne au niveau

de l'évidence. Le système d'ECAO s'appuiera donc sur l'utilisation de nombreuses simulations en parallèle avec des exercices au cours desquels l'étudiant aura l'occasion d'accéder au niveau le plus fin de la décomposition des processus étudiés.

De même la décomposition de chaque discipline et de ses supports devra être conduite par l'enseignant en partant des objets les plus simples pour parvenir aux concepts les plus composés.

Enfin le système d'ECAO comportera des outils conversationnels élaborés, pour que chaque étudiant puisse réaliser tous les dénombrements et vérifications nécessaires à une étude critique de son travail, des modèles qu'il aura construits lui-même ou de ceux qu'on lui aura proposés.

Nous retrouverons ces principes étroitement intégrés dans notre système générateur CASCADE.

4.4 - La stratégie pédagogique

Comment se matérialisent les options pédagogiques, au sein d'un système d'ECAO? Tout d'abord indirectement, au cours du choix des modèles de la discipline et de ses supports, mais également de façon plus visible dans la gestion du travail de l'étudiant sur le système.

C'est cette "gestion" que nous appellerons la stratégie pédagogique.

L'enseignant exprimera ses choix pédagogiques à l'aide d'un certain nombre d'outils que nous mettrons à sa disposition, et, bien que nous ne puissions pas préjuger des décisions du pédagogue, nos options s'accordent avec les hypothèses fondamentales qui président à la définition proposée pour l'ECAO. Nous avons mis l'accent sur des outils pédagogiques qui faciliteront la gestion des activités de CAO en permettant une analyse "personnalisée" du travail de chaque étudiant sur un système de mode non directif. Une très grande liberté est laissée à l'enseignant dans la description de sa stratégie pédagogique et surtout dans les modifications qu'il devra constamment y apporter, même en cours de fonctionnement du système.

4.5 - L'environnement matériel

Dans tout système pédagogique, il faut absolument prendre en considération le contexte dans lequel travaillera l'étudiant. La relation psychologique Homme-Machine aura une influence fondamentale sur le travail de l'étudiant. Cette relation s'exprime à deux niveaux. Tout d'abord, en fonction du matériel (périphériques d'ordinateur) mis à la disposition de l'étudiant, ensuite à travers les communications directes entre l'étudiant et la machine : langage utilisé, messages émis par la machine, interactions graphiques, projections de diapositives, etc...

Il nous a paru indispensable que l'enseignant pénétre au plus profond de cette relation Homme-Machine et prenne lui-même les deux principales décisions :

- Choix de la configuration du poste de travail ;
- Choix des langages de communication et définition détaillée de toutes les ressources d'édition.

Des outils généraux ont été développés de telle sorte que le système CASCADE influe le moins possible sur ces décisions d'ordre pédagogique.

5. LA MODELISATION

=====

L'analyse que nous avons conduite jusqu'à présent sur le plan pédagogique nous a permis de préciser ce qu'est l'Enseignement par la Conception Assistée par Ordinateur. Nous avons également énoncé les principales hypothèses pédagogiques qui caractérisent notre approche. Nous allons prolonger notre étude en proposant un modèle d'un système d'ECAO et de son environnement. Cette modélisation sera conduite dans le cadre des hypothèses pédagogiques que nous avons choisies en essayant d'en réduire le moins possible la portée.

Cette nouvelle étape va nous permettre de proposer une méthodologie de la conception de systèmes d'ECAO indépendamment de la discipline qu'ils illustrent [54]. Cette méthodologie nous conduira à définir les grandes lignes d'un système d'aide à la construction de systèmes d'ECAO.

Notre deuxième chapitre, également consacré à la recherche pédagogique, réalisera une approche structurée de notre problème en mettant les techniques de la science des systèmes et de l'informatique au service de la pédagogie sans sacrifier les options concrètes que nous avons définies. Ce n'est qu'au troisième chapitre que nous aborderons les détails techniques de la construction du prototype que nous avons réalisé.

CHAPITRE II

=====

MODELISER L'ENVIRONNEMENT

1. MODELE D'UN SYSTEME D'ECAO ET DE SON ENVIRONNEMENT
2. DU TRAVAIL DE L'ETUDIANT AU MODELE POTENTIEL DES MODELES POTENTIELS
3. LES RESSOURCES DE SIMULATION
4. L'INTERFACE ENTRE L'ETUDIANT ET LE SYSTEME
5. LA STRATEGIE PEDAGOGIQUE
6. LE MODELE COMPLET

1. LE MODELE D'UN SYSTEME D' ECAO ET DE SON ENVIRONNEMENT

Nous allons rechercher au sein d'un système d'ECAO des sous-ensembles pourvus d'une logique qui leur soit propre de telle sorte que chaque étape de la conception d'un système corresponde à la définition de l'un de ces éléments.

La structure type que nous avons choisie peut être décomposée en trois ensembles fonctionnels, selon la figure III.

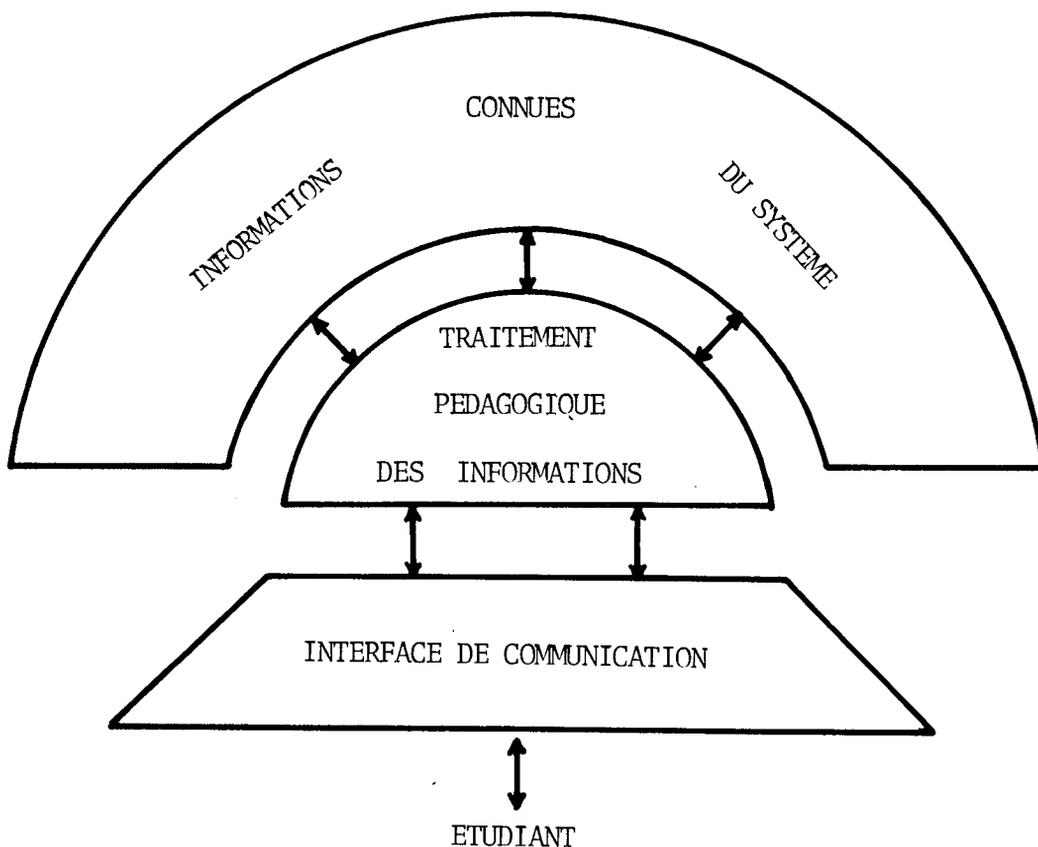


Figure III : Structure globale d'un système d'ECAO.

Au sein de chaque élément de cette structure, nous pouvons à nouveau délimiter des sous-ensembles fonctionnels caractéristiques :

Dans l'interface de communication, nous distinguerons trois composantes : le langage qui regroupe toutes les interactions dans le sens étudiant vers le système, les ressources d'édition qui réunissent tous les supports des informations transmises à l'étudiant par le système, et enfin le poste de travail.

Les informations seront regroupées en trois banques de données : l'une représente le travail déjà réalisé par chaque étudiant, la deuxième constitue le modèle de la discipline et de son support, la troisième regroupe les algorithmes de calcul et de simulation dont dispose le système.

La stratégie pédagogique assure le traitement de toutes ces ressources dont la structure est représentée à la fig. IV et dont nous allons examiner en détail les caractéristiques.

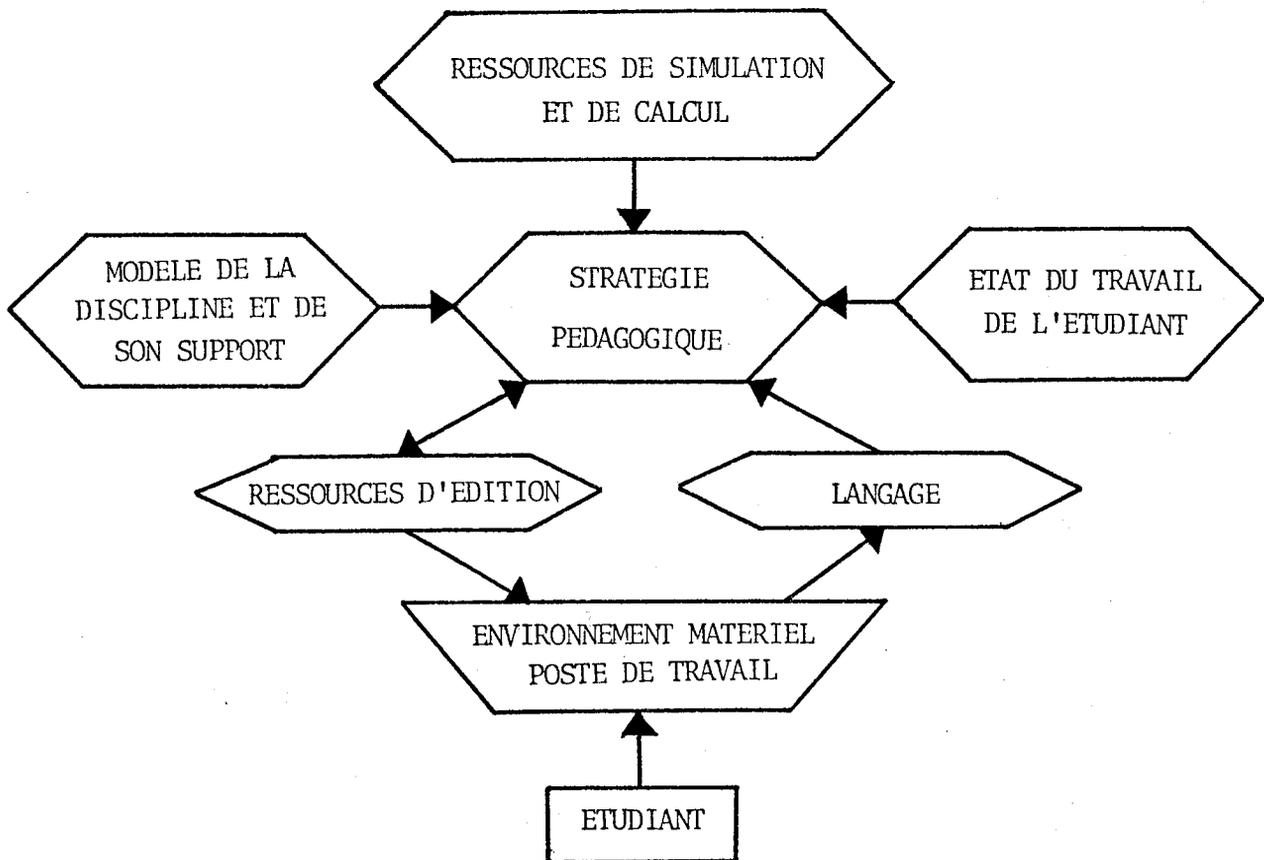


Figure IV : Modèle d'un système d'ECAO

2. DU TRAVAIL DE L'ETUDIANT AU MODELE POTENTIEL DES MODELES POTENTIELS =====

Au sein d'un système d'ECAO, le processus de conception assistée a besoin d'avoir accès à une trace du travail réalisé par l'étudiant. De même, le processus de gestion pédagogique utilise cette information pour s'adapter à chaque cas particulier. Concevoir la banque de données qui représente le travail de l'étudiant est donc une tâche ardue qui interfère au plus haut degré avec les processus de résolution de problèmes et de gestion pédagogique du système. C'est pourquoi, nous proposons à l'enseignant d'aborder cette étape à l'aide des outils conceptuels nouveaux que nous allons définir dans cette étude.

2.1. LE TRAVAIL DE L'ETUDIANT

Il est nécessaire de mémoriser toutes les caractéristiques de l'objet conçu par l'étudiant au cours de son travail de conception, mais il est également fort utile de connaître les ressources pédagogiques que l'étudiant aura consultées : textes de cours, exercices d'illustration, tests d'entraînement, etc... Or, dans un système d'ECAO fondamentalement NON DIRECTIF, la consultation de ces ressources doit être libre et la démarche de chaque étudiant travaillant sur des exercices de conception doit pouvoir être personnelle. L'ensemble des structures possibles pour représenter le travail d'un étudiant est donc fortement combinatoire, c'est pourquoi, l'enseignant ne devra pas s'efforcer de rechercher la structure idéale représentant l'approche jugée optimale de son système d'ECAO. Il lui faudra plutôt recenser l'ensemble des approches pédagogiquement acceptables et définir tous leurs degrés de liberté.

Chaque base de données qui représentera le travail de chaque étudiant aura donc non seulement un contenu spécifique, mais aussi une structure qui lui sera propre. L'intérêt pédagogique d'un tel choix est indéniable. Les informations personnalisées que contiendront ces bases permettront en effet une analyse très fine du travail de l'étudiant. Chaque structure sera le reflet de la démarche qu'aura suivie l'étudiant dans son travail sur le système et de la méthode d'apprentissage qu'il aura utilisée. On pourra également y retrouver les interactions étroites qui auront eu lieu entre l'activité de conception assistée réalisée par l'étudiant et les consultations directes qu'il aura effectuées auprès des

ressources pédagogiques du système. Le dépouillement de ces renseignements devrait certainement apporter sa contribution à la recherche en didactique.

La tâche de l'enseignant comprendra donc trois étapes :

- Etudier tout d'abord l'ensemble des structures technologiquement et pédagogiquement acceptables pour représenter l'objet que l'étudiant aura à concevoir ;
- Préciser l'ensemble des ressources pédagogiques mises à la disposition de l'étudiant et les diverses liaisons qui les caractérisent (ce travail peut conduire à la définition d'un véritable modèle de la discipline concernée) ;
- Préciser enfin les interactions les plus probables entre les phases de conception des principaux sous-ensembles de l'objet à concevoir et les ressources pédagogiques susceptibles d'apporter des éclaircissements à ce sujet.

Nous pouvons représenter ces trois composantes au sein d'une même structure, c'est ce que nous appellerons le MODELE POTENTIEL du système d'ECAO.

2.2. LE MODELE POTENTIEL

Cette notion a été introduite par J.C. Sabonnadière et J.C. Latombe [36], lors de la conception du système ESPACE. C'est ce même concept que nous reprenons ici en élargissant sa portée au niveau de toute une discipline.

Définition : Le modèle potentiel d'une discipline est l'ensemble structuré des ressources pédagogiques de cette discipline associé aux modèles les plus généraux des objets qui seront conçus dans le cadre de son apprentissage au moyen d'un système d'ECAO.

Le travail de chaque étudiant sera représenté par une banque de données construite selon l'une quelconque de toutes les structures compatibles avec le modèle potentiel.

L'enseignant pourra donc concevoir la gestion du travail de l'étudiant à travers l'écran que constitue le modèle potentiel, sans jamais avoir

besoin de connaître les détails de la structure de liste qui représentera à chaque instant ce travail dans les mémoires de l'ordinateur. Au lieu de concevoir et gérer une banque de données pour chaque étudiant qui travaille sur le système, l'enseignant n'a plus qu'à spécifier les caractéristiques du modèle potentiel qu'il aura choisi. Mais cette tâche est assez inhabituelle pour un enseignant, c'est pourquoi nous allons analyser de plus près les caractéristiques d'un modèle potentiel, et les principales étapes de sa conception.

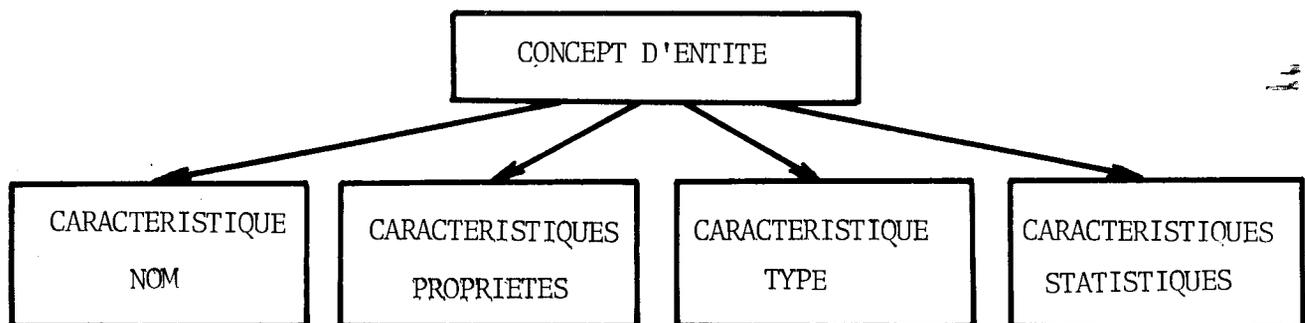
D'autre part, si nous voulons assurer automatiquement la gestion informatique de toute banque représentant le travail d'un étudiant en ne connaissant que le modèle proposé par l'enseignant, nous allons être amenés à définir plus en détail les propriétés que devront posséder les modèles potentiels pour que cette gestion soit possible. Un nouveau concept : le modèle potentiel des modèles potentiels, va servir de support à cette approche.

2.3. LE MODELE POTENTIEL DES MODELES POTENTIELS

Définition : L'ensemble des concepts dont l'utilisation permet la description d'un modèle potentiel, constitue le modèle potentiel des modèles potentiels.

Huit concepts sont à la base de cette description : L'entité, la liaison, la hiérarchie, la sous-hiérarchie, l'occurrence d'entité, l'occurrence de liaison, l'occurrence de hiérarchie, l'occurrence de sous-hiérarchie.

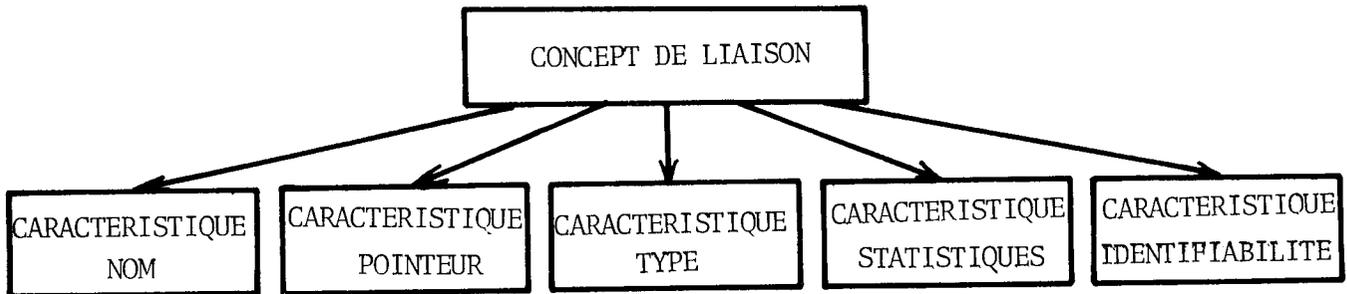
ENTITE :



Une entité permet de représenter soit une partie physique ou fonctionnelle de l'élément à concevoir, exemple ROTOR.FIL.CIRCUIT MAGNETIQUE. REFROIDISSEMENT ; soit un élément pédagogique : RESULTATS AUX TESTS,

STRUCTURE DU COURS, EXERCICE PARTICULIER, etc... Certaines caractéristiques de ce concept font appel à des notions que nous n'avons pas encore définies ; nous en donnerons une explication détaillée ultérieurement.

LIAISON :



Une liaison représente une relation entre deux entités. Une liaison particulière : la liaison hiérarchique, permet de représenter la relation "contenant-contenu" et de définir la notion de hiérarchie. Comme pour l'entité nous détaillerons ultérieurement les caractéristiques de ce concept.

UNE HIERARCHIE est un ensemble d'entités reliées entre elles par des liaisons hiérarchiques de telle sorte que le graphe orienté qu'elles forment soit connexe, sans circuit et possède un seul sommet dont le degré entrant est nul [2]. L'entité correspondant à ce sommet sera appelée la tête de hiérarchie.

Un modèle potentiel ne pourra comporter qu'une seule hiérarchie. Cette hiérarchie ne pourra pas être vide (elle devra comporter au moins un élément : la tête de hiérarchie). Les entités qui n'appartiennent pas à la hiérarchie sont dites "hors hiérarchie".

UNE SOUS-HIERARCHIE est formée par l'ensemble des entités descendant hiérarchiquement d'une entité donnée. Cette entité est appelée tête de sous-hiérarchie. Une sous-hiérarchie est nécessairement un sous-ensemble de la hiérarchie. Les diverses sous-hiérarchies qui seront définies, devront être disjointes (de telle sorte qu'une entité ne puisse jamais appartenir à plus d'une sous-hiérarchie).

NOTION D'OCCURRENCE : les concepts que nous avons définis permettent de décrire les éléments d'un modèle potentiel. D'autre part le travail de l'étudiant sera représenté par une banque de données construite d'après le modèle potentiel. Chaque élément de cette banque de données sera une réalisation concrète de l'un des concepts du modèle potentiel. Nous distinguerons donc au niveau du travail de l'étudiant des occurrences d'entités, des occurrences de liaisons, des occurrences de hiérarchie et de sous-hiérarchies.

NOTION D'IDENTIFIABILITE. cette notion concerne les occurrences d'entités et les occurrences de liaisons : un certain nombre d'occurrences d'une même entité pourront quelquefois avoir un sous-ensemble de leurs caractéristiques identiques (mêmes valeurs pour certaines propriétés, ou liaisons communes vers une autre occurrence). Il apparaîtrait alors fastidieux d'être contraint de préciser la même valeur ou la même liaison pour chacune de ces occurrences. C'est pourquoi, le concepteur peut définir à l'avance les caractéristiques qu'il juge identifiables pour chaque entité. Lorsque l'étudiant travaillera, il pourra ainsi, suivant les besoins, définir des occurrences d'une même entité entièrement distinctes les unes des autres ou bien créer des occurrences d'entité identiques à une occurrence définie au préalable. Ainsi toutes les occurrences créées identiques entre elles auront en commun toutes les propriétés et liaisons déclarées identiques et posséderont chacune en particulier les autres caractéristiques.

Nous allons maintenant détailler les caractéristiques de chaque concept.

2.3.1. Entité

Caractéristique NOM : ce nom (8 lettres) choisi par l'enseignant permettra de se référer à une entité.

Caractéristique TYPE : cette caractéristique pourra prendre l'une des trois valeurs suivantes :

- Tête de hiérarchie
- Tête de sous-hiérarchie
- Standard.

Caractéristique STATISTIQUES : ces caractéristiques sont numériques et comportent deux types de valeurs :

- Le nombre moyen d'occurrences qui pourront être générées pour une même entité au cours du travail de l'étudiant.
- Eventuellement le nombre maximum d'occurrences qu'un étudiant pourra créer pour une même entité lors de son travail.

Caractéristiques PROPRIETES : chaque entité pourra posséder un nombre illimité de propriétés. Chacune sera définie par :

- un nom : 12 caractères au maximum
- un type : à choisir parmi la liste suivante : Entier, Flottant, Logique, Alphanumérique(8 lettres), Tableau d'entiers, Tableau de Flottants, Tableau de logiques, Courbes.

Pour les tableaux et les courbes, le concepteur devra donner leur dimension sous forme de valeur numérique entière.

- des contraintes éventuelles : sous la forme de bornes numériques entre lesquelles devront être comprises les propriétés numériques ou bien sous la forme d'une liste de valeurs interdites ou seules autorisées pour une propriété (sauf pour les logiques).

- la particularité d'être identifiable ou non

Nous avons regroupé ces caractéristiques au sein d'un même diagramme à la figure V.

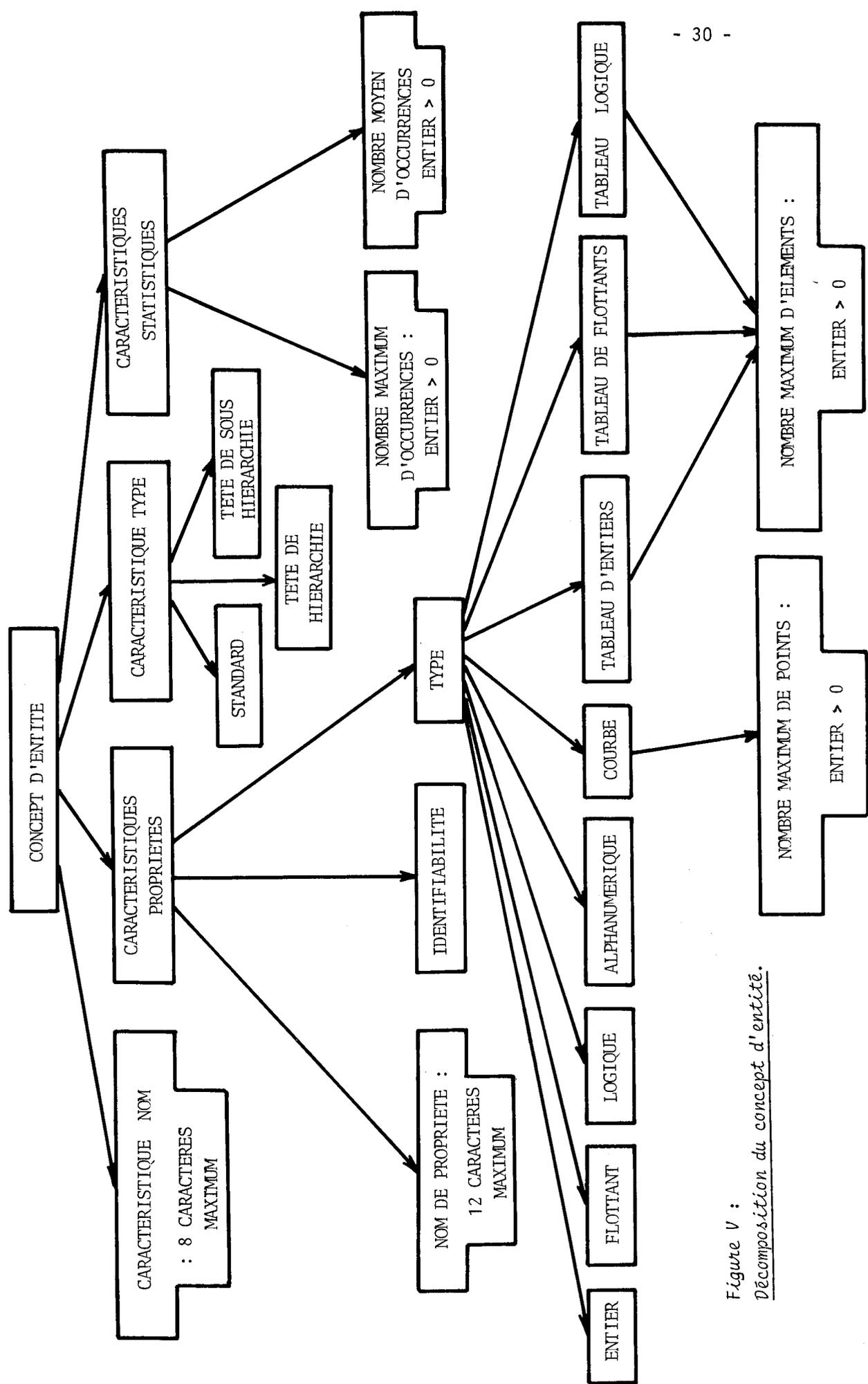


Figure V :
Décomposition du concept d'entité.

2.3.2. Liaison

Toute liaison sera définie par :

Caractéristique NOM (8 caractères) le nom "HIERARCHIQUE" sera commun à toutes les liaisons constituant une hiérarchie.

Caractéristiques POINTEURS :

- le nom de l'entité origine de la liaison.
- le nom de l'entité extrémité de la liaison.

Les liaisons hiérarchiques devront constituer un graphe compatible avec la définition que nous avons donnée du concept de hiérarchie.

Les liaisons non hiérarchiques pourront être de trois types :

Interhiérarchiques : ces liaisons devront relier deux entités de la hiérarchie (elle pourront former des cycles).

Vers la hiérarchie : ces liaisons devront relier une entité hors hiérarchie à une entité de la hiérarchie.

Hors hiérarchie : elles relieront deux entités n'appartenant pas à la hiérarchie.

De même que pour les entités, le concepteur pourra définir pour chaque liaison :

- les caractéristiques STATISTIQUES qui précisent le nombre moyen et le nombre maximum d'occurrences.
- l'identifiabilité.

Nous avons regroupé toutes les caractéristiques des liaisons au sein du diagramme de la figure VI.

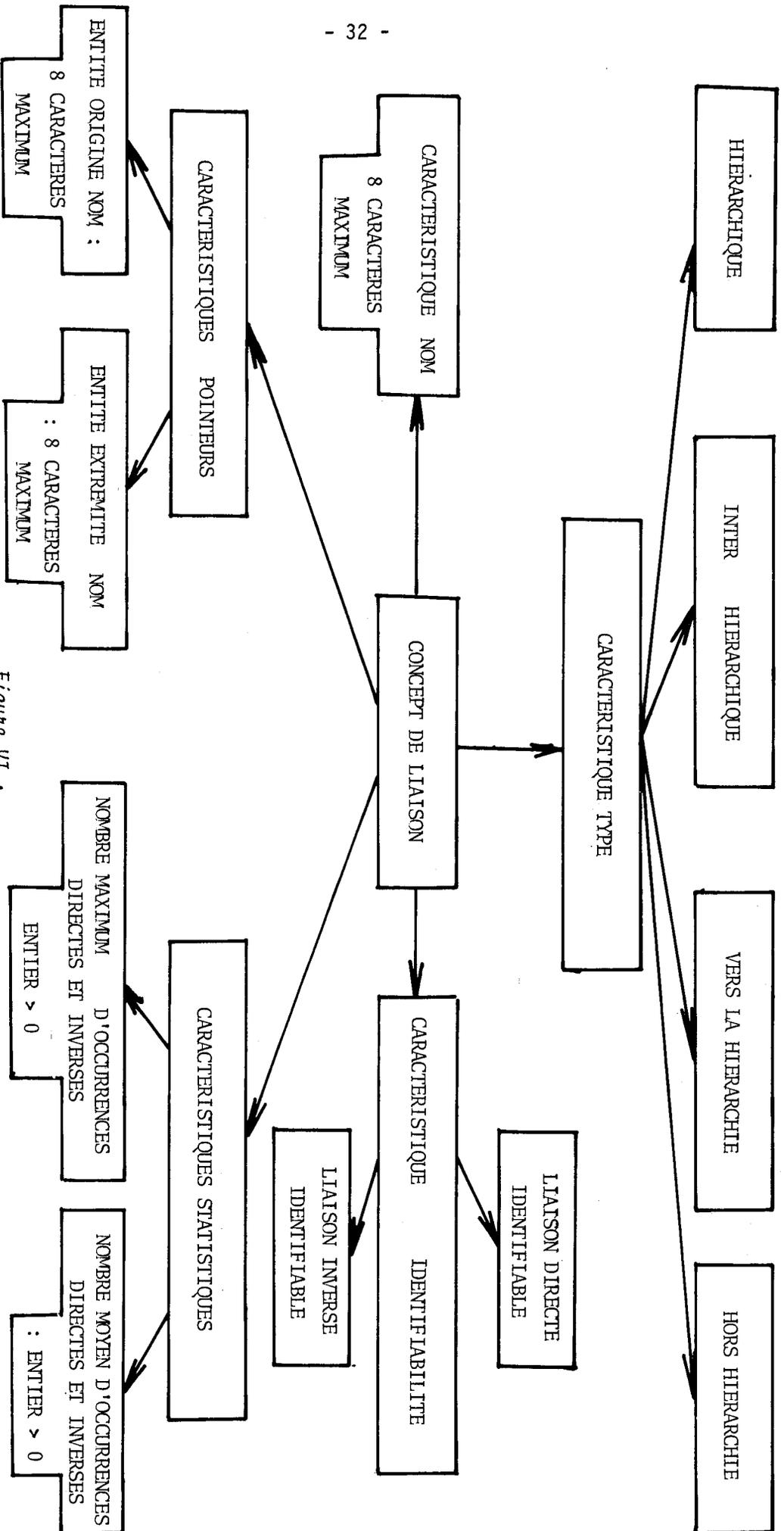


Figure VI :

décomposition du concept de liaison.

2.3.3. Exemples de modèles potentiels

Nous donnerons tout d'abord des exemples de modèles potentiels illustrant la partie conception assistée de l'ECAO. Nous introduirons seulement ensuite un modèle pédagogique complet comportant à la fois des entités destinées à la conception et à la gestion pédagogique.

2.3.3.1. Cas d'un procédé commandé

La discipline illustrée ici est l'automatique et plus particulièrement les problèmes de modélisation de commandes de processus. L'exercice de CAO proposé consiste à concevoir la structure du modèle mathématique de la commande d'un système monovariable ou multivariable que l'on a au préalable identifié. Un sous-ensemble des modèles réalisables pour décrire une commande forme le modèle potentiel que nous présentons à la figure VIII (cette étude a été réalisée en collaboration avec le LAG).*

L'étudiant devra construire l'une des structures cohérentes que l'on peut obtenir à partir de ce modèle. Cette structure sera stockée dans la banque de données qui sera attribuée à chaque étudiant. C'est sur les données numériques de cette banque (propriété de chaque occurrence d'entité) que seront activées les ressources de calcul et de simulation lorsque l'étudiant désirera évaluer les performances de la commande qu'il aura réalisée.

2.3.3.2. Cas de l'identification d'un procédé

Cet exemple illustre comme le précédent, l'automatique. L'activité de conception proposée à l'étudiant, consiste à construire un modèle mathématique d'un procédé que l'on cherche à identifier. Cette identification peut être obtenue par diverses techniques telles que : maximum de vraisemblance, intercorrélacion, moindres carrés généralisés, matrice instrumentale... Ces méthodes nécessitent une adéquation entre le modèle choisi et la technique utilisée. L'Etudiant devra donc construire un modèle cohérent de son procédé, il pourra ensuite lancer les programmes d'identification mis à sa disposition et obtiendra ainsi les valeurs de toutes les caractéristiques du modèle mathématique choisi pour son procédé. Une simulation lui permettra de comparer modèle et procédé.

* Laboratoire d'Automatique de Grenoble.

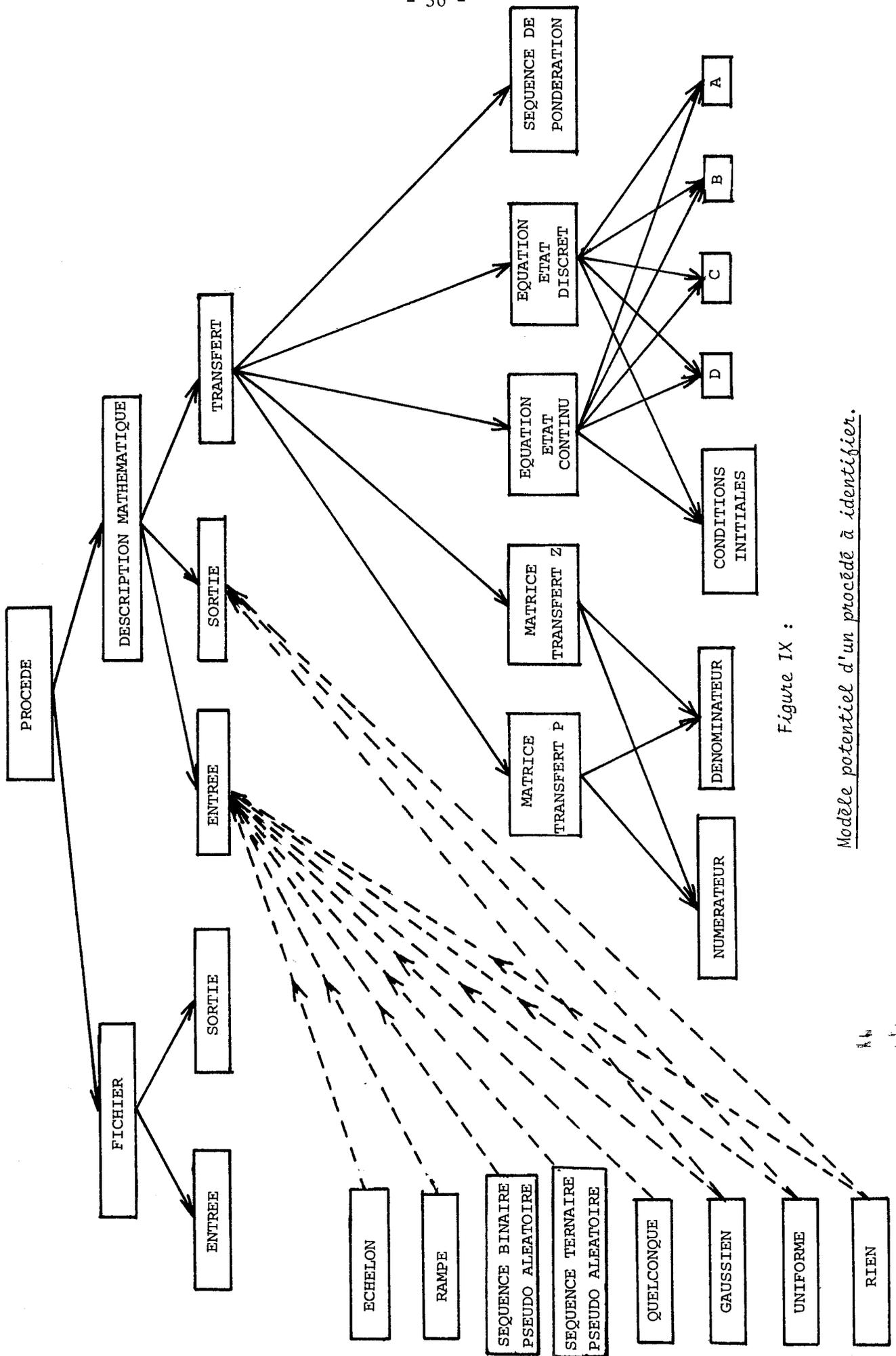


Figure IX :

Modèle potentiel d'un procédé à identifier.

réel et ainsi d'évaluer l'efficacité de la structure sélectionnée. Ce modèle est représenté à la figure IX. Nous remarquerons l'intérêt des éléments hors hiérarchie et des liaisons vers la hiérarchie qui permettent de définir une fois pour toutes les valeurs des entrées et des bruits qui seront communs à tous les procédés (ce travail a été réalisé en collaboration avec le IAG).

2.3.3.3. Exemple de la machine asynchrone

La discipline illustrée est ici l'électrotechnique. L'activité de CAO consiste à concevoir un moteur asynchrone. Nous présentons le modèle potentiel de cette machine à la figure X. (le travail a été réalisé avec la collaboration de M. R. Perret du Laboratoire d'Electrotechnique de l'INPG). Nous pouvons remarquer que dans ce modèle apparaissent des liaisons physiques importantes telles que l'interaction entre le rotor et le stator.

2.3.3.4. Exemple d'un transformateur

Nous présentons à la figure XI, le modèle potentiel du transformateur défini par J.C. Latombe et J.C. Sabonnadière pour le système ESPACE. Ce modèle est bien sûr descriptible avec nos outils et nous pouvons remarquer au passage qu'il n'était pas décrit par une banque de données dans le système ESPACE, mais que sa structure était implicitement contenue dans la logique interne des programmes constituant ce système.

2.3.4. Exemple d'un modèle intégrant conception et gestion pédagogique

L'enseignant peut intégrer dans son modèle potentiel des entités et des liaisons qui lui permettent de stocker des renseignements pédagogiques utiles à la gestion du système. Mais le concepteur peut aller plus loin en définissant un véritable modèle de l'environnement pédagogique du système ECAO qu'il a construit : le modèle de la discipline illustrée et des disciplines annexes.

Nous proposons à la figure XII un modèle simplifié des ressources pédagogiques associées à la conception d'un moteur asynchrone et illustrant l'électrotechnique. Ce modèle n'a en aucune façon, la prétention de constituer un élément de recherche pédagogique, de par son contenu ou

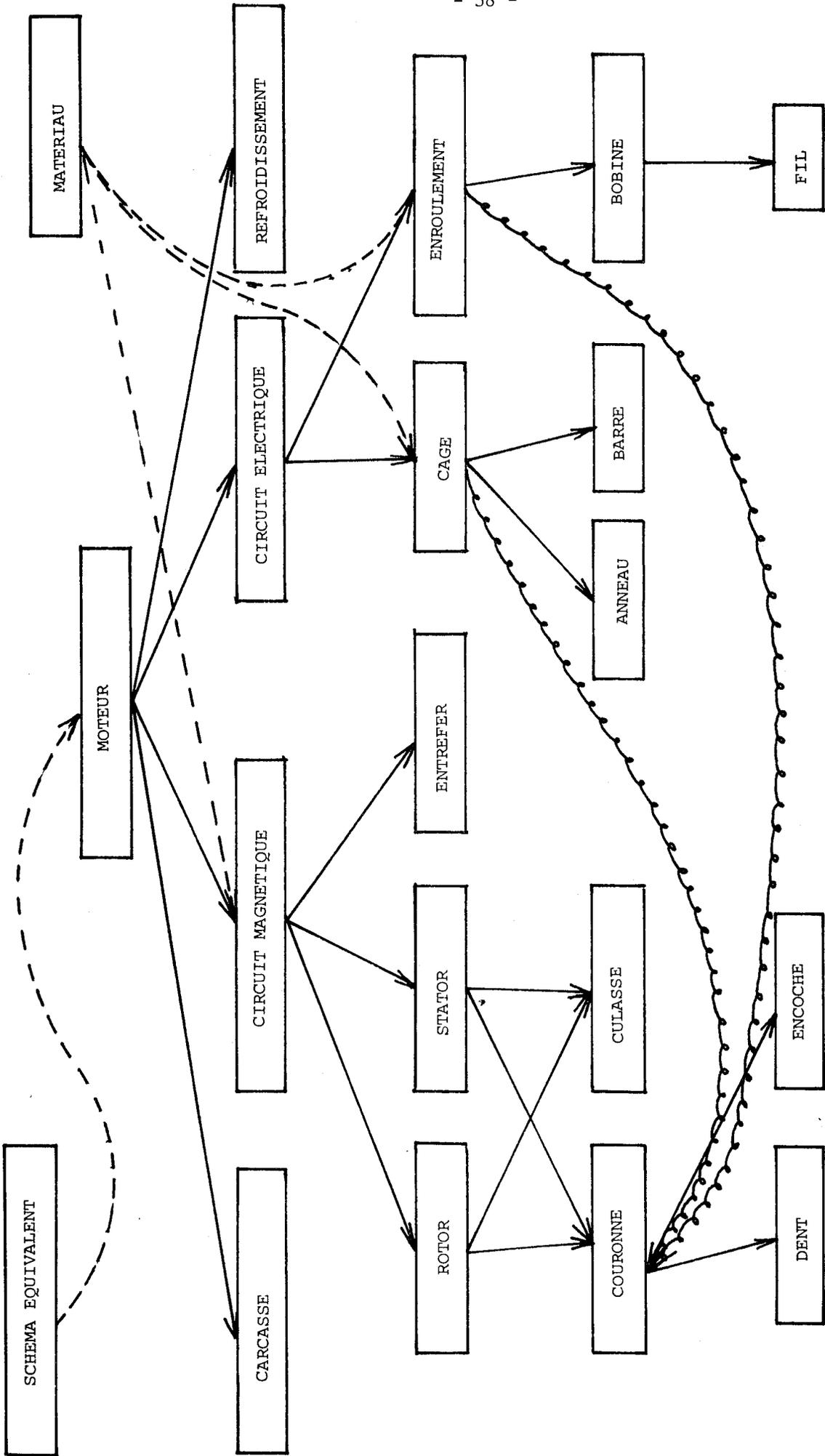


Figure X.

Modèle potentiel d'un moteur asynchrone.

84
14

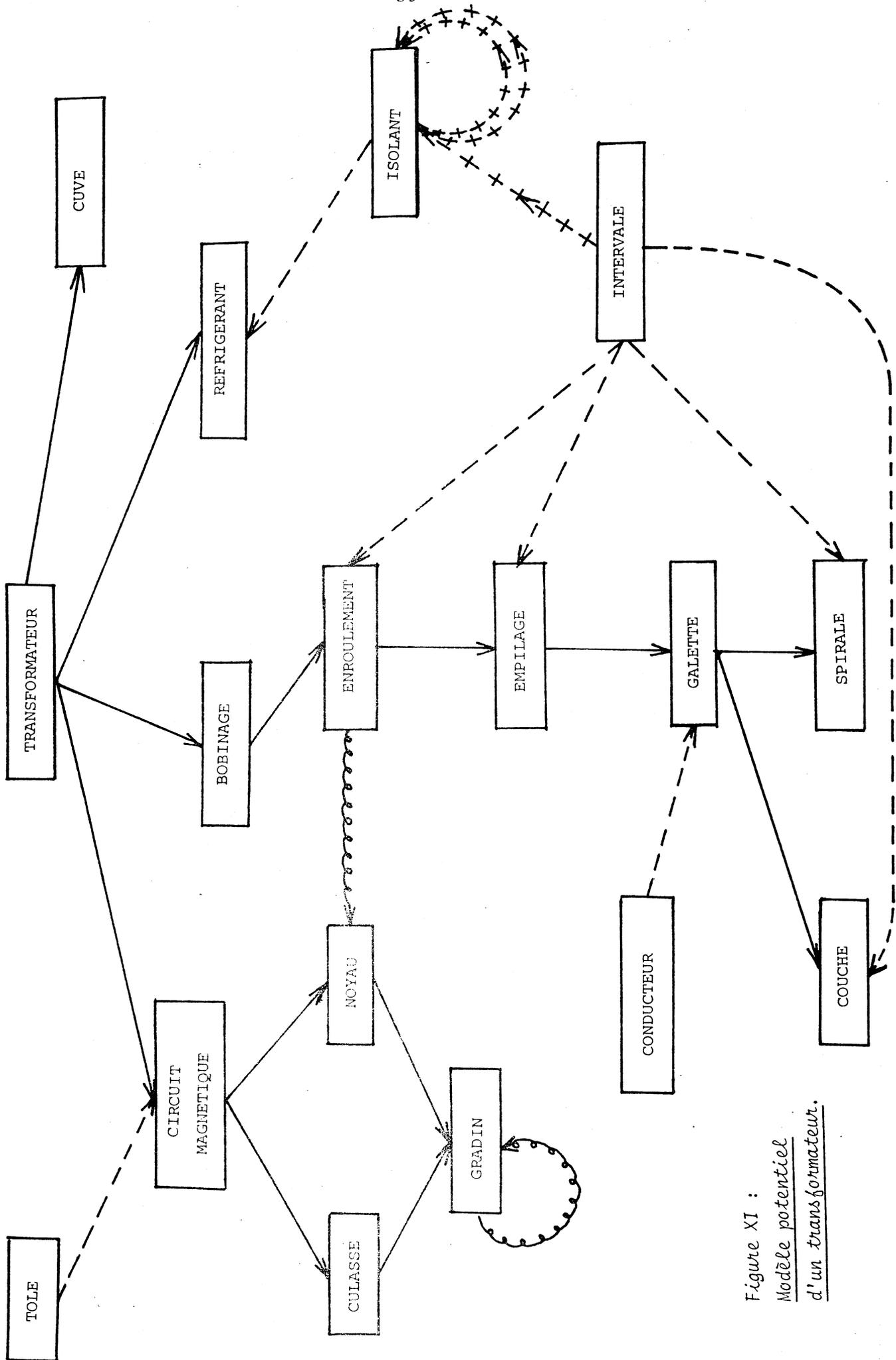


Figure XI :
Modèle potentiel
d'un transformateur.

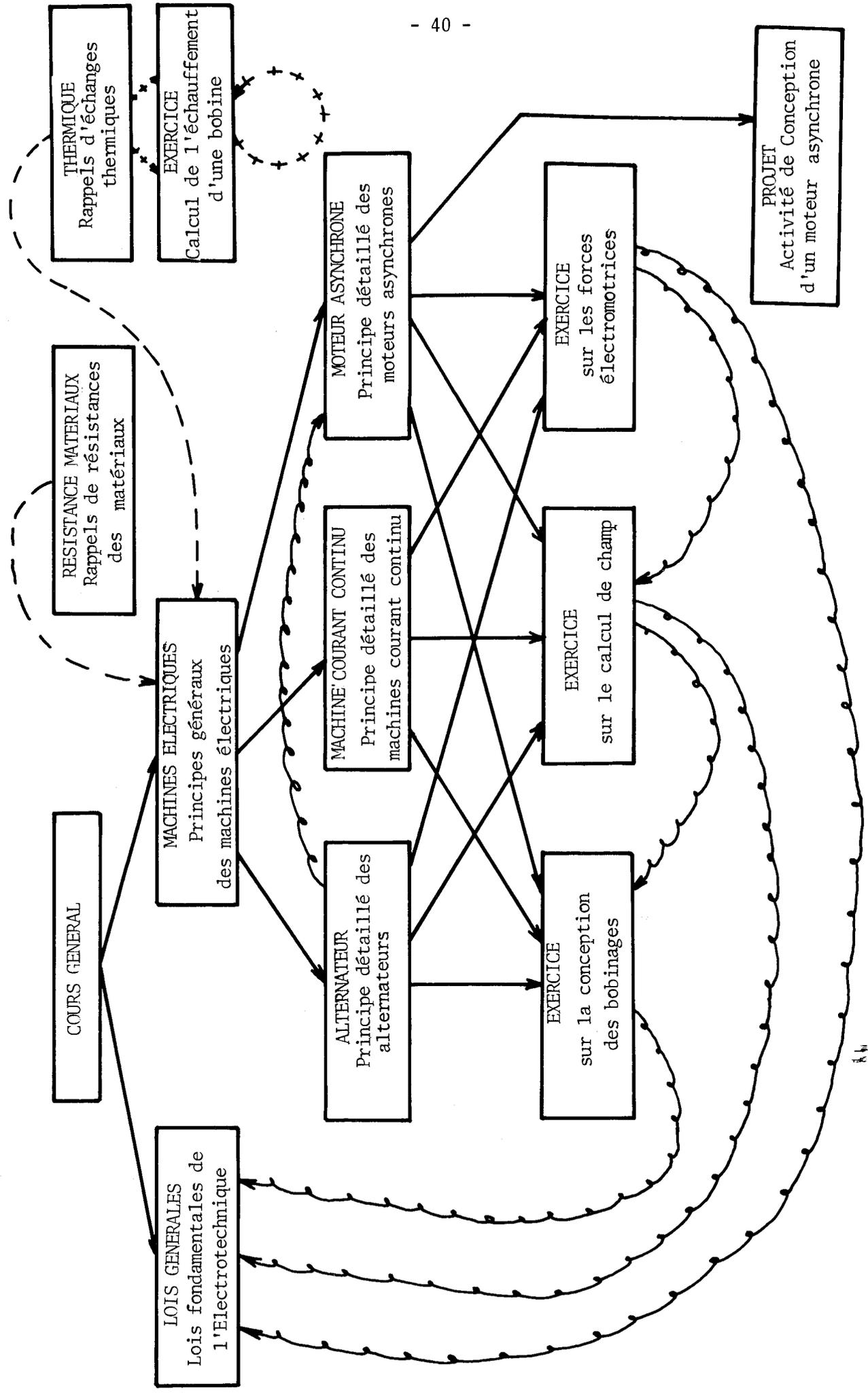


Figure XII.

sa structure. Il a pour but d'illustrer seulement plus concrètement l'interprétation pédagogique des outils conceptuels précédemment développés.

Nous avons utilisé dans cet exemple les liaisons hiérarchiques pour représenter l'enchaînement usuel des diverses ressources.

Les liaisons interhiérarchiques figurent les renvois entre les exercices et les présentations théoriques.

Les entités hors hiérarchies représentent les matières annexes.

Les liaisons hors hiérarchies permettent d'associer des exercices ou des tests à ces disciplines.

L'intégration du modèle potentiel de la discipline et de celui de son support au sein d'une même structure procure à l'enseignement un nouvel outil pédagogique. En effet l'enseignant peut prévoir les liaisons pédagogiques essentielles entre le processus de CAO et la consultation des cours, travaux dirigés ou banques d'informations diverses telles que : caractéristiques de matériaux, diapositives de documentation, etc... Au sein du modèle potentiel la CAO et l'EAO se rejoignent donc concrètement et s'imbriquent étroitement. Nous proposons à la figure XIII, le modèle potentiel complet d'un système d'enseignement de l'électrotechnique orienté vers la conception de moteurs asynchrones.

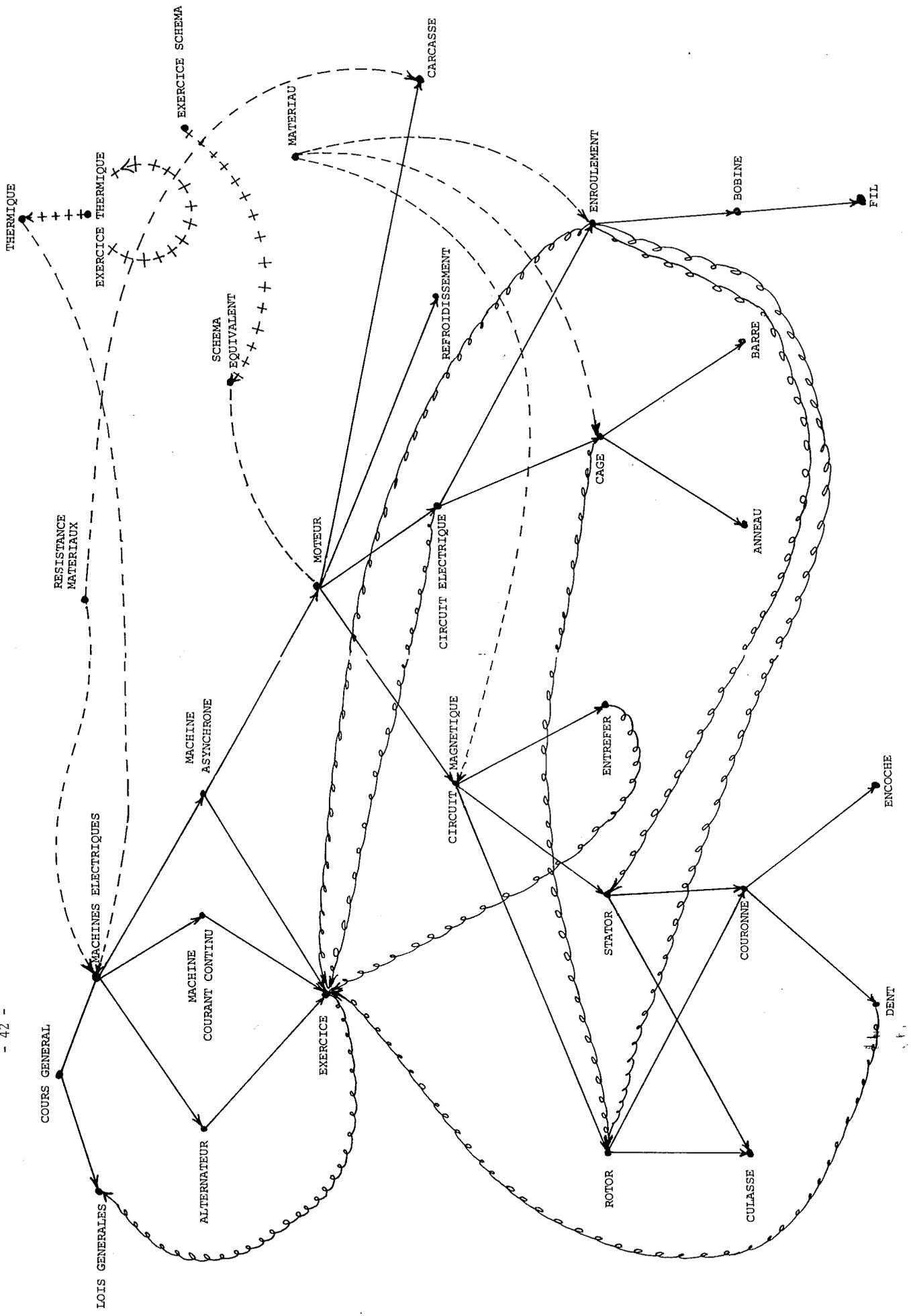


Figure XIII : Le modèle général

3. LES RESSOURCES DE SIMULATION

=====

Au sein du modèle d'un système d'ECAO que nous avons présenté à la fig. IV, les "informations connues du système" sont décomposées en trois sous-ensembles : le modèle potentiel, le travail de l'étudiant (que nous venons de définir) et les ressources de simulation.

Ces ressources seront utilisées par l'étudiant afin d'évaluer les performances de l'objet qu'il aura conçu. L'impact pédagogique de cette phase de conception a été analysé en détail par A. BOLOPION qui en a montré les qualités fondamentales.

Il est donc essentiel que l'enseignant puisse disposer d'une banque d'algorithmes de simulation aussi détaillée que possible. Il paraît impossible que ce soit l'enseignant lui-même qui conçoive chacun des programmes de simulation dont il aura besoin. C'est pourquoi le pédagogue pourra intégrer à son système les ressources algorithmiques très importantes qui ont déjà été conçues pour d'autres besoins. Les sciences de l'ingénieur ont fait des progrès récents dans le domaine de l'utilisation de simulations sur ordinateur au point qu'actuellement chacun peut disposer d'importantes ressources en logiciel. Ces programmes, généralement écrits en FORTRAN, devront pouvoir être facilement adaptés aux besoins d'un système d'ECAO. L'efficacité du système dépendra de l'importance des contraintes qui seront attachées aux caractéristiques des ressources de simulation.

Nous décrirons au chapitre III la solution que nous proposons pour résoudre ce problème.

4. L'INTERFACE ENTRE L'ETUDIANT ET LE SYSTEME

=====

Cette interface est décomposable en trois éléments indépendants : le poste de travail, les ressources d'édition, le langage utilisé par l'étudiant

4.1. LE POSTE DE TRAVAIL

Le choix de la configuration d'un poste de travail doit incomber à l'enseignant qui adaptera aux besoins pédagogiques, le matériel mis à la disposition de l'étudiant.

Au niveau de notre système, nous ne ferons donc aucune hypothèse concernant les périphériques qui composeront un poste de travail. Nous supposerons que tout poste peut être configuré de façon conversationnelle et peut être modifié alors que le système d'ECAO est en fonctionnement. L'étudiant pourra ainsi immobiliser des périphériques coûteux (tels que traceur de courbe, imprimante, lecteur de cartes, projecteur de diapositives) seulement lorsqu'il en aura besoin. Cette souplesse de gestion permettra d'optimiser la configuration d'un ensemble de postes de travail qui disposeront d'un "pool" de périphériques communs.

4.2. LES RESSOURCES D'EDITION

Au cours de la définition du modèle potentiel, nous avons parlé d'entités pédagogiques telles que des cours, des exercices, des banques d'informations. Jusqu'à présent ces entités n'étaient que des concepts dépourvus de leur contenu (textes des cours, énoncés des exercices, etc...). Nous supposerons que tous ces textes sont regroupés dans une ressource unique d'édition et qu'ils sont enregistrés indépendamment des autres éléments du système.

Le contenu de cette ressource commune d'édition est géré, bien entendu, par l'enseignant à l'aide d'un langage très simple qui lui permet de réaliser une mise au point de tous les textes pédagogiques du système de façon autonome sans avoir à retoucher les programmes dans lesquels il les utilise.

4.3. LE LANGAGE MIS A LA DISPOSITION DES ETUDIANTS

Le choix d'un langage de communication mis à la disposition de l'étudiant est fort délicat. Un tel langage peut être optimisé dans le détail lorsque l'on connaît à l'avance la matière à laquelle il est destiné. Mais pour notre propos, nous ne pouvons absolument pas préjuger des mots clés de ce langage ni de sa sémantique. C'est pourquoi, nous ne ferons que des hypothèses sur la syntaxe de ce langage. Nous utiliserons ici essentiellement les résultats des recherches que nous avons conduites au sein de notre équipe sur la définition de langages appropriés à l'ECAO.

- Un tel langage doit vérifier les deux principes fondamentaux suivants :

a) il doit pouvoir être conçu aisément par chaque enseignant construisant un système d'ECAO.

b) il doit pouvoir évoluer indépendamment du système d'ECAO avec lequel il permet de communiquer.

Ces deux contraintes sont remplies par un interpréteur générateur de langage dont nous préciserons les caractéristiques dans le chapitre III.

5. LA STRATEGIE PEDAGOGIQUE

=====

Chaque commande entrée par l'étudiant pose un problème dont l'enseignant doit avoir prévu la résolution. L'ensemble des méthodes utilisées pour résoudre tous les problèmes qu'un étudiant pourra poser au système constitue ce que nous dénommons la stratégie pédagogique.

L'enseignant qui conçoit un système d'ECAO n'est généralement pas un spécialiste des techniques de résolution de problèmes. Or les problèmes que l'ont doit traiter au sein d'un système d'ECAO sont généralement très complexes puisqu'ils imposent de surcroît de nombreuses contraintes pédagogiques à des processus de conception assistée par ordinateur déjà fort élaborés. L'aide que nous devons apporter à l'enseignant dans ce domaine doit concerner tout d'abord la phase d'analyse qu'il devra effectuer. Il nous faudra ensuite rendre facilement accessibles les outils que nous mettrons à sa disposition pour décrire sa stratégie.

Les problèmes que l'enseignant doit résoudre au niveau de la stratégie pédagogique ont de nombreuses caractéristiques communes

En effet, lorsqu'une commande est entrée par un utilisateur il faut successivement :

- tester la cohérence des paramètres de la requête :
 - a) par rapport à la logique propre à cette commande ;
 - b) par rapport au travail précédemment réalisé par l'étudiant :
- choisir ensuite une méthode de résolution adaptée à la structure de l'ébauche que l'étudiant aura réalisée au cours de ses travaux précédents.
- construire l'enchaînement des ressources de simulation.
- récupérer les paramètres nécessaires à tous les calculs.
- lancer finalement l'exécution des simulations sélectionnées.

De façon globale, ce type de traitement peut être décomposé en une longue phase d'analyse et de tests suivie d'une exécution d'algorithmes.

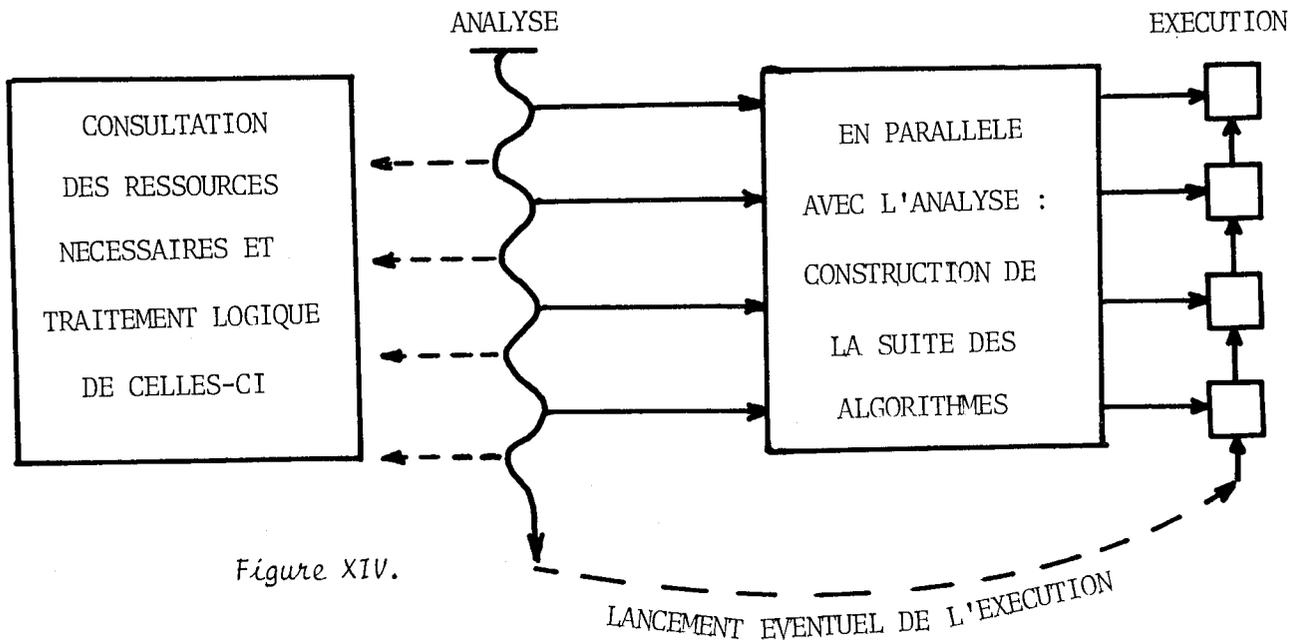


Figure XIV.

5.1. DECOMPOSITION DE LA STRATEGIE PEDAGOGIQUE

Nous pouvons dégager quatre processus fondamentaux au sein de la stratégie pédagogique :

- un processus de consultation et de gestion des ressources ;
- un processus d'analyse des renseignements fournis pour la consultation des données ;
- un processus de construction d'un enchaînement d'algorithmes de calcul et de simulation destiné à résoudre le problème posé ;
- l'exécution de l'enchaînement précédemment constitué.

Une analyse détaillée de chacun de ces processus est nécessaire :

- la gestion et la consultation des diverses ressources du système au sein de la stratégie pédagogique offrent peu de difficultés. En effet cela peut toujours se ramener à un problème de gestion de banques de données. La consultation du travail de l'étudiant, vu à travers la lentille que constitue le modèle potentiel, pourra s'effectuer à l'aide d'un langage très simple aisément assimilable par le concepteur. La gestion des ressources d'éditions présente un peu plus de difficultés lorsque l'on veut assurer une "personnalisation" des messages ainsi qu'un contrôle assez

fin de leur fréquence d'édition. La possibilité de construire une suite de messages munis chacun d'une priorité devrait permettre d'assurer des éditions très souples adaptées au processus de résolution de problèmes. Par ailleurs, il sera possible d'instaurer un dialogue entre le processus d'analyse et l'étudiant qui pourra ainsi intervenir au sein du processus de résolution de problèmes. D'autre part des éditions concernant les caractéristiques du travail réalisé par l'étudiant seront disponibles au sein du langage qu'utilisera l'enseignant pour décrire sa stratégie.

- Le processus d'analyse du problème posé disposera donc de toutes les informations nécessaires. Le traitement de ces informations peut être facilement résolu par l'utilisation d'un langage permettant le traitement d'expressions logiques classiques. Mais la conception du traitement d'un problème complexe ne peut être menée à bien sans une décomposition en sous-problèmes et sans une analyse structurale de la démarche de résolution. Nous nous sommes donc particulièrement attachés à développer des outils destinés à faciliter le travail de l'enseignant en l'incitant à utiliser une décomposition très fine du processus d'analyse. Nous les examinerons de plus près au paragraphe suivant.

- La résolution du problème proprement dit (généralement une simulation) pourra rarement être réalisée par un seul algorithme. Dans ce cas il faudrait en effet disposer un programme associé à chaque problème. Or le nombre de problèmes posés dépend de la structure définie par l'étudiant qui, comme nous l'avons remarqué, peut être fortement combinatoire. Le nombre de programmes nécessaires serait alors excessif. Nous proposons donc à l'enseignant de construire un enchaînement combinatoire de petits programmes de simulation qui pourront ainsi s'adapter à tous les cas possibles. La construction de cet enchaînement devra être réalisée en parallèle avec l'analyse du problème.

- L'exécution de l'enchaînement des programmes sélectionnés au cours de l'analyse est aisée à mettre en oeuvre. Par contre, elle nécessite un traitement délicat sur le plan informatique : le système devra prendre totalement à sa charge la communication des données et ses résultats d'un programme à l'autre sans que cela ne crée des contraintes trop importantes au niveau de la programmation des algorithmes concernés. Une solution technique sera donnée à ce problème au chapitre III.

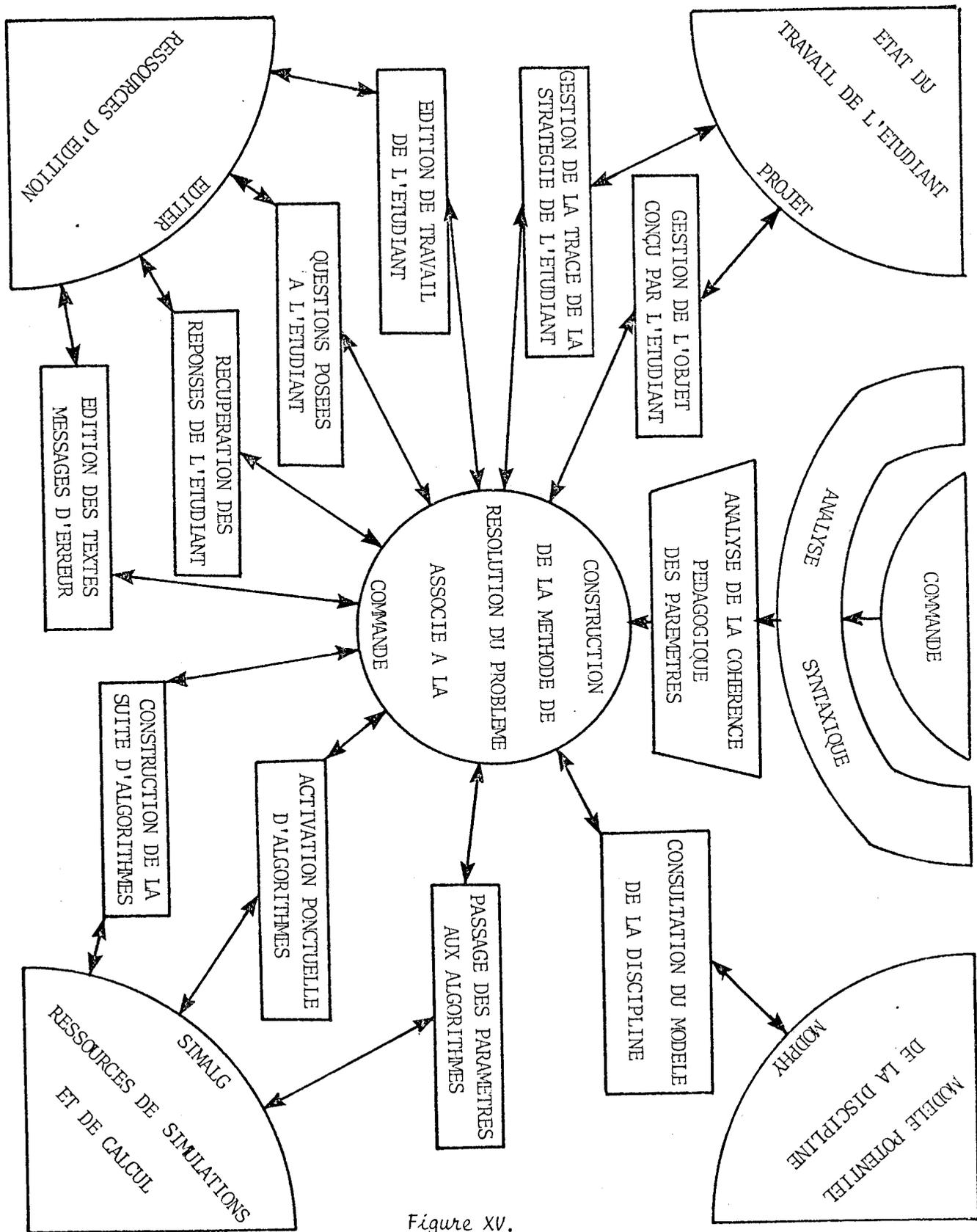


Figure XV.

Nous avons représenté à la figure XV, les diverses interactions entre la stratégie pédagogique et les ressources du système. Nous allons maintenant détailler les outils d'analyse modulaire de problèmes que nous avons développés.

5.2. BLOC ELEMENTAIRE D'ANALYSE STRATEGIQUE

Nous pouvons décomposer la stratégie pédagogique en blocs élémentaires constitués chacun par un sous-ensemble logique des quatre processus que nous venons de définir. Nous remarquerons que les processus d'analyse, de consultation et de construction de la suite d'algorithmes sont simultanés, alors que le processus d'exécution de la simulation est un processus différé, nous les avons représentés à la figure XVI.

Définition

Tout sous-ensemble des trois processus simultanés : analyse, consultation, construction, sera appelé un Bloc Élémentaire d'Analyse Stratégique (BEAS). Il lui sera associé le fragment du processus d'exécution correspondant.

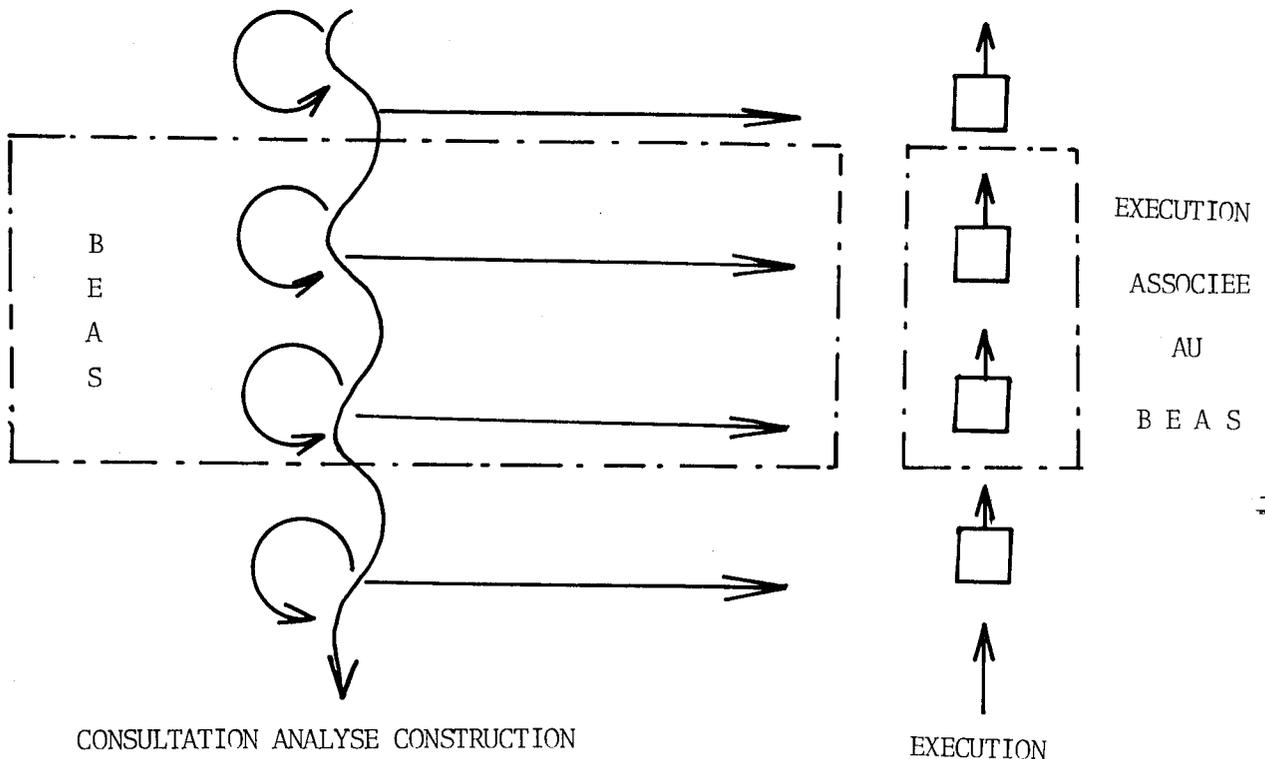


Figure XVI.

Nous supposons que tous les BEAS sont disjoints et nous pourrions donc caractériser chacun d'eux par un nom qui lui sera propre.

Chaque phase de la stratégie pédagogique est représentable macroscopiquement par un enchaînement de BEAS.

La réunion de tous les BEAS et de tous les enchaînements possibles de ces blocs représente donc globalement toutes les stratégies utilisables pour résoudre éventuellement divers problèmes.

Le concepteur devra donc s'attacher à concevoir en premier lieu l'enchaînement des BEAS qui lui seront nécessaires pour résoudre des problèmes. Il pourra définir ultérieurement le contenu détaillé de chaque bloc.

5.3. DIVERS TYPES D'ENCHAINEMENTS DE BEAS

La dualité "analyse exécution" que nous rencontrons entre un BEAS et l'exécution qui lui est éventuellement associée, va induire deux types d'enchaînements de Blocs.

- Enchaînement avec exécution différée.: Nous supposons qu'une succession de BEAS construit la même suite d'algorithmes qui ne sera activée qu'à la fin de tous les blocs enchaînés : figure XVII.

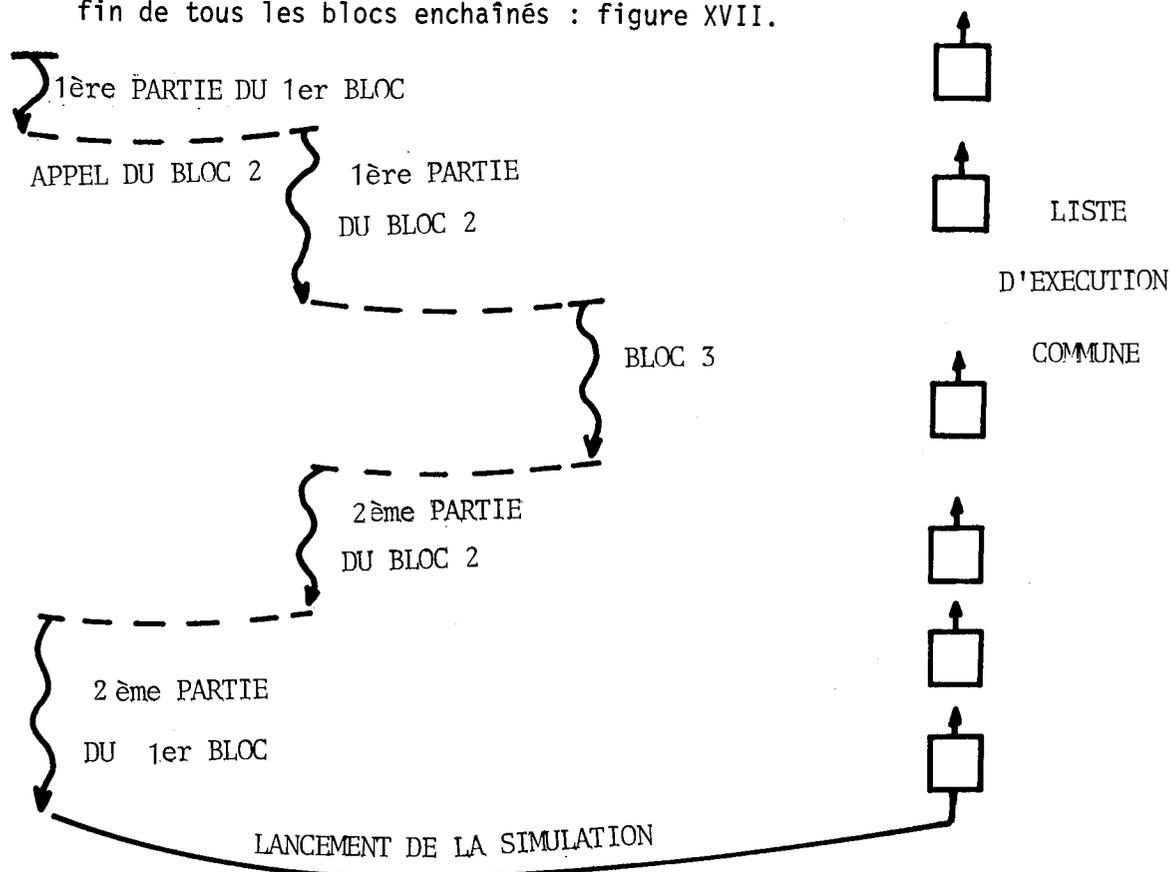


Figure XVII : Enchaînement avec exécution différée.

- Lancement avec exécution immédiate : La construction de la solution d'un problème complexe doit nécessiter un certain nombre de calculs intermédiaires dont peut dépendre la structure de la suite d'algorithmes nécessaires à la résolution. Des exécutions immédiates sont parfois nécessaires. Le deuxième type d'enchaînement des BEAS permet un tel traitement. Nous avons représenté à la figure XVIII un enchaînement alterné d'analyse exécution qui illustre la complexité des stratégies que l'on peut construire avec ces outils.

5.4. DESCRIPTION DE LA STRATEGIE PEDAGOGIQUE

Nous avons défini un langage qui permet à l'enseignant de décrire sa stratégie pédagogique conformément aux principes que nous venons de dégager. Le détail de ce langage sera abordé au chapitre suivant.

6. LE MODELE COMPLET

Le modèle que nous proposons pour décrire un système d'ECAO (fig. IV p. 23), permet de matérialiser chaque étape logique du processus de conception d'un tel système. L'enseignant pourra disposer d'un outil adapté à la construction de chacun de ces éléments.

a) Le langage de commande : A l'aide du générateur de langage, l'enseignant décrira le langage de commande qu'il aura conçu. Une mise au point pédagogique de ce langage pourra être effectuée indépendamment des autres composants du système. L'interpréteur de langages prendra alors à sa charge l'analyse syntaxique de toute commande entrée par l'étudiant.

b) Les ressources de simulation : L'enseignant devra sélectionner les algorithmes de simulation et de calcul nécessaires à son système. Les programmes correspondants pourront être écrits par ailleurs, chacun indépendamment des autres ressources du système.

c) Le modèle potentiel de la discipline et de son support sera décrit par l'enseignant à l'aide d'un langage approprié : le Langage MODPHY. Le système CASCADE activé par ce langage gèrera la banque de données correspondante.

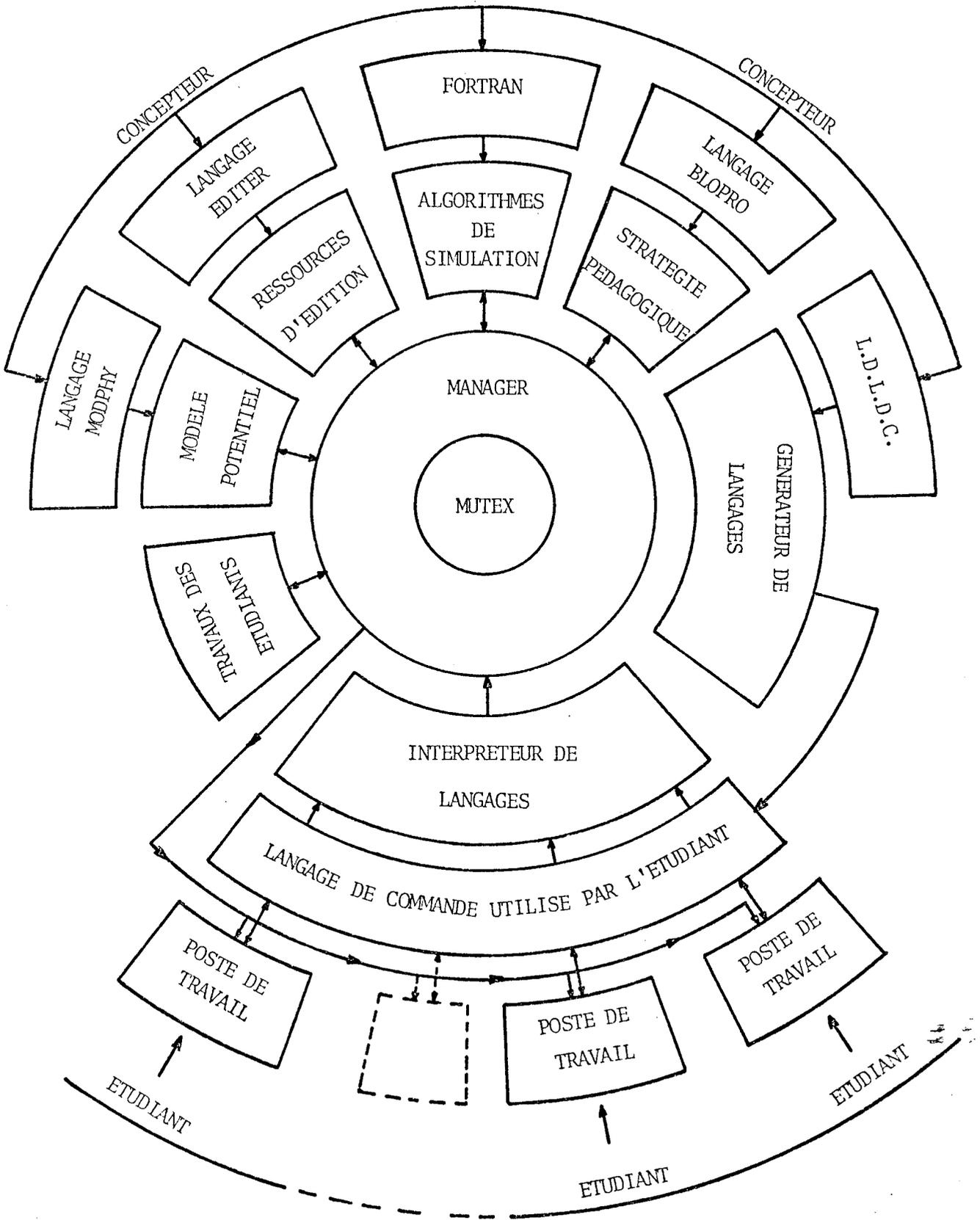


Figure XIX. Le système CASCADE

d) La banque des ressources d'édition sera définie et gérée par le concepteur, grâce au langage EDITER. Le système CASCADE prendra alors à sa charge la gestion automatique de tous les textes préalablement entrés dans cette banque.

e) La stratégie pédagogique consulte en particulier toutes les ressources précédentes et gère la banque PROJET. Elle sera décrite par le concepteur à l'aide d'un langage de programmation : le langage BLOPRO. Il permettra de définir séparément chaque BEAS et de décrire globalement leur enchaînement.

f) La communication entre les divers composants d'un système décrit à l'aide de CASCADE, sera prise en charge par le noyau de CASCADE : le macrosuperviseur MANAGER qui assurera la gestion globale de toutes ces ressources.

LA STRUCTURE GENERALE DU SYSTEME CASCADE

Une analyse méthodique des caractéristiques d'un système d'ECAO nous a permis de proposer un modèle général d'un tel système.

En parallèle, une étude approfondie de la technique de conception d'un système d'ECAO nous a conduit à particulariser chaque étape de sa construction, à en extraire les principes généraux pour pouvoir définir ses composantes immuables.

La conception d'un outil informatique prenant à sa charge toute la systématique d'une telle approche est dès lors possible : Nous présentons donc en conclusion de ce chapitre la structure générale du système CASCADE (fig.XIX) dont nous aborderons la description détaillée au chapitre suivant.

CHAPITRE III
=====

LE SYSTEME CASCADE

I - <u>INTRODUCTION</u>	
II - <u>LES BESOINS EN INFORMATIQUE DE BASE</u>	
1. Mini-Ordinateur ou Ordinateur	
2. Les options du système CASCADE.	
III - <u>LES LANGAGES DU SYSTEME CASCADE</u>	
1. Terminologie et notations.	
IV - <u>DEFINITION D'UN MODELE POTENTIEL</u>	MODPHY
V - <u>CREATION DES RESSOURCES D'EDITIONS</u>	EDITER
VI - <u>DEFINITION D'UN LANGAGE DE COMMANDE</u>	LDLDC
VII - <u>DESCRIPTION DE LA STRATEGIE PEDAGOGIQUE</u>	BLOPRO
VIII - <u>DEFINITION DES MODULES DE SIMULATION</u>	

1. INTRODUCTION

=====

Le système que nous avons défini dans les chapitres précédents est fondé sur de nombreuses hypothèses pédagogiques. Leur vérification implique la construction d'un prototype et l'expérimentation de son fonctionnement. Dans ce chapitre, nous allons préciser les caractéristiques du système CASCADE en introduisant les contraintes de réalisation dont nous nous étions jusqu'alors affranchis.

La plupart des options pratiques que nous avons choisies lors de la réalisation de notre prototype sont étayées sur les résultats que A. Bolopion a établis au cours de l'utilisation du système ESPACE [42] et sur notre expérience personnelle de ce système [41].

Sur le plan informatique un travail d'équipe nous a permis de définir en détail les outils de base que nécessite l'implantation d'un système d'ECAO sur une petite configuration. Ces outils ont été élaborés par M. Campmas [45] sur un ordinateur Télémécanique T1600. Nous les avons intégrés dans notre système en cherchant à évaluer leurs performances et à les utiliser au maximum de leurs possibilités. Une justification détaillée des options pratiques fondamentales de notre système se trouve dans les deux thèses que nous avons citées plus haut.

Tenant compte des travaux de nos prédécesseurs nous ne présenterons que très brièvement les fondements des principaux choix techniques de notre système :

Le choix du matériel et du logiciel est toujours lié à des critères techniques et économiques, cependant les options pédagogiques que nous avons choisies ont déterminé les principales caractéristiques du matériel sur lequel nous avons implanté notre prototype, et ont beaucoup influé sur la réalisation du prototype lui-même.

2. LES BESOINS EN INFORMATIQUE DE BASE

=====

2.1. MINI-ORDINATEUR OU ORDINATEUR

De nombreux critères président au choix d'un ordinateur ou d'un mini-ordinateur. Nous comparerons succinctement ces deux types de matériels en nous fondant sur les critères pédagogiques et informatiques que nous avons retenus.

2.1.1. Décentralisation

Le développement de l'ECAO tel que nous le concevons passe nécessairement par la construction de systèmes décentralisés conçus au sein même des établissements d'enseignement où ils seront utilisés. Cette décentralisation va dans le même sens que celle qui s'est annoncée sur le plan du matériel :

- Développement de réseaux d'ordinateurs accessibles à partir de nombreux points et,

- Création de petits centres de calcul qui assurent également quelquefois la fonction de terminaux lourds pour les réseaux.

2.1.2. Facilité d'accès pour le concepteur

L'interaction étroite que CASCADE crée entre le concepteur et son système rend le mini-ordinateur plus agréable d'utilisation, ne serait-ce que par la démystification que produit l'accès direct à une machine. D'autre part certaines phases de conception nécessitent l'utilisation de périphériques élaborés (lecteur de carte, dérouleurs de bandes magnétiques, imprimante rapide). Le concepteur devra donc intervenir sur les lieux mêmes du centre de calcul, ce qui peut être délicat à réaliser dans l'utilisation d'un gros ordinateur et surtout d'un réseau.

2.1.3. Souplesse de gestion

La gestion d'un système d'ECAO pose toujours des problèmes d'interférence entre des emplois du temps toujours très rigides et une utilisation de l'ordinateur non moins planifiée.

L'enseignement aura de fortes chances de voir aboutir ses requêtes auprès de la gestion très souple d'un mini-ordinateur, alors qu'il devra se plier aux exigences d'un gros système d'exploitation.

2.1.4. Temps de réponse et pédagogie

Le temps de réponse d'un système d'ECAO est une notion très difficile à définir de manière détaillée. Il paraît évident au premier abord que les performances d'un ordinateur puissant dépassent largement celles d'un mini-ordinateur, on pourrait en conclure que seul un grand centre de calcul peut répondre aux exigences d'un système d'ECAO élaboré. Mais il faut prendre en considération d'autres facteurs :

Le coût élevé de l'utilisation d'un gros centre de calcul vient introduire une contrainte économique dont on doit tenir compte. Des procédures complexes sont quelques fois nécessaires pour utiliser un système important (lancement du système, chargement des programmes),

elles consomment un temps non négligeable et alourdissent l'utilisation pédagogique de l'ordinateur. Enfin, le temps de réponse n'est pas en lui-même un critère d'évaluation des performances pédagogiques d'un système d'EAO. La définition des ratios significatifs est encore du domaine de la recherche. L'expérience seule pourra amener un élément de réponse à travers la comparaison de l'efficacité pédagogique d'une même application d'EAO implantée sur deux configurations différentes.

2.1.5. Adaptation des postes de travail

La définition d'un poste de travail ne peut pas se faire indépendamment du système d'exploitation sous lequel se déroule l'EAO. Or un gros centre de calcul est souvent muni d'un système d'exploitation plus rigide qui rend difficile l'utilisation optimale de certains périphériques (notamment les terminaux graphiques qui nécessitent des vitesses de transmission élevées). L'utilisation d'un mini, voire d'un micro-ordinateur affranchit généralement le concepteur de cette contrainte.

Les critères pédagogiques que nous avons considérés, nous font préférer le mini-ordinateur à une grosse configuration de matériel. Nous allons voir qu'il en est autrement pour les critères informatiques.

2.1.6. Capacité mémoire

La faible capacité mémoire d'un mini-ordinateur contraint généralement le concepteur à restreindre la complexité des problèmes traités ou à utiliser des méthodes de segmentation lorsqu'elles sont applicables. Cette restriction inévitable augmente généralement un temps de réponse déjà lourd.

2.1.7. Temps de traitement pour les tâches complexes

La construction d'un système d'EAO nécessite des analyses complexes de graphes qui demandent beaucoup d'opérations. Sur petit ordinateur le temps de traitement sera assez long et le concepteur devra quelquefois prendre patience. Sur une grosse machine ce défaut ne sera pas autant ressenti : les tâches les plus lourdes ne dépasseront que très rarement la seconde.

2.1.8. Logiciels de base et langages évolués

Le langage évolué le plus couramment disponible sur petit ordinateur est "FORTRAN". Or ce langage est mal adapté au traitement des chaînes de caractères et ne permet pas une réservation dynamique de mémoires ; en outre une programmation structurée est plus facilement réalisable dans des langages de niveau plus élevé tels que PL1, ALGOLW, LISP, disponibles sur les gros ordinateurs : l'écriture d'un système tel que CASCADE en FORTRAN ou dans un langage de plus bas niveau, est en fait fastidieux.

La solution de facilité consisterait donc à opter pour l'implantation de CASCADE sur un gros ordinateur. Nous avons cependant choisi l'autre direction en faisant prévaloir nos options pédagogiques sur les avantages techniques que présentait une grosse machine.

2.1.9. Temps partagé ou multi-programmation ?

Pour gérer un enseignement assisté par ordinateur, un système multi-utilisateurs est nécessaire. Mais il faut choisir entre un système multi-programmation ou en temps partagé. Un système en temps partagé est indispensable lorsque le principal type d'utilisation de l'ordinateur nécessite le passage de longs programmes.

Nous remarquerons que sous le système CASCADE, de longs calculs peuvent être effectués mais ils sont toujours réalisés par une succession de nombreux programmes distincts qui s'enchaînent mutuellement. Un système de multi-programmation est donc compatible avec nos besoins en calcul. D'autre part une gestion en temps partagé, de construction plus complexe, risque de grever le temps de réponse de notre système lorsque ce dernier ne sera pas trop surchargé. C'est pourquoi nous avons opté pour l'utilisation d'un système multi-programmation.

2.2. LES OPTIONS DU SYSTEME CASCADE

Après avoir choisi la machine sur laquelle nous allions implanter nos programmes : le T 1600 ou SOLAR de la Télémécanique Electrique,-et défini le système d'exploitation que nous utiliserions : le système multi- consoles MUTEX, il nous restait à préciser les grandes options de notre prototype : Portabilité, Modularité, Interprétation.

2.2.1. Portabilité

Définir un système totalement portable et implantable sur mini-ordinateur est assez délicat. Une solution simple consiste à le programmer intégralement en langage "FORTRAN". Mais avec ce langage il est impossible de s'affranchir totalement des particularités de la machine sur laquelle on travaille notamment dans trois cas précis :

- Relations avec le système multi-utilisateurs MUTEX |45|
- Gestion des mémoires auxiliaires (DISQUE) GESTRA |45|
- Traitement des langages du système. INTGEN |45|

Pour chacun de ces cas, nous avons défini une interface très détaillée avec les programmes écrits en langage de niveau inférieur (PL 1600) et nous avons restreint ces programmes à un strict minimum : seulement 5/100 des programmes ont été écrits en langage PL 1600 et nous pensons pouvoir encore réduire leur volume pour une version ultérieure.

2.2.2. Modularité

Tout le système CASCADE a été décomposé en modules de faible taille (tous inférieurs à 20 K octets) de telle sorte que la mise au point du prototype soit aisée et que le système ainsi conçu puisse évoluer facilement. Le système CASCADE se compose d'une centaine de modules écrits en langage FORTRAN. Seuls trois modules ont été programmés en PL 1600.

2.2.3. Interprétation

Le système CASCADE comporte quatre langages. La portabilité est plus difficile à obtenir avec des techniques de compilation qu'avec une interprétation. Une précompilation générant du FORTRAN était

possible [11] mais aurait introduit de la lourdeur dans l'emploi conversationnel de notre système surtout au niveau de la stratégie pédagogique qui est très évolutive. Pour permettre des modifications aisées et ponctuelles au sein de tous les programmes écrits dans les langages de CASCADE, nous avons préféré utiliser des techniques d'interprétation qui sont également plus adaptées à une utilisation conversationnelle.

3. LES LANGAGES DU SYSTEME CASCADE

=====

La conception d'un système d'ECAO avec CASCADE peut être décomposée en cinq étapes :

- Définition du modèle potentiel
- Définition des ressources d'édition
- Définition du langage mis à la disposition de l'étudiant
- Définition de la stratégie pédagogique
- Définition des modules de simulation.

Nous étudierons dans un premier temps les langages qui permettent de procéder à la création de chacun de ces éléments et nous proposerons en parallèle des extraits de la construction d'un système d'ECAO illustrant la conception de moteurs asynchrones.

3.1 TERMINOLOGIE ET NOTATIONS

Dans les paragraphes qui suivent, nous allons décrire quatre langages de commande qui possèdent tous une structure arborescente.

Nous représentons ces langages à l'aide de tableaux dans lesquels nous adopterons les notations suivantes :

a) Nom de commande

Le premier élément de toute phrase d'un langage doit être mot-clé du langage, c'est-à-dire un ensemble de caractères alphanumériques composé au minimum d'un caractère, sans blanc, et dont seuls les trois premiers caractères, lorsqu'ils existent, sont analysés.

Nous noterons ceci en écrivant les trois premiers caractères du nom en majuscules soulignées, les caractères suivants étant reproduits en minuscules.

Exemple : CREer SUPprimer

b) Modificateur

C'est le premier élément de toute branche d'une commande : c'est-à-dire, soit un mot-clé du langage, soit le modificateur vide (absence de modificateur).

Nous utiliserons la même notation que pour le nom de commande.

Exemple : ENTité

Le modificateur vide sera noté par la lettre ϕ .

Exemple : ϕ

c) Mot Ignorable

C'est un mot-clé du langage qui peut être omis au sein d'une commande. Nous le représenterons par ses trois premières lettres en majuscules mises entre crochets suivies des autres lettres en minuscules.

Exemple : [POU]r [MAX]imum

d) Séparateur

C'est un caractère non alphanumérique (= < > etc...), intervenant dans une commande. Nous le représenterons par un rectangle contenant le caractère.

Exemple : [;] [=]

Le séparateur vide (absence de séparateur) sera noté [ϕ]

e) Paramètre

Les paramètres d'une commande seront notés de la façon suivante :

- entier : <CI>

- réel : <CF>

- alphanumérique : <CA_n>

↑
nombre de lettres retenues.

Les regroupements logiques au sein d'une commande seront figurés par une accolade.

4. LE LANGAGE MODPHY

=====
Ce langage permet la définition d'un modèle potentiel. Nous l'avons conçu pour une utilisation conversationnelle, mais un mode de type "MONITEUR" est disponible lorsque le concepteur préfère enregistrer la description de son modèle sur cartes, bande magnétique ou disque.

Le système prend à sa charge l'analyse de la cohérence de chaque commande entrée par l'enseignant qui reste libre de décrire son modèle comme il le désire.

A tout moment le concepteur dispose de commandes d'édition qui lui permettent de visualiser le modèle décrit.

Le concepteur peut également demander une analyse globale du modèle qu'il a construit. Une carte de la structure entrée lui est alors fournie par le système qui contrôle les propriétés de cette structure : connexité de la hiérarchie, absence de cycle, compatibilité des liaisons par rapport à la hiérarchie, disjonction des sous hiérarchies, etc...

Nous présenterons tout d'abord la syntaxe du langage MODPHY. Nous proposerons alors un extrait de la construction du modèle potentiel d'un moteur asynchrone. Nous donnerons enfin des exemples de détection d'erreurs et d'analyse globale du travail du concepteur par le système.

4.1. SYNTAXE DU LANGAGE MODPHY

4.1.1. Structure des instructions

Chaque instruction de ce langage est décomposable en trois champs (les deux derniers pouvant être absents selon les instructions).

a) Le champ ACTION

qui définit le type de l'action que va entraîner la commande.

b) Le champ REFERENCE

qui définit le concept concerné (Entité, Liaison, Hiérarchie ou Sous-hiérarchie).

c) Le champ CARACTERISTIQUE

qui précise la caractéristique qui est concernée par la commande.

Nous avons reproduit aux pages 29 et 32 (Ch. II) la décomposition des deux concepts principaux qui sont à la base de la construction d'un modèle : l'entité et la liaison. Les commandes du Langage MODPHY vont permettre de créer des réalisations de ces concepts et de définir précisément leur caractéristiques.

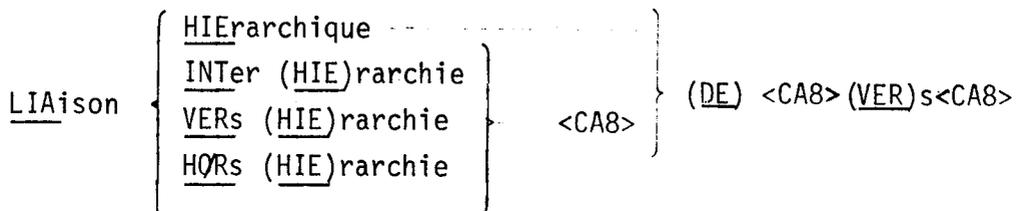
4.1.2. Notation

Nom d'une entité : chaque entité est définie par un nom de 1 à 8 lettres que nous représenterons par <NOMENT> .

Référence à une liaison

Une liaison est définie par :

- son type : HIERARCHIQUE, INTER-HIERARCHIQUE, VERS HIERARCHIE, HORS HIERARCHIE.
- Son nom (1 à 8 caractères) sauf pour les liaisons hiérarchiques dont le nom est confondu avec le type.
- Le nom de l'entité origine.
- le nom de l'entité extrémité.



Exemples de liaisons :

- LIAISON HIERARCHIQUE DE MOTEUR VERS CIRCMAGN
- LIAISON INTER HIERARCHIQUE BOBINER DE ENROULEM VERS STATOR
- LIAISON VERS HIERARCHIE REPRESEN DE SCHEMAEQ VERS MOTEUR

Nous noterons une référence à une liaison par <REFLIA>.

Remarque : l'énoncée du type d'une liaison est obligatoire lors de la création de cette liaison ; il devient facultatif au sein des autres commandes.

Nom de propriété :

C'est un mot de 1 à 12 caractères que nous représenterons par <NOMPRO>.

Types de propriété :

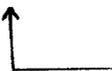
Les types reconnus par MODPHY sont :

<u>ENTier</u>	variable entière simple
<u>FLOttant</u>	variable réelle simple
<u>LOGique</u>	variable logique simple
<u>ALPhanumérique</u>	variable comportant de 1 à 8 caractères

TABleau { ENTier
FLOttant
LOGique } DIMension <CI>

La dimension doit être une constante entière positive

COUrbe DIMension <CI>



définit le nombre maximum de points de la courbe

Nous noterons l'un quelconque de ces types par <TYPRO>

4.1.3. Les instructions du langage

* Création d'entités

CREer <CA8> [STATistiques <CI>] ;

Exemples :

CREER MOTEUR ;
CREER ROTOR STA 2 ;

La caractéristique statistique définit le nombre moyen d'occurrences de l'entité susceptibles d'être créées pendant le travail de l'étudiant. Si cette caractéristique est absente, le système l'initialisera à 4 ;

* Création de liaison

CREer <REFLIA> [IDentifiable { ORIGine
EXTrémité }] [STATistique <CI><CI>] ;

Exemples :

CREER LIAISON INTER HIERARCHIQUE BOBINER DE ENROULEMENT VERS STATOR ;
CREER LIAISON HIERARCHIQUE DE ENROULEMENT VERS BOBINE IDENTIFIABLE EXTREMITÉ ;
(la notion d'identification a été définie au chapitre II)
CREER LIAISON VERS HIERARCHIE CONSULTE DE EXERCICE VERS MOTEUR STATISTIQUE 4 6 ;

L'instruction précédente crée la liaison "CONSULTE" de type "Vers Hiérarchie" et précise qu'en moyenne chaque structure représentant le travail d'un étudiant pourra comporter 4 liaisons "CONSULTE" partant d'une occurrence d'exercice et 6 liaisons "CONSULTE" arrivant à une occurrence de moteur (ceci permet d'optimiser la banque de données représentant le travail de l'étudiant).

* Création de propriété

CREer POUr <NOMENT><TYPRO><NOMPRO> [IDentifiable] ;

Rappel : La prépondérance d'une hiérarchie impose que les extrémités de toute liaison inter-hiérarchique appartiennent à une même hiérarchie.

* Libération d'une hiérarchie :

LIBerer { HIErarchie ;
SOUs HIErarchie <NOMENT> ;

Les notions d'appartenance et de prépondérance de cette hiérarchie sont supprimées par cette instruction

* Analyse de la hiérarchie :

ANALyser [LA] HIErarchie ;

Permet d'analyser la cohérence du modèle potentiel créé.
Détection de toutes les incohérences logiques qui pourraient exister.
Édite une carte détaillée du modèle entré.

* Traitement des contraintes sur les nombres d'occurrences

Positionnement des contraintes d'entités :

LIMiter POUr <NOMENT> NOMbre [OCCurrence] [A] <CI> ;

Exemple :

LIMITER POUR NOYAU NOMBRE OCCURRENCE A 3 ;

Positionnement des contraintes de liaison :

LIMiter POUr <REFLIA> NOMbre [OCCurrence] { DIRecte
INVerse } [A] <CI> ;

Exemple :

LIMITER POUR LIAISON RENVOI DE EXOFEM VERS EXOCHAMP NOMBRE DIRECT A 2 ;

LIMITER POUR LIAISON HIERARCHIQUE DE CIRCMAgn VERS NOYAU NOMBRE DIRECT A 3 ;

Suppression des contraintes d'occurrence :

LIBérer POUr <NOMENT> NOMbre [OCCurrences] ;

LIBérer POUr <REFLIA> NOMbre [OCCurrences] { DIrectes
[INverses] ;

* Traitement des contraintes de propriétés :

Bornes

BORner POUr <NOMENT> PROpriété <NOMPRO> (ENTre) <CF>

[ET] <CF> [PREcision] <CF> ;

Exemple :

BORNER POUR MOTEUR PROPRIETE FREQUENCE ENTRE 0.0 ET 1000. PRECISION 0.00001 ;

Listes d'admission ou d'exclusion (cf CH II page 30)

Commande de définition d'une liste :

INTerdire }
 } POUr <NOMENT> PROpriété <NOMPRO> ;
AUToriser }

Cette instruction précise la propriété à laquelle va se rapporter la liste. Le système émet alors les caractères => il suffit alors d'entrer la liste des valeurs (Format libre).

Le nombre n'est pas limité.

La liste doit être terminée par OK.

Exemple :

AUTORISER POUR ENCOCHE PROPRIETE TYPE ;

=> FONDPLAT

=> FONDROND

=> OK

Suppression des bornes ou d'une liste d'exclusion ou d'admission pour une propriété :

SUPprimer POUr <NOMENT> PROpriété <NOMPRO> $\left. \begin{array}{l} \text{CON} \text{traintes} \\ \text{BOR} \text{nes} \end{array} \right\}$;

Exemple :

SUPPRIMER POUR MOTEUR PROPRIETE FREQUENCE BORNES ;

* Suppression de l'identifiabilité :

Seules les entités sont concernées par cette commande car l'identifiabilité d'une liaison n'est pas définie dans l'absolu mais par rapport à son entité origine ou extrémité.

Toutes les caractéristiques (Propriétés et Liaisons) d'une entité déclarées identifiables sont remplacées par des caractéristiques non identifiables grâce à la commande :

SUPprimer POUr <NOMENT> IDentifiabilité ;

* Ajustement statistique des nombres moyens d'occurrences :

Pour optimiser la gestion du travail de l'étudiant, le concepteur peut donner le nombre moyen des occurrences de liaisons et d'entités qui seront créées lors du travail de l'étudiant.

Cas des entités

STATistique POUr <REFLIA> NOMbre [OCCurrence] [A] <CI> ;

Cas des liaisons

STATistique POUr <REFLIA> NOMbre [OCCurrence] {DIRecte
INVerse} [A] <CI> ;

* Edition du modèle créé

Edition du nom des entités créées :

EDIter ENTités CREées ;

Edition du nom des entités appartenant à une hiérarchie ou toutes les hiérarchies.

EDIter HIErarchie ;

EDIter SOUs HIErarchie ;

* Edition des caractéristiques d'entité

EDIter POUr <NOMENT> TOUtes {PROpriétés
LIAisons} ;

Edition des contraintes

d'entités :

EDIter POUr <NOMENT> CONtraintes ;

de liaison :

EDIter POUr <REFLIA> CONtraintes {DIRectes
INVerses} ;

de propriétés :

EDIter POUr <NOMENT> PROpriétés <NOMPRO> CONtraintes ;

4.2 - Exemple du moteur asynchrone

Nous avons reproduit ci-dessous, des extraits de la définition du modèle potentiel d'un moteur asynchrone.

```
>
  CREER MOTEUR ;
OK
>
  CREER CIRCMAGN ;
JK
>
  CREER CIRCELEC ;
LK
>
  CREER ENRULLEM ;
JK
>
  CREER BOBINE ;
LK
>
  CREER FIL ;
LK
>
  CREER CAGE ;
LK
```

Ce travail doit être fait pour toutes les entités.

```
>
  LISTER ENTITES CREEES ;
  ENTITES CREEES
MTEUR      CIRCMAGN  CIRCELEC  ENRULLEM  BOBINE    FIL        CAGE
BARRE      ANNEAU    EXERCICE  ENTREFER  SCHEMAEG  MATERIAU  CULASSE
STATOR     ROTOR        CLUCORNE  DENT      ENCOCHE
JK
```

Définition des propriétés :

```
>
  CREER POUR EXERCICE PROPRIETE ENTIERE NOMBRECONSUL ;
JK
>
  CREER POUR SCHEMAEG PROPRIETE FLUTTANTE R1 ;
LK
>
  CREER POUR SCHEMAEG PROPRIETE FLUTTANTE R2 ;
JK
```

Ce travail doit être fait pour toutes les propriétés de chaque entité.

Le concepteur peut demander à tout moment l'édition des propriétés déjà définies.

```
>
  EDITER POUR ENCOCHE TOUTES PROPRIETES ;
ENTITE : ENCOCHE
PROPRIETES FLOTTANTES :
                                     NCM
LARGEUR.HAUT
LARGEUR.MOY
LARGEUR.FCND
OUVERTURE
HAUT.ISTHME
HAUT.CALE
HAUT.MOY
HAUT.FCND
PERMEANCE
PROPRIETES ALPHANUMERIQUES :
                                     NCM
                                     TYPE
```

JK

>

```
  EDITER POUR SCHEMAEQ TOUTES PROPRIETES ;
ENTITE : SCHEMAEQ
PROPRIETES FLOTTANTES :
                                     NCM
R1
R2
X1
X2
XM
```

JK

Création des liaisons.

```
>
  CREER LIAISON HIERARCHIQUE DE CIRCUMAGN VERS ENTREFER ;
JK
>
  CREER LIAISON VERS HIERARCHIE CONSULTE DE EXERCICE VERS MOTEUR ;
JK
>
  CREER LIAISON INTER HIERARCHIQUE INSERE DE ENROULEMENT VERS ROTOR ;
JK
>
  CREER LIAISON VERS HIERARCHIE REPRESENTE DE SCHEMAEQ VERS MOTEUR ;
JK
>
  CRE LIA VER HIE CONSTITU MATERIAU ROTOR ;
JK
>
  CRE LIA VER HIE CONSTITU DE MATERIAU VERS ENROULEM ;
JK
```

Ce travail doit être fait pour toutes les liaisons.

Le concepteur peut demander à tout moment l'édition des liaisons déjà créées

```

>
EDITER PLUS MOTEUR TOUTES LIAISONS ;
ENTITE : MOTEUR
LIAISONS DIRECTES :
                                NOM      VERS
                                HIERARCH  CIRCMAGN
                                HIERARCH  CIRCELEC
LIAISONS INVERSES :
                                NOM      DE
                                CONSULTE  EXERCICE
                                REPRESEN  SCHEMAEQ

```

```

OK
>
EDITER PLUS ROTOR TOUTES LIAISONS ;
ENTITE : ROTOR
LIAISONS DIRECTES :
                                NOM      VERS
                                HIERARCH  CULASSE
                                HIERARCH  COURONNE
LIAISONS INVERSES :
                                NOM      DE
                                INSERE   ENROULEM
                                CONSTITU  MATERIAU
                                HIERARCH  CIRCMAGN

```

OK

```

>
DECLARER MOTEUR TETE HIERARCHIE PREPONDERANTE ;

```

OK

Après définition de la tête de la hiérarchie, il est possible d'analyser toute la structure entrée, le système commence par dresser une carte des liaisons directes définies dans l'ordre chronologique avec lequel les entités ont été créées.

```

>
ANALYSER HIERARCHIE ;
*****
DEBUT ANALYSE :LISTE DES ENTITES ET DES LIAISONS
*****
MOTEUR
    LIAISON    VERS
    HIERARCH   CIRCMAGN
    HIERARCH   CIRCELEC
CIRCMAGN
    LIAISON    VERS
    HIERARCH   ENTREFER
    HIERARCH   ROTOR
    HIERARCH   STATOR
CIRCELEC
    LIAISON    VERS
    HIERARCH   CAGE
    HIERARCH   ENROULEM
ENROULEM
    LIAISON    VERS
    INSERE     ROTOR
    HIERARCH   REBINE
REBINE
    LIAISON    VERS
    HIERARCH   FIL
FIL

```

Le système organise ensuite la hiérarchie par niveaux.

```
*****
CONSTRUCTION DE LA HIERARCHIE
*****
***** NIVEAU HIERARCHIQUE *****
ALTEUR
EXERCICE
SCHEMAEQ
MATERIAL
***** NIVEAU HIERARCHIQUE *****
CIRCMAGN
CIPCELEC
***** NIVEAU HIERARCHIQUE *****
ENRCULEM
CAGE
ENTREFER
STATOR
RLTOR
***** NIVEAU HIERARCHIQUE *****
BOBINE
BARRE
ANNEAU
CULASSE
COURONNE
***** NIVEAU HIERARCHIQUE *****
FIL
DENT
ENCLCHE
```

Puis il contrôle la correction des liaisons non hiérarchiques et précise les liaisons interhiérarchiques contrôlées vis-à-vis de la hiérarchie

EN CENTRE LES LIAISONS PAR RAPPORT A LA HIERARCHIE;

```
ALTEUR
CIRCMAGN
CIPCELEC
ENRCULEM
INSERE      RLTOR      CONTROLE
BOBINE
FIL
CAGE
BARRE
ANNEAU
EXERCICE
ENTREFER
SCHEMAEQ
MATERIAL
CULASSE
STATOR
RLTOR
COURONNE
DENT
ENCLCHE
```

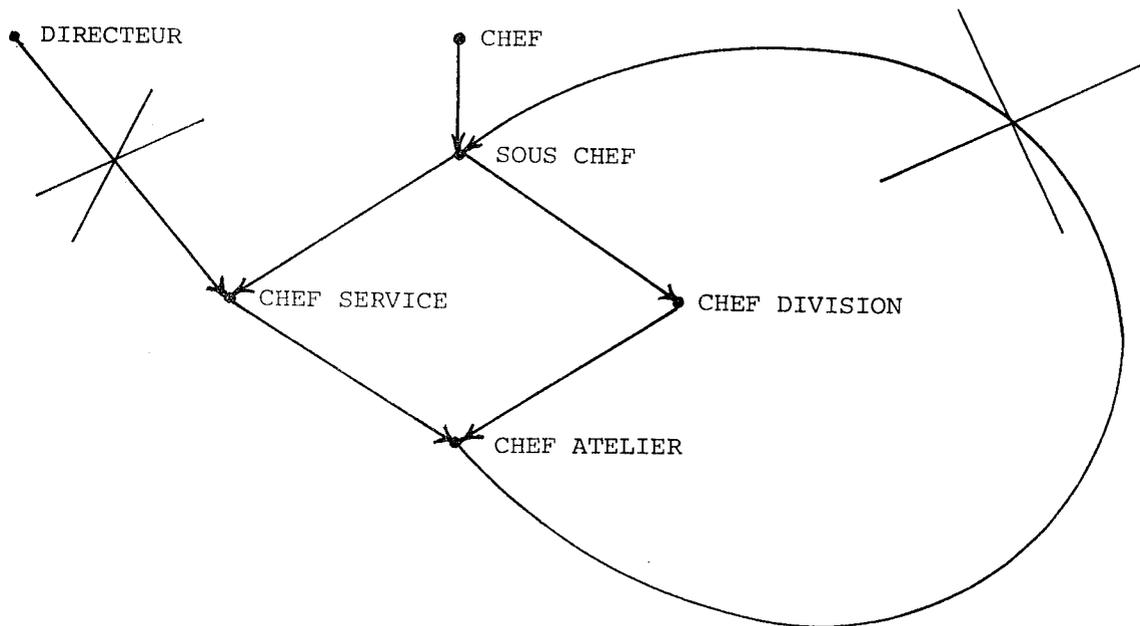
Une analyse statistique de la structure entrée est alors conduite, entité par entité avec optimisation de la banque de donnée décrite et préparation de tous les tableaux inverses nécessaires à la compilation des requêtes de langage BLOPRO

```
*****  
ANALYSE STATISTIQUE  
*****  
ENTITE : MLTEUR  
ENTITE : CIRCMAGN  
ENTITE : CIRCFLEC  
ENTITE : ENRLULEM  
ENTITE : BOBINE  
ENTITE : FIL  
ENTITE : CAGE  
ENTITE : BARRE  
ENTITE : ANNEAU  
ENTITE : EXERCICE
```

4.3 - Exemples d'erreurs détectées par MODPHY

Nous allons travailler sur une structure très simple en illustrant les erreurs que détecte le système. D'une manière générale toute incompatibilité avec le modèle potentiel des modèles potentiels, est détectée de telle sorte que le concepteur puisse décrire son modèle dans l'ordre qui lui convient.

L'ensemble choisi a la structure suivante :



>
EDITER ENTITES CREEES ;
ERREUR : IL N EXISTE AUCUNE ENTITE
>
>
>
CREER POUR CHEF PROPRIETE QUALITE ;

ERREUR= > *erreur de syntaxe, la caractéristique type*
COMMANDE MAL FORMULEE *a été oubliée.*

>
CREER POUR CHEF PROPRIETE ALPHANUMERIQUE QUALITE ;
ERREUR : ENTITE CHEF INEXISTANTE
>
>
>

CREER CHEF ;

JK

>
CREER CHEF
ERREUR : ENTITE CHEF DEJA CREEE
>
>
>

EDITER POUR CHEF PROPRIETE QUALITE CONTRAINTE ;
ERREUR : PROPRIETE QUALITE INEXISTANTE
>
>
>

CREER POUR CHEF PROPRIETE ALPHANUMERIQUE QUALITE ;

JK

>
AUTORISER POUR CHEF PROPRIETE QUALITE ;

- => GENTIL
- => AIMABLE
- => GENEREUX
- => OK

L'utilisateur ne pourra donner que les valeurs citées à la propriété qualité de l'entité chef.

>
>
>
CREER SOUSCHEF ;
JK
>
CREER LIAISON HIERARCHIQUE DE CHEF VERS SOUSCHEF ;
JK

>
CREER LIAISON HIERARCHIQUE DE SOUSCHEF VERS CHEF ;
ERREUR : LIAISON INVERSE DEJA CREEE
>

En effet les liaisons hiérarchiques ne peuvent former des boucles orientées.

LIMITER POUR SOUSCHEF NOMBRE OCCURRENCES A -2 ;
ERREUR : NOMBRE INCORRECT(< 0)

>
>
>

LIMITER POUR SOUSCHEF NOMBRE OCCURRENCES A 2 ;

OK
>
>
>

EDITER PLUR SOUSCHEF CONTRAINTES ;
SOUSCHEF
TYPE DE L ENTITE

STANDARD
NOMBRE MAXIMUM D OCCURENCES 2
NOMBRE MOYEN D OCCURENCES 7

APPARTENANCE

PAS DEFINIE (HIERARCHIE A ANALYSER)

OK
>

ANALYSER HIERARCHIE ;
ERREUR : IL N Y A PAS DE TETE DE HIERARCHIE

C'est en effet le concepteur qui doit donner au système le nom de l'entité tête de hiérarchie.

DECLARER CHEF TETE HIERARCHIE PREPONDERANTE ;

OK

La prépondérance de la hiérarchie impose que les occurrences reliées par des liaisons interhiérarchique, appartiennent à la même hiérarchie.

>

CREER CHEFATEL ;

OK
>

CREER CHEFSERV ;

OK
>

CREER CHEFDIVI ;

OK
>

CREER LIAISON HIERARCHIQUE DE SOUSCHEF VERS CHEFDIVI ;

OK
>

CREER LIAISON HIERARCHIQUE DE SOUSCHEF VERS CHEFSERV ;

OK
>

CREER LIAISON HIERARCHIQUE DE CHEFDIVI VERS CHEFATEL ;

OK
>

CREER LIAISON HIERARCHIQUE DE CHEFSERV VERS CHEFATEL ;

OK
>

CREER LIAISON HIERARCHIQUE DE CHEFATEL VERS SOUSCHEF ;

OK

Cette dernière liaison crée une boucle orientée au sein de la hiérarchie. Cette boucle va être détectée lors de l'analyse globale de la hiérarchie.

>

ANALYSER HIERARCHIE ;

DEBUT ANALYSE :LISTE DES ENTITES ET DES LIAISONS

CHEF

LIAISON VERS
HIERARCH SCUSCHEF

SCUSCHEF

LIAISON VERS
HIERARCH CHEFDIVI
HIERARCH CHEFSERV

CHEFATEL

LIAISON VERS
HIERARCH SOUSCHEF

CHEFSERV

LIAISON VERS
HIERARCH CHEFATEL

CHEFDIVI

LIAISON VERS
HIERARCH CHEFATEL

CONSTRUCTION DE LA HIERARCHIE

***** NIVEAU HIERARCHIQUE *****

CHEF

***** NIVEAU HIERARCHIQUE *****

IL EXISTE UNE BUCLE ORIENTEE DANS LA HIERARCHIE

>

SUPPRIMER LIAISON HIERARCHIEQUE DE CHEFATEL VERS SOUSCHEF ;

JK *(supprime la liaison qui créait la boucle orientée).*

>

CREER DIRECTEUR ;

OK

>

CREER LIAISON HIERARCHIQUE DE DIRECTEUR VERS CHEFSERV ;

JK *(créé une deuxième racine à la hiérarchie).*

>

ANALYSER HIERARCHIE ;

DEBUT ANALYSE :LISTE DES ENTITES ET DES LIAISONS

CHEF

LIAISON VERS
HIERARCH SCUSCHEF

SCUSCHEF

LIAISON VERS
HIERARCH CHEFDIVI
HIERARCH CHEFSERV

CHEFATEL

CHEFSERV

LIAISON VERS
HIERARCH CHEFATEL

CHEFDIVI

LIAISON VERS
HIERARCH CHEFATEL

DIRECTEL

LIAISON VERS
HIERARCH CHEFSERV

CLNSTRUCTION DE LA HIERARCHIE

***** NIVEAU HIERARCHIQUE *****

CHEF

DIRECTEU

ERREUR : LA HIERARCHIE A LA RACINE DIRECTEU EN TROP

>
SUPPRIMER LIAISON HIERARCHIQUE DE DIRECTEUR VERS CHEFSERV ;

OK *supprime la liaison erronée.*

>
CREER LIAISON VERS HIERAR CONTACTE DE CHEFDIVI VERS CHEFSERV ;

OK
La liaison ainsi créée devrait être de type inter-hiérarchique : l'analyse de hiérarchie va le déceler.

ANALYSER HIERARCHIE ;

DEBUT ANALYSE : LISTE DES ENTITES ET DES LIAISONS

CHEF

LIAISON VERS
HIERARCH SLUSCHEF

SLUSCHEF

LIAISON VERS
HIERARCH CHEFDIVI
HIERARCH CHEFSERV

CHEFATEL

CHEFSERV

LIAISON VERS
HIERARCH CHEFATEL

CHEFDIVI

LIAISON VERS
HIERARCH CHEFATEL
CONTACTE CHEFSERV

DIRECTEU

CLNSTRUCTION DE LA HIERARCHIE

***** NIVEAU HIERARCHIQUE *****

CHEF

DIRECTEU

***** NIVEAU HIERARCHIQUE *****

SLUSCHEF

***** NIVEAU HIERARCHIQUE *****

CHEFSERV

CHEFDIVI

***** NIVEAU HIERARCHIQUE *****

CHEFATEL

***** NIVEAU HIERARCHIQUE *****

OK CONTROLE LES LIAISONS PAR RAPPORT A LA HIERARCHIE ;

CHEF

SLUSCHEF

CHEFATEL

CHEFSERV

CHEFDIVI

CONTACTE CHEFSERV

ERREUR : ENTITE CHEFDIVI ORIGINE APPARTIENT A LA HIERARCHIE

>
CREER SECRETAIRE ;

OK

>
CREER LIAISON HIERARCHIQUE CHEF SECRETAIRE IDENTIFIABLE ORIGINE ;
ERREUR : UNE LIAISON HIER DIRECTE NE PEUT ETRE IDENTIFIABLE

En effet une occurrence de hiérarchie doit être arborescente.

>
LIMITER POUR LIAISON HIE CHEF SOUSCHEF NOMBRE OCCURRENCES INVERSES A 2
ERREUR NB D OCCURRENCES LIMITE NECESSAIREMENT A 1;

Pour cette même raison (arborescence de l'occurrence de hiérarchie) : une liaison inverse hiérarchie a un nombre d'occurrences limité automatiquement à 1 par le système

5. LE LANGAGE EDITER

=====

Ce langage permet d'insérer et d'éditer les textes nécessaires à un système d'ECAO :

Messages, textes de cours, énoncés d'exercices. Chaque ligne de texte est repérable par un numéro compris entre 1 et 999. Toutes les lignes d'un même système d'ECAO sont regroupées dans une ressource commune qui porte le nom du système (que nous noterons <NOMSYS>).

- Entrée de texte :

ENTrer TEXte [POUr] [SYStème] <NOMSYS> ;

Le contrôle est alors rendu à l'utilisateur qui peut entrer un nombre de lignes quelconque dans n'importe quel ordre, chaque ligne étant précédée de son numéro.

L'envoi d'une ligne blanche arrête cette procédure.

Cette insertion de texte peut être réalisée à tout moment, même lorsque les étudiants utilisent le système.

- Edition des textes :

EDiter MESsage <CI> DU SYStème <NOMSYS> ;

EDiter TEXte DE LIGne <CI> A LIGne <CI> DU SYStème <NOMSYS> ;

EDiter TOUt TEXte DU SYStème <NOMSYS> ;

Exemple :

Entrée de l'énoncé de l'exercice associé au calcul d'un schéma équivalent en cours de conception d'un moteur asynchrone.

ENTRER TEXTE POUR SYSTEME ASYNCHRONE ;

```
800 *****
801 *****EXERCICE ILLUSTRANT LA REPRESENTATION D UN MOTEUR *****
802 *****          PAR SON SCHEMA EQUIVALENT          *****
803 *****
804 '
805 '
806 L EQUATION D UN MOTEUR ASYNCHRONE PEUT SE METTRE SOUS
807 LA FORME SUIVANTE :
808  $U = (KSTA + J*A11)*ISTA + J*A*IKPT$ 
809  $0 = J*A*ISTA + (KRET/G + J*A22)*IRLT$ 
810 AVEC LES NOTATIONS CLASSIQUES:
811     RSTA=RESISTANCE DU STATOR
812     RRGT=RESISTANCE DU ROTOR ;
813     REACTANCE A11 ;
814     REACTANCE A ;
815     REACTANCE A22
816 *****
817 PREMIERE QUESTION :
818 CALCULEZ LES ELEMENTS DU SCHEMA EQUIVALENT CLASSIQUE
819 VEULEZ VOUS FAIRE CE CALCUL AVEC LES VALEURS NUMERIQUES
820 DU MOTEUR QUE VOUS AVEZ CONSTRUIT ? (OUI OU NON) ?
821 QUEL EST LE NCM DE VOTRE MOTEUR ?
822 LE MOTEUR CITE N EXISTE PAS
823 VALEUR DE LA RESISTANCE DU STATOR :
824 VALEUR DE LA RESISTANCE DU ROTOR :
825 VALEUR DE LA REACTANCE A11 :
826 VALEUR DE LA REACTANCE A :
827 VALEUR DE LA REACTANCE A22 :
828 QUELLE EST LA VALEUR DE R1 ?
829 QUELLE EST LA VALEUR DE R2 ?
830 QUELLE EST LA VALEUR DE X2 ?
831 QUELLE EST LA VALEUR DE XM ?
832  $R1 = RSTATOR$ 
833  $R2 = A11**2 * RRGT/(A**2)$ 
834  $X2 = A11**2 * A22/ A**2 - A11$ 
835  $XM = A11$ 
836 $$$$$$
837 DEUXIEME QUESTION
838 DETERMINER LE COURANT A VIDE ET LE FACTEUR DE PUISSANCE
839 A VIDE SACHANT QUE :
840     V = 220 VOLT
841 QUELLE EST LA VALEUR DU COURANT ?
842  $I0 = V/SQRT(R1**2 + XM**2)$ 
843 QUELLE EST LA VALEUR DU COSPHI ?
844  $COSPHI = COS(ARCTG(XM/RSTATOR))$ 
845 FIN DE L EXERCICE
846 LE MOTEUR CITE EXISTE DEJA
847 LE CIRCUIT MAGNETIQUE CITE EXISTE DEJA
848 LE CIRCUIT ELECTRIQUE CITE EXISTE DEJA
849 OK
850 VOULEZ VOUS CONSULTER L EXERCICE SUR LES SCHEMAS EQUIVALENTS
851 ACCESSIBLE PAR LA COMMANDE : EXERCICE SCHEMA EQUIVALENT ;
852 ( OUI NON ) ?
853 EN CHOISISANT ARBITRAIREMENT X1 = 0
```

6. - LANGAGE DE DEFINITION DE LANGAGE DE COMMANDE

Ce langage a été défini par Messieurs LATOMBE et CAMPAS
L'interpréteur générateur correspondant a été réalisé en deux versions :
l'une compatible avec l'IBM 360 CPCMS sous le système ESPACE, l'autre
compatible avec le T 1600 ou SOLAR. Nous en donnons ici une présentation
très succincte.

- Début de commande (cf § 3.1 page 61)

COMmande <CA3> $\left\{ \begin{array}{l} \phi ; \\ \text{ETI} \text{quette} <CI> ; \end{array} \right.$

- Fin de commande :

FIN DE COMmande ;

- Définition de modificateurs (cf § 3.1 page 62)

MODificateur <CA3> $\left[\begin{array}{l} \phi \\ \text{ETI} \text{quette} <CI> \end{array} \right] \text{NIV} \text{eau} <CI> ;$

- Définition de séparateur (cf § 3.1 page 62)

SEParateur <CA1> ;

- Branchements

BRAncher $\left\{ \begin{array}{l} \text{SI} \left\{ \begin{array}{l} \text{SE} \text{Parateur} <CA1> \\ \text{NON FIN DE COM} \text{mande} \\ \text{MOD} \text{ificateur IN} \text{Connu} \\ \text{ER} \text{Reur DE CON} \text{version} \end{array} \right\} \text{ETI} \text{quette} <CI> ; \\ \text{ETI} \text{quette} <CI> ; \end{array} \right.$

Exemple de la machine asynchrone

Nous avons retenu à titre d'exemple, trois commandes du langage mis à la disposition de l'étudiant construisant une machine asynchrone.

a) Nous supposerons que l'enseignant veut permettre à l'étudiant de définir au sein d'une même commande (créer) soit :

- le nom du moteur, du circuit magnétique, du circuit électrique ;
- le nom du rotor, du stator, et de l'entrefer d'un circuit magnétique ;
- le rotor seul.

A cette commande correspondent trois traitements logiques : l'enseignant demandera donc suivant les cas, l'exécution de l'un des trois programmes correspondants de la stratégie pédagogique (CREMOTGL, ROSTAEN, CREROTOR) (Ces programmes seront conçus par l'enseignant et décrits grâce au langage BLOPRO, nous les détaillerons au paragraphe suivant).

Nous supposerons que l'enseignant veut libeller la commande de la façon suivante :

CREer {
 MOTeur <CA8> (AYA)nt (CIR)cuit MAGnétique <CA8> (CIR)cuit ELEctrique <CA8> ;
 ROTor <CA8> {
 Φ ;
 STAtör <CA8> ENTrefer <CA8> (POU)r (CIR)cuit MAGnétique <CA8> ;

<CA8> représente des paramètres alphanumériques de 8 caractères maximum.
 Φ représente l'absence de modificateur

Description de cette commande à l'aide du LDLDC.

COMMANDE CREER ;

EMPLER 599 ; *(permet de lancer le superviseur MANAGER.)*

MODIFICATEUR MOTEUR NIVEAU 1 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 1 ;

MLT IGNORABLE AYANT ;

MLT IGNORABLE CIRCUIT ;

MODIFICATEUR MAGNETIQUE NIVEAU 2 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 3 ;

MLT IGNORABLE CIRCUIT ;

MODIFICATEUR ELECTRIQUE NIVEAU 3 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 5 ;

INITIALISER PARAM ALPHA SUR 2 CASES DEPUIS 60 A CREMOTGL ;
*(Permet de communiquer à MANAGER le nom du programme de la stratégie
pédagogique (CREMOTGL) qui sera activé).*

MODIFICATEUR ROTOR NIVEAU 1 ;

INITIALISER PARAM ALPHA SUR 2 CASES DEPUIS 60 A CROTOR ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 1 ;

BRANCHER SI NON FIN DE COMMANDE ETIQUETTE 1 ;

*(Si le caractère fin de commande ;) n'est pas rencontré, la commande
doit continuer selon la description donnée au bloc d'étiquette 1)*

BLOC ETIQUETTE 1 NIVEAU 1 ;

MODIFICATEUR STATLR NIVEAU 2 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 3 ;

MODIFICATEUR ENTREFER NIVEAU 3 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 5 ;

MLT IGNORABLE PLUR ;

MLT IGNORABLE CIRCUIT ;

MODIFICATEUR MAGNETIQUE NIVEAU 4 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 7 ;

INITIALISER PARAM ALPHA SUR 2 CASES DEPUIS 60 A RUSTAEN ;

FIN DE COMMANDE ;

b) L'étudiant disposera entre autres des deux commandes suivantes :

EVALuer SCHéma (EQU)ivalent (DE) MOTeur < > ;

et

EXercice SCHéma (EQU)ivalent ;

Nous donnons ci-dessous le détail de leur définition à l'aide du LDCDC.

COMMANDE EVALUER ;

MODIFICATEUR SCHEMA NIVEAU 1 ;

MLT IGNORABLE EQUIVALENT ;

MLT IGNORABLE DE ;

MODIFICATEUR MOTEUR NIVEAU 2 ;

RANGER PARAMETRE ALPHANUMERIQUE SUR 2 CASES DEPUIS 1 ;

INITIALISER PARAM ALPHA SUR 2 CASES DEPUIS 60 A CALSCHEM ;

EMPILE 599 ;

FIN DE COMMANDE ;

COMMANDE EXERCICE ;

EMPILE 599 ;

MODIFICATEUR SCHEMA NIVEAU 1 ;

MLT IGNORABLE EQUIVALENT ;

INITIALISER PARAMETRE ALPHA SUR 2 CASES DEPUIS 60 A ENONCE ;

FIN DE COMMANDE ;

Nous étudierons les programmes de la stratégie pédagogique associée à chacune de ces commandes (CALSCHEM, ENONCE) dans le paragraphe suivant :

7. DESCRIPTION DE LA STRATEGIE PEDAGOGIQUE

=====
Nous avons montré au chapitre II comment la stratégie pédagogique entre en relation avec toutes les ressources du système et comment on peut la décomposer en une succession de Blocs Elémentaires d'Analyse Stratégique : les BEAS.

La description de chaque BEAS sera réalisée dans le langage BLOPRO que nous allons présenter au paragraphe suivant. Au sein de ce langage nous avons cherché à faciliter la communication entre la stratégie et les autres composants du système : (Modèle potentiel, travail de l'étudiant, ressources d'édition, langage de commande, ressources de simulation) et leur enchaînement.

Par contre nous n'avons pas développé d'outil puissant de calcul arithmétique ou logique. En effet, ceux-ci sont généralement rejetés au niveau des modules de simulation. Il reste cependant possible d'implanter des outils de calcul plus efficaces au sein de la stratégie pédagogique ; mais la technique d'interprétation utilisée deviendrait alors insuffisante et un précompilateur s'avèrerait nécessaire. Nous avons envisagé cette solution sur laquelle nous reviendrons en conclusion.

7.1 - SYNTAXE DU LANGAGE BLOPRO

Tout programme (Bloc Elémentaire d'Analyse Stratégique) écrit dans le langage BLOPRO doit commencer par :

DEbut (BLO)c <CA8> ;

Il doit se terminer par :

FIN (DE) (BLO)c ;

Cette instruction rend le contrôle à l'utilisateur (Cf. ARRêt).

Etiquette :

Toutes les instructions (sauf Début et Fin) sont étiquetables. Les étiquettes, placées après l'instruction concernée et précédées du mot clé ETIquette, peuvent avoir de 1 à 8 caractères alphanumériques.

Constantes :

Entier : nombre entier de format libre précédé de son signe ; si le signe est omis le nombre est considéré comme positif.

Flottant : nombre réel simple précision conforme à la syntaxe FORTRAN en format libre F ou E.

Alphanumérique : Suite de caractères alphanumériques dont le nombre maximum sera toujours indiqué (8 ou 12). Les caractères en sus seront ignorés et les caractères manquants complétés par des blancs à droite.

Logique : L'une des deux constantes :
VRAi
FAUx

Dans cette description du langage, nous représenterons ces diverses constantes respectivement par CI, CF, CAn, CL .

↑
_____.
nombre maximum de caractères.

7.1.1. Les variables

Toutes les variables reconnues par BLOPRO sont des tableaux caractérisés par un nom variable suivi d'un indice entre parenthèses.

7.1.1.1. Les noms des variables

Ils sont décomposables en un préfixe et un suffixe :

a) Type d'une variable : Le préfixe

On retrouve les quatre types usuels.

La première lettre d'une variable (son préfixe) caractérise son type :

- Variables entières commençant par I ;
- Variables réelles commençant par F ;
- Variables alphanumériques commençant par A ;
- Variables logiques commençant par L.

b) Portée d'une variable

Les trois portées que peuvent avoir les variables du langage correspondent aux trois fonctions bien distinctes suivantes :

- Accès aux paramètres d'une commande entrée par un étudiant ;
- Traitement local au sein de la stratégie pédagogique ;
- Passage des valeurs aux modules de simulation.

1) Les variables paramètres

Elles sont caractérisées par le suffixe PAramètre. Elles permettent de traiter les paramètres rangés dans une commande (cf. Ch. III page 86)

2) Les variables locales

Elles sont caractérisées par le suffixe LOcal. Elles permettent de garder les informations en mémoire dans le contexte d'un "BEAS". Elles sont compatibles avec un appel récursif des blocs.

3) Les variables Relais

Elles sont caractérisées par le suffixe RElais. Elles sont accessibles à partir de tous les BEAS appartenant à un même enchaînement et de tous les modules de simulation appelés par cet enchaînement. Comme leur suffixe l'indique, elles servent de relais entre les banques de données et les modules de simulation.

Les noms des variables reconnues de BLOPRO sont donc :

ILO, IPA, IRE, FLO, FPA, FRE, ALO, APA, ARE, LLO, LRE.

Remarque : Les variables de type logique n'existent pas au sein de la description d'un langage. La variable LPA n'est donc pas reconnue par BLOPRO.

7.1.1.2. Les indices des variables

Les indices des tableaux doivent être des constantes entières positives.

Le programmeur peut donner le nombre maximum de variables dont il aura besoin pour chacun des tableaux ILO, FLO, ALO, LLO, IRE, FRE, ARE, LRE. Ceci doit être fait une fois pour toute au début de la description de la stratégie pédagogique (une commande spéciale est prévue à cet effet).

Dans le cas où le programmeur ommettrait de préciser ces valeurs, le système BLOPRO réserve par défaut 50 variables pour chacun des types précédents.

Le nombre des variables paramètres est limité à 50. L'indice désigne le numéro de la case dans laquelle a été rangé le paramètre. Le préfixe permet de préciser au système la nature du paramètre cherché.

7.1.1.3. Variable Alphanumérique de Référence

Les BEAS peuvent s'enchaîner (cf. ANALyser & ACTiver page 96). Les variables à suffixe PARAMètre et RELais, sont accessibles à tous les blocs appelés et ce passage de variable est à rapprocher des "COMMON" de fortran. Mais, pour permettre un appel récursif, le programmeur peut passer au Bloc appelé une variable alphanumérique nommée REFérence dont la portée est limitée au bloc appelé. (Ce paramètre sera souvent le nom d'une occurrence de la banque de données qui permettra ensuite un accès aisé aux valeurs numériques correspondantes).

De même les modules de simulation peuvent être lancés par la stratégie pédagogique (cf. EMPiler et EXEcuter page 95) et on peut leur transmettre une variable alphanumérique REFérence.

7.1.2. Les expressions arithmétiques

Peu de calculs sont généralement nécessaires au sein de la stratégie pédagogique (les calculs sont en effet réalisés par les modules de

simulation) c'est pourquoi nous avons limité nos expressions arithmétiques à deux opérands :

- Les opérands sont des variables ou des constantes ;
- Les opérateurs sont PLUS, MOINS, MULTIPLIER, DIVISER.

7.1.3 Les expressions logiques : elles peuvent avoir 3 types d'opérands :

Les opérands logiques : variables ou constantes logiques : les opérateurs étant alors : ET, OU, NON.

Les opérands Alphanumériques : variables ou constantes alphanumériques : EG, NE (pour EGAL et NON EGAL).

Les opérands numériques : variables ou constantes entières ou flottantes : les opérateurs étant : IN (inférieur) IE (inférieur ou égal) SU (Supérieur) SE (Supérieur ou égal) EG (égal) NE (non égal).

7.1.4. Instructions d'affectation simple :

Arithmétique

Variable Entière ou Réelle = Expression arithmétique ;

Exemple :

FLO(13) = ILO(3) PLUS FRE(18) ;

Logique

Variable Logique = Expression logique ;

Exemple :

LLO(4) = NON LRE(3) OU LLO(5) ;

Alphanumérique

Variable Alphanumérique = Constante ou variable alphanumérique ;

Exemple :

ALO(8) = CETOUBON ; ALO(3) = REF ;

Remarque : Les variables à suffixes PARAMÈTRE et la variable RÉFÉRENCE ne peuvent pas apparaître en partie gauche d'une instruction d'affectation.

7.1.5. Instruction d'affectation externe

Cette instruction permet de donner le contrôle au poste de travail, en cours de stratégie pédagogique, pour récupérer une constante (entière, réelle, logique, alphanumérique).

Exemple :

FL0(2) = EXT ;

7.1.6. Instructions de branchement

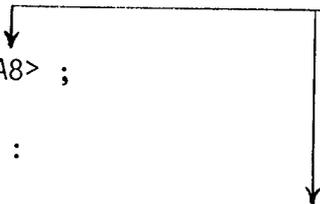
Simple :

ALLer [A] <CA8> ;

Conditionnelle :

SI (<Expres-Logique>) ALLer [A] <CA8> ;

étiquette alphanumérique



Si l'expression logique est égale à VRAI, on dérouté le programme sur l'instruction correspondant à l'étiquette.

Si elle est égale à FAUX, on passe en séquence.

7.1.7. Arrêt d'exécution

ARRêt ;

Cette instruction redonne le contrôle au poste de travail

7.1.8. Appels de modules de simulation

Deux appels de modules sont possibles (cf. CH II page 50).

Appel avec exécution immédiate

EXEcuter MODule <CI> [REFérence <VCA>] ;

Les modules sont repérables par un numéro de 1 à 999. La référence est une variable ou Constante Alphanumérique.

Exemple :

EXE MOD 3 REF TRANSF01 ;

Appel avec exécution différée

Cette instruction construit une pile de modules munis de leur référence. Cette pile sera exécutée suivant un mode lié au type d'appel du bloc dans lequel elle est construite.

EMPiler MODule <CI> [REFérence <VCA>] ;

Exemple :

EMPILER MODULE 5 REFERENCE APA(10) ;

7.1.9. Enchaînements de BLOCS

Deux modes d'appel sont possibles (cf CH II page 50)

Appel avec exécution immédiate.

ACTiver BLOc <CAS> [REFérence <VCA>] ;

Les blocs sont repérés par leur nom (Constante Alphanumérique 8). L'exécution de la pile de modules de simulation éventuellement construite durant l'exécution du bloc sera lancée lorsque l'instruction RETour ; sera rencontrée.

Exemple : ACTIVER BLOC ANACOM ;

Appel avec exécution différée :

ANalyser BLOc <CA8> [REFérence <VCA>] ;

L'exécution de la pile éventuellement construite au sein du BLOC appelé ne sera pas lancée lorsque l'instruction retour sera rencontrée. Les modules empilés au sein de ce bloc seront insérés dans la pile de modules du bloc appelant et seront exécutés avec ceux-ci.

Exemple :

ANALYSER BLOC ANAPEDAG REFERENCE COURGENE ;

7.1.10. Instruction de Retour de BLOC :

RETour ;

Rend le contrôle au bloc appelant et lance l'exécution d'après le mode d'appel du bloc concerné.

7.1.11. Instructions de gestion des ressources d'édition :

a) Stockage

A chaque message est associée une référence alphanumérique qui permet de personnaliser ce message, et une priorité qui permet d'ordonner les messages entre eux (cf. CH.II page 44)

Les messages munis de leur référence peuvent être stockés dans une pile pour différer leur édition.

STOcker MESsage <CI> REFérence <VCA> PRIorité <CI> ;

b) Edition immédiate de message

EDIter MESsage <CI> [REFérence <VCA>] ;

édite de façon immédiate le message muni de sa référence éventuelle.

Exemple :

```
EDITER MESSAGE 3 REFERENCE APA(1) ;  
EDITER MESSAGE 182 ;
```

c) Edition différée de messages

```
EDIter TOUs MESSages [PRIorité <CI>] ;
```

Edite la liste des messages précédemment stockés dans la pile de messages et possédant une priorité supérieure ou égale à la constante entière positive donnée (si la priorité n'est pas indiquée tous les messages sont édités).

d) Edition de texte

```
EDIter TEXte [DE] [LIGne] <VCI> [A] [LIGne] <VCI> ;
```

Edite les lignes de texte ainsi spécifiées.

Exemple :

```
EDITER TEXTE 150 ILO(15) ;
```

7.1.1.2. Instructions de gestion du travail de l'étudiant :

Notation : Ces instructions font référence au modèle potentiel. Nous représenterons par :

- <ENT> le nom d'une entité du modèle potentiel
- <LIA> le nom d'une liaison du modèle potentiel
- <PRO> le nom d'une propriété d'une entité du modèle
- <VCA> <VCI> ... le nom d'une variable ou constante du type spécifié

a) Création

- Création d'occurrence simple

```
CREer <ENT> <VCA> ;
```

Exemple :

CREER MOTEUR MOT001 ;

crée l'occurrence spécifiée dans le cas où celle-ci n'existe pas déjà.

Si le nombre d'occurrences a été limité (cf. Notion d'occurrence : CH.II page 28) la création n'est réalisée que si le nombre maximum n'est pas atteint.

- Création d'identique

CREer <ENT> <VCA> IDEntique <VCA> ;

Exemple :

CREER ROTOR RT001 IDENTIQUE RT 002 ;

Cette instruction est soumise aux mêmes contrôles que la création simple en imposant de surcroît l'existence préalable de l'occurrence servant de modèle.

Toutes les propriétés et liaisons déclarées identifiables au sein du modèle potentiel seront donc communes aux deux occurrences.

- Création de liaison

CREer LIAison <LIA> [DE] <ENT> [VERs] <ENT> [ENTre] <VCA> [ET] <VCA> ;

Exemple :

CREER LIAISON INSERE DE BOBINE VERS ROTOR ENTRE BOB1 ET R2 ;

Si l'une des extrémités possède des identiques et si la liaison a été déclarée identifiable, elle sera également créée sur les occurrences identiques concernées. Dans le cas où le nombre d'occurrences de liaison est limité, ce contrôle est fait automatiquement. Dans le cas où la liaison est contrôlée par rapport à une hiérarchie (hiérarchie prépondérante) le test d'appartenance à une hiérarchie commune est fait systématiquement.

b) Suppression :

- Suppression des occurrences

SUPprimer <ENT> <VCA> ;

Toutes les liaisons concernant l'occurrence sont alors supprimées.
Si cette occurrence possédait des identiques, celles-ci ne sont pas affectées.

- Suppression des liaisons

SUPprimer LIAison <LIA> [DE] <ENT> [VERs] <ENT> [ENTre] <VCA> [ET] <VCA> ;

Si l'une des extrémités possède des identiques et si la liaison est identifiable, celle-ci est supprimée aussi sur toutes les occurrences identiques concernées.

c) Entrée de données :

Notation : nous noterons par <VCIFLA> une variable ou constante de type entier, réel, logique ou alphanumérique.

Propriétés simples

PROpriété <PRO> [DE] <ENT> <VCA> = <VCIFLA> ;

Exemple :

PROPRIETE RESISTANCE DE FIL FCU005 = 3.1583 ;

Il doit y avoir concordance entre le type de la propriété et celui de la variable ou de la constante.

Propriétés tableaux

TABleau. <PRO> (<VCI>) [DE] <ENT> <VCA> = <VCIFL> ;

Exemple :

TABLEAU REPONSE(4) DE EXERCICE EX01 = APA(10) ;

L'indice du tableau doit être un entier positif. Les dépassements de tableaux sont testés.

Propriété Courbes

COUrbe <PRO> (<VCF>) [DE] <ENT> <VCA> = <VCF> ;

Exemple :

COURBE PERTKILOINDUC (1.5) DE TOLE SILIC = FLO(8) ;

Les points erronés ne peuvent être effacés que lorsqu'on connaît exactement leur abscisse dans le cas contraire on peut effacer toute la courbe grâce à la commande :

EFFacer COUrbe <PRO> [DE] <ENT> <VCA> ;

d) Editions :

Sortie directe d'une variable ou constante

SORTir <VCIFLA> ;

Exemple :

SORTIR COUCOU ;

fait écrire sur le poste de travail la suite de caractères : COUCOU

Sortie de propriété

SORTir PROpriété <PRO> [DE] <ENT> <VCA> ;

Exemple :

SORTIR PROPRIETE NBREPONJUSTE DE EXERCICE EX03 ;

Si le nom de la propriété représente un tableau ou une courbe, toutes les valeurs de ce tableau ou de cette courbe sont éditées.

Sortie de toutes les propriétés d'une occurrence

Sortir Toutes Propriétés [DE] <ENT> <VCA> ;

Exemple :

SORTIR TOUTES PROPRIETES DE ROTOR APA(19) ;

Sortie des noms de toutes les occurrences d'une entité

Sortir NOMs [DE] TOUT <ENT> ;

Sortie des noms de toutes les occurrences d'une entité appartenant à une hiérarchie donnée.

Sortir NOMs [DE] TOUT<ENT> APPartenant [A] <ENT> <VCA> ;

Exemple :

SORTIR NOM DE TOUT EXERCICE APPARTENANT A CHAPITRE CH2 ;

Sortie des noms de toutes les liaisons d'une occurrence

Sortir NOMs [DE] TOUT LIAisons [DE] <ENT> <VCA> ;

e) Affectation avec référence à la banque de données

Ces instructions permettent de lire des informations au sein de la banque de données où est stocké le travail de l'étudiant et de ranger ces informations dans les variables à suffixe RELAIS.

Remarque : affectation dans les variables relais logiques : lorsque la partie droite de l'affectation est en contradiction avec le modèle potentiel (cf. CH.III page 64) la variable logique sera affectée à FAUX.

LRE(CI) = $\left\{ \begin{array}{l} \text{PRO} \text{propriété} \langle \text{PRO} \rangle \text{ [DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \text{TAB} \text{leau} \langle \text{PRO} \rangle \langle \text{VCI} \rangle \text{ [DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \text{EXI} \text{ste} \text{ OCC} \text{urrence} \langle \text{VCA} \rangle ; \\ \text{EXI} \text{ste} \text{ OCC} \text{urrence} \langle \text{VCA} \rangle \text{ [DE]} \langle \text{ENT} \rangle ; \\ \text{EXI} \text{ste} \text{ OCC} \text{urrence} \langle \text{VCA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ APP} \text{artenant} \text{ [A]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \text{EXI} \text{ste} \text{ LIAI} \text{son} \langle \text{LIA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ [VERs]} \langle \text{ENT} \rangle \text{ [POUr]} \text{ OR} \text{igine} \langle \text{VCA} \rangle ; \\ \text{EXI} \text{ste} \text{ LIAI} \text{son} \langle \text{LIA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ [VERs]} \langle \text{ENT} \rangle \text{ [POUr]} \text{ EX} \text{T} \text{rém} \text{ité} \langle \text{VCA} \rangle ; \\ \text{EXI} \text{ste} \text{ LIAI} \text{son} \langle \text{LIA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ [VERs]} \langle \text{ENT} \rangle \text{ [ET]} \langle \text{VCA} \rangle ; \end{array} \right.$

Exemples :
 LRE(1) = EXI OCC R001 ;
 LRE(2) = EXI OCC R001 DE ROTOR ;
 LRE(3) = EXI OCC R001 DE ROTOR APP A MOTEUR AS03 ;
 LRE(4) = EXI LIA INSERE DE CAGE VERS ROTOR POU EXT R001 ;
 LRE(5) = EXI LIA INSERE DE CAGE VERS ROTOR ENTRE CAG2 ET R001 ;

IRE(CI) = $\left\{ \begin{array}{l} \text{PRO} \text{propriété} \langle \text{PRO} \rangle \text{ [DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \text{TAB} \text{leau} \langle \text{PRO} \rangle \langle \text{VCI} \rangle \text{ [DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \text{NOM} \text{bre} \text{ OCC} \text{urrence} \text{ [DE]} \langle \text{ENT} \rangle ; \\ \text{NOM} \text{bre} \text{ LIAI} \text{son} \langle \text{LIA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ [VERs]} \langle \text{ENT} \rangle \text{ [POUr]} \text{ OR} \text{igine} \langle \text{VCA} \rangle ; \\ \text{NOM} \text{bre} \text{ LIAI} \text{son} \langle \text{LIA} \rangle \text{ [DE]} \langle \text{ENT} \rangle \text{ [VERs]} \langle \text{ENT} \rangle \text{ [POUr]} \text{ EX} \text{T} \text{rém} \text{ité} \langle \text{VCA} \rangle ; \end{array} \right.$

FRE(CI) = $\left\{ \begin{array}{l} \underline{\text{PROpriétés}} \langle \text{PRO} \rangle \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \underline{\text{TAB}1\text{eau}} \langle \text{PRO} \rangle \langle \text{VCI} \rangle \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \underline{\text{COUrbe}} \langle \text{PRO} \rangle \langle \text{VCF} \rangle \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \end{array} \right.$

Remarque : Pour les courbes une interpolation parabolique est réalisée automatiquement par le système à l'aide des trois points de la courbe les plus rapprochés du point cherché.

ARE(CI) = $\left\{ \begin{array}{l} \underline{\text{PRO}priété} \langle \text{PRO} \rangle \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \underline{\text{TETe}} \underline{\text{HIE}rarchie} \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \underline{\text{TETe}} \underline{\text{SOUs}} \underline{\text{HIE}rarchie} \underline{[DE]} \langle \text{ENT} \rangle \langle \text{VCA} \rangle ; \\ \underline{\text{NOM}} \langle \text{VCI} \rangle \underline{[OCC}urrence] \underline{[RELIée]} \underline{\text{ORIGine}} \langle \text{VCA} \rangle \underline{[PAR]} \underline{\text{LIAison}} \langle \text{LIA} \rangle \underline{[DE]} \langle \text{ENT} \rangle \underline{[VERS]} \langle \text{ENT} \rangle ; \\ \underline{\text{NOM}} \langle \text{VCI} \rangle \underline{[OCC}urrence] \underline{[RELIée]} \underline{\text{EXTrémité}} \langle \text{VCA} \rangle \underline{[PAR]} \underline{\text{LIAison}} \langle \text{LIA} \rangle \underline{[DE]} \langle \text{ENT} \rangle \underline{[VERS]} \langle \text{ENT} \rangle ; \end{array} \right.$

Remarque 1 : Les deux dernières instructions permettent de récupérer le nom des occurrences reliées à une occurrence donnée.

ARE(3) = NOM 2 OCC REL EXT R001 PAR LIA INSERE DE CAGE VERS ROTOR ;

Ceci permet de récupérer le nom de la deuxième cage insérée dans le rotor R001.

Remarque 2 : Lorsque le nom cherché n'existe pas (contradiction avec la stratégie minimale) ou lorsqu'il y a une incohérence entre la commande et la structure construite, la valeur rangée dans ARE est &INCONU&.

7.2 - EXEMPLES DE STRATEGIES PEDAGOGIQUES

Les exemples que nous allons traiter sont extraits d'un système d'ECAO appliqué au moteur asynchrone dont nous avons déjà décrit :

- le modèle potentiel ,
- une partie des ressources d'édition ,
- un sous-ensemble du langage de commande.

7.2.1. Exemple associé à la commande créer

Nous allons voir comment l'utilisation de la stratégie pédagogique permet de regrouper des actions élémentaires définies au niveau du modèle potentiel : en effet, il peut paraître fastidieux pour l'étudiant d'élaborer son moteur, élément par élément, puis de relier chacun de ces sous ensembles par des liaisons. La stratégie pédagogique permet de construire un écran entre le modèle potentiel et la perception qu'en aura l'étudiant à travers le langage de commande que l'enseignant a mis à sa disposition.

Nous avons décrit au § VI la commande :

CREER MOTEUR < > AYANT CIRCUIT MAGNETIQUE < > CIRCUIT ELECTRIQUE < > ;

dans laquelle le nom du moteur était rangé dans le tableau des paramètres en 1 celui du circuit magnétique en 3 et celui du circuit électrique en 5. Au sein du BEAS CREMOTGL qui est associé à cette commande nous aurons donc accès à ces noms dans les variables APA(1), APA(3), APA(5).

La stratégie (très simple) consistera à tester si ces composants existent déjà, à éditer un message d'erreur dans ce cas, et dans le cas contraire, à créer les occurrences de MOTEUR CIRCMAGN CIRCELEC en leur donnant les noms fournis en paramètre, puis à relier les éléments ainsi définis par les liaisons hiérarchiques prévues au sein du modèle potentiel.

Dans ce cas précis nous avons regroupé au sein d'une même commande cinq opérations élémentaires qu'aurait dû faire l'étudiant s'il travaillait en liaison directe avec le modèle potentiel :

```
DEBUT BLOC CREMOTGL ;
OK
>
LRE(1)=EXISTE OCCURRENCE APA(1) DE MOTEUR ;
JK
>
LRE(2)=EXISTE OCCURRENCE APA(3) DE CIRCUMAGN ;
OK
>
LRE(3)=EXISTE OCCURRENCE APA(5) DE CIRCELEC ;
OK
>
SI(LRE(1))ALLER A ERRMOT ;
OK
>
SI(LRE(2))ALLER A ERRMAGN ;
OK
>
SI(LRE(3))ALLER A ERRELEC ;
OK
>
CREER MOTEUR APA(1) ;
JK
>
CREER CIRCUMAGN APA(3) ;
JK
>
CREER CIRCELEC APA(5) ;
OK
>
CRE LIA HIE DE MOTEUR VER CIRCUMAGN ENT APA(1) ET APA(3) ;
OK
>
CRE LIA HIE MOTEUR CIRCELEC APA(1) APA(5) ;
OK
>
EDI MES 850 ;
JK
>
RETOUR ;
OK
>
EDITER MESSAGE 846 REFERENCE APA(1) ETIQUETTE ERRMOT ;
OK
>
ARRET ;
OK
>
EDITER MESSAGE 847 REFERENCE APA(3) ETIQUETTE ERKMAGN ;
OK
>
ARRET ;
OK
>
EDITER MESSAGE 848 REFERENCE APA(5) ETIQUETTE ERRELEC ;
JK
>
ARRET ; RETOUR ; FIN DE BLOC ;
OK
```

846 LE MOTEUR CITE EXISTE DEJA

847 LE CIRCUIT MAGNETIQUE CITE EXISTE DEJA

848 LE CIRCUIT ELECTRIQUE CITE EXISTE DEJA

7.2.2. Exemple associé au calcul d'un schéma équivalent d'un moteur asynchrone.

Nous supposons (§VI) que l'étudiant dispose des deux commandes suivantes :

- EVALUER SCHEMA EQUIVALENT DE MOTEUR < > ;
- EXERCICE SCHEMA EQUIVALENT ;

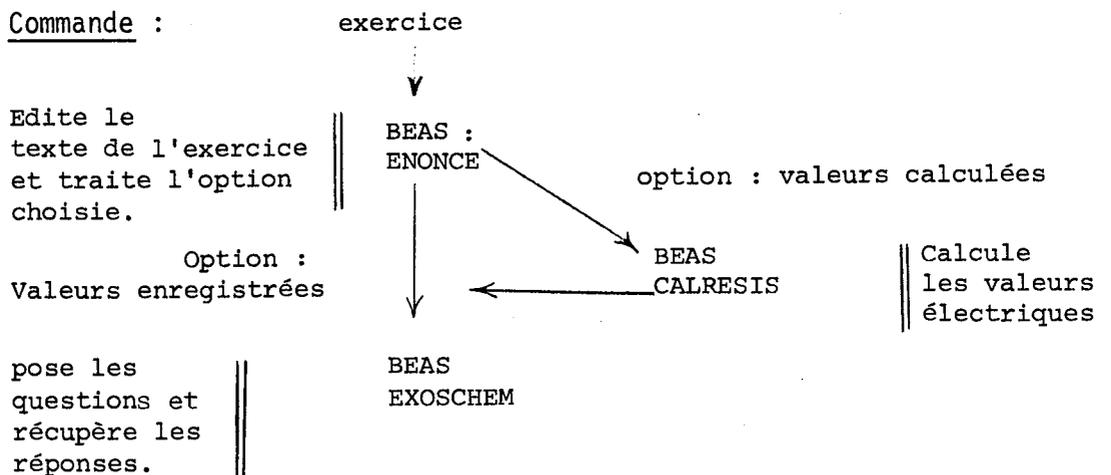
La stratégie que nous associons à ces deux commandes est la suivante :

- * La consultation de l'exercice est libre
- * Lorsqu'un étudiant consulte l'exercice :
 - Le système lui demande s'il préfère que l'exercice porte sur des valeurs enregistrées ou sur celles de son propre moteur.
 - Dans ce dernier cas il y a exécution du BEAS qui calcule les valeurs électriques associées au moteur de l'étudiant et édition de ces résultats.
 - Dans le premier cas il y a simplement édition des valeurs préenregistrées.
 - Le calcul des bonnes réponses est alors effectué (par un module FORTRAN de la banque d'algorithmes).
 - Les questions de l'exercice sont alors posées.
 - Une réponse juste entraîne la question suivante tandis qu'une erreur déclenche l'édition de la formule littérale et de la valeur numérique exacte puis l'énoncé de la question suivante.
 - La propriété nombre de consultations de l'exercice est alors incrémentée (pour contrôle ultérieur).
 - Si l'étudiant a choisi d'effectuer l'exercice en relation avec son propre moteur, la liaison CONSULTE entre l'exercice et le moteur concerné est alors créée.

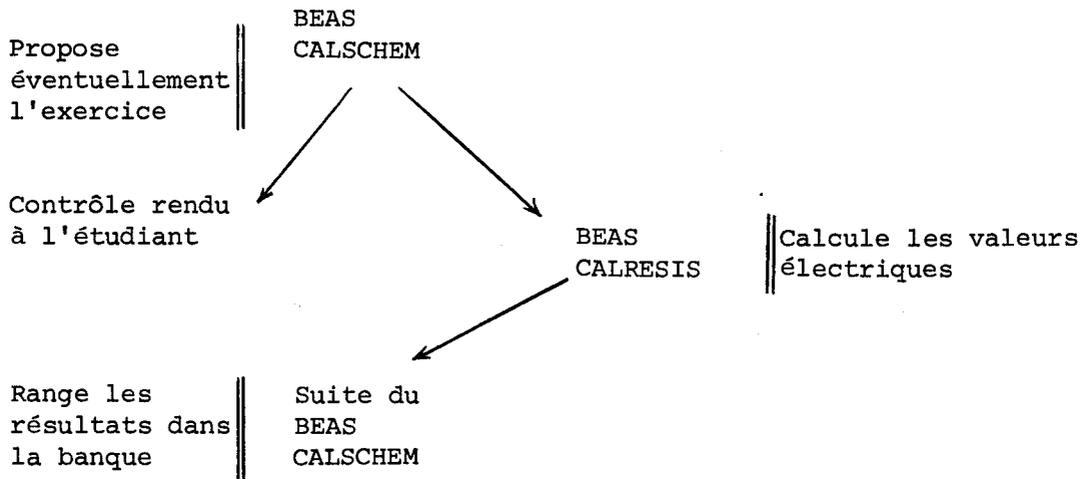
* Lorsqu'un étudiant demande l'évaluation automatique du schéma équivalent à son moteur :

- Le système vérifie s'il a déjà consulté l'exercice : dans ce cas le calcul est réalisé.
- Sinon l'exercice lui est proposé.
- Si l'étudiant choisit l'exercice, le contrôle lui est rendu et il peut frapper la commande exercice. Sinon le système lance l'étape du contrôle.
- Le contrôle qui précède le calcul consiste à tester si la structure construite par l'étudiant comporte un schéma équivalent (nécessité pour pouvoir stocker les résultats en fin de calcul).
- Dans le cas où la structure n'est pas correcte, la stratégie la complète en créant l'occurrence ou la liaison manquante et lance alors le calcul.
- Pour calculer le schéma il est nécessaire d'activer le BEAS qui évalue les valeurs électriques du moteur (déjà utilisé dans l'exercice) et le module qui détermine les caractéristiques du schéma correspondant (également utilisé dans l'exercice).
- Après ce calcul, les valeurs sont éditées puis stockées dans la banque de données réservée à l'étudiant et le contrôle lui est alors rendu.

Nous allons montrer comment cette stratégie peut être décrite à l'aide du langage BLOPRO. De façon globale nous avons décomposé cette stratégie de la manière suivante :



Commande : évaluer schéma équivalent



Nous ne décrivons pas le bloc CALRESIS dont nous ne préciserons que l'interface d'entrée (Nom du moteur concerné), et l'interface de sortie : (valeurs des grandeurs électriques cherchées).

Note :

- Les textes encadrés rappellent le contenu des ressources d'édition concernées.
- Les textes inscrits en italiques ont été rajoutés sur le listing à titre d'explication.

>
DEBUT BLOC ENCENCE ;
OK
>
EDITER TEXTE DE LIGNE 800 A LIGNE 820 ;
OK

```
800 *****  
801 *****EXERCICE ILLUSTRANT LA REPRESENTATION D UN MOTEUR *****  
802 ***** PAR SON SCHEMA EQUIVALENT *****  
803 *****  
804 '  
805 '  
806 L EQUATION D UN MOTEUR ASYNCHRONE PEUT SE METTRE SOUS  
807 LA FORME SUIVANTE :  
808  $U = (R_{STA} + J \cdot A_{11}) \cdot I_{STA} + J \cdot A \cdot I_{ROT}$   
809  $D = J \cdot A \cdot I_{STA} + (R_{ROT} / G + J \cdot A_{22}) \cdot I_{ROT}$   
810 AVEC LES NOTATIONS CLASSIQUES:  
811 RSTA=RESISTANCE DU STATOP  
812 RROT=RESISTANCE DU ROTOR ;  
813 REACTANCE A11 ;  
814 REACTANCE A ;  
815 REACTANCE A22  
816 *****  
817 PREMIERE QUESTION :  
818 CALCULEZ LES ELEMENTS DU SCHEMA EQUIVALENT CLASSIQUE  
819 VUULEZ VOUS FAIRE CE CALCUL AVEC LES VALEURS NUMERIQUES  
820 DU MOTEUR QUE VOUS AVEZ CONSTRUIT ? (OUI OU NON) ?
```

>
ALL(1)=EXT ;
OK *permet de lire la réponse de l'étudiant.*
>
SI(ALL(1) EG CUI) ALLER A ONRECUPERE ;

OK
>
FRE(1)=0.1 ;
OK
>
FRE(2)=1.11E-03 ;
OK
>
FRE(3)=32.0 ;
OK
>
FRE(4)=1.0
OK
>
FRE(5)=3.55E-02 ;
OK

} Initialisation des valeurs pour l'exercice
dans le cas où l'étudiant a
choisi la réponse NON à la question posée.

>
ACTIVER BLOC EXOSCHEM REFERENCE DIRECTE ;
OK *On lance l'exécution de l'exercice en précisant que c'est l'option*
> *préenregistrée qui a été choisie (REFERENCE DIRECTE).*
RETOUR ;
OK

Dans le cas où l'étudiant fait l'exercice avec les caractéristiques de son propre moteur on lui demande de donner le nom du moteur concerné.

EDITER MESSAGE B21 ETIQUETTE ENRECUPERE ;
OK

P21 QUEL EST LE NOM DE VOTRE MOTEUR ?

>
ARE(1)=EAT ;
OK
> *lecture du nom du moteur donné par l'étudiant*

LRE(1)=EXISTE OCCURRENCE ARE(1) DE MOTEUR ;

OK
>
SI(LRE(1)) ALLER A ENCALCULE ;

OK
>
EDITER MESSAGE B22 REFERENCE ARE(1) ;
OK

B22 LE MOTEUR CITE N EXISTE PAS

>
ARRET ;
OK
> *Contrôle rendu à l'étudiant*

ACTIVER BLOC CALRESIS REFERENCE ARE(1) ETIQUETTE ENCALCULE ;
OK
> *Appel du bloc qui calcule les valeurs électriques du moteur dont le nom est APA(1).*

ACTIVER BLOC EXOSCHEM REFERENCE ARE(1) ;
OK
> *Lancement de l'exercice proprement dit en indiquant qu'il est réalisé sur les caractéristiques du moteur APA(1) (REFERENCE APA(1)).*
FIN DE BLOC ;

Description du BLOC EXOSCHEM

Nous supposons que le bloc CALRESIS fournit après exécution l'interface de sortie suivante :

Valeur de la résistance du STATOR dans FRE(1)

Valeur de la résistance du ROTOR dans FRE(2)

Valeur des réactances A11 A A22 dans FRE(3) FRE(4) FRE(5)

Dans tous les cas le bloc EXOSCHEM trouve donc dans ces variables les données nécessaires à l'exercice.

```

DEB BLC EXCSCEM ;
OK
>
EDI MES 823 ;
JK
>
SCR TIR FRE(1) ;
OK
>
EDITER MESSAGE 824 ;
JK
>
SCR FRE(2) ;
JK
>
EDITER MESSAGE 825 ;
JK
>
SCR FRE(3) ;
JK
>
EDITER MESSAGE 826 ;
JK
>
SCR FRE(4) ;
JK
>
EDI MES 827 ;
JK
>
SCR FRE(5) ;
OK
>
EXECUTER MODULE 630 ;
JK
>
Ce module calcule les valeurs cherchées R1 R2 X2 Xm et les range dans
cet ordre dans les variables
FRE(6) FRE(7) FRE(8) FRE(9) CF § 8.
>
EXECUTER MODULE 631 ;
OK
>
Ce module calcule les valeurs de I à vide et du Cosφ et les range dans
cet ordre dans les variables
FRE(10) FRE(11).
>
EDITER MESSAGE 854 ;
OK
854 EN CHUISISSANT ARBITRAIREMENT X1 = 0
>
EDI MES 828 ;
OK
828 QUELLE EST LA VALEUR DE R1 ?
>
FLL(1)=EXT ;
JK
lecture de la réponse de l'étudiant (en format libre)
>
FLL(2)=FRE(6) PLU 0.01 ;
OK
La bonne réponse est encadrée à ± 0.01
>
FLL(3)=FRE(6)MCI 0.01 ;
OK
>

```

823 VALEUR DE LA RESISTANCE DU STATUR :
824 VALEUR DE LA RESISTANCE DU RETOR :
825 VALEUR DE LA REACTANCE A11 :
826 VALEUR DE LA REACTANCE A :
827 VALEUR DE LA REACTANCE A22 :

Edition des données sur lesquelles l'étudiant devra faire l'exercice (valeurs préenregistrées ou particulières selon l'appel).

854 EN CHUISISSANT ARBITRAIREMENT X1 = 0
--

828 QUELLE EST LA VALEUR DE R1 ?

La bonne réponse est encadrée à ± 0.01

```
LLL(1)=FLC(1) IE FLD(2) ;
OK
> test de la réponse de l'étudiant.
LLD(2)=FLC(1) SE FLC(3) ;
OK
>
SI(LLL(1) ET LLD(2)) ALLER A QUEST2 ; Réponse exacte : passage à la
OK question suivante
>
EDI MES 832 ; Réponse fausse : édition la valeur théorique.
OK
> E32 R1 = RSTATCR
>
SDR FRE(6) ;
OK Edition la valeur numérique exacte.
>
EDI MES 829 ETI QUEST2 ; Passage à la question suivante.
:
: Les 6 questions de l'exercice sont traitées d'une manière similaire.
: On passe directement à la fin de la 6ème question :
:
>
SI(REFERENCE EG DIRECTE) ALLER A ACCUMULE ETIQUETTE CEFINI ;
OK
> Si l'étudiant a fait l'exercice sur les valeurs préenregistrées, la
> propriété nombre de consultations est incrémentée.
>
>
LRE(1)=EXISTE LIA CONSULTE EXERCICE MOTEUR ENTRE EXDSCHEM REFERENCE ;
OK
> Est-ce que l'étudiant avait déjà fait l'exercice en liaison avec ce
> moteur si oui la liaison existe déjà, elle n'est donc pas créée.
>
>
SI(LRE(1)) ALLER A ACCUMULE ;
OK
>
CREEC LIAISON CONSULTE EXERCICE MOTEUR ENTRE EXDSCHEM REFERENCE ;
OK
> Création de la liaison pédagogique CONSULTE.
>
IRE(1)=PROPRIETE NOMBRECONSUL DE EXERCICE EXDSCHEM ETI ACCUMULE ;
OK
> Récupération de la valeur précédente du nombre de consultations.
>
>
ILL(1)=IRE(1) PLUS 1 ;
OK
>
PROPRIETE NOMBRECONSUL DE EXERCICE EXDSCHEM = ILD(1) ;
OK
> La nouvelle valeur est rangée.
>
>
EDI MES 845 ;
OK 845 FIN DE L EXERCICE
>
RETOUR ;
OK Contrôle rendu au bloc appelant (cad ENONCE).
>
FIN DE BLOC ;
```

Description du bloc CALSCHEM associé à la commande :
CALCULER SCHEMA EQUIVALENT DE MOTEUR < > ;

```
DEBUT BLOC CALSCHEM ;
OK
>
IRE(1)=PROPRIETE NOMBRECONSUL DE EXERCICE EXCSHEM ;
OK
>      Lecture du nombre de consultation de l'exercice.
>
SI(IRE(1) SE 1) ALLER A CNCALCULE ;
OK
>
>
EDITER TEXTE DE LIGNE 851 A 853 ;      L'exercice est proposé à l'étudiant.
OK
>      VOULEZ VOUS CONSULTER L EXERCICE SUR LES SCHEMAS EQUIVALENTS
>      ACCESSIBLE PAR LA COMMANDE : EXERCICE SCHEMA EQUIVALENT ;
>      ARE(1)=EXT ; ( OUI NON ) ?
OK
>      Lecture de sa réponse
>
SI (ARE(1) EQ OUI )ALLER A CEFINI;
OK
>      Il a choisi l'exercice : le contrôle lui est rendu.
>
LRE(1)=EXISTE OCCURRENCE APA(1) DE MOTEUR ;
OK
>      Vérification de l'existence du moteur cité.
>
SI(NON LRE(1)) ALLER A ERRMOTEUR ;
OK
>
ARE(1)=NOM 1 OCC RELIEE EXTREM APA(1) LIAISON REPRESENTE SCHEMAEQ MOTEUR ;
OK
>
>      Recherche du nom du schéma équivalent qui représente le moteur concerné
>
>
>
ALD(1)=ARE(1) ;
OK
>      Sauvegarde de ce nom dans une variable dont la portée est limitée au bloc.
>
>
SI(ARE(1) NE "&INCENUE")ALLER A CNCALCULE ;
OK
>      Si ce schéma équivalent existe, ses caractéristiques sont calculées.
>
LRE(1)=EXISTE OCCURRENCE SCHEMA DE SCHEMAEQ ;
OK
>      Sinon le système en crée un auquel il donne le nom de "schéma".
>      (il faut donc tester si l'occurrence "schéma" n'existe pas déjà).
>
SI(LRE(1)) ALLER A CRELIAISON ;
OK
>      Si cette occurrence existe seule la liaison est créée.
>
```

```
CREER SCHEMAEQ SCHEMA ;
OK
> Sinon le schéma est créé
>
CREER LIAISON REPRESENTE SCHEMAEQ MOTEUR SCHEMA APA(1) ETIQ CRELIAISON ;
OK
> puis la liaison est créée.
>
>
ALL(1)=SCHEMA ;
OK
> Sauvegarde du nom du schéma équivalent avant d'appeler le bloc CALRESIS
> qui peut détruire ARE(1).
>
ACTIVER BLOC CALRESIS REFERENCE APA(1) ETIQUETTE ONCALCULE ;
OK
> Le nom du moteur (APA(1)) est communiqué au bloc de calcul qui fournit les
> résultats dans les variables
> FRE(1), FRE(2), FRE(3), FRE(4), FRE(5).
>
EXECUTER MODULE 630 ;
OK
> Lancement de l'exécution du module 630 qui récupère les valeurs des
> caractéristiques électriques et donne les résultats dans les variables
> FRE(6), FRE(7), FRE(8), FRE(9) Cf. § 8.
>
EDITER MESSAGE 854 ;
OK
> Indique que l'on a choisi X1 = 0
>
PROPRIETE X1 DE SCHEMAEQ ALD(1)=0. ;
OK
> Rangement de la valeur de la propriété.
>
>
EDITER MESSAGE 832 ;
OK
>
PROPRIETE R1 DE SCHEMAEQ ALD(1)=FRE(6) ;
OK
>
>
>
SCR FRE(6) ;
OK
>
EDITER MESSAGE 833 ;
OK
> Rangement et
> Édition de
> chaque propriété.
>
PROPRIETE R2 DE SCHEMAEQ ALC(1)=FRE(7) ;
OK
>
>
SCR FRE(7) ;
OK
>
EDITER MESSAGE 834 ;
OK
```

```
>
PROPRIETE X2 DE SCHEMAEQ ALO(1)=FRE(8) ;
OK
>
SORTIR FRE(8) ;
JK
>
EDITER MESSAGE 835 ;
OK
>
PROPRIETE XM DE SCHEMAEQ ALO(1)=FRE(9) ;
OK
>
SORTIR FRE(9) ;
JK
>
EDITER MESSAGE 850 ;
JK
>
RETOUR ;      Contrôle rendu à l'étudiant
OK
>
ARRET ETIQUETTE CEFINI ;
JK
>
EDITER MESSAGE 822 REFERENCE APA(1) ETI ERRMOTEUR ;
JK
>
ARRET ;      Cas où le moteur APA(1) n'existait pas
OK
>
FIN DE BLOC ;
```

7.3. DETECTION DES ERREURS ET AIDE A LA MISE AU POINT

- Toute incompatibilité entre la stratégie pédagogique et le modèle potentiel est bien sûr détectée lors de la description d'un BEAS.
- De même toute erreur de programmation dans le langage BLOPRO est décelée et un message explicite est donné à l'utilisateur (références non résolues, dépassement de tableau, validité des paramètres numériques etc...)
- A l'exécution le système traite les incompatibilités qui peuvent se produire entre la stratégie pédagogique et la logique propre de la banque de données construite par l'étudiant. Un programme (MANAGER) ausculte le travail réalisé et arrête l'exécution dès que toute incohérence est découverte. Un message en clair est envoyé à l'étudiant (qui doit en faire part à l'enseignant). L'enseignant doit alors corriger la stratégie erronée.

Afin de faciliter la tâche de l'enseignant dans la phase de mise au point de la stratégie pédagogique, nous avons mis à sa disposition trois types d'exécution de cette stratégie.

a) Exécution normale : c'est celle que nous avons illustrée jusqu'à présent, elle consiste à :

- programmer les BEAS (sur cartes perforées, sur disques, ou conversationnellement).;
- les faire analyser par le système BLOPRO qui génère alors le code qui sera interprété plus tard ;
- lancer l'exécution des BEAS à partir du langage de commande de l'étudiant (mode normal) ;

b) Exécution en pas à pas : Les deux premières étapes restent identiques à celles du cas précédent mais l'exécution de chaque BEAS se fait en pas à pas et l'enseignant peut ainsi contrôler point par point l'évolution de la stratégie (en cours de réalisation).

c) Exécution immédiate : Ce mode permet une compilation suivie d'une exécution immédiate de chaque instruction décrite. L'accès conversationnel avec la banque de données qui représente le travail de l'étudiant est donc immédiat. (Ce mode peut notamment être utilisé pour un contrôle du travail d'un étudiant). Toutes les instructions ne sont pas disponibles dans ce mode. (En effet certaines variables ne peuvent plus être accessibles : variables paramètres, et les "ALLER A" n'ont plus de sens).

8. DEFINITION DES MODULES DE SIMULATION

=====

Un module de simulation doit être un programme principal FORTRAN comportant un "COMMON Blanc" de 1K mots :

Ce "COMMON" contient toutes les informations nécessaires au système CASCADE, les variables PARAMETRES et les variables RELAIS. Le programmeur n'a pas à en connaître l'organisation. Il a en effet accès à ces valeurs par l'intermédiaire de FONCTIONS systèmes programmées qui sont rangées dans la bibliothèque commune du système.

Accès aux PARAMETRES des COMMANDES

IPA (<CI>) est une fonction qui fournit le paramètre entier qui possède l'indice donné en argument.

FPA (<CI>) fournit de même un paramètre réel.

La subroutine APA (<CI1>, <CI2>, <TABLEAU D'ENTIER>) lit le paramètre alphanumérique rangé à partir de la case <CI1> ayant <CI2> caractères (généralement 8 ou 12) et le recopie dans le tableau d'entiers donné en argument.

Accès aux variables RELAIS

Suivant le même principe, le programmeur peut accéder aux variables relais :

IRE (<CI>) fournit le relais entier d'indice <CI>

FRE (<CI>) fournit le relais flottant d'indice <CI>

LRE (<CI>) fournit le relais logique d'indice <CI>

La subroutine ARE (<CI1>, <TABLEAU D'ENTIER>) range le relais alphanumérique (8 caractères) dans le tableau d'entiers donné en argument.

L'opération inverse peut être effectuée (rangement dans les variables relais des résultats du module simulation) grâce aux sousroutines.

```
RANIRE (<CI>, <VALEUR ENTIERE>) ;  
RANFRE (<CI>, <VALEUR FLOTTANTE>) ;  
RANLRE (<CI>, <VALEUR LOGIQUE>) ;  
RANARE (<CI>, <TABLEAU D'ENTIER>);
```

Début du module

Tout module devra commencer par l'instruction :

```
CALL INTERF (<TABLEAU D'ENTIER>).
```

Cette sousroutine prend à sa charge la résolution de tous les problèmes d'interface entre le module et la stratégie pédagogique qui l'a appelé. Elle range la variable référence éventuelle (paramètre alphanumérique d'appel de module cf page 93) dans le tableau d'entiers qui lui est passé en argument.

Fin de module

Tout module devra se terminer par l'instruction CALL ENCHNE(0) qui rend le contrôle à la stratégie pédagogique.

Exemple :

Ce module (630) est appelé au cours du bloc EXOSCHEM (cf. CH.III p: 112).
Il calcule les caractéristiques du schéma équivalent d'un moteur dont on connaît les grandeurs électriques.

```

      DIMENSION ITABLO(4)
      COMMON III(1024)
C *****
C
C      RECUPERATION DE L INTERFACE D ENTREE
C
C *****
C      CALL INTERF(ITABLO)
C *****
C
C      RECUPERATION DES PARAMETRES DE CALCUL
C      A L AIDE DES FONCTIONS SYSTEMES
C
C *****
C      RSTATO=FRE(1)
C      RROTOR=FRE(2)
C      ALFA11=FRE(3)
C      ALPHA=FRE(4)
C      ALFA22=FRE(5)
C *****
C
C      CALCUL DES VALEURS CHERCHEES
C
C *****
C      ACARRE=ALPHA*ALPHA
C      A1CARR=ALFA11*ALFA11
C      R1=RSTATO
C      R2=A1CARR*RROTOR/ACARRE
C      X2=A1CARR*ALFA22/ACARRE
C      XM=ALFA11
C *****
C
C      PASSAGE DES RESULTATS A LA STRATEGIE PEDAGOGIQUE
C
C *****
C      CALL RANFRE(6,R1)
C      CALL RANFRE(7,R2)
C      CALL RANFRE(8,X2)
C      CALL RANFRE(9,XM)
C *****
C
C      ON REND LE CONTROLE A LA STRATEGIE PEDAGOGIQUE
C
C *****
C      CALL ENCHNE(0)
      END
```

9. CONCLUSION

Au cours de ce chapitre, nous venons de présenter en détail l'interface entre le système CASCADE et ses utilisateurs : enseignants et étudiants. Un certain nombre d'exemples ont été cités pour permettre d'apprécier concrètement le travail qui reste à la charge de l'enseignant. Néanmoins, afin de ne pas alourdir cette présentation, nous n'avons pas évoqué les problèmes que nous avons rencontrés ni mentionné les méthodes (généralement classiques) utilisées pour les résoudre. Ces éléments techniques sont regroupés dans une notice destinée aux utilisateurs du système.

En fait, la difficulté majeure de l'élaboration de CASCADE résidait dans le volume considérable de la programmation et donc l'organisation très stricte nécessitée par la conception et la gestion de la centaine de programmes FORTRAN qui composent le système. Ce nombre qui peut paraître élevé provient en partie de la faible taille mémoire de notre ordinateur (32 K mots dont 20 K occupés par le système MUTEX). Une configuration comportant une mémoire centrale de 48 K mots associée à une unité de disque de 5 millions d'octets nous paraît suffisante pour accroître sensiblement les performances et minimiser les difficultés de réalisation.

Notre prototype fait largement appel aux outils informatiques de base conçus et réalisés par M. Campmas : MUTEX, GESTRA, INTGEN. En effet, au sein de l'équipe, notre travail constituait la première évaluation de ces outils en mode opérationnel. Nous avons pu ainsi préciser les temps de réponse et parachever la mise au point de ce logiciel. C'est pourquoi l'utilisation de ces programmes qui n'était pas toujours nécessaire, et quelquefois alourdissait même le fonctionnement du système, devra être revue dans une optique nouvelle orientée vers l'optimisation de CASCADE (c'est particulièrement le cas du système générateur dont l'emploi dans un contexte multi-utilisateur n'a qu'un intérêt limité). Le temps de réponse obtenu est

assez élevé lors de la construction du système (temps moyen d'interprétation et d'exécution d'une instruction des langages BLOPRO et MODPHY : 1,5 seconde), mais demeure très raisonnable lors de l'utilisation du système par les étudiants (temps d'exécution moyen d'une instruction de la stratégie pédagogique : 0,1 seconde). Nous pensons qu'une utilisation plus judicieuse des outils que nous avons définis, associée à un accroissement de la capacité mémoire (48 K mots de mémoire centrale) permettra de diviser ces temps dans un rapport de 4 ou 5 dans la phase de conception d'un système par le professeur et dans un rapport de 2 à 3 lors de son utilisation par les étudiants.

Nous n'avons pu effectuer une évaluation pédagogique des outils que nous proposons, en effet, la constante de temps en éducation est grande, d'autant plus que des mesures sur notre système ne peuvent être réalisées qu'avec l'aide d'enseignants déjà bien occupés par ailleurs. Ce travail d'évaluation qui va maintenant commencer permettra d'acquérir les données concrètes qui, aujourd'hui, nous font défaut dans la critique de notre approche et l'évaluation des performances du système construit.

CONCLUSION

Après la construction du prototype d'Enseignement Assisté "ESPACE" par J.C.LATOMBE* et J.C.SABONNADIÈRE* [36] et son utilisation durant trois années scolaires par les étudiants de l'ENSEGP** sous la conduite d'A.BOLOPION*, il devenait nécessaire de tirer les conséquences de cette expérience dont les résultats [42] validaient l'ensemble des options pédagogiques qui avaient présidé à la définition du système ESPACE.

Dans cette perspective, l'extension de ce type de pédagogie à d'autres activités que le projet de construction d'ensembles techniques : travaux dirigés, cours, mini-projets... allait permettre de proposer une méthode d'enseignement générale applicable à de nombreuses disciplines et notamment aux sciences de l'ingénieur.

Après avoir dégagé les grandes lignes d'une nouvelle forme d'enseignement Assisté que nous avons expérimentée, nous avons précisé

* : Chercheur au Laboratoire d'Electrotechnique et de Génie Physique

** : Ecole Nationale Supérieure d'Electrotechnique et de Génie Physique.

les ressources informatiques nécessaires pour qu'un enseignant puisse réaliser par lui-même l'application de cette méthode à sa propre discipline.

Enfin, la définition d'une méthodologie générale de la construction de systèmes d'EAO nous a permis de spécifier un système de construction assistée destiné au professeur : le système C.A.S.C.A.D.E.

Ce système impose à l'enseignant de travailler dans un cadre didactique rigoureux et, prenant à sa charge la réalisation des tâches fastidieuses, l'affranchit des problèmes informatiques. Le professeur peut alors concentrer ses efforts sur la partie de la conception de son futur système d'EAO qui est spécifique de sa discipline.

Un prototype du système CASCADE a été implanté sur l'ordinateur Télémecanique 1600 de l'ENSEGP. Sa programmation a été facilitée par les outils informatiques de base développés par M. CAMPMAS dans le cadre du système MUTEX [45]. Ce prototype est actuellement achevé ; une phase de test détaillé lui confèrera la grande fiabilité que nécessite toujours l'utilisation de l'ordinateur à des fins pédagogiques.

La conception de système d'EAO sera facilitée par ce nouvel outil dont il faut maintenant mesurer l'impact sur le monde enseignant (en incitant des professeurs à construire des applications d'EAO avec CASCADE). Il faudra ensuite l'adapter aux besoins exprimés par les critiques que son utilisation ne manquera pas de soulever. Cette tâche très délicate nécessitera encore beaucoup de travail mais permettra de faire passer le système CASCADE de l'utilisation expérimentale au stade opérationnel. (Une version plus optimisée et totalement transportable est dorés et déjà à l'étude).

Nous pensons qu'alors le système CASCADE constituera un outil original de recherche en didactique et que la souplesse des stratégies pédagogiques qu'il permet de définir débouchera à plus long terme sur une meilleure connaissance des processus d'apprentissages et donc sur une meilleure utilisation de l'ordinateur en enseignement.

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

- |1| J.J.ALLAN III. Foundations of the many manifestations of CAD :
Proceedings of IFIP working conf on principles of
CAD.
Eindhoven, Oct.72 - Pages 27 à 58.
- |2| F.HARARY. Théorie des Graphes Orientés, Modèles structuraux :
Proc. of UCODI Summer School Geneve 1975 (à paraître).
- |3| R.J.SHAVELSON. Stanford center for research and development in
teaching : The Theory of directed graphs : Some
applications to research on teaching. (Research and
Development Memorandum N.71). February 1971 36 pages.
- |4| C.BERGE. Optimisation d'une stratégie par la théorie des
graphes. Les méthodes d'apprentissages : Proc. of UCODI
Summer School Geneve 1975 (à paraître).
- |5| A.KAUFMANN, M.COOLS, Th.DUBOIS.
Stimulation inventive dans un dialogue homme-machine
utilisant la méthode des morphologies.
Communication au Symposium de Glarus 25-27 oct.73 -
8 pages.
- |6| A KAUFMANN. Ebauche d'une théorie de l'opérateur humain.
Learning management based on formal models of behavior
and aptitudes in Computer-Assisted System of
instruction.
Proc. UCODI Summer School Ang. 74.
- |7| P.RAFINEJAD and J.C.SABONNADIÈRE.
Finite-element computer programs in design of electro-
magnetic devices. Intermag Conference Paper 17-20
April 1975 - London England (à paraître dans IEEE
Transactions on Magnetics - Sept.76).
- |8| P.SILVESTER and P.RAFINEJAD.
"Curvilinear finite elements for two dimensional
saturable magnetic fields : IEEE Transactions on
Power Apparatus Systeme. Vol. PAS-93. 1974 pp 1861-1870.

- [9] J.C.LATOMBE. Artificial Intelligence in Computer Aided Design :
The TROPIC system.
Proc. IFIP Working Conference 5.2. on CAD systems.
Feb.76 (Pub by North-Holland Publish CO). (à paraître).
- [10] S.LEBLANC. Une expérience de recyclage dans l'industrie au moyen
de l'enseignement assisté par ordinateur.
ACTA. ELECTRONICA. 17.3.74 pages 297-318.
- [11] D.ROOS. ICES System : General Description. MIT Department
of Civil Engineering Report R.67-49. September 1967.
- [12] F.ADAM, R.JACOUD, M.JACQUES, J.J.PELLETTREAU, J.PENNE, J.P.RAMBAUT,
and J.P.TOULOUSE.
A current CAI project : the OPE System.
Proc. IFIP Conf. Stockholm Aug. 1974 p. 875 à 879.
- [13] R.JACQUART, P.REGNIER, F.R.R.VALETTE.
GERMINAL toward a general and integrated system for
CAD. Journées sur les systèmes généraux pour la CAO
Toulouse 1974.
- [14] R.W.TINKER, I.L.AVRUNIN.
The COMRADE executive system : National Computer
Conference 1973 - pp. 331-337.
- [15] J.R.ABRIAL, J.P.CAHEN, J.C.FAVRE, D.PORTAL, G.MAZARE et R. MORIN.
Projet SOCRATE.
USMG Mathématiques Appliqués-Informatique. Nouvelles
spécifications. Sept.72. Grenoble.
- [16] P.RAFINEJAD. Systeme FLUX : un système interactif d'aide au calcul
des champs électromagnétiques. Rapport Général
Equipe ECAO - Laboratoire Electrotechnique - INP
Grenoble 1975.
- [17] B.M. KARUNARATINE.
Contribution à l'étude informatique des réseaux de transp
transport d'énergie. Thèse de 3° Cycle.
Grenoble. Juin 74 - 88 pages.
- [18] J.C.SABONNADIÈRE, PH.AURIOL.
Surtensions de manoeuvre dans les réseaux HT et THT.
Principales méthodes de calcul des régimes transitoires
dans les réseaux THT. Revue Générale de l'Electricité.
Nov.1973. t.82, n°11 - p. 718 à 727.
- [19] D.ROOS, J.E.RODRIGUEZ.
Theoretical foundations for the CAD Systems ; AFIPS,
Proc. SJCC /vol 23/1963.

- | 20 | R.JACQUART, Ph.REGNIER, F.R.VALETTE.
GERMINAL un système intégré et conversationnel pour
la CAO:Automatisme, T XIX n°3 mars 1974.
- | 21 | B.GOTT. The scope of computer-aided design : Proceedings of
IFIP Working Conf. on principles of CAD.
Eindhoven. Oct.72. Pages 1 à 25.
- | 22 | B.DAVID. Pour une généralisation des systèmes CAO approches et
applications. Thèse 3ème Cycle. Oct.75 - Grenoble.
- | 23 | R.L.KLAUS. New Concepts in interactive computing and their
relations ship to computer-Aided Design : Proceeding
of IFIP Working Conference on principles of CAD
Eindhoven. Oct.72. Page 80 à 111.
- | 24 | D.ALBERT. The Plato IV system in use : a progress report. Proc.
of IFIP Conf. Marseille. Sept.1975 (O.Lecarme, R.Lewis
editors). p. 181-186
- | 25 | F. MORRISON. Planning a large scale computer assisted instruction
installation : the TICCIT experience.
Proc of IFIP Conf. Marseille. Sept.1975.
(O.Lecarme, R.Lewis editors), P. 187-190.
- | 26 | FRANK, KELBERT, LANSKY, PROCHNOV.
Lehrmaschinen in kybernetischer und pädagogischer
Sicht. Klett-Oldenbourg 1966.
- | 27 | M. LANSKY. A cybernetic Model of Group learning.
VII Congres Intern. de Cybernetique. Namur 1973.
p. 677-695.
- | 28 | M.QUERE. Modèle Mathématique d'un Système d'enseignement.
Proc IFIP Marseille. Sept.75. (O.lecarme and R.Lewis
editors). p.895 à 898 T.1.
- | 29 | MAC KENZIE, M.ERAUT, H.C.JONES.
L'art d'enseigner et l'art d'apprendre. UNESCO
Paris 1971 - 239 pages.
- | 30 | T.H.GAWAIN. Some reflexions on education of creativity in enginee-
ring. IEEE transaction on education. Nov.1974.
p. 189-192.
- | 31 | A.BODE. Universitaet Karlsruhe Institut Fuer Informatik
LEGIS-SHUELER-HANDBUCH (Fuer Die 86700) 32 pages.

- [32] R.M.HARKNES, P.B.MORICE.
The computer in the teaching of engineering sciences.
Proc.IFIP. Marseille. Sept.75 - p 975 à 978 T2.
- [33] PLOTZ, DIAMOND, SCHLESINGER.
Computer Assisted Simulation of the CHART Method of
teaching.
Proc of UCODI Summer School. Aug. 74.
- [34] F.N.BLONDEL, J.F.BOYER, G.FLANDRIN, G.LATGE, J.P.LEVY, I.N'GOSSO,
N.SALAME, PH.SARTORY and B.VARET.
Enseignement assisté par ordinateur en médecine.
Expérience en Hématologie.
Proc IFIP, Marseille. Sept.75. p 519 à 522.
- [35] J.C.LATOMBE et J.C.SABONNADIERE.
Elaboration d'un système informatique intégré pour
l'assistance à l'enseignement et à la conception en
Electrotechnique. Revue Française d'Automatique.
Informatique et Recherche Opérationnelle. Juin 1972.
pp. 141-164.
- [36] J.C.LATOMBE. Etude d'un système pédagogique d'assistance à la
conception en Electrotechnique. Thèse de Docteur-
Ingénieur.
USMG. Grenoble. Nov. 1972 - 158 pages.
- [37] Equipe ECAO de l'Institut National Polytechnique de Grenoble :
Problem Solving : objectifs, activités, contraintes
technologiques. Cours présenté à l'Ecole d'été
UCODI.
Pugnochiuso Bari Italie. Août. Sept.73.
- [38] J.C.LATOMBE. E.S.P.A.C.E. : A computer aided instruction system
oriented towards educational problem solving activi-
ties. R.G.U. 74 Conference on computer based learning
systems, Hambourg, August 1974, 10 pages.
- [39] A.BOLOPION, M.CAMPAS, J.C.LATOMBE, and J.C.SABONNADIERE.
A new approach to C.A.I. by interactive design of
electrical engineering systems. I.E.E.E. Transactions
on education, May 1975, pp. 87-93.
- [40] Equipe ECAO de l'Institut National Polytechnique de Grenoble.
ESPTRA : utilisation de l'application transformateur.
Manuel d'utilisation de ESPACE/CP/CMS, Volume 8, 1975.
- [41] Equipe ECAO de l'Institut National Polytechnique de Grenoble.
FORCE : Utilisation de l'application réseaux électriques
(Régime permanent). Manuel d'utilisation de
ESPACE CP/CMS, Volume 10, 1974, 103 pages.

- [42] A. BOLOPION. Mise en oeuvre et expérimentation pédagogique d'un système d'enseignement assisté par ordinateur en électrotechnique.
Thèse Docteur Ingénieur USMG Grenoble. Oct. 75.
110 pages.
- [43] A. BOLOPION, J.C. SABONNADIÈRE.
Une expérience d'intégration de l'ordinateur dans la formation d'ingénieurs. Conférence on "The future progress of electrical engineering",
Prague (Tchécoslovaquie), Octobre 1975, 19 pages.
- [44] A. BOLOPION, J.C. SABONNADIÈRE.
Bilan de quelques années d'expérimentation pédagogique d'un système d'E.C.A.O destiné à la formation des ingénieurs. I.F.I.P. 2ème conférence Internationale Informatique et Enseignement, Marseille, Septembre 1975, 6 pages.
- [45] M. CAMPMAS. Conception et réalisation de logiciels destinés à faciliter l'implantation de systèmes d'enseignement assisté sur mini-ordinateur.
Thèse Docteur Ingénieur. INP Grenoble. Juin 76.
90 pages.
- [46] M. CAMPMAS, J.C. LATOMBE.
Un programme général pour l'interprétation de langages d'application : Automatisation. Tome XXI
Jan-Fév. 76 p. 12 à 18.
- [47] A. BOLOPION, J.C. LATOMBE, PH. MASSE.
Problem Solving Methods and Computer used as a tool to teach power system and apparatus science.
4th. ICEE. Shiraz. Iran. May 74.
- [48] Ph. MASSE, J.C. SABONNADIÈRE.
Les graphes un support conceptuel puissant pour la construction de systèmes d'E.C.A.O.
Proc. of UCODI. Summer School. Aug 75. Genève.
- [49] Ph. MASSE, J.C. SABONNADIÈRE.
A computer aided instruction system for network engineering education. Conférence on "The future progress of electrical engineering", Prague.
(Tchécoslovaquie). Octobre 1975. 9 pages.
- [50] Ph. MASSE, J.C. SABONNADIÈRE.
CASCADE une aide à la conception de systèmes d'enseignement assisté par ordinateur. Panorama de la nouveauté informatique en France. Nov. 76. 8 pages.
(à paraître).

- |51| J.L.COULOMB, Ph.MASSE, J.C.SABONNADIÈRE.
How to build your own CAD system : the CASCADE program
and its application to Finite Element Program Design.
ICCAD Inter. Conf. on Numerical methods in electrical
and Magnetic Field Problem. Italie. Juin 76.
- |52| J.MELESE.
Analyse Modulaire des Systèmes de Gestion A.M.S.
Edit. Hommes et techniques 1972. 233 pages.
- |53| H.SIMON.
Computer Assisted learning in Pairs : effects on the
cognitive learning performance and effective attitude.
R.G.U. 74. Conference Hambourg. Aug. 74.
- |54| H.SIMON.
La Science des systèmes EPI editeur. Paris 1974 p. 154.
- |55| D.G.HULL and J.L.DOWELL.
PROJECT C.BE. Computer Based Education.
Program FIGHTER.
University of TEXAS at Austin.
- |56| KINGSTON POLYTECHNIC.
MSc degree in computer aided engineering systems design
Programme de cours du Kingston Polytechnic
(Angleterre), Juillet 1974.

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,
VU le rapport de présentation de

- . Monsieur R. JACOUD, Professeur à l'Université de
PARIS VII
- . Monsieur J.C. SABONNADIÈRE, Maître de Conférences à
l'Institut National Poly-
technique de GRENOBLE

Monsieur M A S S É Philippe

est autorisé à présenter une thèse en soutenance pour l'obtention
du diplôme de DOCTEUR-INGENIEUR, spécialité "ELECTROTECHNIQUE".-

Fait à Grenoble, le 16 Septembre 1976

Le Président,

Ph. TRAYNARD
Président
de l'Institut National Polytechnique

