E.A.O. et enseignement de la programmation : une maquette de didacticiel

Jésus Zambrano

▶ **To cite this version:**

Jésus Zambrano. E.A.O. et enseignement de la programmation : une maquette de didacticiel. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1984. Français. NNT : . tel-00311951

HAL Id: tel-00311951
https://theses.hal.science/tel-00311951

Submitted on 22 Aug 2008

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

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

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR DE 3ème CYCLE
«Informatique»

par

Jésus ZAMBRANO

E.A.O. ET ENSEIGNEMENT DE LA PROGRAMMATION :

UNE MAQUETTE DE DIDACTICIEL.

Thèse soutenue le 30 octobre 1984 devant la commission d'examen.

G. VEILLON	Président
J.P. FINANCE	Examinateurs
Ph. JORRAND	
M. MILCHBERG	
P.C. SCHOLL	

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : D. BLOCH

Vice-Président : René CARRE
Hervé CHERADAME
Marcel IVANES

PROFESSEURS DES UNIVERSITES :

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	E.N.S.I.E.G.
BESSON Jean	E.N.S.E.E.G.
BLIMAN Samuel	E.N.S.E.R.G.
BLOCH Daniel	E.N.S.I.E.G.
BOIS Philippe	E.N.S.H.G.
BONNETAIN Lucien	E.N.S.E.E.G.
BONNIER Etienne	E.N.S.E.E.G.
BOUVARD Maurice	E.N.S.H.G.
BRISSONNEAU Pierre	E.N.S.I.E.G.
BUYLE BODIN Maurice	E.N.S.E.R.G.
CAVAIGNAC Jean-François	E.N.S.I.E.G.
CHARTIER Germain	E.N.S.I.E.G.
CHENEVIER Pierre	E.N.S.E.R.G.
CHERADAME Hervé	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.M.C.P.P.
COHEN Joseph	E.N.S.E.R.G.
COUMES André	E.N.S.E.R.G.
DURAND Francis	E.N.S.E.E.G.
DURAND Jean-Louis	E.N.S.I.E.G.
FELICI Noël	E.N.S.I.E.G.
FOULARD Claude	E.N.S.I.E.G.
GENTIL Pierre	E.N.S.E.R.G.
GUERIN Bernard	E.N.S.E.R.G.
GUYOT Pierre	E.N.S.E.E.G.
IVANES Marcel	E.N.S.I.E.G.
JAUSSAUD Pierre	E.N.S.I.E.G.
JOUBERT Jean-Claude	E.N.S.I.E.G.
JOURDAIN Geneviève	E.N.S.I.E.G.
LACOUME Jean-Louis	E.N.S.I.E.G.
LATOMBE Jean-Claude	E.N.S.I.M.A.G.

.../...

LESSIEUR Marcel	E.N.S.H.G.
LESPINARD Georges	E.N.S.H.G.
LONGEQUEUE Jean-Pierre	E.N.S.I.E.G.
MAZARE Guy	E.N.S.I.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.A.G.
PARIAUD Jean-Charles	E.N.S.E.E.G.
PAUTHENET René	E.N.S.I.E.G.
PERRET René	E.N.S.I.E.G.
PERRET Robert	E.N.S.I.E.G.
PIAU Jean-Michel	E.N.S.H.G.
POLOUJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIERE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

PROFESSEURS ASSOCIES

BASTIN Georges	E.N.S.H.G.
BERRIL John	E.N.S.H.G.
CARREAU Pierre	E.N.S.H.G.
GANDINI Alessandro	U.E.R.M.C.P.P.
HAYASHI Hirashi	E.N.S.I.E.G.

PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis
Chatelin Françoise

PROFESSEURS E.N.S. Mines de Saint-Etienne

RIEU Jean
SOUSTELLE Michel

CHERCHEURS DU C.N.R.S.

FRUCHART Robert	Directeur de Recherche
VACHAUD Georges	Directeur de Recherche

.../...

ALLIBERT Michel	Maître de Recherche
ANSARA Ibrahim	Maître de Recherche
ARMAND Michel	Maître de Recherche
BINDER Gilbert	
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DEPORTES Jacques	
DRIOLE Jean	Maître de Recherche
GIGNOUX Damien	
GIVORD Dominique	
GUELIN Pierre	
HOPFINGER Emil	Maître de Recherche
JOUD Jean-Charles	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Dore	Maître de Recherche
LASJAUNIAS J.C.	
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche
PIAU Monique	
PORTESEIL Jean-Louis	
THOLENCE Jean-Louis	
VERDILLON André	

CHERCHEURS du MINISTERE de la RECHERCHE et de la TECHNOLOGIRE (Directeurs et Maîtres de Recherches, ENS Mines de St. Etienne)

LESBATS Pierre	Directeur de Recherche
BISCONDI Michel	Maître de Recherche
KOBYLANSKI André	Maître de Recherche
LE COZE Jean	Maître de Recherche
LALAUZE René	Maître de Recherche
LANCELOT Francis	Maître de Recherche
THEVENOT François	Maître de Recherche
TRAN MINH Canh	Maître de Recherche

PERSONNALITES HABILITEES à DIRIGER des TRAVAUX de RECHERCHE (Décision du Conseil Scientifique)

ALLIBERT Colette	E.N.S.E.E.G.
BERNARD Claude	E.N.S.E.E.G.
BONNET Rolland	E.N.S.E.E.G.
CAILLET Marcel	E.N.S.E.E.G.
CHATILLON Catherine	E.N.S.E.E.G.
CHATILLON Christian	E.N.S.E.E.G.
COULON Michel	E.N.S.E.E.G.
DIARD Jean-Paul	E.N.S.E.E.G.
EUSTAPOPOULOS Nicolas	E.N.S.E.E.G.
FOSTER Panayotis	E.N.S.E.E.G.

.../...

GALERIE Alain	E.N.S.E.E.G.
HAMMOU Abdelkader	E.N.S.E.E.G.
MALMEJAC Yves	E.N.S.E.E.G. (CENG)
MARTIN GARIN Régina	E.N.S.E.E.G.
NGUYEN TRUONG Bernadette	E.N.S.E.E.G.
RAVAINE Denis	E.N.S.E.E.G.
SAINFORT	E.N.S.E.E.G. (CENG)
SARRAZIN Pierre	E.N.S.E.E.G.
SIMON Jean-Paul	E.N.S.E.E.G.
TOUZAIN Philippe	E.N.S.E.E.G.
URBAIN Georges	E.N.S.E.E.G. (Laboratoire des ultra-réfractaires ODEILLON)
GUILHOT Bernard	E.N.S. Mines Saint Etienne
THOMAS Gérard	E.N.S. Mines Saint Etienne
DRIVER Julien	E.N.S. Mines Saint Etienne
BARIBAUD Michel	E.N.S.E.R.G.
BOREL Joseph	E.N.S.E.R.G.
CHOVET Alain	E.N.S.E.R.G.
CHEHIKIAN Alain	E.N.S.E.R.G.
DOLMAZON Jean-Marc	E.N.S.E.R.G.
HERAULT Jeanny	E.N.S.E.R.G.
MONLLOR Christian	E.N.S.E.R.G.
BORNARD Guy	E.N.S.I.E.G.
DESCHIZEAU Pierre	E.N.S.I.E.G.
GLANGEAUD François	E.N.S.I.E.G.
KOFMAN Walter	E.N.S.I.E.G.
LEJEUNE Gérard	E.N.S.I.E.G.
MAZUER Jean	E.N.S.I.E.G.
PERARD Jacques	E.N.S.I.E.G.
REINISCH Raymond	E.N.S.I.E.G.
ALEMANY Antoine	E.N.S.H.G.
BOIS Daniel	E.N.S.H.G.
DARVE Félix	E.N.S.H.G.
MICHEL Jean-Marie	E.N.S.H.G.
OBLED Charles	E.N.S.H.G.
ROWE Alain	E.N.S.H.G.
VAUCLIN Michel	E.N.S.H.G.
WACK Bernard	E.N.S.H.G.
BERT Didier	E.N.S.I.M.A.G.
CALMET Jacques	E.N.S.I.M.A.G.
COURTIN Jacques	E.N.S.I.M.A.G.
COURTOIS Bernard	E.N.S.I.M.A.G.
DELLA DORA Jean	E.N.S.I.M.A.G.
FONLUPT Jean	E.N.S.I.M.A.G.
SIFAKIS Joseph	E.N.S.I.M.A.G.
CHARUEL Robert	U.E.R.M.C.P.P.
CADET Jean	C.E.N.G.
COEURE Philippe	C.E.N.G. (LETI)

.../...

DELHAYE Jean-Marc	C.E.N.G. (STT)
DUPUY Michel	C.E.N.G. (LETI)
JOUVE Hubert	C.E.N.G. (LETI)
NICOLAU Yvan	C.E.N.G. (LETI)
NIFENECKER Hervé	C.E.N.G.
PERROUD Paul	C.E.N.G.
PEUZIN Jean-Claude	C.E.N.G. (LETI)
TAIEB Maurice	C.E.N.G.
VINCENDON Marc	C.E.N.G.

LABORATOIRES EXTERIEURS

DEMOULIN Eric	C.N.E.T.
DEVINE	C.N.E.T. (R.A.B.)
GERBER Roland	C.N.E.T.
MERCKEL Gérard	C.N.E.T.
PAULEAU Yves	C.N.E.T.
GAUBERT C.	I.N.S.A. Lyon

A Ankeke

TABLE DES MATIERES

I L'enseignement de la programmation et l'enseignement assisté par ordinateur. I - 1

I.1 L'enseignement de la programmation. I - 1
I.2 L'enseignement assisté par ordinateur. I - 10
I.3 Conclusions. I - 16

II Spécification pédagogique de la maquette. II - 1

II.1 Sujets abordés dans la maquette. II - 2

II.1.1 Notion d'algorithme. II - 2
II.1.2 Notion d'analyse descendante. II - 3
II.1.3 Modèles d'algorithme servant de support à l'exposition des sujets choisis. II - 4

II.2 Les moments pédagogiques. II - 11

II.2.1 Sensibilisation au contexte de travail. II - 12
II.2.2 Présentation de connaissances. II - 16
II.2.3 Consolidation des connaissances. II - 18
II.2.4 Typologie d'erreurs. II - 20

II.3 Les approches pédagogiques. II - 22

II.3.1 La lecture d'un document d'analyse. II - 22
II.3.2 La simulation. II - 27

II.4 Conclusions. II - 28

III Les solutions E.A.O. III - 1

III.1 Stratégie pour la définition d'une solution E.A.O. III - 2
III.2 Le choix des différents moyens E.A.O. mis en oeuvre. III - 3
III.3 Structuration de la maquette du didacticiel. III - 4

III.3.1 Solutions E.A.O. spécifiques au didacticiel. III - 5
III.3.2 Spécification des dialogues. III - 7

III.3.2.1 Dialogue de sensibilisation. III - 7
III.3.2.2 Dialogue de présentation. III - 12
III.3.2.3 Dialogue de consolidation. III - 20

III.4 Synthèse sur les fonctions E.A.O. nécessaires. III - 22

III.4.1 Pour la lecture d'algorithmes. III - 22
III.4.2 Pour la simulation. III - 28

III.5 Méthode de production du didacticiel. III - 30

III.5.1 Etape de préparation pédagogique. III - 30
III.5.2 Etape de transition pédagogie-informatique. III - 31
III.5.3 Etape de préparation infomatique. III - 31

III.6 Les leçons. III - 33
III.7 La leçon-machine. III - 52
III.8 Conclusions. III - 55

IV Considérations sur l'expérience réalisée. IV - 1

IV.1 Quelques résultats de l'expérience. IV - 4
IV.2 Quelques perspectives pour la poursuite de la recherche. IV - 9
IV.3 Conclusions. IV - 13

ANNEXES.

ANNEXE I Eléments de la notation algorithmique.
ANNEXE II Production des leçons.

BIBLIOGRAPHIE

Enseignement de la Programmation et Enseignement assisté par Ordinateur :

Une maquette de didacticiel

REMERCIEMENTS

Je tiens à remercier Monsieur G. VEILLON Professeur à l'I.N.P.G., Directeur de l'E.N.S.I.M.A.G., pour l'honneur qu'il me fait de présider le jury de cette thèse.

Je remercie Monsieur Jean Pierre FINANCE, Professeur à l'Université de Nancy I d'avoir accepté de participer à mon jury.

Je prie Monsieur P. JORRAND, Directeur de Recherche au C.N.R.S., de trouver ici l'expression de ma gratitude. Je lui suis reconnaissant d'avoir bien voulu accepter de faire partie du jury.

Je remercie Monsieur P.C. SCHOLL, Professeur à l'U.S.M.G., de m'avoir acueilli dans son équipe de recherche et d'avoir accepté la responsabilité de ces recherches. Je tiens à souligner l'importance qu'ont eu pour moi, la rigueur et l'esprit critique avec lesquels Mr. SCHOLL a suivi ce travail depuis le début, il m'a appris beaucoup.

Je tiens à remercier tout particulièrement Monsieur Jean Pierre PEYRIN, maître assistant à l'U.S.M.G. pour l'immense intérêt qu'il a porté à mon travail, pour le dévouement dont il a fait preuve en m'aidant et en me soutenant dans ma tâche, et aussi pour les encouragements qu'il m'a prodigués. Depuis le début de mes études en France, j'ai pu apprécier sa constance dans le travail et son humeur toujours égale. Je lui dois beaucoup.

Je voudrais remercier aussi chaleureusement Monsieur Mauricio MILCHBERG, Professeur à l'U.S.M.G qui, depuis que j'étais au Vénézuela, m'a toujours conseillé et guidé dans mon travail et ma recherche. Il aveillé personnellement à ma formation d'enseignant, et il a suivi, en l'orientant, l'évolution de ma carrière. Je le remercie pour tout le temps et les conseils qu'il m'a donnés, et qui m'ont permis d'arriver là où je suis aujourd'hui.

Je remercie également Monsieur Georges FAFIOTTE, enseignant à l'U.S.S.G., pour ses conseils et critiques en tant qu'expert en E.A.O. et spécialiste des matériels informatiques.

Je remercie mon ami Philippe MORAT, pour toute l'aide matérielle et morale, ainsi que pour cet "accueil positif" qu'il a donné toujours dans toutes les occasions.

Je veux remercier infiniment Mme. Annie FAUSTINI pour toute sa collaboration et l'aide qu'elle m'a apportée dans la rédaction et la correction de cette thèse.

Je tiens également à remercier mes amis Sophie LEMOINE, Sylvie PAINVIN, Alain LUCCI, Jean-Michel ADAM et Heloise MOITTA-MONTE ainsi que mes compatriotes pour tout l'appui et la gentillesse qu'ils m'ont temoignés.

Un mot de remerciement pour Mme. BRUNET d'une part et le personnel de la scolarité de l'I.N.P.G., d'autre part, pour leur accueil et leur attention envers moi au cours de ces dernières années.

Monsieur Daniel IGLESIAS a assuré avec Monsieur Claude ANGUILLE et tout le personnel le tirage de ce travail et je les en remercie.

Je veux remercier, enfin, toutes les personnes qui, d'une façon ou d'une autre, m'ont aidé à la reálisation de ce travail. Pour tous "muchas gracias".

Cette recherche a été effectué pendant que j'étais boursier du Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela.

CHAPITRE I : L'enseignement de la programmation et l'enseignement assisté par ordinateur.

I L'enseignement de la programmation et l'enseignement assisté par ordinateur.

Nous abordons, dans ce premier chapitre, quelques aspects de l'enseignement de la programmtion et de l'enseignement assisté par ordinateur (E.A.O), afin de présenter au lecteur le point de départ de notre travail.

I.1 L'enseignement de la programmation.

Après de nombreuses années d'investigation, la communauté informatique s'accorde à l'heure actuelle sur le fait qu'un enseignement de la programmation doit traiter à la fois des méthodes propres à la construction des programmes et des techniques qui permettent la description de programmes, en tenant compte de contraintes imposées par l'environnement informatique utilisé et des qualités attendues du produit programme.

Par ailleurs, on cherche de plus en plus à situer la programmation dans une démarche informatique qui comporte les étapes essentielles suivantes : décision d'informatiser une situation donnée; élaboration d'un cahier des charges, analyse d'un problème conduisant à sa spécification en termes informatiques, construction d'un programme, mise en oeuvre de tests, rédaction de documents, intégration dans le site d'utilisation, exploitation et maintenance.

Ce consensus peut être observé en analysant un certain nombre d'ouvrages destinés aux étudiants, qui selon l'auteur, le public visé et le niveau considéré mettent plus particulièrement l'accent sur tel ou tel point.

Nous omettons sciemment de citer ici les ouvrages plus particulièrement centrés sur un langage particulier (Basic par exemple) destinés à un public autodidacte et dont le nombre considérable s'explique par l'explosion actuelle de l'informatique grand public; ces ouvrages nous paraissent typiques d'une conception ancienne et dangereuse de l'apprentissage de la programmation.

La plus grande partie des textes qui s'appuient sur une conception méthodique de la programmation traitent d'algorithmique et de représentation des données ([BoM-83], [BiC-84], [ClB-84], [LPS-83], [Luc-83], [Sch-84], [Duc-84], et quelques classiques anglo-saxons tels que par exemple, [Knu-69], [Wir-73], [AHU-83]).

D'autres ouvrages se consacrent plus spécifiquement à l'algorithmique ([Ars-80], [Ars-83], [Lau-83]); ou à certains aspects plus théoriques de la programmation ([Gri-83], [Dij-76], [BeB-83], [Liv-78]) ou à l'étude de structures de données particulières [CoP-83]. Enfin on voit apparaître en France des ouvrages destinés particulièrement à des élèves du secondaire.

Par ailleurs, il faut aussi considérer les textes qui concernent plus particulièrement la production de logiciels selon une méthode particulière (Warnier [War-74], Jackson [Jac-75], Aladin [RiR-81], ...).

Cette analyse bibliographique nous permet de résumer ici quelques éléments fondamentaux sur l'activité de la programmation qui doivent éclairer la conception de son enseignement :

a) L'activité de la programmation repose sur une distinction claire entre l'analyse, la construction proprement dite, et la vérification progressive de la correction.

* L'analyse du problème conduit du cahier des charges, formulé en langue naturelle dans des termes qui relèvent du domaine d'application envisagé, à une spécification informatique qui exprime les relations précises entre données et résultats en cherchant à faire abstraction au maximun de contraintes informatiques spécifiques.

La forme de cette spécification est extrêmement variable selon l'application considerée, l'existence de normes précises dans l'environnement humain de programmation envisagé, et la culture même des programmeurs.

Les méthodes qui permettent de conduire une telle analyse s'appuient sur des principes généraux tels que l'analyse descendante et quelques techniques de décomposition (séquentialisation, analyse par cas, répartition, induction)[ANA-84].

* La construction proprement dite d'un programme conduit d'une spécification à un texte rédigé dans un langage de programmation.

C'est dans cette phase qu'on cherche à utiliser de la manière la plus adéquate un ensemble de techniques répertoriées de représentation du contrôle ou des données. C'est aussi dans cette phase que l'on tient compte de particularités de l'environnement informatique visé.

On doit donc savoir maîtriser la diversité des matériels, des langages, des systèmes existants. Ici encore on dispose de méthodes générales telle que la construction ascendante de solutions à partir de composants déjà existants.

* La vérification progressive de la correction de la solution élaborée consiste à assurer à chaque étape du processus de programmation la cohèrence des solutions envisagées par rapport aux spécifications.
La nature de ces vérifications est liée à la précision des formalismes employés.

b) Le produit programme, c'est-à-dire le résultat du travail du programmeur, doit être apprecié en termes de divers facteurs :

- fiabilité.-

le programme répond aux spécifications.

- sécurité.-

l'exploitation du programme est exempte de pannes : arrêt brutal dû a des raisons diverses, notamment des erreurs de l'utilisateur, des erreurs de transmission, des actions malveillantes.

- robustesse.-

l'exécution peut continuer en cas d'accident (erreur de manipulation, erreur de données, ...).

- adaptabilité.-

le programme doit pouvoir être modifié à des coûts raisonnables en cas de changement du cahier des charges.

- maintenabilité.-

le programme doit pouvoir être corrigé et ceci à des coûts raisonnables, lors de la découverte d'erreurs.

Par ailleurs le produit programme doit consister non seulement en un texte final rédigé dans un langage de programmation, mais aussi des divers documents nécessaires à son exploitation et à sa maintenance:

- document d'analyse qui reflète les diverses étapes de l'analyse et explicite les décisions qui ont été prises.

- document technique qui précise le contexte concret de travail utilisé (logiciel et matériel), l'historique du programme (différentes versions), les éléments de tests et d'intégration.

- document utilisateur.

L'ensemble de ces documents destinés à une lecture ultérieure doivent nécessairement être très structurés, très clairs et formulés dans des termes précis et compréhensibles pour les personnes auxquelles ils sont destinés (lisibilité).

A partir de ce constat sur l'activité de la programmation, nous analysons maintenant quelques problèmes spécifiques à la mise en oeuvre de l'enseignement de la programmation. Nous pourrons alors justifier la recherche des solutions de type E.A.O. pour aider à cet enseignement.

On doit tout d'abord constater que le public auquel s'adresse l'enseignement de la programmation présente des caractéristiques particulières.

On observe à l'heure actuelle une très forte demande pour un tel apprentissage émanant de personnes de tous âges et de toutes professions.

Ce public très hétérogène n'a, en général, aucune culture informatique, ou du moins ne partage pas une culture standard comme cela se passe dans d'autres disciplines, du fait de l'absence de l'informatique dans les programmes de la scolarité obligatoire. De plus, " le cours de programmation est le seul cours, pris par un très grand pourcentage d'étudiants, où l'on essaie d'apprendre des méthodes générales pour la résolution des problèmes" [Gri-74].

Enseigner à résoudre des problèmes implique non seulement la description des outils et des méthodes employés, illustrée par de nombreux exemples, mais aussi et surtout la mise en évidence des différents choix faits tout au long du processus qui va de l'énoncé à l'algorithme. Ceci implique aussi le besoin de stimuler la créativité tout en gardant une attitude systématique dans la manière d'aborder le travail.

Les organismes qui proposent un tel enseignement sont donc confrontés à un double problème :

- un problème de masse (nombre important de demandeurs) qui se traduit par un problème de moyens pour l'enseignement (nombre d'enseignants, matériel informatique, structures d'accueil.

- un problème de culture de base qui se traduit notamment par une difficulté à adapter facilement l'initiation au public.

Une deuxième observation générale est liée à l'importance du travail conceptuel dans l'activité de programmation qui s'oppose constamment aux aspects techniques de la discipline et aux relations psychologiques particulières du public, vis-à-vis des machines (aspects ludiques, comportements artisanaux,...).

On doit à ce propos rappeler ici quelques mécanismes intellectuels qui conditionnent l'aisance avec laquelle l'étudiant pourra s'approprier la programmation :

a) Capacité d'abstraction.-

D'une part elle permet de modéliser la réalité de la situation que l'on cherche à informatiser. D'autre part, elle sert à maîtriser la complexité des problèmes et des solutions [Dij-72].

b) Capacité d'analogie.-

Pour trouver une solution, on peut être conduit à considérer des problèmes analogues plus simples, où une solution est connue.

L'analogie peut s'établir en mettant en rapport des éléments d'ordre différents : méthodes, démarche à suivre, données, résultats, etc., du problème dont la solution est connue et du problème à résoudre. Mais ce rapport peut ne pas être établi de façon directe, ce qui peut nous amener à une généralisation d'un problème "guide" (on passe de l'examen d'un ensemble limité d'objets à celui d'un ensemble plus étendu incluant le premier) ou nous amener à une particularisation d'un problème "guide" (on passe d'un ensemble d'objets donnés à un ensemble plus restreint).

Les analogies trouvées peuvent être de niveaux bien différents. On peut arriver à des degrés de précision assez importants comme dans le cas de certains modèles mathématiques.

c) Aptitude au raisonnement formel.-

La construction de programmes s'appuie notamment sur l'utilisation de divers formalismes adaptés à chacune des étapes du processus de programmation.

Le programmeur doit accepter de recourir à des formalismes précis. De plus il doit savoir choisir le formalisme le plus adéquat pour une situation donnée. En outre, la compréhension des langages de programmation repose sur une bonne vision de la notion de langage.

Enfin, le travail de vérification des programmes repose de plus en plus (au fur et à mesure que des méthodes pragmatiques sont développées) sur des raisonnements formels dont l'objet est la définition des propriétés des textes manipulés au cours de la production d'un programme.

L'importance de ces mécanismes intellectuels apparaît dès le début de l'apprentissage de la programmation : en effet on doit y introduire simultanément les notions de processus, d'algorithme et d'analyse descendante, ainsi que les notations pour les exprimer.

On peut citer, par exemple, les points suivants :

I - les étudiants ont des difficultés à comprendre la différence entre le processus engendré à partir d'un programme (phénomène dynamique) et le texte de l'algorithme qui est une représentation statique de ce processus. Ceci explique en partie les problèmes rencontrés en général par l'enseignant pour faire raisonner les étudiants formellement sur ces notions. (ces difficultés sont accrues lorsque le texte décrit un processus parallèle).

I - l'introduction des notations posent des problèmes diver selon la démarche employée.
Lorsque l'enseignant s'appuie directement sur un langage de programmation, il se voit forcé de présenter nombre de détails syntaxiques (voire sémantiques) qui n'ont pas normalement lieu d'être traités dès l'initiation.
Pour pallier à cet inconvénient on peut songer à employer un pseudo langage sans insister sur sa définition syntaxique. Ceci est vite incompatible avec la nécessité d'introduire, le plus tôt possible, l'idée que la compréhension des programmes doit s'appuyer sur des raisonnements formels.

I - la plupart des étudiants manquent d'habitude pour concevoir des systèmes complexes. L'analyse descendante qui répond à ce problème, est le plus souvent présentée à partir d'exemples simples : ceci est dû à l'importance du nombre d'étudiants, aux limitations en temps, et à l'insuffisance des ressources informatiques et pédagogiques.

Sur la base des ces observations que nous venons de faire sur la programmation et les problèmes posés pour son enseignement, nous avons pu mieux cerner où pouvait se situer un apport de l'E.A.O. dans un tel enseignement.

L'existence d'un certain consensus sur l'enseignement de l'algorithmique permet d'envisager la conception de didacticiels sur ce thème.

L'hétérogénéité du public visé conduit naturellement à chercher une réponse au problème par l'E.A.O. dont une caractéristique reconnue est l'adaptation au profil particulier de chaque étudiant (bases, rythmes,..).

Enfin, l'évolution rapide et constante de l'informatique conduit à une rénovation permanente des enseignements et impose aux enseignants d'intégrer le plus tôt possible les nouveaux développements de la discipline dans leur enseignement. Ceci pose de sérieux problèmes en raison du temps nécessairement limité, consacré à l'enseignement.

L'E.A.O. apparaît encore comme une solution qui doit permettre de décharger les enseignants de certains aspects de leurs cours au profit de nouveaux thèmes. Ceci est tout à fait clair dans le cas de la programmation lorsque l'on observe son évolution, que ce soit par la consécration actuelle de certaines formes de programmation (programmation fonctionnelle ou logique) ou par la multiplication d'environnements informatiques spécialisés (progiciels, SGBD, systèmes C.A.O.,..).

I.2 Enseignement assisté par ordinateur.

Nous voulons dans ce paragraphe situer notre travail sur la base d'une analyse des problèmes qui se posent en E.A.O. à l'heure actuelle.

Nous pourrons alors préciser le projet de recherche dans lequel s'insère notre reflexion, énumérer les objectifs de la thèse et fixer le contexte dans lequel elle a été réalisée.

L'examen de l'évolution de l'E.A.O. ([Sim-80], [IFI-75], [IFI-81], [Smi-82],...) nous conduit à souligner plusieurs aspects significatifs dans ce domaine interdisciplinaire :

Tout d'abord il est clair que le développement de l'E.A.O. suit directement l'évolution technologique en informatique.

A l'origine l'E.A.O. était surtout envisagée comme une solution à des problèmes pédagogiques propres à des minorités pour lequelles on acceptait (pour des raisons sociales ou politiques) des coûts relativement importants : retards scolaires de groupes socioculturels particuliers, formes d'enseignement destinées aux handicapés,...etc..

Cette problématique a été totalement transformée par l'avènement de la microinformatique, de la télématique et de l'informatique individuelle (cf.[BeF-82]).

L'E.A.O. devient maintenant un phénomène de masse soumis à l'impact des medias, dans une société où l'informatique d'une part, et les problèmes d'éducation d'autre part, ont une place prédominante. De plus, les enjeux culturels et économiques directement attachés à l'E.A.O. prennent une importance de plus en plus considérable : l'introduction de l'informatique dans le système éducatif français, par exemple, est largement amorcé; de nombreuses entreprises recherchent des solutions E.A.O. à leurs problèmes de formation; les sociétés de service en informatique développent leurs activités en E.A.O; les pouvoirs publics multiplient les ac-

tions d'initiation ou de normalisation en ce domaine (par exemple, les actions de l'A.D.I., les actions du centre mondial de l'informatique, ou des actions mises en oeuvre par des organismes internationaux tels que UNESCO ou I.B.I.).

Par ailleurs, L'E.A.O. est un domaine d'application de l'informatique très particulier dans la mesure où il est extrêmement difficile de modéliser les processus d'apprentissage, et où le métier d'enseignant est peu normalisé et très individualisé.

Le rejet, presque exagéré, par la communauté d'enseignants, des premières tentatives en E.A.O. est significatif à cet égard : il est dû notamment (outre les coûts prohibitifs de production) à l'extrême simplicité des modèles pédagogiques qui en étaient la base (cf. Skinner, Pressey, Crowder).

Le regain d'intérêt actuel pour l'E.A.O peut, en outre, s'expliquer par le fait que les développements actuels de l'informatique (intelligence artificielle, postes de travail ergonomiques) permettent d'espérer des progrès réels en matière de réflexion pédagogique.

Quoi qu'il en soit, l'E.A.O. apparaît aujourd'hui de manière sensible dans trois grands domaines : le système éducatif, la formation dans les entreprises, l'informatique grand public.

Ce qui caractérise l'E.A.O. dans le système éducatif, est un effort centralisé d'équipement des établissements, de formation des enseignants, de développement d'outils et de normalisation de ces outils : dans le contexte du plan 100 000 microordinateurs, création de centres académiques sur l'informatique et ses applications pédagogiques, constitution d'equipes académiques de formation; définition d'un plan national pour l'E.A.O.; projet DIANE (logiciels et normes).

L'objectif de cette informatisation est, à terme, de rendre effective l'intégration de l'informatique dans la pédagogie en s'appuyant sur une participation active des enseignants.

Le développement de l'E.A.O. dans la formation technique assurée au sein des entreprises est moins centralisé (tout en étant appuyé par les pouvoirs publics).
Il est le fait de très grandes entreprises en relation avec des sociétés de services spécialisées.
Les moyens informatiques mis en oeuvre peuvent être assez importants, mais les considérations d'ordre pédagogique sont soumises à des contraintes de coûts et de rentabilité immédiate.

L'apparition de l'E.A.O. dans l'informatique grand public suit les tendances inaugurées par les jeux électroniques. C'est le domaine où les contraintes d'ordre commercial sont les plus apparentes. Les didacticiels envisagés sont donc d'intérêt très large, portant sur des sujets ponctuels. Sur le plan pédagogique l'effort privilégie la forme par rapport au contenu.

On observe ainsi une grande diversité d'objectifs et d'approches dans l'industrie naissante de l'E.A.O. Le phénomène E.A.O. doit donc être analysé dans le contexte plus large de l'informatisation de la societé.

Tout d'abord on peut observer la pression exercée par l'informatique individuelle dans tous les secteurs de la vie culturelle, sociale et économique.

Dans le cas de l'éducation, on ne sait pas dire, à l'heure actuelle, quel peut être le rôle précis de ce type d'informatique, ni évaluer son influence sur les systèmes d'éducation.

Une orientation possible est l'intégration progressive de l'informatique dans les systèmes existants sans pour autant les modifier fondamentalement (c'est notamment la tendance des pouvoirs publics).

Une autre orientation est d'envisager à terme la transformation des systèmes éducatifs actuels sur la base d'un principe d'éducation individualisée grâce à l'ordinateur (cf.[Pap-83]).

A côté de ces incertitudes, on observe que la mise à la disposition de matériels informatiques de grande diffusion précède largement une réflexion suffisante sur les objectifs et les besoins pédagogiques, bien qu'elle ait été amorcée par des expériences importantes : PLATO [Har-81] (première expérience industrielle de vente de didacticiels par un grand constructeur de systèmes informatiques); LOGO [Pap-83] (approche de l'E.A.O. fondée sur le principe que l'apprentissage découle d'une attitude active de l'enfant renforcée par la programmation dans un environnement adapté); LSE puis DIANE [ADI-83] (systèmes auteurs destinés à permettre une participation active des enseignants cohérente avec des soucis de normalisation).

Enfin sur le plan technologique, les moyens existants peuvent répondre aux besoins pédagogiques déjà reconnus; mais le développement industriel de l'E.A.O. reste très ralenti pour deux raisons majeures :

I - l'insuffisance de la réflexion sur les relations entre les fonctions pédagogiques et ces moyens.

I - l'importance des coûts des moyens technologiques

Examinons quelques tendances actuelles dans la façon de travailler en E.A.O..

Une première approche est de proposer des langages spécialisés (langages - auteurs) aux enseignants en leur confiant le soin d'intégrer dans leur pédagogie les didacticiels qu'ils auront conçus (LSE, COURSEWRITER, SUPERPILOT, EGO, ARLEQUIN, ...).

Une deuxième approche est celle de l'"enfant programmeur". Le langage BASIC, par exemple, a été conçu à l'origine dans un contexte d'enseignement. Il en est de même pour LOGO [Pap-80], [Pap-83]. (L'utilisation de ces deux langages s'est ensuite étendue à des contextes autres que celui de l'enseignement).

Une troisième approche consiste à proposer aux enseignants et aux élèves des produits finis diffusés par des maisons d'édition ou des sociétés de service spécialisées.

Une quatrième orientation propose une démarche générale qui intègre les diverses aspects de la conception, de la production, de l'exploitation et de l'évaluation des didacticiels.
Le souci primordial dans cette approche est de tenir compte des compétences spécifiques des divers intervenants selon une typologie précise (concepteur, auteur, enseignant utilisateur, apprenant, informaticien) en fournissant à chacun les moyens les plus adaptés à leur fonction. Ceci se traduit par la définition de systèmes d'E.A.O. qui intègrent ces moyens et qui sont accompagnés de méthodes appropiées. On s'approche ainsi d'une véritable notion d'atelier de production de didacticiel.

Le système MOSAIQUE a été conçu dans cet esprit au sein du laboratoire de génie informatique de l'IMAG (avec le support de l'ADI) autour d'un projet de production d'un didacticiel de taille importante (soutien à un enseignement de français langue étrangère, cf.[Ada-83], [Pai-81], [Sco-83]).

Le système DIANE est un autre exemple de cette approche : partant d'une synthèse de fonctionnalités E.A.O. repérées par l'expérience d'un grand nombre d'auteurs (cf. [ADI-81]), l'A.D.I. en a organisé et financé la production avec un souci de normalisation et un objectif de grande diffusion à un niveau national.

Quelle que soit l'approche utilisée, on constate l'apport de techniques informatiques et pédagogiques dans le développement de l'E.A.O.

Sur le plan informatique on adapte directement des formes usuelles d'utilisation de l'informatique (techniques d'interactivité, simulation, bases d'informations) en intégrant les medias les plus divers (textes, graphiques, images, sons) et les moyens de communication (réseaux).

Pour développer systèmes et didacticiels on s'appuie sur les techniques de base du génie logiciel.

Sur le plan pédagogique, on s'efforce d'adapter les techniques pédagogiques actuelles (pédagogie active, pédagogie par objectifs,..) pour concevoir, spécifier et développer les produits.

Pour maîtriser la relation entre informatique et pédagogie, la tendance actuelle est d'utiliser des démarches analogues à celle de l'intelligence artificielle.

On peut ainsi conjecturer que l'E.A.O. peut devenir une discipline à part entière fondée sur des méthodes issues de l'informatique et de la pédagogie. Cette discipline devra tenir compte des besoins d'industrialisation des produits E.A.O. et devra fournir les réponses aux problèmes ainsi posés.

I.3 Conclusion.

Le travail que nous présentons ici est rattaché à un projet initié à Carry le Rouet en 1982 pour développer à l'echelle nationale de nombreux didacticiels pour l'enseignement de l'informatique.

Ce projet est patronné par le Ministère de l'Education Nationale et par l'Agence de l'Informatique. Les produits devront être développés en utilisant le système DIANE (qui n'est devenu opérationnel qu'à l'été 1984).

Nous nous sommes appuyés sur une réflexion pédagogique sur l'enseignement de la programmation, menée à l'Université de Grenoble depuis une dizaine d'années [LuS-75], [Sch-79], [PeS-82], [LPS-83], et nous avons repris la démarche E.A.O. conçue et expérimentée dans le projet MOSAIQUE.

Le projet de l'équipe est de produire un didacticiel de grande taille conçu comme une ressource supplémentaire à un enseignement de la programmation.

La réalisation que nous présentons ici, en est une étape préliminaire. Son objectif est d'explorer des possibilités pédagogiques nouvelles apportées par l'utilisation de l'ordinateur dans un tel enseignement. Il s'agit donc d'une maquette du futur didacticiel.

Pour la concevoir, nous avons sélectionné quelques sujets en algorithmique (notion d'algorithme et d'analyse descendante).

Pour la réaliser nous avons choisi d'utiliser un système existant issu du projet MOSAIQUE (système VOYELLES [Lcc-83]). Ce système a dû être adapté aux besoins spécifiques de notre projet et nous avons dû réaliser quelques outils supplémentaires.

L'ensemble de ces réalisations est décrit dans les deux chapitres suivants.

CHAPITRE II : Spécification pédagogique de la maquette.

II Spécification pédagogique de la maquette.

La maquette réalisée est un soutien à l'action de l'enseignant autant qu'à l'étudiant. Ce soutien doit permettre à l'élève entre autres :

a) d'acquérir des connaissances de façon individuelle.

b) de travailler le temps qu'il veut pour dissiper les doutes qui ont pu lui rester du cours correspondant.

c) de se concentrer sur un sujet.

Quant à l'enseignant, elle doit lui permettre de présenter ce qui est difficile à montrer dans un cours traditionnel.

L'élaboration de la maquette s'appuie sur le choix des éléments suivants :

- les sujets traités dans le didacticiel.
- Les moments pédagogiques de l'apprentissage de ces sujets.
- Les approches pédagogiques utilisées pour enseigner ces sujets.

II.1 Sujets abordés dans la maquette.

La maquette de didacticiel présentée ici a été conçue comme un soutien à une initiation à la programmation.

Le travail de l'étudiant avec ce didacticiel doit ainsi compléter ce qui est developpé par l'enseignant en classe.

Deux thèmes ont été retenus comme se prêtant particulièrement à une expérience d'E.A.O. : La notion d'algorithme et la notion d'analyse descendante.

Chaque thème est constitué de plusieurs sujets dont la liste sera décrite dans le paragraphe III.3 ("Structuration de la maquette").

II.1.1 Notion d'algorithme.

La compréhension de la notion d'algorithme par l'élève n'est pas évidente. D'une part, elle concerne plusieurs éléments (d'ordre différent), d'autre part, il est difficile de montrer la séparation de ces éléments; et finalement, chacun de ces éléments met en rapport des processus et/ou des mécanismes abstraits difficiles à montrer concrètement à l'élève.

En outre tous ces aspects interviennent dès que l'on travaille sur un algorithme qui est en effet la représentation statique de différents processus dynamiques.

La notion d'algorithme est mieux assimilée par les étudiants, s'ils peuvent discerner clairement :

- les techniques de description des algorithmes qu'ils découvrent en même temps que la notion même d'algorithme.

- le travail d'analyse que résume l'algorithme :
  compréhension des relations entre le problème posé et l'algorithme. C'est l'étude du travail du programmeur.

- le travail d'exécution exprimé par l'algorithme :
  compréhension de la relation entre l'algorithme et l'obtention du résultat. C'est l'étude du travail de la machine.

II.1.2. Notion d'analyse descendante

Dès le début de l'initiation, l'étudiant doit être confronté à l'analyse descendante des problèmes pour la construction d'un programme.

Avant d'être lui-même en situation pour pratiquer ce genre de méthode, il paraît bon de lui présenter, par des exemples, des documents d'analyse décrivant les diverses étapes conduisant d'un énoncé à une solution.

Nous allons considérer un document d'analyse comme la description d'un algorithme dont les composants doivent montrer des différents niveaux d'abstraction de la démarche suivie pour arriver au programme correspondant.

Un objectif important dans une telle présentation est que l'étudiant cerne bien ce que l'on entend par "niveau d'abstraction" et qu'il voit les divers modes d'expression utilisés (français, formules mathématiques ou logiques, notation algorithmique, etc.).

L'analyse permet pour chaque niveau d'abstraction de construire un document qui décrit un ensemble d'opérations spécifiques sur des données abstraites.

Cet ensemble d'opérations est soumis à des décompositions au prochain niveau ("plus bas") d'abstraction, et le processus se poursuit jusqu'à ce qu'on arrive à un niveau où tout est exprimé moyennant des primitives élémentaires de la notation employée.

Nous décrivons maintenant le modèle d'algorithmes qui a été utilisé, ainsi que les types de problèmes sur lesquels nous avons travaillé pendant l'expérience.

II.1.3 Modèles d'algorithmes servant de support à l'exposition des sujets choisis.

Les algorithmes proposés dans la maquette illustrent essentiellement la composition conditionnelle et la composition itérative pour les deux types d'exercices présentés.

Deux méthodes d'analyse sont proposées à l'étudiant pour les compositions précédentes, tendant à résoudre la difficulté de compréhension des sujets : l'analyse par cas et le traitement séquentiel.

L'analyse par cas

L'exécution d'actions dans un processus dépend très souvent de la valeur de certaines variables déterminant l'état courant du processus.

L'analyse par cas aide à construire une solution en analysant chacun des cas possibles déterminés pour le problème à résoudre, et en associant à chaque cas une action qui exprime la solution.

Cet outil d'analyse, impose un rigueur systématique qui oblige à :

- préciser la liste de cas possibles et les exprimer.

- préciser les actions qui résolvent le problème.

- établir une exclusion mutuelle entre les cas définis.

L'analyse par cas definit donc, une partition du domaine des données où on fait correspondre les cas et les actions afin de résoudre le problème.

L'analyse peut être faite de deux manières:

- On énumère les différents cas possibles d'après le domaine des données et à chaque cas, on associe une action.

- On sépare les actions appropiées pour résoudre le problème et on associe à chacune de ces actions la condition nécessaire et suffisante pour que l'action soit solution du problème.

Pour l'analyse par cas nous nous appuyons sur les compositions sélective et conditionnelle [LPS-83], qui font préciser l'ensemble de cas possibles et les actions correspondantes; elle est illustrée dans notre maquette à travers deux problèmes dont les énoncés sont les suivants :

. "Maximum de deux valeurs"
Déterminer la plus grande de deux valeurs données. (présentation des notations)

. "Maximum de trois valeurs"
Déterminer la plus grande de trois valeurs données, (on suppose que les trois valeurs sont différentes).
(présentation de l'analyse par cas)

Le traitement séquentiel

Cette méthode (cf.[Sch-79], [LPS-83]) utilise un modèle abstrait de "machine séquentielle" qui permet de décrire les schémas de traitement séquentiel en faisant abstraction de la représentation de la séquence donnée.

Le traitement séquentiel de l'information fait intervenir une machine séquentielle d'accès, et une machine séquentielle de traitement.

Le modèle nous permet de faire la manipulation des algorithmes de la façon la plus simple pour illustrer les idées les plus importantes des sujets concernés.

Une machine d'accès séquentiel permet d'accéder séquentiellement aux éléments d'une suite portés par un ruban.

La machine vue de l'extérieur présente :

- une fenêtre appelé "CARACTERE COURANT" (en abrégé CC)

- deux boutons appelés "DEMARRER" et "AVANCER" (en abrégé DEM et AV)

L'état de la machine est composé :

. d'un indicateur de file vide
. d'un élément courant
. d'un indicateur de fin de file

Le répertoire de primitives de la machine d'accès [PeS-82], [LPS-83] est composé des actions de modifications et de consultations suivantes :

- Modification

DEMARRER (DEM): Donne accès au premier élément de la file. Ce premier élément est accesible par la fenêtre.

AVANCER (AV) : Donne accès au prochain élément de la file. Cet élément devient accessible par la fenêtre.

- Consultation

ELEMENT COURANT (EC): Donne accès à l'élément present dans la fenêtre. (Il n'y a pas de modification d'état)

FIN de FILE (FDF) : Indique si la fin de la file est atteinte .

FILE VIDE (F_VIDE) : Indique si la file est vide.

Le répertoire de la machine de traitement est composé des actions suivantes :

VISITER :

C'est l'action élémentaire appliquée à tous les éléments de la file.

INITIALISATION DES VISITES (INIT_V) :

Action appliquée en prologue du traitement séquentiel.

TERMINAISON DES VISITES (TERM_V) :

Action appliquée en épilogue du traitement séquentiel.

TRAITEMENT FILE VIDE (T_VIDE) :

Traitement à suivre pour le cas où la suite est vide.

Un cas particulier est la machine compteur (cf.III.3.2.1) qui représente un compteur dont la valeur est montrée dans la fenêtre. Elle présente le répertoire d'actions de modification suivantes :

MISE A ZERO (MAZ) : initialise à zéro le compteur.

AJOUTER (AJ) : augmente de "1" la valeur du compteur.

La combinaison des répertoires de ces machines, d'après les caractéristiques de la suite de l'information traitée, nous a permis de définir un ensemble de schémas (cf.[Mon-81], [PeS-82], [Sch-79], [Zam-81]). Ces schémas s'appuient sur les deux formes de compositions itératives (répéter et tantque).

Nous décrivons le schéma qui a été utilisé dans la maquette, qui permet le parcours d'une file munie d'une marque caractérisant la fin de la file (cf.annexe I). La file est vide si elle ne comporte que la marque.

* Schéma de traitement séquentiel

```
DEM
choix
  FDF : T-VIDE
 ¬FDF :
        INIT-V
        répéter
           VISITER (EC)
            AV
        jusqua FDF
        TERM-V
fchoix
```

Tous les schémas sont exprimés à l'aide de la notation algorithmique MEFIA [Mor-79], [Sch-79]. Nous décrivons en annexe I les éléments de la notation utilisés dans ce travail .

L'étude des algorithmes de traitement séquentiel des informations conduit à maîtriser le mécanisme d'itération. Nous donnons ici, comme exemples, quelques énoncés qui conduisent à l'assimilation progressive de l'itération (cf.[LuS-75], [PeS-82], [LPS-83]).

- "Compter les 'A'" :
  Le ruban d'une machine caractère porte un texte non vide terminé par un '.'. On veut compter le nombre d'occurences du caractère 'A'.
  (présentation de la notation algorithmique)

- "Fréquence d'apparition du caractère 'A'" :
  Le ruban de la machine caractère porte un texte non vide terminé par un '.'. On veut calculer le pourcentage du nombre de 'A' par rapport au nombre total de caractères différents de l'espace.
  (présentation du schéma de traitement séquential)

- "Compter les 'LE'" :
  On veut compter le nombre d'occurences du couple 'LE' dans un texte terminé par un point.
  (présentation de la méthode d'analyse)

II.2 Les moments pédagogiques.

Il est important de distinguer les différents moments pédagogiques dans le développement d'un sujet [ADI-81]. Ceci est clairement apparu dans l'analyse pédagogique du didacticiel de français langue étrangère support du projet MOSAIQUE (cf.[Sco-83], AbS-84]). Nous nous appuyons de manière classique, sur trois moments pédagogiques, qui correspondent à des objectifs différents mais dont la mise en pratique fait appel à des moyens communs. Ces moments sont : la sensibilisation, la présentation, et la consolidation.

Dans la progression pédagogique, chaque moment est consacré à un aspect différent et présente aussi différents niveaux de difficultés.

L'enseignant peut faire travailler l'élève à travers une progression, soit en lui suggérant le travail à faire, soit en le lui imposant. Néanmoins, dans la maquette, l'élève peut avoir une certaine liberté pour travailler sur des exercices concernant un sujet particulier, au niveau choisi par lui (et par l'enseignant) sans qu'il ait besoin de suivre toute la progression.

L'étudiant se trouve dans un environnement d'apprentissage où il réalise un travail guidé mais sans être soumis à la pression d'un cours traditionnel, due entre autres aux points suivants :

-L'enseignant doit suivre un horaire pour achever son programme.

-L'étudiant suivre le rythme imposé par l'enseignant.

L'intérêt d'un tel environnement devient important dans la mesure où l'étudiant peut travailler à son propre rythme pour assimiler et fixer ses connaissances.

L'étudiant peut donc travailler de manière indépendante sur ce qui lui est offert à chaque moment, en confrontant les difficultés, en temps voulu, selon le niveau du sujet choisi.

II.2.1 Sensibilisation au contexte de travail

Dans ce moment pédagogique, on introduit la catégorie des problèmes sur lesquels on désire travailler avec l'étudiant. On s'efforce de s'appuyer sur les acquis de l'élève.

Au cours de la sensibilisation, on cherche à créer un intérêt pour le thème, tout en attirant l'attention de l'apprenant sur les idées de base qu'il devra maîtriser.

Pour chacun des deux sujets qui ont été choisis, la sensibilisation est assurée à deux niveaux différents. On distingue les activités qui sont faites en séance avec l'enseignant, et celles qui font l'objet de l'E.A.O.

a) Notion d'algorithme.

La sensibilisation à la notion d'algorithme consiste à faire participer l'étudiant à l'élaboration d'une suite d'actions décrivant un processus, soit déjà connu par lui, soit déduit par lui à partir de l'énoncé d'un problème (un jeu, par exemple).
On confronte l'étudiant à des situations qui réclament de lui une réaction en fonction des notions sur lesquelles on travaille.

Un exemple fréquemment utilisé, et dû à Dijkstra [Dij-71], est le suivant :

> "Ecrire un algorithme permettant de décrire comment éplucher un nombre suffisant de pommes de terre qui se trouvent dans un panier. Il faut tenir compte du fait que le panier peut être vide à un moment donné."

En tenant compte du degré de précision de l'énoncé, il faut écrire un texte décrivant la succession d'actions. Cela implique l'utilisation d'une notation et d'un répertoire bien précis d'actions élémentaires.

Le texte produit doit être exécutable : un "exécutant réel" doit être capable de l'interpréter et l'exécuter, sans ambiguité ni imprécision.

Les objectifs de la sensibilisation sont en rapport avec:

a) La notion d'action :

Comment interpréter un problème en termes d'une suite d'actions.
Quelle est la spécification de chaque action.
(Toute action doit être réalisable a priori)

b) La notation algorithmique :

Comment exprimer chaque action décrivant le processus de manière précise et claire, et comment décrire les objets manipulés pour chaque action.

c) La notion d'exécution :

Quel est l'effet produit pour chaque action et pour l'ensemble complet d'actions.

La sensibilisation, selon qu'elle est faite en classe ou à travers l'E.A.O., mène à différentes activités.

En classe, on fait participer l'étudiant à la construction d'un algorithme - solution d'un problème posé appartenant à la catégorie de problèmes à laquelle on s'intéresse.
Cela consiste, en général, en une intéraction orale classique (élève - enseignant) à partir des questions concernant les aspects décrits ci-dessus.

Dans notre maquette, nous bénéficions de la sensibilisation faite en classe, et nous utilisons l'E.A.O. pour mettre l'étudiant en rapport, de façon plus concrète, avec la catégorie de problèmes.

L'étudiant, grâce à un poste de travail qui lui est offert peut manipuler un ensemble de touches fonctions, représentant les actions qu'on veut lui faire comprendre et en même temps il peut observer l'effet produit par chaque action ainsi que l'ordre d'exécution correspondant.

On met en évidence le principe de conception d'algorithmes.

L'activité de l'étudiant pendant une session de travail est montrée par l'affichage d'une trace.
Cette trace a deux rôles selon qu'elle est analysée par l'auteur-enseignant, ou par l'étudiant.

Pour l'auteur, elle lui permettra de suivre la progression de l'élève depuis le début de la session, et d'évaluer son activité face au problème posé.

Pour l'élève, elle lui permettra de faire l'analyse du processus, et de reconnaître les rapports entre le problème, le répertoire d'actions, la composition des actions et la structuration d'un algorithme.

b) Notion d'analyse descendante.

La sensibilisation à l'analyse descendante a pour but de mettre l'élève en rapport avec la concrétisation d'un (ou plusieurs) niveau(x) d'abstraction présent(s) dans un algorithme.
Cette concrétisation montrera, en particulier, le passage d'un niveau courant à un niveau plus bas et aussi la description des actions traduisant ce changement.

On cherche à induire chez l'étudiant cette démarche pour les différents niveaux d'abstraction d'un algorithme.

Dans la classe, la sensibilisation confronte l'étudiant à un problème (un énoncé, ou une solution algorithmique d'un problème exprimé à un certain niveau) pour qu'il participe à la construction d'une solution qui montre un niveau de description compréhensible par la machine.

Les objectifs de la sensibilisation en classe sont inhérents à :

- la description d'un niveau d'abstraction
  (passage de "que faire" à "comment le faire")

- l'application de la même démarche à tous les niveaux.

- l'intégration des compositions algorithmiques dans
  l'expression d'un niveau de l'analyse.

Du point de vue E.A.O., la sensibilisation a les mêmes objectifs qu'en classe, mais utilise des moyens différents.
De façon plus concrète on confronte l'étudiant aux différents niveaux surgissant d'une analyse qui mène à une solution.
Ces niveaux sont représentés, dans la maquette, par des visualisations à l'aide d'encadrements dans l'algorithme correspondant affiché sur une partie de l'écran et la description de l'ensemble d'actions traduisant le cadre dans une autre partie de l'écran.

II.2.2 Présentation de connaissances.

Il s'agit de présenter à l'étudiant les éléments fondamentaux du sujet abordé et leurs relations. Dans notre contexte on présente à l'élève des modèles informatiques pour résoudre les problèmes auxquels il a été sensibilisé.

Pour les thèmes que nous avons retenus, il est important de montrer les aspects suivants:

Pour la notion d'algorithme :

- La notation algorithmique utilisée.
- L'analyse qui fait la relation entre l'énoncé et l'algorithme.
- L'exécution de l'algorithme qui fait la relation entre l'algorithme et la machine.

Pour l'analyse descendante :

- L'étude de la structure de l'algorithme
- L'étude des composants de l'algorithme

à divers niveaux d'abstraction.

La présentation d'un sujet concerne aussi un ensemble d' activités qui peut varier selon qu'elle est developpée en séance par l'enseignant, ou qu'elle l'est dans une activité d'E.A.O.

Comme on a déjà dit (cf.II.1.1), il est difficile de faire comprendre à l'élève les différents aspects liés à la notion d'algorithme. D'une part, elle montre dans un seul texte beaucoup d'aspects concernant les relations entre l'algorithme et son expression, l'algorithme et l'analyse d'un problème et l'algorithme et le processus d'exécution qu'il représente; et d'autre part, on ne dispose pas, en général, de moyens pour présenter en classe ces aspects de manière convaincante.

Par exemple, l'étudiant a beaucoup de difficulté à comprendre la relation existant entre un algorithme (représentation statique) et son exécution (processus dynamique). Une autre difficulté se présente pour l'élève avec la notion d'état et ses changements (un algorithme change la valeur des variables). Bien qu'on comprenne comment un algorithme manipule des données particulières pour produire un résultat, la compréhension de ce qui se passe lorsqu'on généralise aux autres données n'est pas toujours évidente pour l'étudiant.

L'environnement d'E.A.O. est utile car il permet à l'étudiant de mettre en évidence, à travers ses activités, les rélations existant entre l'algorithme proprement dit du problème que l'on veut résoudre et l'exécution du processus concerné.

Quant à l'analyse descendante, on essaie de visualiser la structure d'un algorithme.

On veut montrer à l'étudiant que la structure d'un algorithme peut être considérée sous deux angles complementaires :

- Comprendre comment l'algorithme a été construit

- Comprendre comment l'algorithme pourra être exécuté

On veut que l'étudiant connaisse les divers éléments qui peuvent composer un algorithme (notamment en relation avec la notation algorithmique).

En résumé, le présentation d'un algorithme vise à développer chez l'étudiant les aptitudes suivantes :

- Capacité de reconnaître les composants d'un algorithme donné.

- Capacité de reconnaître les relations entre ces composants (niveau d'abstraction, lien entre structure de programme et structure de données).

II.2.3 Consolidation des connaissances

En ce moment pédagogique, on cherche à :

a) Fixer les connaissances présentées.

b) Fixer les relations avec d'autres connaissances déjà acquises.

c) Offrir un environnement d'expérimentation, pour évaluer les acquis.

On distingue deux buts précis : d'une part consolider les connaissances qui ont été présentées à l'élève et d'autre part vérifier s'il les a comprises.

En classe, pour fixer les connaissances on se sert, en général, d'une série d'exercices à résoudre englobant le même genre de difficultés dont la solution a été presentée à l'étudiant [PeS-82].
Pour vérifier que l'étudiant a bien compris, les moyens utilisés sont, entre autres :

- Des questionnaires sur le sujet à consolider.

- Des exercices où on a glissé intentionnellement des erreurs dans le but de les faire découvrir.

- Des exercices d'application sur un problème type.

- Des exercices de modification.
  On donne à l'élève une solution juste et on lui demande de la modifier pour en donner une autre.

En E.A.O., pour la consolidation, on dispose d'autres moyens :

- L'individualisation du travail dans ce moment est un facteur particulièrement important.
  L'élève a besoin de réagir devant ses propres erreurs à son propre rythme.

- le contrôle de l'activité de l'élève.
  Dans notre maquette on se sert de moyens simples :

  * QCM ou tests sur un sujet fait à un point déterminé du dialogue (à la manière d'un résumé).

  * Questions-réponses tout au long du déroulement du dialogue pour évaluer des aspects importants (de façon à orienter la compréhension).

- A travers certaines fonctionnalités de base, notamment l'affichage avec des attributs d'aspect, on met en évidence des points qui ont été présentés et qu'on veut consolider chez l'étudiant (Exemple : la composition d'actions dans un algorithme, des niveaux d'abstraction, etc.).
  Ces attributs d'aspects sont utilisés aussi pour le renforcement des réponses.

II.2.4 Typologie d'erreurs

La sollicitation de réponses de l'élève laisse toujours ouverte la possibilité de lui faire commettre des erreurs au moment de fournir une réponse.

Nous proposons ici une typologie d'erreur pour les différents moments pédagogiques.

Dans la maquette, les erreurs propres à la sensibilisation sont en rapport avec l'activité de l'élève lorsqu'il manipule la machines caractères et la machine compteur.

On distingue trois types :

Type 1 : Touche - fonction mal employée.
Le système attend une réponse particulière et réceptionne une réponse dont la valeur est juste (vis-à-vis des valeurs possibles) mais n'est pas la réponse attendue.

Type 2 : Tout emploi des touches est interdit.
L'élève a atteint la fin de la file (caractère terminal affiché dans la fenêtre) et appuie sur une touche quelconque.

Type 3 : Touche incorrecte.
L'élève frappe une réponse (touche - fonction) fausse involontairement.

Quant aux erreurs propres à la consolidation, on distingue quatre types :

Type 4 : Réponse non numérique.
Le système attend une réponse numérique.

Type 5 : Réponse en dehors des valeurs possibles.
Le système attend une réponse numérique particulière et en receptionne une numérique mais en dehors de l'ensemble des valeurs possibles.

Type 6 : Réponse déjà donnée.
Le système attend une réponse parmi un ensemble possible et l'étudiant en propose une déjà donnée (pas attendue par le système).

Type 7 : Réponse incorrecte.
L'élève frappe une mauvaise réponse.

II.3 Les approches pédagogiques.

Les approches pédagogiques permettent aborder les sujets de manière particulière (cf.[APS-81]). Pour enseigner les divers sujets nous avons travaillé ici sur la base de deux approches pédagogiques : la simulation et la lecture d'un document d'analyse.

Des travaux dans le domaine de l'apprentissage ont montré combien il était important d'intégrer l'étudiant dans son propre processus d'apprentissage. C'est par la participation active de l'élève qu'on a le plus de chance de réussir cette intégration.

Nous faisons participer activement l'étudiant, à travers les différentes approches utilisées auxquelles nous faisons correspondre des outils E.A.O. particuliers.

Pour la simulation nous disposons de la machine caractères et de la machine compteur.

Pour la lecture nous disposons d'un mécanisme qui nous permet d'avoir accès aux différents niveaux d'un algorithme. Chaque niveau, montre des composants selon un mode de lecture qui a été préalablement défini.

II.3.1 Lecture d'un document d'analyse.

Un document d'analyse réunit plusieurs types de textes qui décrivent la structure de l'algorithme, la structure des données manipulées, les éléments d'information propres aux invariants du processus, etc..

La lecture par l'élève d'un tel document consiste, d'une part parcourir sa structure et d'autre part, déchiffrer chaque composant de cette structure. Elle permet aussi de percevoir les rapports existant entre les différents composants d'un algorithme, ainsi que les différents niveaux d'abstraction qui y sont présents, selon les

objectifs de l'enseignant. Cela implique différentes manières de lire le document.

La lecture est basée sur l'identification des composants et leur interprétation.

On conçoit donc un enseignement à travers cette approche en considérant concrètement :

- la notion d'algorithme et certains aspects tels que :

  . Syntaxe et sémantique de la notation algorithmique.

  . Différents types de compositions d'actions.

  . Les structures des données.

- l'analyse et la construction d'algorithmes qui met en oeuvre des méthodes d'analyses précises.

- les langages de programmation qui mettent en oeuvre des constructions différentes permettant d'exprimer les algorithmes.

- la vérification d'algorithmes qui met en oeuvre les assertions propres à chaque action représentée dans l'algorithme.
  Une telle lecture devra manipuler essentiellement deux types de langages (informatique et logique) pour mettre en évidence, d'une part la correction de chaque composant de l'algorithme, et d'autre part la correction de l'algorithme complet, d'après les rapports existant entre ces composants (et éventuellement les données manipulées).

- la documentation des algorithmes qui met en oeuvre des commen-

taires précis sur chacune des actions représentées dans l'algorithme.
Cette lecture manipule aussi, deux types de langage (naturel et informatique) pour mettre en évidence la signification des composants et les actions correspondantes.

Dans notre maquette notre but, à travers la lecture d'un document d'analyse est que l'étudiant puisse faire correspondre à chaque partie de l'analyse faite, une idée claire et précise de la construction de l'algorithme correspondant et de sa signification.

L'activité de l'élève est centrée sur la structure de l'algorithme. Ceci le conduit à manipuler les actions représentées, ce qui peut impliquer une composition et/ou décomposition d'actions pour arriver à leur compréhension.

Cette approche est particulièrement bien adaptée aux moments de présentation et de consolidation.
On s'appuie sur un mécanisme qui permet d'accéder aux divers composants d'un algorithme. Nous en donnerons la spécification plus loin (III.3.1).

Nous distingons trois types de lecture selon qu'on veut travailler sur la construction d'un algorithme, l'identification des composants dans l'algorithme ou la sémantique de celui-ci.

Type 1 : On présente les divers pas conduisant de l'énoncé d'un problème à un algorithme. L'étudiant découvre ainsi l'analyse qui est faite et il certifie les moyens d'expression utilisés.

Type 2 : On parcourt la structure de l'algorithme.
L'étudiant identifie les différents composants et les rattache à l'énoncé.

Type 3 : A partir d'un algorithme on essaie de découvrir l'enchaînement d'actions qu'il représente. Une manipulation d'assertions est prévue.

On peut considérer aussi plusieurs modes de lecture associés à ces types. Ces modes de lecture dépendent des composants auxquels on accède lors de la lecture d'un algorithme. On peut s'intéresser à lire un algorithme (selon les types décrits ci-dessus) en analysant les composants:

- du point de vue syntaxique
- en référence à un schéma précis présent dans l'algorithme.

Nous parlons donc de mode syntaxique, et mode schématique; et un algorithme pourra être lu, par exemple, selon le mode schématique du type 1 pour montrer les divers pas de l'analyse faite pour construire une solution en utilisant un des schémas possibles de traitement séquentiel de l'information.
Les trois types de lecture permettent de diversifier les dialogues selon que l'on veut:

- montrer l'analyse d'un algorithme
- faire identifier les composants d'un algorithme
- faire découvrir la sémantique des composants d'un algorithme.

Cette diversification des dialogues est importante car elle induit chez l'étudiant une façon systématique de lire un

algorithme, en considérant seulement les éléments concernés propres à chaque type de lecture.

On distingue ainsi, par exemple, que le dialogue pour faire découvrir la sémantique des composants d'un algorithme, met en oeuvre :

- une décomposition de ces composants qui montre soit les éléments essentiels d'information, soit les rapports entre ces éléments.

- le processus concerné et

- le rapport de ce composant avec une organisation de l'information.

D'autre part, le dialogue pour montrer la construction d'un algorithme peut nous amener, par exemple, à :

- considérer les informations élémentaires manipulées

- établir des actions précises sur chaque information élémentaire.

- décrire chacune des actions en utilisant la notation.

- composer ces actions.

II.3.2 La simulation.

Cette approche met à la disposition de l'élève certains outils qu'il pourra utiliser et qui lui permettront de tirer ses propes enseignements.

Ces outils correspondent à des modèles simulés sur l'ordinateur qui sont directement en rapport avec une catégorie particulière de problèmes. Ainsi, par exemple, un modèle simulé d'une balance permettra-il à l'élève, de déduire de sa propre activité la démarche à suivre pour des problèmes mettant en évidence des relations d'ordre.

Dans notre maquette nous offrons concrètement à l'élève un modèle de la machine caractère et de la machine compteur, lui permettant de découvrir à travers son activité le principe du traitement séquentiel.

Cette approche est particulièrement bien adaptée au moment pédagogique de sensibilisation.

II.4 Conclusions.

On a montré comment aborder l'analyse pédagogique de manière systématique.

Bien que l'on se sait appuyé sur les moments pédagogiques simples, et qu'on n'aie exploré que deux approches pédagogiques, on a mis en évidence la complexité d'une telle analyse.
Cette analyse a, par ailleurs, permis de montrer des exemples d'apports spécifiques de l'E.A.O. dans une initiation à l'algorithmique.

La simulation est tout à fait compatible avec une pédagogie active. Dans notre contexte, elle permet de répondre aux besoins ressentis par les étudiants, de concrètiser les processus dynamiques mis en jeu par un algorithme, tout en controlant leur activité à l'aide de modèles de machines cohérents avec le souci méthodologique caractérisant l'initiation à la programmation.

Dans un didacticiel en vrai grandeur, cette approche peut être utilisée systématiquement tout au long de l'introduction aux notions qui font appel à des mécanismes dynamiques : modèles de machines illustrant les divers types de contrôle de l'exécution, puis les représentation des structures de données de base.
Ce dernier point a été étudié, par ailleurs, dans une maquette réalise par H.MONTE [Mon-84].

La lecture des algorithmes est un élément essentiel dans la maîtrise de la programmation.
Elle est toute fois difficilement explicitée dans les approches traditionnelles et ceci par des raisons materielles évidentes.

L'auteur d'un livre ou l'enseignant en cours, peut difficilement exploiter cette activité dans toutes les dimensions que nous avons soulignées, autrement qu'à l'occasion d'un ou deux exemples.
Ce qu'il en reste à l'élève est assez diffus tant qu'il ne le pratique pas suffisament. L'E.A.O. est à ce titre un outil particulièrement adapté.

Comme la simulation, l'approche lecture pourra être utilisée systématiquement dans un didacticiel en vraie grandeur.
Les expériences préalables de H.MONTE [Mon-84] laissent penser que son intérêt est très grand pour aborder la incompréhension des représentations de données en relation avec les types abstraits correspondants d'une part, et les modèles de représentation de l'information sur les ordinateurs d'autre part.

CHAPITRE III : Les solutions E.A.O.

III Les solutions E.A.O.

Après avoir considéré les sujets que doit traiter la maquette, la caractérisation des moments pédagogiques et les approches pédagogiques pour aborder ces sujets, nous étudions, maintenant, comment utiliser ces éléments dans la construction d'une solution E.A.O.

L'élaboration de cette solution dépend du choix :

- D'une stratégie pédagogique.

- Des différents moyens E.A.O. mis en oeuvre.

Nous cherchons à partir d'une stratégie pédagogique préalablement étudiée pour aborder les sujets, à mettre en oeuvre des moyens E.A.O. qui, d'une part, traduisent la stratégie, et dont l'utilisation, d'autre part, ne présente pas de difficultés supplémentaires pour les étudiants.

III.1 Stratégie pour la définition d'une solution E.A.O.

Elle est basée sur les points suivants:

a) La présence de l'enseignant.
   On choisit l'E.A.O. comme un complément à l'activité de l'enseignant en classe.

b) La progression dans l'enseignement, caractérisée par les trois moments pédagogiques.

c) La liberté donnée à l'élève.
   En classe il est important de suivre une progression dans l'enseignement . L'E.A.O. permet de bénéficier des intérêts d'une pédagogie active.
   La liberté offerte par le didacticiel permet à l'élève de choisir son travail. Même s'il est préférable de suivre une progression, aucune contrainte ne lui est imposée.

d) Les approches pédagogiques qui particularisent l'apport de l'E.A.O.

Pour mettre en oeuvre cette stratégie, nous avons suivi la méthode de production du didacticiel issue du projet projet MOSAIQUE (cf.[Ada-83]) et nous avons utilisé les fonctions mises à notre disposition dans ce système. Ce dernier ayant été construit uniquement à partir des besoins révélés par l'analyse pédagogique d'un didacticiel de français langue étrangère (pour des raisons exposées au chapitre I), nous n'avons utilisé qu'une partie de ces fonctions, et par contre nous avons du réaliser les fonctions spécifiques à notre maquette.

III.2 Le choix des différents moyens E.A.O. mis en oeuvre

Dans notre maquette, nous utilisons un poste de travail peu sophistiqué. L'intérêt d'une démarche systématique en est renforcé.
Le choix de moyens E.A.O. s'appuie sur l'ensemble de fonctions pédagogiques réperées en cours de l'analyse pédagogique. Il permet de traduire ces fonctions en termes des fonctions disponibles sur le poste de travail élève.

Parmi les moyens choisis on retrouve:

- Une organisation particulière de l'écran pour chaque activité, d'après l'approche pédagogique.

- L'encadrement, les attributs d'aspect (clignotement, inverse-video, intensité, etc.) et l'animation qui permettent : la désignation d'éléments, la mise en relation de composants, et la gestion de dialogues.

- La sonnerie : renforcement de phases importantes du dialogue.

- La définition de touches-fonctions : base pour l'activité de l'élève.

Une description des différents moyens utilisés sera donnée plus loin.

III.3. Structuration de la maquette de didacticiel.

Nous abordons cette question sur la base des principes énoncés dans le rapport de l'ADI [ADI-81] qui avaient été déjà utilisés de manière systématique dans le système MOSAIQUE [Ada-83]
Le didacticiel est représenté par un ensemble de dialogues.
L'analyse d'un sujet peut conduire à une succession de dialogues.

Nous avons utilisé dans la maquette deux sujets qui ont été décrits dans le chapitre II (cf.II.1). Nous les illustrons dans la liste de thèmes suivante :

- Notion d'expression algorithmique
- Notion d'état
- L'analyse par cas
- La machine caractère
- L'analyse descendante
- Le traitement séquentiel

Une dialogue développe un sujet pour un moment pédagogique.

La progression des leçons, vis-à-vis de moments péagogiques n'est pas rigide. On peut avoir des sujets où il n'y a qu'un seul moment pédagogique pertinent dans le soutien E.A.O.

Du point de vue de l'élève, la maquette lui permet de travailler un sujet sous l'approche pédagogique choisie par lui; en outre, il pourra travailler de façon indépendante dans les différents moments pédagogiques qui lui sont offerts.

En ayant consideré les éléments pédagogiques intervenant dans l'élaboration des leçons, nous allons par la suite décrire les mécanismes E.A.O. qui ont été mis en oeuvre pour les différentes leçons.

III.3.1 Solutions E.A.O. spécifiques au didacticiel.

Deux solutions E.A.O. ont été mises en oeuvre : Une solution basée sur un modèle simulé et une solution basée sur la définition des accès aux différentes parties de la structure d'un algorithme.

Leur utilisation provoque des réactions au niveau affichage (clignotement, sonnerie, etc.) qui sont propres aux actions produites à travers les solutions

Dans la solution basée sur un modèle simulé, on met à la disposition de l'élève un objet (une machine, un appareil,etc), simulé dans l'ordinateur, que l'on fait fonctionner par l'intermédiaire des touches - fonctions.
Son fonctionnement implique des données spécifiques.

Chaque touche - fonction engendre une action du répertoire d'actions élémentaires de l'objet simulé qu'on veut que l'étudiant manipule.
Le modèle donne la possibilité donc de mettre en rapport la notion d'action nommée et les touches fonctions qui permettent la communication avec le modèle.

Dans la maquette, nous utilisons deux modèles pour aborder les catégories de problèmes traitées : la machine caractères et la balance à deux plateaux.

Le premier permet d'aborder des problèmes concernant le traitement séquentiel. Le deuxième permet d'aborder ceux concernant l'analyse par

cas : on peut établir la comparaison entre n objets, en termes des comparaisons successives entre deux objets. Cette comparaison s'appuie sur la composition conditionnelle.

La deuxième solution concerne l'accès aux composants d'un document d'analyse. Ce document d'analyse que l'on présente à l'étudiant, est représenté d'une façon interne par une structure d'information. On y accède par l'intérmediaire d'un ensemble de requêtes.

Lorsque l'on veut accéder à un composant de l'algorithme, celui-ci est affiché à côté de l'algorithme, ses parties les plus importantes (vis-à-vis de la lecture) étant encadrées.

III.3.2 Spécification des dialogues.

III.3.2.1 Dialogues de sensibilisation.

Pour la notion d'algorithme le dialogue est centré sur le modèle simulé de la machine caractère.

Un tel dialogue est caractérisé par:

- L'utilisation de touches-fonctions

- L'affichage des traces de l'activité de l'élève

- L'affichage sur l'écran des actions qui sont provoquées par l'utilisation des touches fonctions correspondantes.

Dans ce dialogue il n'y a aucune contrainte sur l'utilisation du modèle, c'est-à-dire que l'étudiant est libre d'utiliser les touches à sa convenance. Cela entraîne qu'il n'y a aucun ordre préétabli dans l'interaction.

Au début du dialogue, les machines caractères et compteur sont présentés sur l'écran. Lorsque l'élève appuie sur la touche correspondant à ce qu' il veut faire, le modèle se met en fonctionnement, en affichant les conséquences de l'action correspondante.

a) Données.-

Les exercices propres à ce dialogue manipulent des textes présentés comme des suites des caractères.

Une suite est interprétée comme étant portée sur le ruban de la machine caractère.

Au début d'une session, une suite de caractères (texte) est choisie aléatoirement et le modèle attend l'activité de l'élève.

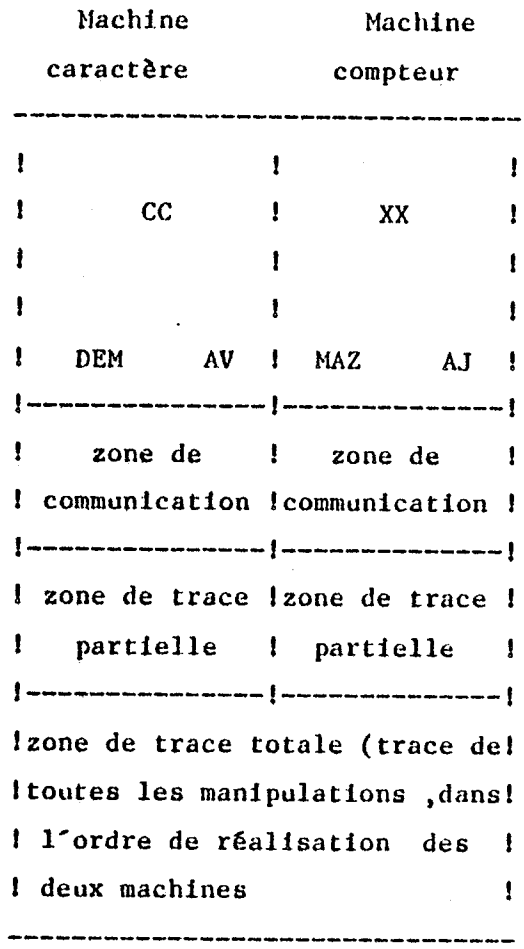
Comme donnée supplémentaire, l'étudiant aura un problème à résoudre posé par l'enseignant (par exemple : Compter les caractères d'un texte), afin d'utiliser le modèle et d'arriver à une solution.

b) Requêtes disponibles.-

La seule requête disponible est la requête d'arrêt 'ARR'.

c) Spécification d'écran.-

Le seul type d'écran prévu est celui qui montre le modèle qui a été simulé pour travailler dans ce moment pédagogique.
Nous utilisons comme modèles simulés la machine caractère et la machine compteur..



La figure ci-dessus présente:

- Les machines affichées sur l'écran.

- Des zones de traces de l'activité de l'élève.

On distingue pour la machine caractère:

. Une fenêtre qui montre le caractère courant.

- Le bouton de démarrage 'DEM' pour mettre le premier caractère de la suite dans la fenêtre.

- Le bouton d'avancement 'AV' pour mettre le prochain caractère de la suite dans la fenêtre.

Deux touches fonctions 'DEM' et 'AV' (sur le clavier) qui sont en correspondance avec les "boutons" 'DEM' et 'AV' affichés sur l'écran.

Les indicateurs de fin de file et de file vide sont représentés par une primitive de sonnerie qui est activée aux moments précis.

Pour la machine compteur :

- Une fenêtre montrant la valeur courante du compteur.

- Un bouton de mise à zéro 'MAZ' qui initialise à zéro le compteur.

- Un bouton pour ajouter 'AJ' qui augmente de '1' le compteur.

Deux touches fonctions 'MAZ' et 'AJ' (sur le clavier) qui sont en correspondance avec les "boutons" 'MAZ' et 'AJ' affichés sur l'écran.

d) Manipulations attendues (élève) et réactions (système)

Dans une session, après qu'un texte a été choisi au hasard, l'étudiant pourra parcourir à volonté le texte, en utilisant les touches fonctions.
Le système ,pour sa part, face à l'activité de l'élève aura plusieurs réactions possibles :

Appuyer sur une touche fonction du clavier provoque :

* Soit dans le cas d'une utilisation correcte, le clignotement du bouton correspondant sur l'écran et la mise à jour de l'information dans la fenêtre de la machine utilisée.

* Soit dans le cas d'une mauvaise utilisation, l'apparition d'un message accompagné d'une sonnerie, indiquant les raisons pour lesquelles, les touches ne peuvent pas être utilisées. (Exemple : essayer d'avancer sans avoir démarré).

Dans tous les cas, tant que la zone des traces (partielle ou totale) n'est pas pleine, l'affichage du nom de l'action correspondante est fait (exemple: affichage de DEM dans la zone de trace de la machine caractère et dans la zone de trace totale).

Appuyer sur la requête 'ARR' provoque l'affichage d'une question pour savoir si l'élève veut continuer avec un autre texte ou non. Si la réponse est négative la session est interrompue.

f) Traces et caractérisation du travail fait par l'élève et/ou l'enseignant.

L'enseignant s'appuiera sur le modèle pour faire travailler l'élève et pour évaluer son activité.

Tout le processus étant affiché sur l'écran, on pourra suivre la démarche et détecter les points de défaillance de l'élève. (Les erreurs commises par l'élève sont signalées aussi en inverse-vidéo sur chaque zone de trace au moment précis (dans l'ordre).

Le dialogue de sensibilisation pour l'analyse descendante est centré sur l'utilisation du mécanisme d'accès et utilise les mêmes éléments des dialogues que pour la présentation.

III.3.2.2 Dialogue de présentation.

Il est caractérisé par:

- Des possibilités d'accès à la structure de l'algorithme.
- Des possibilités d'accès et mémorisation des composants d'un algorithme.

A la différence des dialogues de sensibilisation, dans ceux-ci il y a des effets prévus, c'est à dire que les composants auxquels on peut accéder ont été préalablement définis par l'auteur.

Un dialogue de présentation consiste à suivre un événement préalablement définis.
Il peut être considéré comme un "tourne page intelligent" puisqu'il permet, lorsque l'élève le demande de :

- mémoriser des étapes de présentation de l'algorithme.
- afficher des composants déjà mémorisés.
- accéder aux différents niveaux dans la stucture de l'algorithme.

Au début du dialogue, l'algorithme est affiché, et l'élève a la liberté d'utiliser l'ensemble des requêtes lui permettant de réaliser les actions décrites ci-dessus.

a) Données.-

L'algorithme sur lequel travaille l'étudiant a dû être défini aupara-

vant, ainsi que les différents modes de lecture. L'étudiant n'a qu'à choisir ce mode.

b) Requêtes disponibles.-

Requêtes d'accès :

MONTER 'MON' : Fait apparaître sur l'écran de référence la partie inférieure d'un algorithme qui ne rentre pas complètement.

DESCENDRE 'DES' : Fait apparaître sur l'écran de référence la partie supérieure d'un algorithme qui ne rentre pas complètement.

BLOC <nom du bloc> : L'apprenant tape le nom du bloc auquel il veut accéder. Le bloc est affiché sur l'écran d'évolution en lui montrant à son tour les blocs (identifiés) le constituant.

SORTIR 'SOR' : Permet l'accès au niveau plus haut (au contexte du bloc affiché). Affiche le contexte en question sur l'écran d'évolution.

On distingue dans les requêtes d'accès deux types de fonctions différentes :

Un type de fonction représenté par les requêtes MONTER et DESCENDRE dont le rôle est de modifier la position du curseur dans l'algorithme (choisir, le long de l'algorithme, la partie souhaitée)

Un autre type de fonction représenté par les <noms des blocs> et par la requête SORTIR dont l'interprétation dépend du point de vue de l'apprenant et de celui de l'auteur.

Pour l'auteur, la fonction représente un changement de niveau. Elle permet de visualiser ce qu'il veut montrer à l'étudiant.

Pour l'apprenant, la fonction montrera, au début, une page qui tourne; au fur et à mesure de la progression dans son utilisation, il devra comprendre le changement de niveau que l'auteur veut lui montrer.

Requêtes de liens :

LISTE 'LIS' : Affiche sur l'écran de référence la liste des images (composants) qui ont été mémorisées préalablement dans la session.

ALGORITHME 'ALG' : Affiche sur l'écran de référence l'algorithme

PRECEDENTE 'PRE' : Affiche sur l'écran de référence l'image précédente à celle qui est affichée sur l'écran d'évolution.

MEMORISEE 'MEM' : L'apprenant tape le nom de l'image qu'il veut visualiser et elle est affichée sur l'écran de référence.

Autres requêtes :

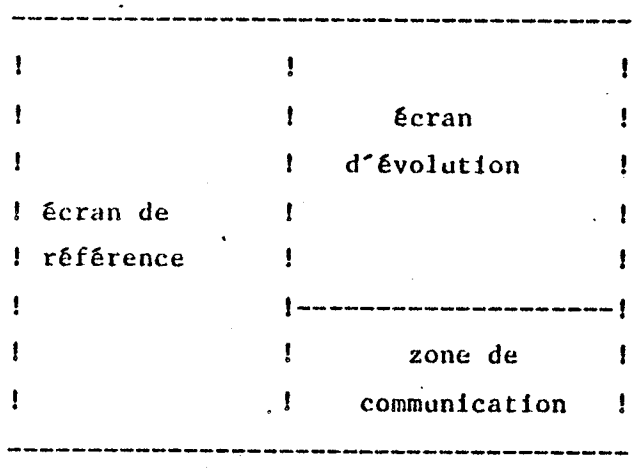
ARRETER : Provoque l'arrêt complet de la session.
'ARR'

ENREGISTRER: L'apprenant, à la demande du système, tape un nom
'ENR' pour l'image affichée sur l'écran d'évolution.
Si le nom tapé existe déjà, la nouvelle image
substitue celle qui existait.

COMPTRENDU : Une trace de l'activité de l'élève est affiché sur
'COM' l'écran d'évolution.

MENU : Le menu de requêtes est affiché sur la zone de
'MEN' communication.

c) Spécification d'écran.-



L'écran de référence permet de visualiser l'algorithme servant de base au déroulement du dialogue.

L'écran d'évolution permet la visualisation des différents composants de l'algorithme affiché à gauche. Il représente un niveau d'abstraction particulier.

La zone de communication permet: l'affichage des questions ,la réception des réponses, l'affichage des messages et commentaires et l'affichage des menus.

Certaines options ont été définies sur cette disposition, selon la nature du dialogue:

Option 1: L'écran de référence montre un énoncé.
L'algorithme est affiché en dessous de l'énoncé pas à pas selon l'analyse présentée dans l'écran d'évolution.

Option 2: L'écran d'évolution est utilisé comme zone d'évaluation (répondre, par exemple, à un questionnaire affiché).
La zone de communication est étendue,(toute la partie inférieure de l'écran réel), pour commenter les réponses de l'apprenant.

Cette disposition d'écran est utilisée dans des situations de présentation et consolidation. Elle permet d'établir des comparaisons entre les différents composants des algorithmes, ce qui facilitera le processus de déduction des relations existantes.

d) Manipulations attendues (élève) et réactions (système).

Dans une session après que le mode de lecture a été choisi, l'étudiant pourra accéder, à volonté, aux différents composants de l'algorithme d'après le mode choisi.
Le système, face à l'activité de l'élève, aura plusieurs réactions possibles.

Prenons, par exemple, l'algorithme suivant correspondant au problème "compter les 'A' " (cf.II.1.1.3), afin de montrer quelques manipulations et réactions.

```
{ Le texte donné comporte au moins un caractère autre que le '.' }

DEM
NBA <- 0

tantque CC =/= '.' faire
{ NBA est le nombre d'occurrences du caractère 'A' rencontrés
  avant CC }

 si CC = 'A' alors
    NBA <- NBA + 1
 fsi
 AV

ftantque
{ NBA est le nombre d'occurrences du caractère 'A' qui précedent
le '.' final }
[] <- NBA
```

Si le mode structural a été choisi les blocs suivants sont affichés :

```
            ------------
(a)         ! DEM      !
            ! NBA <- 0  !
            ------------

            ----------------------------
            ! tantque CC =/= '.' faire !
            !   si CC = 'A' alors      !
(b)         !      NBA <- NBA + 1      !
            !   fsi                    !
            !  AV                      !
            ! ftantque                 !
            ----------------------------

            ------------
(c)         ! [] <- NBA !
            ------------
```

Si le nom "(b)" est donné, les blocs suivants sont affichés (toujours sur l'écran d'évolution) :

```
                  ------------
(a)       tantque ! CC =/= '.'! faire
                  ------------

                  --------------------
                  ! si CC = 'A' alors !
                  !    NBA <- NBA + 1 !
(b)               ! fsi               !
                  ! AV                !
                  --------------------

                ftantque
```

Si, par la suite, on choisit le nom "(b)", les blocs suivants sont affichés:

```
            --------------------
            ! si CC = ´A´ alors !
  (a)       !    NBA <- NBA + 1 !
            ! fsi               !
            --------------------

            ------
  (b)       ! AV !
            ------
```

Si le nom "(a)" est donné, on obtient:

```
               ------------
  (a)       si ! CC = ´A´ ! alors
               ------------

             ------------------
  (b)        ! NBA <- NBA + 1 !
             ------------------

            fsi
```

Finalement, donner le nom "(b)", affiche le bloc

```
             ------------------
             ! NBA <- NBA + 1 !
             ------------------
```

Dans le mode de lecture choisi, le dialogue consiste simplement à montrer (encadrer) les composants de l´algorithme.

L´élève a la liberté d´accéder aux composants qu´il veut et d´utiliser toutes les requêtes qu´il a à sa disposition.

e) Traces et caractérisation du travail fait par l'élève et/ou l'enseignant.

. L'élève s'appuie sur l'ensemble de requêtes pour faire ce qu'il veut sur la structure de l'algorithme.

. A travers la requête COMPTRENDU l'enseignant (et l'élève) pourra suivre la démarche de l'activité réalisée.

III.3.2.3 Dialogues de consolidation.

Au moment de consolidation, un dialogue est caractérisé par :

* Une liste de problèmes appartenant à la catégorie de problèmes qui ont été presentés (représentés eventuellement par les algorithmes correspondants) pour fixer les connaissances.

* Des possibilités d'accès sous contrôle aux composants de l'algorithme.

* L'apparition des questions tout au long du dialogue.

* L'apparition des QCM, pour faire le résumé sur l'évaluation du sujet. (exemple: un test sur les variables utilisées).

En général, le dialogue se déroule comme pour la présentation (cf.III.3.2.2) ; la différence réside dans le contrôle sous lequel l'étudiant travaille.

Lorsque la réponse attendue de l'élève est un bloc, l'algorithme est affiché avec un numéro identifiant chaque ligne.

Une réponse de l'élève impliquera donc, le nom (numéro) de la ligne et celui du composant dans la ligne (1er,2ème,..)

Exemple de dialogue:

```
1 DEM
2 NBC <- 0
3 tantque CC '.' faire
4 NBC <- NBC + 1
5 AV
6 ftantque
```

Combien d'actions y a-t-il dans
la composition itérative ? ___

(on encadre la composition itérative).

Lesquelles ? ___

Les autres spécifications restent les mêmes que pour les dialogues de présentation.

III.4 Synthèse sur les fonctions E.A.O. nécessaires

Les objectifs pédagogiques ayant été fixés pour chacune des leçons, notre intérêt était la mise en oeuvre d'une solution qui permettait d'atteindre ces objectifs en utilisant les différentes ressources disponibles (système et matériel).

Nous donnons par la suite, quelques considérations sur les fonctionnalités E.A.O nécessaires et la façon dont elles ont été mises en oeuvre.

Nous montrerons, pour chaque approche, les besoins et de quelle façon on a satisfait ces besoins.

III.4.1 Pour l'approche lecture

. Besoin 1 : Identification et/ou discrimination d'un élément de l'algorithme.

. Outil : L'encadrement et une combinaison des attributs d'aspects, avec des rôles pédagogiques précis, notamment:

. Cadre double intensité :
Il permet d'identifier pendant la lecture un composant de l'algorithme.

. Cadre inverse-vidéo :
Il permet de signaler ces composants de l'algorithme susceptibles d'être disjoints pour une éventuelle substitution.

. Cadre inverse-vidéo-clignotant :
Il permet de renforcer une réponse comprenant plusieurs composants de l'algorithme (Exemple : une composition itérative).

. Besoin 2 : Moyen de communication

Nous avons employé l'encadrement en double intensité comme moyen pour identifier pendant la lecture les composants intéressants selon les modes de lecture.
Lorsqu'on demande à l'élève une réponse qui implique de distinguer un composant de l'algorithme, il ne dispose pas d'un outil lui permettant d'afficher directement des cadres; il devrait pouvoir utiliser le même moyen pour exprimer sa réponse.

. Outils : Deux situations se présentent correspondant aux moments pédagogiques de présentation et consolidation.

Pour la présentation on propose à l'étudiant un nom pour chaque cadre (bloc) affiché pendant la lecture d'un algorithme. L'élève doit donner le nom du cadre qu'il veut désigner. Les encadrements correspondant aux blocs internes du bloc signalé sont affichés par la machine (voir Exemple 1).

Pour la consolidation, étant donné qu'on exige une réponse (en général un cadre que l'élève, avec le poste de travail utilisé, ne peut pas afficher), on propose à l'élève des coordonnées pour chaque composant de l'algorithme affiché permettant de les identifier (voir exemple 2).

La réponse de l'élève est donnée en deux temps :

1) il donne le nom de la ligne.
   Le système affiche en inverse-vidéo les composants de cette ligne.

ii) si la ligne a plus d'un composant, le système demande le nom du composant (1er., 2ème.,,)

```
          ------------
a)        ! DEM      !
          ! NBA <- 0  !
          ------------

          ----------------------------
          ! tantque CC =/= '.' faire !
          !  si CC = 'A' alors       !
          !       NBA <- NBA + 1     !
b)        !  fsi                     !
          !  AV                      !
          ! ftantque                 !
          !                          !
          ----------------------------

          -------------
c)        ! [] <- NBA !
          -------------
```

Exemple 1 .-
Chaque bloc a son nom (a,b et c) permettant l'accès.

```
1       DEM
2       NBA <- 0

3       tantque CC =/= '.' faire

4        si CC = 'A' alors
5               NBA <- NBA + 1
6        fsi
7        AV
8       ftantque

9       [] <- NBA
```

Exemple 2 .-
Chaque composant a son identification.
Si on veut désigner l'élément CC = 'A'
par exemple, on donne comme réponse
4 (le système affiche en inverse -
vidéo les trois composants :

si
CC = 'A'
alors

et puis on donne comme réponse 2
(2ème composant de la ligne).

Cette solution reste cohérente avec les types de réponses utilisées auparavant, qui ont été basées sur des réponses numériques ou des mots ("oui", "non", "a",..).

. Besoin 3 : Montrer les composants de l'algorithme à un moment donné sous différents points de vue.

Notamment:

- Des niveaux d'abstraction
- Des module
- Des schémas
- Des compositions d'actions

. Outils : L'utilisation de deux écrans (un écran de référence et un écran d'évolution) d'une part, et l'encadrement et l'utilisation des attributs d'aspects d'autre part, nous ont permis de mettre en évidence ces points, en affichant sur l'écran de référence l'algorithme correspondant et sur l'écran d'évolution les parties concernées, notamment :

- l'encadrement des composants correspondant au niveau d'abstraction immédiatement inférieur du niveau courant.
- l'encadrement d'un module signalé dans l'algorithme.
- le clignotement des composants d'un schéma particulier.
- la structure de chaque composition en remarquant les composants intéressants par rapport à la lecture qu'on fait.

La mise en oeuvre de cette solution a impliqué la réalisation de certaines fonctionnalités pour mémoriser des états d'un algorithme à un moment donné.
L'affichage sur les deux écrans simultanément permet de faire des comparaisons de ces deux niveaux affichés.

. Besoin 4 : Visualisation de l'algorithme.
On a , très souvent, besoin de visualiser un algorithme complet.
La taille de l' écran utilisé ne permet pas, en général, cet affichage.
Il faut donc, délimiter la portion de l'algorithme qu'on veut afficher à chaque instant.

. Outil : On simule le mode de défilement type "scroll" de l'algorithme et on n'affiche que la partie qui nous intéresse à un moment donné.

. Besoin 5 : Montrer différents aspects propres aux sujets et à la méthode suivie pour les apprendre.
Nous avons besoin, notamment, de montrer le rapport existant entre l'analyse d'un algorithme et sa construction.

. Outil : Une disposition d'écran (définition des zones d'affichage) nous a permis de montrer au fur et à mesure de l'analyse d'un problème, la construction de l'algorithme correspondant.
Cette disposition de l'écran est utilisée pour la présentation et la consolidation.
Elle permet de proposer les trois types d'activités de lecture.

Le système étant construit pour une manipulation des messages et des zones sur l'écran, il ne suffit pas pour manipuler aisément la structure algorithmique et certains rapports entre ses composants. On a dû créer, du point de vue système, des outils pour satisfaire ces besoins, notamment :

. Besoin : Manipuler et faire manipuler des composants de la structure algorithmique sous différents points de vue.

. Outil : Nous avons créé une structure de données pour représenter les algorithmes par niveaux. Cela permet, non seulement, de traiter les composants algorithmiques mais aussi de faire leur affichage correspondant.
Cet outil nous a mené à conditionner le système (modification de l'environnement) pour la manipulation de cette structure.

L'outil a été créé en utilisant les mêmes fonctionnalités de base du système VOYELLES [Lcc-83] et certaines de ses structures de données.

De nouvelles fonctionnalités ont été créées pour permettre la mémorisation de scènes (états de la lecture), l'accès et l'affichage des différents niveaux de l'algorithme.

III.4.2 Pour la simulation

. Besoin : Mettre l'élève en activité.
La manipulation facilite l'acquisition des connaissances par l'élève.
La connaissance du modèle de machine séquentiel est une bonne aide pour la compréhension du traitement séquentiel de l'information.

. Outils : On a fait la réalisation du modèle simulé en conservant les mêmes spécifications du modèle théorique; cela veut dire qu'on a affiché sur une partie de l'écran la machine que l'on veut faire utiliser à l'élève.

Pour son fonctionnement on a défini un ensemble de touches - fonctions simulant les boutons de la machine réelle

Les attributs d'aspects, notamment l'inverse - vidéo - clignotant, servent à montrer la machine "en marche".

Différentes zones d'affichage ont été définies pour montrer les traces partielles et la trace totale de l'activité de l'élève.

Une zone de communication permet d'afficher les messages de réaction du modèle et certains commentaires.

. Besoin : montrer l'activité de l'élève d'après les touches - fonctions utilisées.

. Outils : Une trace, représentant cette activité, est affichée pendant l'utilisation du modèle.

III.5 Méthode de production du didacticiel.

Nous avons utilisé un méthode qui s'appuie sur celles qui ont été mises en oeuvre dans le projet MOSAIQUE [Ada-83]) pour la production des programmes informatiques.

Dans cette méthode nous distinguons trois étapes :

III.5.1 Etape de préparation pédagogique.

On considère les éléments suivants :

a) Fixation des objectifs:
Consiste à préciser quelle partie du sujet on va travailler.

b) Choix du moment pédagogique:
On fixe le niveau d'acquisition de connaissance sur lequel on veut faire travailler l'élève.

c) Le choix de l'exercice:
On doit trouver un problème bien adapté tant aux objectifs qu'au moment pédagogique.

d) L'analyse pédagogique:
Définir ce qu'il faut montrer et comment le montrer.

Exemple:

a) : Montrer l'analyse par cas.

b) : Dialogue de présentation.

c) : MAXIMUM DE DEUX VALEURS.

d) "que" => . la partition du domaine des données
. les actions correspondantes

"comment" => Approche de lecture (type 1: lire l'algorithme pour découvrir comment il est construit à partir de l'énoncé).

III.5.2 Etape de transition pédagogie-informatique

Elle concerne l'analyse des possibilités de réalisation de la leçon. Cette étape peut devenir importante selon que l'auteur, faisant partie de l'équipe pédagogique, participe ou non à la préparation informatique. La non participation peut demander une interaction importante entre la partie pédagogique et la partie informatique afin que cette dernière puisse établir les niveaux de précision pour la réalisation des dialogues correspondants.

Cette étape permet aussi la vérification des outils dont on dispose pour faire la réalisation. Elle est particulièrement importante lors de la définition de nouvelles fonctionnalités non prévues dans le système.

III.5.3 Etape de préparation informatique.

Elle comporte deux aspects :

a) La définition d'un document leçon (dialogue) qui consiste à décrire:

. La définition des écrans

. La description du déroulement

. La codification du dialogue.

b) Intégration du code au système:

On travaille sur l'ordinateur.

On doit faire :

- La définition des messages et zones
- La transcription du code

Une leçon est achevée lorsque on met en rapport tous les éléments du système, c'est-à-dire:

- Les modules
- Le code (la leçon codifiée)
- Les données

Ces trois éléments intégrés dans un seul programme, dont le nom est celui de la leçon, constituent la leçon à présenter à l'étudiant.

III.6 Les leçons

Une leçon est caracterisée par:

- le sujet
- le moment pédagogique
- l'approche pédagogique
- la stratégie pédagogique
- les moyens E.A.O.
- exercice choisi (énoncé)
- le déroulement (étapes fondamentales)

Le tableau suivant montre les leçons qui ont été réalisées dans la maquette.
Il montre les éléments de base de la structuration de la maquette.

		Moments pédagogiques		
		Sensibilisation	Présentation	Consolidation
S	AC		MAX3V (LT1)	
U	AC		MAX2P (LT1)	
J				MAX2C (LT1)
E	TS		CONTA (LT1)	
T	TS		FREQA (LT3)	FREQA
S	AD			
	TS	MACHI (S)		
	TS			CLES2 (LT3)
	TS		CLES1 (LT3)	
	AC	BALAN (S)		

où :

AC = Analyse par cas,

TS = Traitement séquentiel,

AD = Analyse descendante.

MAX3V = "MAXIMUM DE TROIS VALEURS"
MAX2P = "MAXIMUM DE DEUX VALEURS" (Présentation)
MAX2C = "MAXIMUM DE DEUX VALEURS" (Consolidation)
CONTA = "COMPTER LES 'A'"
FREQA = "FREQUENCE DES CARACTERES 'A'"
CLES1 = "COMPTER LES 'LE'" (Présentation)
CLES2 = "COMPTER LES 'LE'" (Consolidation)
MACHI = "MACHINE CARACTERES"
BALAN = "BALANCE"

Une ligne de ce tableau réunit un sujet, le nom de la leçon et l'approche sous laquelle la leçon est présentée.

Les colonnes, quant à elles montrent, soit la liste des sujets abordés dans la maquette, soit les leçons prévues pour chacun des moments pédagogiques.

Un aspect à remarquer dans la maquette est le rapport existant entre le moment et l'approche pédagogique utilisée pour un sujet. Cela montre, par exemple, qu'un sujet peut être présenté sous une approche et consolidé sous une autre.

Nous allons, par la suite, décrire ces leçons en donnant leurs caractéristiques vis-à-vis de la description faite précédemment.

Leçon : MAX2P

Sujet : L'analyse par cas

moment pédagogique : Présentation

Approche pédagogique : Lecture type 1
(Enoncé -> Algorithme)

Stratégie pédagogique :

- Utilisation des données du problème pour introduire la notion de partition.

- Association de l'analyse par cas avec la notion de partition.

Moyens E.A.O. :

- Configuration de l'écran pour ce type de lecture.

- Visualisation de la notion de partition à l'aide d'encadrements et de clignotements.

Exercice choisi : " MAXIMUM DE DEUX VALEURS "

" Etant donné deux nombres entiers, déterminer celui qui a la valeur la plus grande.
On suppose les deux nombres diffé - rents) "

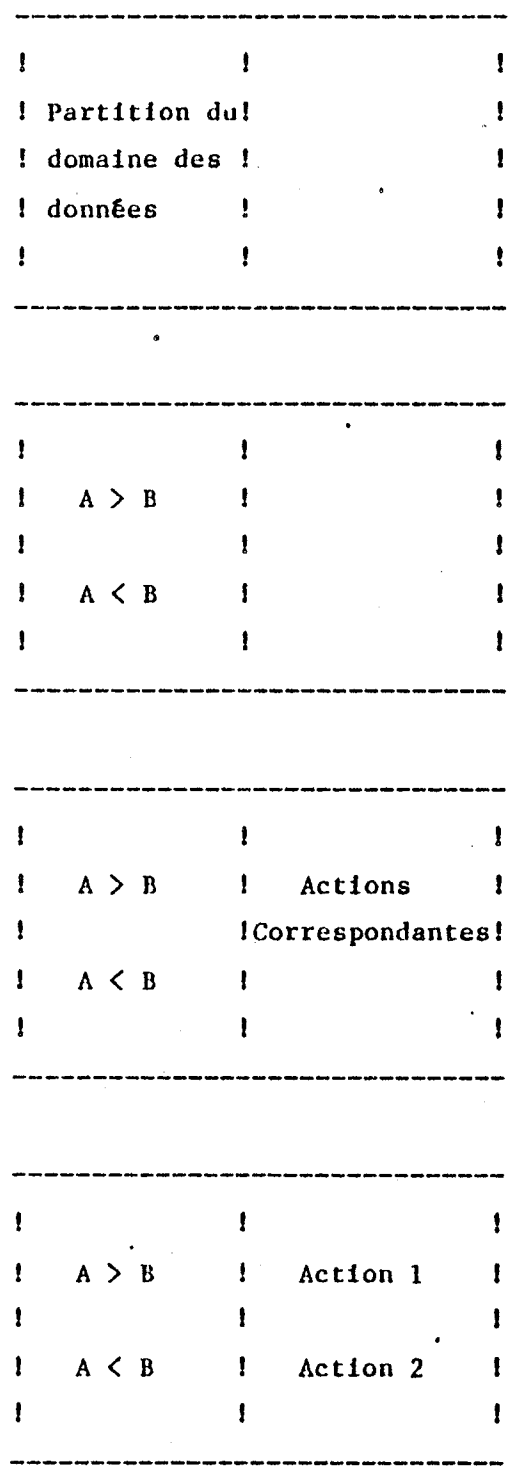
Déroulement :
On montre les algorithmes découlant de deux analyses faites.

* Première analyse : La construction est fondée sur une partition du domaine de données.

- Apparition de l'énoncé.

- Apparition des assertions montrant l'état initial et un état final incomplet. On demande à l'élève de le compléter.

- Association de "boîtes nommées" avec des expressions. On montre ci-dessous les différents passages du déroulement montrant le raffinement pour obtenir les expressions correspondantes.


Partition du domaine des données
A > B
A < B
A > B
Actions Correspondantes
A < B
A > B
Action 1
A < B
Action 2

A > B		MAXI <- A
A < B		MAXI <- B

- Association de l'analyse faite et de l'algorithme correspondant (on montre la notation algorithmique).

- Affichage de l'algorithme.

* Deuxième analyse : La construction est fondée aussi sur une partition du domaine des données.

- On affiche d'abord les actions possibles (pour le problème)

	MAXI <- A
	MAXI <- B

et on déduit les cas correspondants. La situation tourne autour de la question :"Sous quelles conditions, ces actions sont-elles justes ?".

- On affiche l'algorithme correspondant sur l'écran de référence.

```
{ A=a , B=b }

choix

      . A > B : MAXI <- A

      . A < B : MAXI <- B

fchoix

{ MAXI = valeur la plus grande de a et b }
```

Leçon : MAX2C

Cette leçon illustre le même exercice que MAX2P mais s'utilise dans un autre moment pédagogique.

Sujet : L'analyse par cas

moment pédagogique : Consolidation

Approche pédagogique : Lecture type 1
(énoncé ---> algorithme)

Stratégie pedagogique :

- Modification de l'énoncé d'un problème connu, pour faire déduire à l'élève la réponse.

Moyens E.A.O :

- Disposition de l'écran pour la lecture.

- Visualisation de la notion de partition à l'aide d'encadrements et de clignotements.

- Communication des réponses de l'élève.

Exercice choisi : " MAXIMUM DE DEUX VALEURS "

Même énoncé sauf que les deux valeurs peuvent être égales.

Déroulement :

- Affichage de l'algorithme de la leçon MAX2P (pour deux valeurs différentes). C'est un rappel de la situation.

- Interaction avec l'élève ; on demande une solution. Deux types de réponses sont acceptés:

à deux cas.
à trois cas.

(L'élève doit les donner toutes les deux dans l'ordre qu'il veut !)

- On fait travailler l'élève d'abord sur les cas et puis sur les actions correspondantes.

- Les réponses données pour la partition sont memorisées pour être utilisées après dans les actions correspondantes.

Erreurs possibles

Type 4 : "Nombre de cas en dehors des valeurs possibles"

Type 5 : "Valeur non numérique"

Type 6 : "Cas redondants"

Type 7 : "La partition n'est pas correcte"

Réaction du programme

- Analyse de réponse aux différents niveaux.

Pour tous les niveaux, s'il y a des erreurs:

. On efface la réponse donnée

. On la redemande.

. On affiche le message propre à l'analyse faite.

Leçon : MAX3V

Sujet : L'analyse descendante

Moment pédagogique : Présentation

Approche pédagogique : Lecture type 1
(énoncé ---> algorithme)

Stratégie pédagogique :

- Utilisation des connaissances acquises de l'exercice "MAXIMUM DE DEUX VALEURS". (Analogie)

- Application systématique de la démarche montrée précédemment pour déterminer le maximum de deux valeurs dans ce problème.

- Utilisation des relations existant entre les définition d'une partition dans un ensemble et la notion de réduction de cet ensemble.

Moyens E.A.O :

- Visualisation de la notion de partition à l'aide d'encadrements et de clignotements.

- L'animation illustre les éléments présents dans chaque cas.

- Le clignotement des cadres nous permet de montrer l'aspect réduction.
  En faisant clignoter un cas encadré de la partition (qui représente un ensemble d'informations), nous affichons les cas qui en découlent (sous ensembles d'information obtenus)

- La disposition de l'écran pour la lecture type 1.

Exercice choisi : " MAXIMUM DE TROIS VALEURS "

" Etant donné trois nombres entiers,
déterminer le plus grand "

Déroulement :

- Apparition de l'énoncé.

- Apparition de trois noms de variables (A,B,C) et une boîte nommée " MAXI (A,B) ".

- Le système montre le maximum de A et B en termes de partition.

- Le système montre le maximum des deux valeurs (entre C et celui qui a résulté de la détermination précédente).

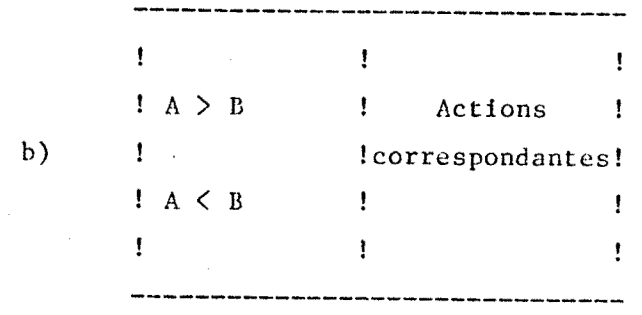
- On montre la séquence des scènes suivantes:

a)

Partition du domaine des données	Actions correspondantes

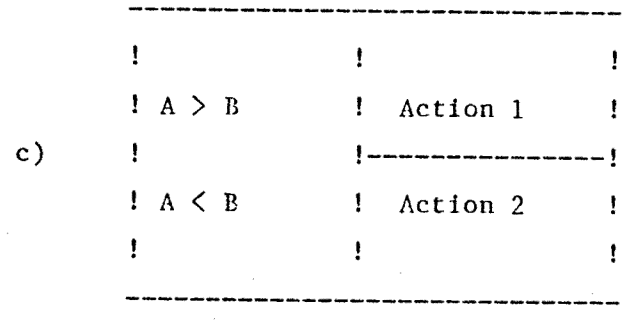
On fait clignoter la boite "partition du ...." et on affiche :

b)



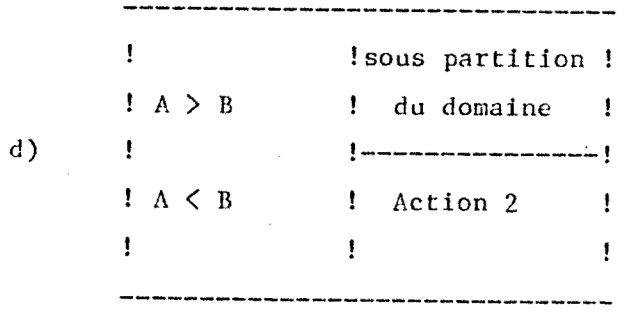
Idem pour " Actions .." et on affiche :

c)



"Action 1" représente à son tour une partition, on affiche :

d)



la même démarche continue :

```
        --------------------------------
        !              ! A > C !Action ! { Visualisation de
        ! A > B        ! A < C !Action !   l'analyse descen-
e)      !              !---------------!   dante et aspect
        ! A < B        !   Action 2    !   réduction }
        !              !               !
        --------------------------------
```

```
  .                     .
  .                     .
  .                     .
```

Raffinement à tous les niveaux. On obtient :

```
        ----------------------------------
        !              ! A > C !MAXI <- A!
        ! A > B        ! A < C !MAXI <- B!
n)      !              !-------!---------!
        ! A < B        ! B > C !MAXI <- B!
        !              ! B < C !MAXI <- C!
        ----------------------------------
```

- Construction de l'algorithme correspondant

{ A=a, B=b, C=c, a=/=b, b=/=c, a=/=c }

choix

. A > B : { Le résultat est le maximum de A et C }

choix

. A > C : MAXI <- A

. A < C : MAXI <- C

fchoix

. A < B : { Le résultat est le maximum de B et C }

choix

. B > C : MAXI <- B

. B < C : MAXI <- C

fchoix

fchoix

{ MAXI = plus grande valeur de a, b et c }

Lecon : CONTA

Sujet : Le traitement séquentiel

Moment pédagogique : Présentation

Approche pédagogique : Lecture type 1
(énoncé ---> algorithme)

Stratégie pédagogique :

- Etablir le rapport de façon immédiate entre le modèle machine caractère et le principe du traitement séquentiel.

- Etudier les aspects "accès" et "traitement" séparément.

- Montrer ces aspects dans le schéma du traitement séquentiel.

Moyens E.A.O :

- Configuration de l'écran pour la lecture type 1.

- Visualisation d'un composant de l'algorithme à un certain niveau d'abstraction à l'aide d'encadrements et de clignotements.

Exercice choisi : " COMPTER LES 'A' "

Déroulement :

- Apparition de l'énoncé du problème " COMPTER LES 'A' ".

- Test de compréhension de l'énoncé.
Affichage d'un texte-exemple et sollicitation d'une réponse pour la question : "Pour ce texte,

combien y a-t-il de caractères "A" ?

- Commentaires sur l'objectif du problème.

- Sur l'écran d'évolution se développe l'analyse du problème, ainsi :

  - Analyse du traitement d'un caractère quelconque (affichage des cas possibles).

  - Accès au prochain caractère.

  - Application de l'action élémentaire à tous les caractères.

  - Détermination de la condition d'arrêt.

- On montre pas à pas la construction en utilisant la notation algorithmique sur l'écran de référence.

- On montre l'analyse faite par rapport à l'accès et traitement du premier caractère.

- On montre le raffinement du traitement d'après le modèle simple d'accumulation.

- Une fois que la construction est finie, on fait un renforcement du schéma de traitement séquentiel

  - Accès aux caractère

  - Traitement des caractères

Réactions du programme

- Analyse de réponses au moment des sollicitations par simple correspondance avec la réponse juste.

Leçon : FREQA

Sujet : Le schéma séquentiel et l'analyse descendante.

Moment pédagogique : Présentation et Consolidation.

Approche pédagogique : Lecture type 3.
(Algorithme ---> énoncé)

Stratégie pédagogique :

- Faire découvrir ce que fait l'algorithme présenté.

- Identification du schéma de traitement séquentiel.

Moyens E.A.O :

- Configuration de l'écran pour ce type de lecture.

- Encadrement des actions.
  Cela permet de faire abstraction des niveaux concernés par la lecture.

- Clignotement.

- Tests d'évaluations basés sur un arbre de décisions préalablement établi.

Exercice choisi : " FREQUENCE D'APPARITION DU CARACTERE A "

" Le ruban de la machine caractères porte un texte non vide terminé par un "." . On veut savoir le pourcentage du nombre de 'A' par rapport au nombre de caractères distincts de l'espace. "

Déroulement :

- Apparition de la consigne:
  " DECOUVRONS CE QUE FAIT CET ALGORITHME ".

- Apparition de l'algorithme sur l'écran.

- On pose des questions sur la composition répétitive de l'algorithme; on s'intéresse à montrer les deux actions de la composition.

- Renforcement des réponses.

- Apparition du schéma de traitement séquentiel.

- Identification des parties.

- Abstraction sur l'action VISITER
  (Effacement de tout sauf l'action VISITER).

- Consigne: " C'est une analyse par cas ".

- Lecture de l'action correspondante.
  (Composition sélective : si)

- Consolidation de l'analyse par cas.

- Interprétation du "si" comme un "choix"
  (Substitution dans l'algorithme).

- affichage des tests sur les variables manipulées et sur la compréhension de l'algorithme.

- Consigne : "L'algorithme est-il correct ? ".

- Interaction sur la consigne.

Erreurs possibles

Type 4 : "Nombre de cas en dehors des valeurs possibles"
Type 5 : "Valeur non numérique"
Type 6 : "Cas redondants"
Type 7 : "La partition n'est pas correcte"

Réactions du programme

- Une analyse de réponse par simple correspondance avec la réponse juste.

- S'il y a une erreur : message "non" et on repose la question.
  L'analyse de réponse implique l'affichage des messages propres à chaque situation.

III.7 La leçon-machines

La leçon qui suit est particulièrement importante car elle est la base de plusieurs exercices pour le moment pédagogique de sensibilisation.

Un exercice dans notre contexte fait référence à un problème posé par l'enseignant, indépendamment du didacticiel, dont la résolution requiert l'utilisation de ce dernier.

On présente la machine caractère et la machine compteur et on fait travailler l'élève sur des problèmes spécifiques concernant la notion d'algorithme.

Cette leçon sert d'outil pour aborder des problèmes en rapport avec le traitement séquentiel, (plutôt qu'une leçon dans le sens de celles qui ont été décrites précédemment) qu'on peut utiliser, dans un contexte de travail dirigé.

Leçon : MACHI

Sujet : Le traitement séquentiel.

Moment pédagogique : Sensibilisation

Approche pédagogique : Simulation

Stratégie pédagogique :

- Confrontation de l'élève face à un outil de travail.
- Liberté d'utilisation pour lui laisser découvrir, déduire et apprendre.
- Faire résoudre un problème appartenant à la catégorie de problèmes souhaitée.

Moyens E.A.O :

- Configuration de l'écran pour ce genre de manipulation.

- Utilisation du mécanisme de simulation pour la manipulation du modèle.

Exercice choisi : Un exercice dont le but est l'utilisation de la machine caractère et/ou la machine compteur. Exemple " COMPTER LES 'A' ".

" Le ruban de la machine caractères porte un texte non vide terminé par un "." . On veut savoir le nombre d'occurrences du caractère 'A'."

Déroulement :

- L'étudiant a la liberté de manipuler le modèle pour parcourir la suite d'information (texte) donnée.

- Touches fonctions disponibles

Machine caractère : DEM , AV
Machine compteur : MAZ , AJ

Requêtes possibles : " ARR "

Erreurs possibles

Type 1 : "Avant d'avancer, il faût démarrer"
Type 2 : "Avant d'ajouter, il faut mettre à zéro le compteur"
Type 3 : "Appuyez la touche correcte S.V.P"

Réactions du programme :

- L'utilisation de touches-fonctions provoque l'affichage de trois traces de l'activité de l'apprenant :

  trace de son activité sur la machine caractère.
  trace de son activité sur la machine compteur.
  trace générale.

- A chaque utilisation d'une touche-fonction, sur le clavier, apparaît un clignotement de la touche correspondante dans la machine affiché sur l'écran; et ensuite l'affichage :

  - soit du caractère courant dans la fenêtre.
  - soit du compteur mis à jour.

- S'il y a une erreur, cela provoque une sonnerie et l'affichage des messages correspondants.

III.8 Conclusions.

Ce chapitre a permis de montrer les éléments d'une démarche systématique pour la réalisation d'un didacticiel à partir d'une analyse pédagogique préalable.

La spécification informatique des leçons s'appuie sur une typologie précise des diverses fonctions informatiques nécessaires. Celles-ci sont déduites des fonctions pédagogiques issues de l'analyse, en tenant compte de l'environnement disponible.

Nous avons utilisé un poste de travail classique qui n'est pas particulièrement prévu pour notre contexte précis (parce que nous avons choisi de ne pas inclure dans cette maquette l'étude du poste de travail, sujet d'une maquette à venir).

Nous avons ainsi toujours utilisé des fonctionnalités E.A.O. simples. Dans le cas des fonctions de communication, nous avons constaté que leur simplicité diminuait l'ergonomie du poste de travail, mais permettait, tout de même, de satisfaire aux besoins pédagogiques.

Dans le cas des fonctions d'analyse de réponse, le souci de ne pas investir sur ce thème, nous a conduit à limiter les dialogues de consolidation à des interactions très simples.

Une première observation est que la méthode de production utilisée permet de mieux maîtriser les diverses étapes de construction des leçons et surtout que l'analyse pédagogique est à la base de la formulation employée pour spécifier les leçons.

Nous n'avons pas employé, dans le cadre de cette maquette, toutes les situations possibles que l'on peut déduire de la typologie de moments et d'approches et de la liste de sujets choisis pour chaque thème.

Toutefois, les leçons effectivement réalisées, sont suffisamment significatives pour envisager la production dans de bonnes conditions d'un didacticiel en vraie grandeur.

On peut notamment souligner que sur la base de notre expérience, la réalisation d'une nouvelle leçon compatible avec le contexte de la maquette peut être assurée très rapidement.

Nous estimons donc avoir atteint notre objectif d'exploration préliminaire à une réalisation en vraie grandeur.

CHAPITRE IV : Considerations sur l'expérience réalisée.

IV Considération sur l'experience réalisée.

Introduction.-

La production du didacticiel de soutien en FLE au cours du projet MOSAIQUE a permis de dégager les principes d'une méthode de production.

Ce premier travail à déjà montré l'efficacité de la méthode pour l'obtention de didacticiels de grande taille.
A partir des ces résultats notre groupe de recherche s'est donné pour objectif de préciser cette méthode de production de manière à pouvoir l'appliquer quels que soient les thèmes d'enseignement et la stratégie pédagogique choisis.

Pour atteindre cet objectif nous avons entrepris une démarche expérimentale : nous voulons explorer succesivement chacun des éléments de la méthode de production, en validant des hypothèses précises par la réalisation de maquettes. Cette démarche expérimentale, itérative par définition, doit contribuer aussi bien à préciser la méthode de production, qu'à affiner le plan de recherche.

A l'issue du travail présenté dans cette thèse, nous nous proposons de tirer des conclusions dans ces deux directions. Pour cela, nous commençons par rappeler brièvement le principe de la méthode de production et les hypothèses sous-jacentes à cette première maquette.

La production est basée sur une nette distinction entre analyse pédagogique et réalisation informatique, et sur la reconnaissance explicite des compétences propres aux différents intervenants d'un processus E.A.O..

L'analyse pédagogique comporte l'analyse du contenu, la définition de moments pédagogiques, le choix d'approches pédagogiques. Cette analyse se fonde, d'une part, sur l'expérience pédagogique et d'autre part, sur une stratégie précise d'intégration de l'E.A.O. dans le processus d'apprentissage.

C'est ainsi que dans notre cas, nous nous sommes appuyés sur une forte expérience d'enseignement du sujet choisi et que nous avons retenu l'utilisation de l'E.A.O. comme ressource complémentaire à un enseignement existant (nous parlons d'un E.A.O. de soutien).

L'analyse pédagogique conduit à définir les fonctions pédagogiques en termes desquelles on peut exprimer l'enseignement envisagé. Par ailleurs on doit aussi décrire la structure d'ensemble de cet enseignement. Il faut ensuite choisir les fonctions pédagogiques qui seront utilisées en E.A.O. et les exprimer en termes informatiques, ce que nous appelons définir une solution E.A.O..

On peut alors considérer le didacticiel comme un ensemble de leçons (dialogues).

La réalisation informatique s'appuie sur une bibliothèque de primitives correspondant aux fonctions pédagogiques.
La production se poursuit par un ensemble de tests : tests informatiques, puis tests pédagogiques.

L'expérimentation et la validation du didacticiel peuvent alors avoir lieu.

On peut enfin envisager l'industrialisation du produit : mise en forme, diffusion.

La maquette que nous avons présentée avait pour objet d'expérimenter la prémière étape de ce processus, à savoir l'analyse pédagogique.

C'est pourquoi, nous avons fixé au départ les point suivants : On ne cherche pas à réaliser un didacticiel complet mais au contraire un ensemble de leçons pertinentes pour notre objectif; on n'explore pas toutes les possibilités de poste de travail (auteur ou élève) mais au contraire on s'appuie sur des postes de travail existents (ceux du projet MOSAIQUE, profitant ainsi immédiatement du savoir faire, acquis antérieurement).

De même on n'envisage pas d'utiliser autrement la culture informatique de l'auteur (qui ici est aussi informaticien).
Enfin on ignore ici les problèmes relatifs à l'industrialisation.

Les éléments que nous avons ignorés dans cette maquette seront l'objet de prochaines expériences spécifiques.

IV.1 Quelques résultats de l'expérience.

Nous cherchons ici à dégager deux types de conclusions selon que l'on considère la méthode de production ou l'utilisation de l'E.A.O. dans l'enseignement de la programmation.

La réalisation de notre maquette confirme, tout d'abord, l'intérêt d'une séparation des phases d'analyse pédagogique et de réalisation informatique.

En premier lieu, cette séparation permet de mener l'analyse pédagogique d'une manière plus claire et plus systématique.

Nous avons retenu une typologie simple en ce qui concerne les moments pédagogiques (sensibilisation, présentation, consolidation) et nous avons choisi deux approches pédagogiques qui nous ont permis d'isoler les apports spécifiques de l'E.A.O. à l'enseignement consideré. C'est ainsi que l'activité de simulation a servi de base pour la sensibilisation à la notion d'algorithme, et que la mise en évidence de différents modes de lecture a permis de traiter l'analyse descendante.

Pour chacune de ces approches nous avons pu définir des fonctions pédagogiques précises, que nous résumons maintenant :

a) la simulation est utilisée comme base d'une pédagogie active.

Une première fonction pédagogique doit permettre à l'élève d'appréhender le phénomène simulé à partir de sa propre activité.

Elle conduit à spécifier les commandes disponibles pour l'élève (cf. II.1.3, III.3.2 : "machine caractère" et "machine compteur").

Une deuxième fonction pédagogique doit permettre à l'élève de prendre du recul sur sa propre activité en relation avec l'enseignement effectué en cours.

Elle conduit à spécifier la documentation mise à disposition de l'élève au cours de l'activité (cf. III.3.2 : "traces" élève).

b) Nous avons voulu placer la lecture d'algorithmes sous l'angle d'une pédagogie active.

Une première fonction pédagogique doit permettre à l'élève d'accéder librement aux divers composants du texte d'un programme. Pour permettre à l'élève d'appréhender les divers modes de lecture que nous avons définis, plusieurs fonctions pédagogiques ont été isolées.

Lorsque la lecture consiste à suivre les étapes de construction de l'algorithme , à partir de l'énoncé, nous offrons à l'élève la possibilité de désigner les textes succesifs et de les mettre en relation.

Ceci conduit à :

* spécifier la représentation arborescente de l'ensemble des textes considerés et le mécanisme d'accès correspondant (cf. II.3, III.4.1).

* définir la manière de présenter ces informations à l'apprenant (choix d'une organisation, de l'écran, cf. III.3.2 (c), III.4.1).

* définir les modes de désignation proposés à l'élève (utilisation d'attributs graphiques,.., cf. III.4.1).

Lorsque la lecture consiste à analyser les étapes d'analyse à partir du texte de l'algorithme, nous offrons à l'élève les moyens correspondants de désignation et accès aux informations (toujours sur la base de la structure arborescente sous jacente) (cf. III.3.2.2).

On voit ainsi comment nous avons cherché à rendre méthodique l'analyse pédagogique par une recherche des fonctions pédagogiques spécifiques à la stratégie d'enseignement envisagée.

Une conséquence immédiate de cette manière de travailler est qu'elle fournit la possibilité de vérifier, en partie, la cohérence de l'analyse pédagogique et donc de valider la spécification du didacticiel (notamment sa complétude).

On peut notamment s'appuyer sur les éléments suivants :

* les fonctions pédagogiques sont réparties selon qu'elles sont assurées par l'enseignant en séance ou par le didacticiel.
  Certaines fonctions doivent être dédiées à la liaison entre les deux formes : par exemple, documentation sur l'activité de l'élève (diverses formes de traces, documents écrits,..), évaluation de cette activité, requêtes donnant accès à des résumés d'informations présentés en séance,...
  Cet aspect des didacticiels a été largement déjà validé lors de la réalisation du didacticiel FLE (cf. [Ada-83], [Pai-81], [Sch-83]). C'est pourquoi une faible place a été consacrée dans notre maquette.

* Le définition des leçons (dialogues) qui doivent constituer le didacticiel peut s'appuyer sur la typologie des moments et des approches pédagogiques considérées, (cf. II, aussi [Sco-83]).

L'approche systématique du travail de spécification pédagogique a des conséquences directes (que nous avons pu vérifier dans notre expérience) sur la réalisation informatique :

- pour décider de la solution E.A.O. dans un environnement particulier, on est alors en mesure de spécifier plus facilement les fonctions informatiques nécessaires et notamment d'isoler celles qui devront être rajoutées à cet environnement.

Dans notre cas, l'environnement choisi (MOSAIQUE) était issu d'une telle démarche à propos d'un enseignement différent, et nous avons dû le compléter par des logiciels spécifiques à notre didacticiel.

- l'existence d'une spécification pédagogique rélativement précise a facilité l'écriture des programmes du didacticiel (temps de programmation, fiabilité des produits). Elle a, par ailleurs, permis de minimiser les difficultés provenant du fait qu'une de nos hypothèses était de ne pas investir du temps sur l'adéquation du poste de travail à l'enseignement considéré (ceci devant être l'objet d'une maquette ultérieure)

Une conclusion générale relative à la problématique de production est qu'il est tout à fait pertinent de chercher à transférer dans le domaine de l'E.A.O., les techniques et les acquis du génie logiciel, ce qui avait été l'hypothèse du projet MOSAIQUE (cf.[Ada-83]). Pour que ce transfert soit effectif, il semble essentiel de développer plus précisement les méthodes d'analyse pédagogique pertinente dans un contexte E.A.O..

Pour terminer ce paragraphe, nous dégageons aussi quelques conclusions relatives à l'utilisation de l'E.A.O. dans l'enseignement de la programmation.

L'analyse pédagogique de la maquette nous a conduit à préciser clairement les fonctions pédagogiques qu'il est préférable d'assurer directement en cours (par exemple : présentation de notions générales, description des notations, construction magistrale d'algorithmes,...), notamment pour éviter l'écueil classique de l'E.A.O. "tourne-pages" où l'élève reste très passif.

Par ailleurs, nous avons pu définir des fonctions pédagogiques qu'il est préférable, au contraire, d'assurer par l'intermédiaire d'un ordinateur, parce qu'elles permettent l'individualisation du travail de l'élève dans une approche active et qu'elles s'appuient sur des moyens

de communication spécifiques qui ne sont pas à l'heure actuelle disponibles à des coûts raisonnables dans une salle de cours. Ceci est bien illustré par les leçons (dialogues) qui ont pour objectif d'exploiter différents modes de lecture d'un algorithme pour une bonne appréhension de la notion d'analyse descendante.

Notre maquette confirme, ainsi, qu'il semble judicieux d'utiliser l'E.A.O. dans l'enseignement de la programmation, et que l'E.A.O. doit être conçu comme une ressource complémentaire à un enseignement en cours.

On peut donc espérer une utilisation de l'E.A.O. permettant d'enrichir effectivement l'enseignement de la programmation.

IV.2 Quelques perspectives pour la poursuite de la recherche.

Les perspectives que nous développons ici, s'insèrent dans la démarche expérimentale de notre groupe de recherche que nous avons décrite au début de ce chapitre.

Elles concernent donc la définition des hypothèses de travail pour de nouvelles maquettes, l'intégration des résultats issus de la réalisation de ces maquettes pour la production de didacticiels en vraie grandeur, et enfin l'élaboration progressive d'un atelier de production de didacticiels.

La validation progressive d'une méthode de production nécessite encore, de s'intéresser à diverses maquettes.

Elles doivent notamment traiter de la phase d'analyse pédagogique, du rôle du poste de travail dans l'élaboration d'un didacticiel, de la recherche d'outils appropriés aux différentes phases de la production, et dans le cas particulier de l'enseignement de la programmation, de la recherche d'apports spécifiques à l'E.A.O..

Nous énumérons ici quelques lignes directrices qui doivent permettre de définir ces maquettes.

En ce qui concerne l'analyse pédagogique on peut orienter la recherche sur les trois axes suivants :

- affiner la typologie concernant les moments et les approches pédagogiques.
  Pour cela il faut réaliser des maquettes traitant de sujets divers dans des disciplines différentes, en suivant la même démarche de réalisation que dans le cas présenté ici.

- étudier les moyens d'expression de la spécification pédagogique.

Un principe important est d'adapter les formes d'expression à la culture des enseignants de la discipline. Ce travail est déjà entammé par la réalisation de maquettes en PROLOG pour l'enseignement du F.L.E. (cf.[Luc-84]).

- concevoir des outils d'aide à la spécification pédagogique basés sur des moyens d'expression précis. On pourra donc s'appuyer ici sur le travail réalisé dans le projet DIANE, en cherchant à approfondir les relations entre les fonctions pédagogiques nécessitées pour une application particulière et les éditeurs disponibles dans un système E.A.O.

En ce qui concerne le rôle du poste de travail dans l'élaboration d'un didacticiel, l'objectif général est d'étudier précisément les conditions dans lesquelles la production des didacticiels et leur exploitation par les enseignants et les élèves peuvent être adaptés au mieux dans le contexte de travail de chaque intervenant (nous parlons d'ergonomie du poste de travail). Cette étude concerne donc la communication homme-machine dans ses aspects materiels, physiologiques, psychologiques et pédagogiques.

* Nous parlons d'abord de la communication élève-machine (poste de travail élève).

On doit étudier les moyens d'expression fournis à l'élève en tenant compte des hypothèses pédagogiques sous-jacentes à l'apprentissage concerné.

On doit distinguer ici :

- les moyens d'expression qui font partie de la culture de l'élève.
  Leur appropriation doit être facile pour ne pas gêner l'apprentissage.

- les moyens d'expression directement liés à la stratégie pédagogique retenue et qui font partie intégrante du sujet enseigné.

Parmi les moyens de communication disponibles dans la technologie existante, on peut citer : clavier, touches fonctions, souris, écrans tactiles, crayons lumineux, tablettes graphiques.

On peut ainsi envisager de réaliser des maquettes dont l'objectif serait :

- d'étudier les relations entre une approche pédagogique et un ensemble de moyens de communication. Par exemple, dans le cas de notre maquette, on peut étudier diverses réalisations de la machine caractères selon qu'on a recours au clavier, à des touches fonctions, ou à un écran tactile.

- d'étudier des fonctions spécifiques du système élève facilitant sa communication avec la machine lorsqu'il utilise un didacticiel. Par exemple, dans le système MOSAIQUE, l'élève dispose d'un éditeur de réponses.

* Quant à la communication auteur-machine (poste de travail auteur), l'objectif est, à terme, de définir un atelier de production de didacticiels. La spécification d'un tel atelier doit s'appuyer sur les systèmes E.A.O. existants, et sur toutes les expériences de réalisation de didacticiels.
Toutes les maquettes dont nous parlons ici doivent donc être analysées dans ce sens.

Il est nécessaire qu'elles portent sur des disciplines différentes et qu'elles s'appuient sur des environnements de production variés (machines utilisées, forme d'expression des didacticiels).

Pour terminer ce paragraphe, nous examinons les suites à donner à notre travail en ce qui concerne l'enseignement de la programmation.
Nous pensons à la réalisation d'un didacticiel en vraie grandeur portant sur un cours complet d'algorithmique du niveau licence d'informatique.

Ce didacticiel doit permettre de fournir un complément E.A.O. à l'enseignement existant sur les points suivants : notion d'algorithme et d'analyse descendante, utilisation des schémas classiques de composition, manipulation d'assertions, structures de liste, notion de table, analyse récurrente et récursivité.

La réalisation s'appuiera sur les résultats du travail présenté ici, sur une maquette faite par H.MOITTA MONTE [Mon-84] sur les structures de données et sur une maquette portant sur la récursivité (qui reste à étudier) .

L'analyse pédagogique devra compléter l'acquis concernant l'enseignement en séance (cf. [LPS-83], [Sch-84]) par une étude systématique des fonctions pédagogiques nécessaires, par la définition de moments et d'approches pédagogiques adaptés aux sujets traités, et par la recherche de solutions E.A.O. adéquates, selon les principes développés dans cette thèse. On pourra par exemple, utiliser le système DIANE pour réaliser le didacticiel.

Par ailleurs, on devra élaborer un plan précis d'expérimentation et de validation dans un contexte réel d'enseignement.

L'état d'avancement de nos travaux ne permet pas encore d'envisager qu'un tel didacticiel apporte un soutien à la construction de programmes. Il faudra donc étudier cette question par une série de maquettes qui restent à définir.
On pourra se baser sur les expériences que nous avons faites autour de l'utilisation d'éditeurs syntaxiques tels que MENTOR [DHK-79a], [DHK-79b], [Zam-81].

IV.3 Conclusions.-

Nous concluons cette thèse en soulignant trois aspects de la contribution que nous pensons avoir apporté par notre travail.

La méthode consistant à étudier une maquette sur la base d'hypothèses précises se révèle fructueuse, et valide l'intérêt d'une démarche expérimentale pour permettre un développement cohérent et bien fondé des applications de l'E.A.O..

Nous avons confirmé l'intérêt qu'il y a, en matière E.A.O., à s'appuyer au maximum sur les acquis méthodologiques et techniques du génie logiciel. Ceci doit encore être exploité notamment pour aborder le problème de la spécification pédagogique des didacticiels et pour élaborer un futur atelier de production.

Enfin, notre maquette laisse augurer des apports spécifiques de l'E.A.O. à l'enseignement de la programmation.

La réalisation d'un didacticiel en vraie grandeur devrait ainsi enrichir cet enseignement.

ANNEXES

ANNEXE I.- Eléments de la notation Algorithmique .

Nous présentons brièvement une description des éléments de la notation algorithmique qui ont été utilisés dans ce travail.

Une notation algorithmique est conçue dans le but d'offrir aux concepteurs d'algorithmes :

- la possibilité d'exprimer des abstractions mises en évidence au cours de l'analyse (informations nommées, actions nommées) [Mor-83].

- des facilités d'expressions des algorithmes (schémas évolués de conception de programmes)

- l'indépendence par rapport à un langage de programmation.

- des facilités de spécification des abstractions manipulées.

a) Caractéristiques principales.

- Le concept d'information nommée :

C'est un outil qui permet de décrire une information et de s'y référer en faisant abstraction d'une partie de ses caractéristiques.

- Le concept d'action nommée :

C'est ce qui permet de décrire la solution algorithmique d'un problème et de s'y référer en faisant abstraction du processus engendré lors de son exécution.

- La notion d'univers :

C'est le regroupement en une même entité (l'univers) des informations nommées et des actions nommées qui les manipulent.

- La notion de type nommé :

C'est un outil permettant de faire abstraction de la représentation d'une information nommée lors de sa spécification.

- La notion d'intervalle nommé :

Cette notion est associée à la gestion d'une table algortihmique; elle sert à spécifier les valeurs possibles des indicatifs de la table et la relation entre une table et les indices qui permettent de la gérer.

- La notion de propriété :

C'est un outil permettant de spécifier, pour une information nommée donnée, les propriétés que doit respecter sa valeur.

b) Description de certains éléments :

- Litteraux et expressions :

On utilise les conventions classiques.

- Instructions :

1.- Affectation :

Le symbole est une fleche vers la gauche

<nom> <-- <expression>

2.- choix :

Cette instruction permet d'établir une relation entre un ensemble de cas et un ensemble d'actions. Les cas envisagés doivent s'exclure mutuellement et correspondre à tous les cas possibles.

choix

<expression logique 1> : <instruction 1>

. .
. .
. .

<expression logique n> : <instruction n>

fchoix

L'exécution d'une telle instruction consiste à exécuter le groupe d'instructions associé à l'expression logique qui prend la valeur vrai. Il y a donc une erreur si aucune des expressions logiques ne prend cette valeur.

Les formes habituelles d'instructions conditionnelles sont admises :

si <expression logique> alors <instruction 1>
sinon <instruction 2>
fsi

si <expression logique> alors <instruction> fsi

3.- Itération :

La forme principale d'itération est la suivante :

itérer

<instruction 1>

arrêt : <expression logique>

<instruction 2>

fitérer

Nous utilisons les schémas particuliers classiques : tantque et répéter (considérés comme des formes particulières du schéma itérer.

tantque <expression logique> faire
<instruction>
ftantque

répéter
<instruction>
jusqua <expression logique>

c) Mise en page :

- Un commentaire est enfermé par des accolades :

{ commentaire }

- Les mots clefs de la notation sont soulignés.

- Le ";" est délimiteur d'instruction indiquant une exécution en sequence. Il peut être omis lorsque se trouve en fin de ligne ou avant un mot clef.

- On réserve l'usage de majuscules pour désigner les objets qui ne sont pas définis dans le texte de l'algorithme.

Schémas du trataiment séquentiel.

a) Parcours d'une file.-

Type 1 (P1) {avec marque}

```
DEM
choix
 FDF : T_VIDE
¬FDF :
        INIT_V
        répéter
         VISITER (EC)
         AV
        jusqua FDF
        TERM_V
fchoix
```

Type 2 (P2) {sans marque}

```
INIT_A
choix
 FILE_VIDE : T_VIDE
¬FILE_VIDE :
              INIT_V
              DEM
              itérer
               VISITER (EC)
              arrêt : FDF
               AV
              fitérer
              TERM_V
fchoix
```

b) Parcours d'une file (variante de traitement)

cas où T_VIDE = INIT_V ; TERM_V

Type 1 (P1') {avec marque}

```
INIT_V
DEM
tantque   FDF faire
 VISITER (EC)
 AV
ftantque
TERM_V
```

Type 2 (P2') {sans marque}

```
INIT_V
INIT_A
choix
 FILE_VIDE :
¬ FILE_VIDE :
                DEM
                itérer
                 VISITER (EC)
                arrêt : FDF
                 AV
                fitérer
fchoix
TERM_V
```

ANNEXE II.- Production des Leçons.

La production d'une leçon implique deux types de travaux : l'un "sur papier" et l'autre sur l'ordinateur (cf.II.3.3).

Le travail "sur papier" concerne l'aspect définition d'un document leçon (cf II.3.3 a) et il est centré sur l'utilisation des formes suivantes :

- DL1 : forme représentant un écran de 24 lignes par 80 caractères.

  Utilisation : Définir la configuration des écrans.
  Définir certains aspects du déroulement des leçons.

- DL2 : forme à deux colonnes nommées : "Affichage" et "Commentaires".

  Utilisation : Ebaucher le déroulement de la leçon et commenter chaque situation pédagogique.

- DL3 : forme à trois colonnes nommées : "nom du message", "longueur de message", "description du message".

  Utilisation : Décrire tous les messages du dialogue.

- DL4 : forme presentant deux colonnes nommées : "nom de la zone", "définition".

  Utilisation : Décrire les zones à utiliser dans le dialogue.

- DL5 : "Information pédagogique".

  Utilisation : Description pédagogique du dialogue.

- DL6 : "Information technique".

  Utilisation : Description de certaines informations techniques.

La codification du dialogue (en PASCAL+) n'utilise aucun formulaire special.

Le travail sur l'ordinateur concerne l'intégration du code au système (cf.II.3.3 b). Il est centré en l'utilisation de l'éditeur de texte -KED- du materiel PLESSEY pour définir les programmes en PASCAL + (Transcription du code).

REFERENCES

Références.-

[AbS-84] ABRY, D; SCHOLL, M
Informatique et Français langue étrangère :
Aspects pédagogiques du projet MOSAIQUE à Grenoble.
CUEF - Université de Grenoble 3 - Janvier 1984.
aussi dans "Le français dans le monde" Juillet 1984.

[Ada-83] ADAM, JM.
Méthodes et outils pour la production de
didacticiels : l'environnement informatique
du projet MOSAIQUE.
Thèse 3ème. cycle - IMAG - 1983

[ADI-81] AGENCE DE L'INFORMATIQUE (A.D.I)
"Rapport E.A.O."
Paris - 1981

[ADI-83] AGENCE DE L'INFORMATIQUE (A.D.I.)
Système DIANE . Enseignement assisté par Ordinateur.
Paris - 1983

[ANA-84] ANNAGRAM
Lettres d'Anna gram 1 à 15.
1979-1984
Groupe de travail de l'AFCET. - 1984

[APS-81] ABRY,D; PAINVIN,S; SCHOLL,M.
Un didacticiel pour l'enseignement du français
langue étrangère.
Rapport de spécification pédagogique
INPG - IMAG - ULLG - CUEF - 1981

[Ars-80] ARSAC, J.
Premières leçons de programmation.
Cedic - Nathen - 1980

[Ars-83] ARSAC, J.
Les bases de la Programmation.
Dunod - 1983

[AHU-83] AHO,A.V.; HOPCROFT,J.E.; ULLMAN,J.D.
Data Structures and Algorithms.
Addison-Wesley - 1983

[BeB-83] BERLIOUX, P. et BIZARD, P.
Algorithmique
Construction, preuve et évaluation des programmes.
Masson - 1983

[Bee-69] BEER, S.
Prospects of the Cybernetic Age.
Survey of Cybernetics.
Illife - London - 1969

[BeF-82] BESTOUGEFF,H; FARGETTE,J.P.
Enseignement et Ordinateur.
Cedic - 1982

[BiC-84a] BIONDI,J; CLAVEL,G.
Introduction à la programmation :
Algorithmique et Langage.
Masson - 1984

[BoM-83] BOUSSARD,J.C; MAHL,R.
Programmation Avancée : Algorithmique et
structures des données.
Eyrolles - 1983

[BuB-79] BURTON, B.J.
An Investigation of Computer Coaching for
Informal Learning Activities.
Int. J. Man Machine Studies Vol. 11 - 1979

[ClB-84b] CLAVEL,G.; BIONDI,J.
Introduction à la programmation :
Structures de données.
Masson - 1984

[CoP-83] COUVERT,A; PEDRONO,R
AERO - Un système d'aide à l'enseignement et à la résolution de problèmes en Recherche Operationnelle.
Techniques et Science Informatiques.
AFCET V2. No.2 - 1983

[DDH-72] DAHL,O; DIJKSTRA,E; HOARE,C.A.R.
Structured programming.
Academic Press - 1972

[DHK-79a] DONZEAU-GOUGE,V.; HUET,G.; KHAN,G.; LANG,B.
Introduction au système MENTOR et ses applications.
IRIA - Laboria - 1979

[DHK-79b] DONZEAU-GOUGE,V.; HUET,G.; KHAN,G.; LANG,B.
The MENTOR Program Manipulation System.
IRIA - Laboria - 1979

[Dij-68] DIJKSTRA, E.
The Structure of the Multiprogramming System.
CACM - 1968

[Dij-71] DIJKSTRA, E.
A short introduction to the art of programming.
EWD 316 - Dept. of Mathematics - Eindhoven - 1971

[Dij-76] DIJKSTRA, E.
A discipline of Programming.
Prentice hall - 1976

[Duc-84] DUCRIN, A
Programmation.
Tome 1 : Du problème à l'algorithme.
Tome 2 : De l'algorithme au programme.
Dunod - 1984

[Enn-83] ENNALS, R.
Beginning micro-PROLOG.
Ellis Horwood - Heinemann
London - 1983

[EPI-83] ASSOCIATION ENSEIGNEMENT PUBLIC _INFORMATIQUE
Le Système LSE.
Dossier EPI No. 3 - Juin 1983

[Flo-67] FLOYD, R.
Assigning Meaning to Programs.
Proc. of a Symp. in Applied Math.
Vol 19, Math. Aspects of Computer Science
Providence, Rhode Island - 1967

[Gri-74] GRIES, D.
What should we teach in an Introductory
Programming Course ?.
TR 74 - 195
Cornell University - Dpt. Comp.Science - 1974

[Cri-81] GRIES, D.
The Science of Programming.
Spinger-Verlag - 1981

[Har-81] HART,S.(Editor)
The PLATO System and Language Study.
Publication of the Language Learning Laboratory
University of Illinois - 1981

[Hoa-69] HOARE C.A.R.
An Axiomatic Basis for Computer Programming.
C.A.C.M. Vol 12 - 1969

[IFI-75] IFIP - WCCE 75
2nd. World Conference and Computer in Education.
Ed. O.Lacarme and R. Lewis
North Holland Publishing Company
American Elsevier Publishing Company - 1975

[IFI-81] IFIP - WCCE 81
3rd. World Conference and Computer in Education.
Ed. R. Lewis and E. Tagg
North Holland Publishing Company - 1981

[INR-81] INSTITUT NATIONAL DE RECHERCHE PEDAGOGIQUE
Dix ans d'informatique dans l'enseignement
secondaire 1970 - 1980.
INRP - 1981

[Jac-75] JACKSON,M.A.
Principles of program design
Academic Press, London - 1975

[Knu-68] KNUTH, D.E.
The Art of Computer programming Vol.1.
Addison Wesley - 1968

[Lau-82] LAURENT, J.P.
Initiation à l'analyse et à la programmation.
Dunod - 1982

[Lcc-83] LUCCI, Alain
Système VOYELLES.
Manuel en préparation.
IMAG. - 1983

[Lcc-84] LUCCI, Alain
Programmation en logique et E.A.O. :
Méthodologie de conception de didacticiels.
travail présenté au Colloque Scientifique 84
Lyon - 1984

[Liv-78] LIVERCY, C.
Théorie des programmes :
Schémas, preuves, sémantique.
Dunod - 1978

[LPS-83] LUCAS, M.; PEYRIN, JP.; SCHOLL, PC.
Algorithmes et représentation des données :
Files, Automates d'états finis
Masson - 1983

[Luc-83] LUCAS, M.
Algorithmes et représentation des données :
Evaluation, arbres, graphes analyse de textes.
Masson - 1983

[LuS-75] LUCAS,M; SCHOLL,P.C.
Propositions pour une initiation à l'algorithmique.
Laboratoire IMAG - USMG - Grenoble - 1975

[MeB-78] MEYER,B.; BAUDOIN,C.
Méthodes de Programmation.
Eyrolles - 1978

[Moi-84] MOITA MONTE, H.
Thèse de docteur ingénieur.
USMG - INPG (en préparation) - 1984

[Mor-79] MORAT,P.
Construction méthodique de programmes : MEFIA.
Etude de quelques primitives de contrôle.
Rapport D.E.A. - INPG - Grenoble - 1979

[Mor-83] MORAT,P.
Une étude sur la base de la Programmation Algorithmique : Notation et Environnement de Travail.
Thèse 3ème. cycle - INPG - Grenoble - 1983

[Pai-74] PAINVIN,S.
Séminaire d'enseignement assisté par ordinateur.
IMAG - 1974

[Pap-80] PAPERT, S.
Computer and learning.
The Computer Age, MIT. Press - 1980

[Pap-83] PAPERT, S.
Mindstorms: Children, Computers and powerful Ideas.
Havester Press, Brighton - 1983

[PeS-82] PEYRIN, J.P; SCHOLL,P.C
Méthodes de programmation.
USMG. Institut de Programmation - 1982

[Pey-83] PEYRIN, J.P.
Programmation systématique.
Notes de cours.
UCV - Caracas - VENEZUELA - 1983

[Pyl-70] PYLYSHYN, Z.
Perspectives on the Computer Revolution .
Prentice Hall, Inc. 1970

[Que-83] QUERE, M.
Langages d'auteurs d'hier et d'aujourd'hui.
A Tout Crin pp. 23-31 - 1983

[RiR-81] RICHARD,C; RICHARD,D.
Initiation à l'algorithmique.
Collection DIA/editions BELIN - 1981

[Ros-67] ROSEN, S.
Programming Systems and Languages.
MC. Graw Hill - 1967

[Sch-78] SCHOLL, P.C.
Le traitement séquentiel : une classe de problèmes
et une méthode de construction de programmes.
Actes Congrès AFCET - Nov. 1978

[Sch-79] SCHOLL, P.C.
Vers une Programmation Systématique : Etude
de quelques méthodes, techniques et outils.
Thèse d'état - Grenoble - 1979

[Sch-84] SCHOLL,PC.
Algorithmique et représentation des données :
Récursivité et arbres.
Masson - 1984

[SCM-80] SCHOLL,PC.; CASERY,C.; MORAT,P.
Introduction à une notation algorithmique.
USMG -INPG - Grenoble 1980

[Sim-80] SIMON,J.C.
L'éducation et l'informatisation de la société.
Rapport au Président de la République
La documentation Française - 1980

[Sco-83] SCHOLL, M.
Etude de méthodes en E.A.O. du français langue étrangère.
Rapport de D.E.A.
I.M.S.S Grenoble 2 - 1983

[Sel-79] SELF, J.A.
Students Models and Artificial Intelligence.
Computers and Education, Vol. 3, No. 4 - 1979

[Smi-82] SMITH, C. (Editor)
Microcomputers in Education.
Ellis Horwood Publishers - 1982

[War--74] Warnier,J.D.
Logical Construction of Programs
Van Nostrand Reinhold Co. New York - 1974

[Wir-71] WIRTH, N.
Program Development by Stepwise Refinement.
CACM - V14 No.4 - 1971

[Wir-73] WIRTH, N.
Systematic Programming : An introduction.
Prentice hall - 1973

[Wir-76] WIRTH, N.
Algorithms + Data = Programs.
Prentice hall - 1976

[Zam-81] ZAMBRANO,J.
Sur un système expérimental de construction de programmes.
Rapport D.E.A
USMG - INPG - 1981

AUTORISATION de SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU le rapport de présentation de Monsieur P.C SCHOLL, Professeur

Monsieur ZAMBRANO Jésus

est autorisé à présenter une thèse en soutenance en vue de l'obtention du titre de DOCTEUR de TROISIEME CYCLE, spécialité "Informatique".

Fait à Grenoble, le 16 octobre 1984

Le Président de l'I.N.P.-G

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,