



HAL
open science

Une initiation à l'analyse descendante compte-rendu d'expérience

Catherine Geoffroy

► **To cite this version:**

Catherine Geoffroy. Une initiation à l'analyse descendante compte-rendu d'expérience. Colloque francophone sur la didactique de l'informatique, Sep 1988, Paris, France. pp.243-260. edutice-00373770

HAL Id: edutice-00373770

<https://edutice.hal.science/edutice-00373770>

Submitted on 7 Apr 2009

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.

**UNE INITIATION À L'ANALYSE DESCENDANTE
COMPTE-RENDU D'EXPÉRIENCE**

Catherine Geoffroy

**Laboratoire CREDI
Ecole Centrale des Arts et Manufactures
Grande Voie des Vignes
92295 - Chatenay-Malabry Cedex**

UNE INITIATION À L'ANALYSE DESCENDANTE COMPTE-RENDU D'EXPÉRIENCE

Catherine Geoffroy

Laboratoire CREDI
Ecole Centrale des Arts et Manufactures
Grande Voie des Vignes
92295 - Chatenay-Malabry Cedex

1. PRÉSENTATION GÉNÉRALE

a) Les objectifs

La conception d'un programme pédagogique fondée sur ses objectifs - par opposition au descriptif de son contenu qui liste les différents points abordés par l'enseignant - privilégie le travail accompli par l'apprenant : les notions et les compétences qu'il doit avoir acquies pour que le "contrat" pédagogique soit rempli. Cette optique sera développée en s'appuyant sur des cas concrets, observés pendant deux années d'expérience.

Les objectifs généraux du cours, conçus à l'origine par C. et P. RICHARD, sont que l'élève énonce le "problème" précisément, liste les informations utiles et les résultats attendus. Puis il devra concevoir une analyse générale, détaillée par affinements successifs, en français structuré, et enfin exprimer sa solution sous la forme concise d'un algorithme.

Le seul pré-requis nécessaire est une maîtrise suffisante de la langue française.

L'axe général de la formation dans laquelle doit s'intégrer le module influera sur l'importance relative des objectifs - non pas sur les objectifs eux-mêmes. Pour un stage, chacun sera spécifié par des critères plus ou moins nombreux, établis en tenant compte du temps imparti (5, 8 ou 10 jours), des "niveaux culturels" des participants, etc.

Ces objectifs s'organisent donc selon deux pôles : la conception d'une solution et sa communicabilité.

Communiquer précisément, c'est avant tout ne pas interpréter abusivement l'énoncé. C'est également construire un dictionnaire exhaustif des informations utilisées, non ambigu et non redondant, mais aussi penser au confort de l'utilisateur (messages clairs, saisies guidées). Bien entendu, il faudra être lisible (reporter l'analyse en

commentaires, choisir des identificateurs imagés, représenter visuellement l'imbrication des actions, ...).

Utiliser une méthode d'analyse descendante, c'est-à-dire partir d'une vue synthétique de la question et procéder par étapes, donc discerner des objectifs intermédiaires et structurer correctement leur enchaînement. Il faudra vérifier, si nécessaire à chaque étape, la cohérence de l'ensemble en simulant des exécutions à la main.

Enfin, l'apprenant devra comprendre et savoir manipuler les notions informatiques de base : variables simples et composées, instructions élémentaires (entrée, sortie, affectation interne), structures et opérateurs logiques.

La poursuite de ces objectifs entraîne certains "bénéfices secondaires" : une structuration de la pensée, le développement des aptitudes à l'organisation et, tout en améliorant l'autonomie intellectuelle, une plus grande efficacité du travail en équipe.

b) Différents contextes et publics

L'expérience présentée porte sur une trentaine de groupes (le plus souvent entre 10 et 30 participants), dans le cadre d'institutions très variées (formations initiales, universitaires ou formations continues, inter ou intra-entreprise), et avec des orientations différentes :

- . formations d'analystes programmeurs ou formation d'une équipe d'informaticiens à une méthode commune avant un projet ;
- . formations d'utilisateurs ;
- . sensibilisation aux problèmes de communication entre utilisateurs et informaticiens, ou formations aux nouveaux métiers "informatique et communication" ;
- . formations d'enseignants (futurs responsables d'atelier informatique dans leur établissement, ou préparation à l'intégration de l'outil informatique aux travaux pratiques des matières enseignées en classes préparatoires).

Le public en formation continue peut parfois être particulièrement hétérogène, et à tous niveaux : certains groupes ont réuni informaticiens expérimentés et débutants ; cadres, chercheurs, techniciens ; utilisateurs, décideurs et chômeurs aux objectifs, aux vécus très divers. De plus, les participants ont arrêté leur précédentes études depuis plus ou moins longtemps, les fourchettes d'âges étant le plus souvent très larges. Bien entendu, la difficulté des exercices et la rapidité de leur enchaînement sont déterminées par les niveaux (initial et attendu) des participants.

Si la motivation personnelle des participants est chose acquise en formation continue, celle des élèves en formation initiale peut varier en fonction de l'image sociale du groupe. L'existence de contrôles, sanctionnés ou non par un diplôme, plus ou moins reconnu, peut également influencer sur le climat dans le groupe.

Le rythme des interventions (par tranche d'une demi-journée minimum) modifie surtout les techniques utilisées pour obtenir une certaine pression dans le groupe. De véritables réflexes sont acquis lorsque les interventions sont groupées sur une semaine à temps complet - mais cette densité même peut parfois provoquer des phénomènes de saturation. En échange, lorsqu'elle s'échelonne sur une période plus longue -un à deux mois- l'intégration des notions abordées peut être consolidée par plus de travail hors cours, le risque étant, surtout si des cours de langage sont suivis parallèlement, de diluer l'impact du message (analyser avant de programmer).

Enfin, un nombre suffisant de participants (une dizaine) est nécessaire pour constituer un véritable groupe et permettre un travail en équipes. Dans ce cadre, nous verrons que l'hétérogénéité du public au niveau du savoir informatique constitue plus souvent une richesse qu'une difficulté supplémentaire.

c) le logiciel d'aide à l'enseignement ALADIN

Conçu et mis au point par C. et P. Richard, le logiciel ALADIN, dont la prise en main est extrêmement facile, permet aux apprenants de concrétiser immédiatement les notions abordées et de mieux admettre la rigueur exigée.

La phase d'"essai-erreur" sur ordinateur permet de se mettre dans le contexte d'un apprentissage naturel et, pour les débutants, de désacraliser l'ordinateur qui n'est plus une "boîte noire" un peu magique, mais une machine dont on peut maîtriser le fonctionnement.

ALADIN est composé d'un éditeur de texte très simplifié et d'un compilateur comprenant un langage structuré élémentaire : une structure alternative (SI...ALORS...SINON, le "sinon" est obligatoire) et une répétitive (TANT QUE...REPETE), trois instructions (LIS, ECRIS, EN...RANGE), des données de type entier, booléen ou caractère, simples ou composées - celles-ci devant être déclarées - et des opérateurs logiques, numériques et de type caractère. Impossible de se tromper : la difficulté ne réside pas dans le langage utilisé.

Surtout, ALADIN comporte tout un environnement de détection d'erreurs : erreur de structure, de syntaxe, mais aussi manque d'initialisation de variables, utilisation d'indices incorrects, etc.

Outre l'aspect ludique apporté par le contact avec la machine, Aladin apporte un renforcement, une fixation des réponses au fur et à mesure de leur apparition. Comme l'a montré Piaget, ce type d'apprentissage par interaction avec l'environnement, aboutit tout

autant à développer la créativité qu'à acquérir un sujet précis. D'autre part, une erreur n'étant que le signe qu'il reste du travail à faire pour que le programme tourne (l'ordinateur ne juge pas l'utilisateur), l'image que l'apprenant a de lui-même, sa confiance en lui ne sont pas entamées. Cette évaluation immédiate offre à chacun une réelle possibilité d'initiative, et l'enseignant peut alors se consacrer exclusivement aux problèmes d'analyse et d'organisation. Le rythme de l'apprentissage peut ainsi être individualisé, même dans un groupe important.

Enfin, le listing présente le travail effectué de manière plus lisible, sa correction est notablement plus rapide que celle d'un algorithme sur papier, et son évaluation est plus complète : les tests effectués, que l'élève est invité à commenter, peuvent également être imprimés.

2. APPROCHE PÉDAGOGIQUE

a) Mise en situation

Il n'existe évidemment pas de stage standard, mais certains moments-clefs jalonnent toujours le déroulement du module :

Le premier jour débute par un tour de table où chacun doit dire en quelques mots s'il a une expérience informatique, et laquelle. Cette prise de contact dynamique avec le groupe - et des participants entre eux - est la première opération de *sensibilisation aux problèmes de communication* et à la nécessité de conventions. C'est ainsi l'occasion de demander à préciser une information, clarifier une expression, ... Nous mettons au point des définitions communes (algorithme, langage informatique, ou ce qui se présente!). Ceci permet d'insister sur l'apport de chacun dans le groupe : les débutants ayant souvent un rôle de "naïf" qui impose aux expérimentés d'éviter tout jargon, et ceux-ci acceptant volontiers d'explicitier leur savoir. Il n'y a donc pas de hiérarchie établie entre novices et expérimentés, mais une certaine complémentarité.

Lors de la *présentation des objectifs*, il est souvent nécessaire d'insister sur l'utilité de la démarche adoptée : indépendance de l'analyse vis à vis d'un quelconque langage, modularité, évolution et adaptation facilitées, communication homme/machine, homme/homme plus fiables. De plus, il est bon d'annoncer la stratégie pédagogique employée et d'explicitier quelques principes qui la justifient. Concevoir et communiquer sont deux facettes d'un même processus qu'il ne peut s'agir d'acquérir comme un savoir technique, mais d'intégrer comme un savoir-faire, une structure de pensée. Bien entendu, ces principaux objectifs seront approfondis au fil des séances, mais la méthode est difficilement fractionnable et doit être abordée d'emblée dans son ensemble. Les notions de base, par contre, ne seront explicitées que lorsque le besoin s'en fera sentir. Par exemple, la définition des données composées (matrice, vecteur) sera expliquée après avoir laissé les

apprenants réfléchir sur un exercice nécessitant l'utilisation d'ensembles de ce type (cf. exemple en annexe c)

Après la *définition des différentes structures utilisées*, un petit exercice de recopie d'un texte volontairement imprécis (par exemple, des alternatives imbriquées où une action est située entre un bloc et un autre) a l'avantage de surprendre l'assistance - en général tout le monde se laisse prendre au piège et interprète le message à sa façon. Le problème de l'analyse est ainsi positionné au sein des problèmes de communication de documents écrits.

Puis nous nous attaquons au premier exercice - extrêmement dirigé, presque une *démonstration* - qui porte de préférence sur un thème de la vie quotidienne pour que tous puissent participer (par exemple, simuler le fonctionnement d'un distributeur automatique de billets). Le sujet est néanmoins suffisamment complexe pour pouvoir justifier le déploiement de la méthode.

L'application suivante, qui sera codée et entrée en machine, ne doit pas présenter, pour les novices, d'importantes difficultés d'analyse. Ils auront en effet à se familiariser avec le clavier, l'utilisation de disques, le fonctionnement général de l'ordinateur. Son sujet est donc choisi de manière qu'il soit possible d'en compliquer l'énoncé à loisir, et que chacun ait de quoi se nourrir !

Le **premier "passage en machine"** provoque généralement une réflexion sur les tests. Les débutants, prenant conscience des possibilités réelles de l'ordinateur, réalisent bien la notion de variable informatique et comprennent mieux les différentes instructions (il n'est pas toujours évident que les ordres de l'algorithme s'adressent tous à la machine, vue de l'intérieur).

Pour contrebalancer l'agitation - et l'inquiétude - régulièrement manifestées lors de ce premier contact avec le logiciel, l'accent sera à nouveau mis sur l'importance de la lisibilité de l'ensemble.

La suite du stage alterne travaux pratiques et travaux dirigés. Ce qu'apprennent les stagiaires résulte pour une grande part de l'activité qu'ils déploient eux-mêmes. Cette méthode "appropriative" est non seulement plus attrayante mais aussi plus efficace : l'apprenant s'implique et s'autonomise.

En **travaux pratiques**, les stagiaires peuvent choisir de travailler soit individuellement, soit en petites équipes de trois participants (3 est le chiffre idéal : à deux, on se fait vite un jargon, à quatre, s'ajoutent des questions d'organisation du travail au sein de l'équipe). Les changements de partenaires sont en tous cas vivement conseillés.

Les **travaux dirigés** ont eux pour fonction de remettre toutes les pendules à l'heure : les différentes solutions envisagées sont discutées, ce qui montre bien qu'une

solution n'est jamais qu'une des solutions possibles, pas la Solution. L'important est qu'à chaque étape, on puisse argumenter les choix qu'on a faits. Il ne s'agit donc pas à proprement parler de corrigés, mais de mises en relief des différentes phases de la méthode descendante.

Il est à noter que les supports de cours (document d'appoint sur les structures de contrôle, exemples d'algorithmes ou de programmes, ...), ont plus pour fonction de rassurer que d'informer (voir annexe d).

Un exercice de synthèse un peu particulier (travail inter et intra équipes), le "4x4", se déroule sur deux demi-journées (ou une journée entière) : le groupe se répartit en un nombre d'équipes multiple de trois, chaque sous-équipe reçoit un projet différent qui doit être réalisé en fin de journée. Mais, à un instant T, prévu à l'avance (environ deux heures), on fait comme si l'équipe devait s'en aller ; chacune passe alors son projet, en l'état, à l'équipe suivante - la règle du jeu consistant à ne transmettre que les informations écrites. Les projets tournent entre les équipes quatre fois dans la journée, et ceux qui ont mis au point le cahier des charges récupèrent ce projet à sa dernière étape. Ils sont alors à même de récapituler les options prises à chacune de ses phases, de repérer les difficultés que d'éventuelles imperfections ou incohérences ont pu entraîner tout au long de son évolution. Cet exercice ne manque jamais de convaincre de l'intérêt d'une bonne documentation.

Ce que l'on touche également du doigt lors de cette journée, toujours éprouvante, est qu'il n'est pas question de s'empêcher d'interpréter, mais de savoir qu'on interprète toujours (nous n'avons pas tous les mêmes évidences). De ce fait, les ambiguïtés sont transformées en questions dont on peut délibérer, et l'on peut déterminer qui, de l'informaticien ou de l'utilisateur, peut décider des options à prendre.

Le dernier jour, nous faisons une synthèse et un bilan de l'enseignement : chacun doit se resituer par rapport aux objectifs déterminés le premier jour, évaluer le chemin qu'il a parcouru. Pour éviter les frustrations, mieux vaut veiller à ce que chacun ait entièrement mis au point au moins un exercice.

Enfin, un questionnaire ouvert (cf. exemple en annexe e) permettra d'évaluer le cours - à chaud - et de recueillir critiques et suggestions.

Les exercices et la méthode d'enseignement ont été mis au point par Patrice Richard.

b) Difficultés couramment rencontrées

*"Les méthodes, ... sont l'essentiel,
et aussi les choses les plus difficiles, celles qui ont
le plus longtemps contre elles les habitudes et la paresse."
F. NIETZSCHE (l'Antéchrist)*

L'objet du cours consiste en une méthode de travail mettant en oeuvre des notions particulières. L'acquisition d'une méthode se manifeste surtout au niveau des attitudes et démarches adoptées par les apprenants, et les questions soulevées portent autant sur la compréhension et l'application des notions que sur l'adhésion aux principes de la méthode.

Les élèves ayant déjà une expérience de programmation ont souvent conçu, après de long efforts, des procédures de construction de solutions qui sont difficiles à remettre en cause. Mais ils agissent à partir de bases plus ou moins implicites, et quand les résultats obtenus ne sont pas ceux escomptés, ils ont recours - plutôt que de réviser leurs prémisses - à des actions rectificatrices ponctuelles, fondées essentiellement sur un certain "feeling". Lorsqu'une confrontation directe, à base d'arguments logiques, se révèle inefficace, insister risque d'accroître la résistance. L'ensemble des connaissances que l'élève a déjà acquises prend à ses yeux une grande importance et lui masque sa démarche intellectuelle elle-même. L'enseignant ne doit pas alors négliger cet acquis (par exemple, un ordinogramme sera accueilli comme un bon outil de communication). L'important est plutôt d'amener l'apprenant à décrire ce qu'il fait, de diriger sa réflexion sur ses propres méthodes de travail. Dans ce domaine, il suffit souvent d'amorcer le changement, sans nécessairement résoudre toutes les difficultés.

Les débutants en revanche, risquent le découragement devant l'avalanche de détails et de concepts inconnus, et admettent plus volontiers certains principes qu'ils assimilent comme autant de repères dans un univers nouveau.

D'une manière générale, il sera bénéfique d'inciter chacun à s'intéresser au travail accompli par les autres.

Tous les apprenants ont du mal à avoir une vue synthétique du problème et à hiérarchiser ses éléments (déterminer ce qui est essentiel, ce qui est du détail). Combien de fois rencontre-t-on en commentaire "comparaison", au mieux en précisant ce qui est comparé, rarement en disant pourquoi on veut le faire ! Cette difficulté est parfois augmentée d'une certaine réticence à accepter d'écrire un premier jet qui peut s'avérer erroné.

Elle peut être également accentuée au début, pour les débutants, par l'ignorance même de ce que l'on cherche à obtenir (qu'est-ce qu'un algorithme, quand doit-on arrêter d'analyser). Il arrive aussi qu'à l'énoncé d'un exercice, se superposent des contraintes données dans le cadre d'un exercice précédent, peut-être tout simplement par désir de le rattacher à quelque chose de connu.

Un autre point d'achoppement régulier est l'expression correcte des conditions de répétitives (surtout lorsqu'elles sont composées ou énoncées de façon négative). Il est probable que la principale difficulté consiste en fait à déterminer le niveau logique auquel appartient chaque question, la tendance générale étant de confondre des conditions de structures emboîtées.

L'application de la célèbre stratégie de "moindre compromission" (Sacerdoti) - ne prendre une décision que lorsqu'on ne peut plus faire autrement - et le travail de réflexion sur le dictionnaire des données (décrire le sens et le rôle de chacune, vérifier qu'il n'y a qu'une donnée par information utile et qu'une information manipulée par donnée) se sont toujours révélés efficaces pour rectifier certaines confusions.

Dans tous les cas, une même erreur pouvant résulter de l'interaction de nombreux facteurs situés à des plans différents (incompréhension, mais aussi saturation, blocage, ...), il faudra pour l'interpréter tenir compte de son contexte (qui l'a faite, et quand).

c) éléments de solutions

Le système d'interactions dans le groupe peut constituer un véritable levier pour modifier - ou maintenir - un comportement, instituer pour norme une certaine exigence de qualité, de rigueur. L'activité de l'enseignant consiste ainsi non seulement à gérer la progression de chaque apprenant par rapport aux objectifs, mais aussi les relations que tous les participants entretiennent entre eux pour accomplir les tâches définies.

Dès le premier jour, changer la disposition des tables dans la salle de manière que personne ne tourne le dos à quelqu'un (ou le moins possible), tend à suggérer une redéfinition des pôles traditionnels de l'enseignement (prof/élève) : les apprenants ne sont plus seulement destinataires mais aussi acteurs de l'action pédagogique, l'enseignant participe aux activités et son rôle consistera plutôt à animer et conseiller l'équipe.

Toute intervention débute (ou se termine) par des remarques générales, adressées au groupe, et destinées d'une part à mieux faire prendre conscience de ce qui est attendu, d'autre part à contribuer à valoriser aux yeux du groupe la qualité du travail accompli.

Parfois nécessaires, des réunions de régulation (1/4h de réflexion en équipes de 3 sur les questions suivantes : quel travail vous a le plus satisfait, pourquoi, quel travail vous a le moins satisfait, pourquoi ; puis 1/2h de comptes-rendus au groupe et discussions) aideront à recadrer l'inquiétude que certains peuvent éprouver : celle-ci apparaît alors

comme le signe d'une prise de conscience des questions soulevées, non plus comme le symbole d'une impossibilité dont les autres seraient témoins et juges. Ces réunions rendent également plus explicite le degré d'investissement de chaque participant et facilitent donc l'établissement de normes dans le groupe.

Lors de cours sur un langage de programmation ou sur un SGBD (dBase) donnés ultérieurement, nous avons pu éprouver les avantages de l'enseignement par projet : il permet à chacun de faire quelque chose de différent, tout en gardant à l'esprit que son module devra s'intégrer dans un tout cohérent. La définition du sujet et une première analyse sont élaborées en commun, chaque partie à réaliser est confiée à un sous-groupe, la réintégration se fera à nouveau avec le groupe entier. Le scénario pédagogique du projet reproduit ainsi physiquement dans la classe la modularité du programme.

3. EVALUATION

Les listing, documentations, et tout travail remis par écrit, donnent lieu à des commentaires par objectif, destinés à renseigner l'apprenant (ou l'équipe) sur les difficultés rencontrées, l'écart qui le sépare encore des objectifs définis.

De plus, tout au long du stage, le travail en équipe aura permis à chacun de comparer son travail à celui des autres, de se positionner au sein du groupe et ainsi de mieux évaluer ses performances individuelles. Cette capacité à s'auto-évaluer fait partie intégrante de l'apprentissage.

Lorsqu'un exercice doit être noté, un repère sur une échelle à 3 ou 5 niveaux est suffisant pour estimer si la capacité peut être considérée comme acquise, (la mesure qui permet de déterminer que l'objectif est atteint reste très subjective). La plupart des objectifs définis sont en effet peu quantifiables, même si les critères sont déterminés de la manière la plus concrète possible. S'il ne s'agit pas d'être objectif, comme Patrice PELPEL le met bien en relief, il faut tout de même être équitable ! En outre, il est nécessaire de veiller à ce que la fonction pédagogique de l'évaluation ne soit pas "contaminée" par la fonction institutionnelle du contrôle, les élèves voulant surtout réussir l'examen.

Des corrections régulières permettent également d'évaluer la manière dont le message a été reçu, de recenser les types d'erreurs et de blocages, de repérer plus systématiquement les points compris et incompris, et ainsi d'apporter certains éléments de correction, d'adapter la prochaine intervention.

Pour chaque exercice, une grille d'évaluation constituée à partir des objectifs détaillés permettra d'uniformiser les corrections. Ce travail systématique favorise une vision synthétique - pas toujours évidente dans le feu de l'action - et permet donc des remarques particulières plus pertinentes, notamment en ce qui concerne l'évaluation de la

progression de chacun. Enfin, au-delà de la compréhension de la méthode, on pourra aussi remarquer le degré d'intérêt qu'elle suscite : du refus à l'acceptation, ou même l'intégration des normes de rigueur.

Ces observations ont tenté de rendre compte de la complexité de cette situation d'apprentissage (interactions de facteurs sociaux, psychologiques, culturels, cognitifs, affectifs), en cherchant à mettre en valeur l'importance des relations développées entre les membres du groupe pour instituer certaines normes de comportement. De plus, l'entraide mutuelle permet aux apprenants de mieux évaluer leurs performances individuelles et bien souvent d'éviter les situations de rejet.

4. ANNEXES

Annexe a : liste des stages effectués

. Formations d'informaticiens (analystes programmeurs) :

FC de la faculté de Villetaneuse - préparation au DUT informatique, groupes d'environ 30 stagiaires

formation intra-entreprise préalable à l'embauche, une dizaine de participants

AFC-EPF - stages de 9 mois donnant lieu à une attestation, groupes d'en général 25 stagiaires

Thomson - formation d'une équipe avant projet, une vingtaine d'informaticiens

. Formations d'utilisateurs :

ECP-FC - formation intra du personnel administratif ; stages inter-entreprises réunissant un public aux objectifs très divers, une dizaine de participants

AFC-EPF stage de 3 mois, groupe d'une vingtaine de stagiaires

Sagittaire - stage de 3 mois, groupe d'une quinzaine de stagiaires

. Communication utilisateurs/informaticiens :

AFP, UCB - stages intra-entreprise, groupes composés pour moitié d'utilisateurs et informaticiens, avec pour objectif que chacun fasse un pas vers l'autre pour mieux se comprendre ;

FC de la faculté de Nanterre - stage "concepteurs médiatiques", une quinzaine de personnes ;

FC de la faculté de Villetaneuse - stage "informatique, enseignement et communication", une vingtaine de stagiaires

. formations initiales :

ECP 1ère année - 40 heures, 360 élèves

faculté de Villetaneuse - DEUG Communication 40 heures, 60 étudiants

. formations d'enseignants :

Académie de Versailles - une trentaine d'enseignants futurs responsables d'atelier informatique dans leur établissement

ECP-FC - intégration de l'outil informatique aux travaux pratiques des matières enseignées en math sup. et math spé. stages d'une semaine, une trentaine de profs de math, physique et chimie

. dictionnaire des sigles utilisés

FC : Formation Continue

AFC-EPF : Association pour la Formation Continue de l'Ecole Polytechnique Féminine

ECP : Ecole Centrale des Arts et Manufactures (ou Ecole Centrale "de Paris")

Annexe b : bibliographie

. Pédagogie générale

Pepel patrice : *se former pour enseigner*, bordas 1986

. Animation de groupe

Jaoui gysa et gourdin marie-claude : *transactions* 1982

Johnson lois v. Et bany mary a.: *Conduite et animation de la classe*, dunod 1974

. Algorithmique

Richard chantal et patrice : *programmation, initiation à la programmation méthodique*, belin 1984

Richard chantal et patrice : *initiation à l'algorithmique, 85 exercices corrigés*, belin 1981

Annexe c

Questionnaire ALADIN
IEC janvier 1988

* Formation(s) précédente(s) o Age :
précisez en quelle(s) année(s)

.....
.....

* Avez-vous une autre expérience informatique ? (laquelle ?)

.....
.....

* Le contenu du module ALADIN était-il dans l'ensemble :

| _____ |
ennuyeux très intéressant

* Précisez ce qui vous a le plus intéressé(e) :

.....
.....
.....
.....

* Précisez ce qui vous a le moins intéressé(e) :

.....
.....
.....
.....

* Quels(s) autre(s) thèmes auriez-vous aimé aborder ?

.....
.....
.....

* Avez-vous rencontré des difficultés ?

| _____ |

aucune

beaucoup

Donnez des exemples :

.....
.....
.....
.....

* Que pensez-vous de la répartition du temps entre cours, travaux dirigés et travaux pratiques ?

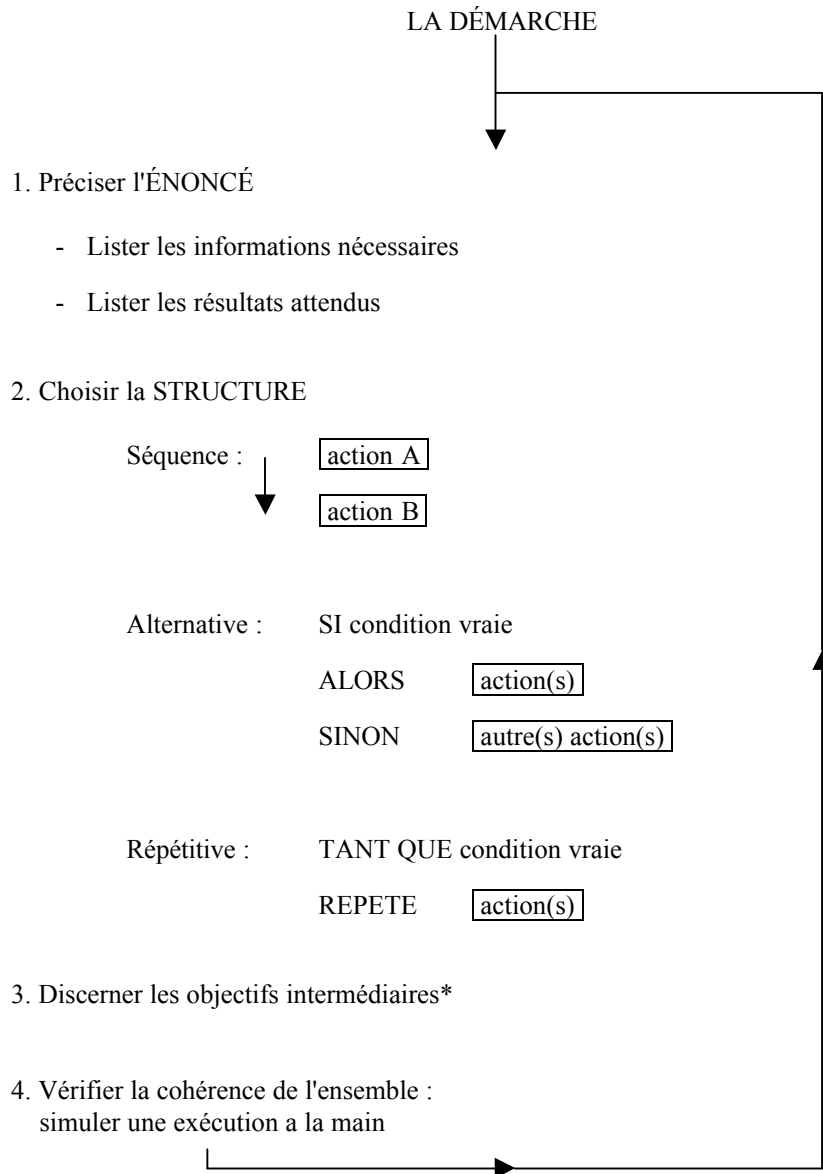
.....
.....
.....
.....

* Que pensez-vous de l'enchaînement des exercices ?

.....
.....
.....
.....

* Commentaires libres

.....
.....
.....
.....

Annexe d

* Chaque sous-problème défini constitue l'énoncé d'une étape suivante.

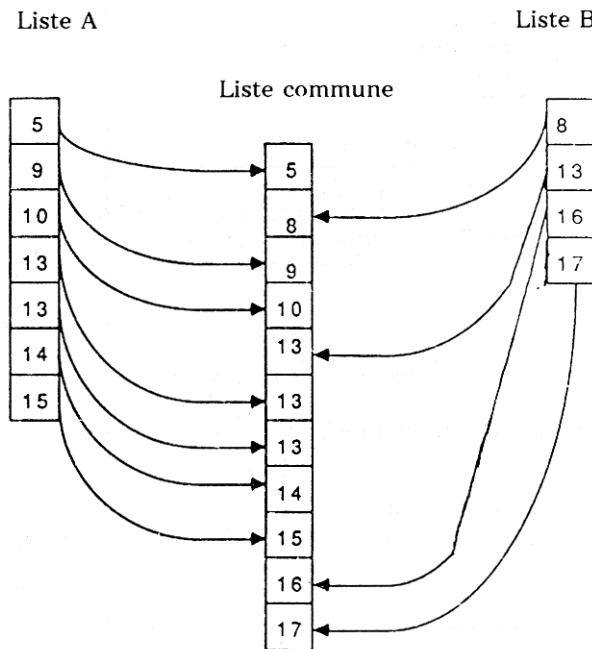
Annexe e

LA COMPÉTITION SPORTIVE INTERNATIONALE

Construire un algorithme qui permette, à partir de deux listes de temps records obtenus par des athlètes de deux pays différents, d'établir une liste commune classée dans l'ordre croissant.

Les deux listes initiales seront saisies dans l'ordre croissant, il vous suffira donc pour confectionner la liste commune de choisir alternativement dans l'une ou l'autre liste initiale.

Exemple :



Enoncé distribué en fin de TP (DUT informatique) pour la séance du lendemain.