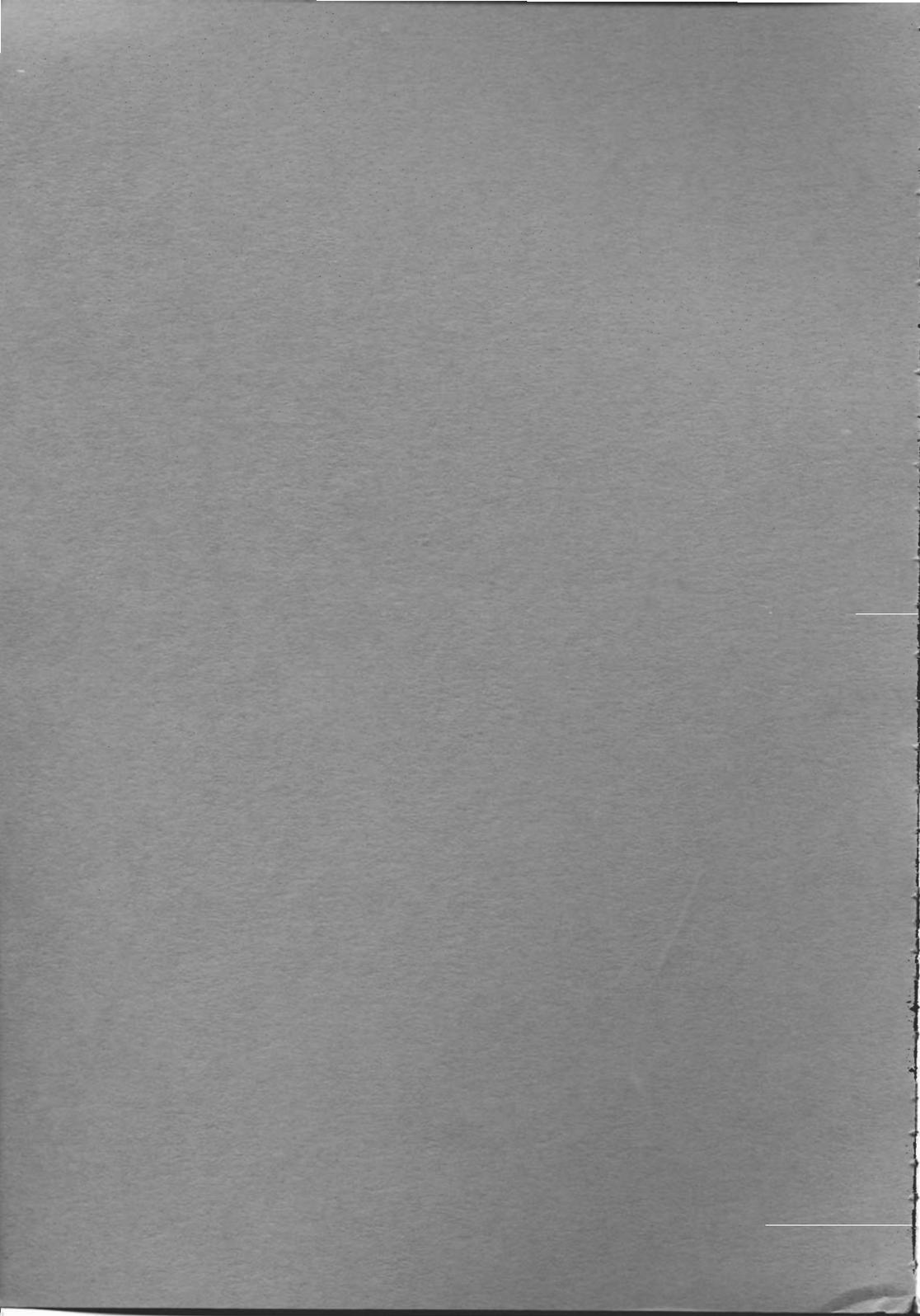


Specif n°12

mai 1990

SOMMAIRE

- Compte rendu de l'A.G. (C. CARREZ)
- Bilan des Commissions de SPECIF
- CNJ Informatique (B. LORHO)
- Protocole d'utilisation de logiciels
- Recherche fondamentale en informatique (M. NIVAT)
- L'informatique dans les premiers cycles scientifiques (M.LUCAS)
- Divers.



SOMMAIRE

- Compte rendu de l'A.G. (C. CARREZ)	3
- Bilan des Commissions de SPECIF.....	9
- CNU Informatique (B. LORHO).....	47
- Protocole d'utilisation de logiciels	53
- Recherche fondamentale en Informatique (M. NIVAT).....	63
- L'Informatique dans les premiers cycles scientifiques (M.LUCAS).....	77
- Divers.....	169

CONSEIL D' ADMINISTRATION DE SPECIF
(1990)

Anciens Présidents : PAIR C. (1986-1988)
COMYN G. (1988-1989)

Président : CARREZ C.

Vice-Président : ARNOLD A.

Membres du C.A. : BARTHET M.F., (Bureau)
BESTOUGEFF H., (Bureau), Secrétaire
BOYAT J.
CHRETIENNE P., Commission Personnel
COT N.,(Bureau) Bulletin
DE SABLET G., Commission Matériel/Logiciel
FAYARD D., Commission Enseignement
HERMAN D.
HERVIER M.
LESCANNE P., Commission Recherche
LUCAS M.
MOSSIERE J., (Bureau)
RENARD G., Relations internationales
RICHIER J.L., Communication interne
ROUSSEAU M.
SCHNEIDER M., (Bureau) Trésorier
SCHOLL P.C., (Bureau)
SIROUX J.
STEEN J.P.
TOURNIER E.
TREHEL M., (Bureau)
VIGNOLLE J.

Bulletin SPECIF : Co-éditeurs : BESTOUGEFF H. , COT N.

Adresse : Bulletin SPECIF
N.COT
EHEI
45, rue des Saints-Pères
75006 PARIS

COMPTE-RENDU DE L'A. G

COMPTE-RENDU DE L'A.G. C. CARREZ

Voici un compte-rendu de l'Assemblée Générale de SPECIF. Il y a d'une part un résumé de la présentation de Carrez, puis les notes qu'il a prises sur les interventions de Finance et Lorho, les votes de l'AG et la structure du CA.

Intervention de Christian Carrez (Vice-Président)

1 Convention avec le SNPLM

La convention entre le SNPLM et le Ministère est signée. Ce n'est pas extraordinaire, mais cela a le mérite d'exister. Elle tente de faire dialoguer les éditeurs et les Universités, de façon à avoir une meilleure compréhension mutuelle des problèmes. En particulier, elle propose que les éditeurs désignent un "correspondant" Education Nationale, à qui nous pourrions nous adresser pour étudier des problèmes précis qui se poseraient dans l'utilisation de certains logiciels dans nos enseignements.

Remarque: les éditeurs s'étonnent de l'absence de commandes en provenance des universités. Qu'en est-il exactement? La réponse semble multiple:

- les logiciels qui ont été achetés continuent à être utilisés, et nous n'avons pas besoin de les renouveler si aucun matériel supplémentaire n'est acheté.
- les enseignants trouvent d'autres solutions à base de mini-ordinateurs et abandonnent les solutions micros.

2 Suppression de la division informatique

C'est la conséquence de la contractualisation. Dans le cadre d'une approche "direction par objectif", au niveau des Etablissements, le Ministère considère qu'aucune discipline ne peut être traitée à part. La conséquence négative pour notre discipline est qu'il n'y a plus un service du Ministère qui prenait à coeur de défendre la discipline lors des discussions budgétaires, des créations d'emplois, etc...

3 Secteur SPI de la DRED

Finalement, le secteur n'a pas été supprimé, sans doute du fait des inquiétudes exprimées par l'ensemble des partenaires, dont SPECIF, le club EEA, les présidents de sections du CNU, etc... Actuellement, ce secteur est dirigé par René Castagnet (électronicien d'Orsay). Jean-Pierre Finance y est consultant.

4 Journée organisée avec le SYNTEC

Le SYNTEC organise avec plusieurs partenaires, dont SPECIF, un colloque de deux jours les 20 et 21 Mars 1990 au CNIT sur les problèmes de la formation des informaticiens. En complément de ce colloque, devrait être annoncé la constitution d'une mission d'observation des emplois et de la formation en informatique. Cette mission aurait la forme d'une association du type loi 1901, subventionné par les Ministères concernés et par les entreprises. Un comité exécutif serait créé dans lequel SPECIF serait représenté.

5 Crédits de Recherche

Il semble que les crédits de recherche seraient à l'avenir attribués soit aux Formations Doctorales, soit aux équipes CNRS. La notion d'équipe recommandée, "qui recouvre le pire et le meilleur" serait supprimé. Il est évident que cela a des implications pour notre discipline, en particulier pour les petites équipes dispersées dans de nombreux Etablissements. Les postes seront de plus en plus difficiles à pourvoir dans ces endroits.

6 Organisation de SPECIF

Il est apparu nécessaire d'améliorer le mode de fonctionnement de SPECIF. Il s'agit de créer une véritable communauté agissante, de multiplier les initiatives responsables, etc... Je proposerai au CA qu'un de ses membres soit plus particulièrement chargé des contacts avec les correspondants.

Notes prises sur l'intervention de Jean-Pierre Finance

Il y a un véritable besoin d'interaction forte avec la communauté des informaticiens. Le Ministère comporte entr'autres, les Directions de la Recherche, de l'Enseignement Supérieur, du personnel, des moyens et planification, etc... La Direction de la recherche comporte 6 Directions Scientifiques, dont le SPI où l'Informatique est représentée par J.P. Finance. La Direction de l'Enseignement Supérieur n'a pas de représentant informaticien pour le moment. Ceci explique peut-être que l'informatique ait été oubliée dans la circulaire sur les ingénieurs Decomps.

La DRED soutiendra les unités associées au CNRS et les formations doctorales.

Un Comité Scientifique international a été créé. Jean Vuillemin en fait partie.

Une commission Michel Demazure vient d'être créée, dont le but est de proposer une politique des moyens informatiques. Laurent Kott et Robert Cori en font partie.

En conclusion, il y a une volonté de banaliser l'informatique. La question posée est comment montrer les problèmes de notre discipline. Il faut se battre ensemble pour cela. Il est nécessaire de continuer les efforts d'équipement et de recruter des personnels, ATOS, en particulier. La communauté doit être facilement mobilisable pour défendre ces problèmes.

Notes prises sur l'intervention de Bernard Lorho

Lors des avancements, le CNU a tenu compte des informations de gestion administrative, et non plus seulement des aspects recherche.

Pour les recrutements, les dossiers des candidats ont été nettement améliorés, ce qui a facilité le travail. Une amélioration du guide est prévue, sur les mêmes idées. Environ 30 % des postes n'ont pas été pourvus, mais CE N'EST PAS NOTOIREMENT PLUS IMPORTANT QUE DANS D'AUTRES DISCIPLINES. La carte géographique a beaucoup d'importance, et la politique des primes accroîtra encore ce phénomène. La différence 24.1 et 24.2 a de moins en moins de sens. Il est recommandé de publier en 24.00, et de mettre un intitulé plus précis sur le poste.

Les jours du CNU sont comptés. Le prochain concours aura lieu comme le précédent. Après cela, il est probable qu'il sera remplacé par une instance locale (régionale ou académique), qui compléterait une instance nationale. Transformations, certains recrutements et certaines promotions seraient régionales. Certains recrutements ou promotions seraient faits sur un contingent national.

SPECIF devrait demander d'arrêter de mettre des postes dans des endroits où ils ne seront pas pourvus. Il devrait également s'interroger sur la carte géographique. Enfin il y a une certaine incompatibilité entre les politiques de la DSUP et de la DRED.

VOTES

Quitus a été donné au conseil d'administration pour sa gestion.

Conseil d'Administration:

Les membres sortants candidats au renouvellement sont les suivants:

Cot, Boyat, Lucas, Renard, Richier, Vignolle

Les membres sortants non candidats au renouvellement sont:

Comyn et Dussauchoy

Nouveaux candidats en remplacement: Arnold et Steen

Membre démissionnaire: Rueher,

candidat proposé en remplacement pour un an: Hervier

Les candidats sont élus à l'unanimité.

Résultat de la réunion du Conseil d'Administration:

Président: Carrez

Vice-Président: Arnold

Secrétaire: Bestougeff

Trésorier: Schneider

Bureau: Cot (Bulletin), Richier (Communication interne), Trehel (enseignement),

Mossiere (recherche), Barthet (matériel/logiciel), P.C. Scholl (personnel).

Responsables: Fayard (enseignement),

Lescanne (recherche),

de Sablet (matériel/logiciel),

Chretienne (personnel),

Renard (relations internationales)

**BILAN DES COMMISSIONS
DE SPECIF**

Commission Matériel /Logiciel (G. DE SABLET)

. Réunion du 18 janvier 1990

Commission Enseignement (D. FAYARD)

. Rapport d'activités 1989

Commission Recherche (P. LESCANNE)

. Rapport SNOWBIRD 1988

. Primes d'encadrement doctorales

Commission Personnel (P. CHRETIENNE)

. Enquête SPECIF

COMMISSION MATERIEL ET LOGICIEL

(G. DE SABLET)

Adresser votre réponse à :

G. de SABLET
IUT PARIS V
Département informatique
143 av. de Versailles
75016 PARIS

18 mars 1990

COMMISSION MATERIEL ET LOGICIEL

Compte-rendu de la commission du 18 janvier

Présents: C. APERGHIS (Marseille), J.-P. RICHIER
(Grenoble), G. de SABLET (Paris)

Excusés: Y. AHRONOWITZ, A. QUERE, ROUINNON, M. ROUSSEAU.

1) RESULTATS DE L'ENQUETE:

L'enquête sur l'utilisation de matériels et logiciels pour l'enseignement de l'algorithmique a permis de collecter 24 réponses; ces réponses proviennent de:

- 6 licence-maîtrise,
- 7 IUT dont 5 ne sont pas des départements informatiques,
- 7 écoles d'ingénieurs ou assimilés,
- 3 MIAGE
- 1 DESS

-- Langages: C a été cité 9 fois et Pascal 12 fois, bien que n'étant pas concernés par le questionnaire; Lisp 6 fois, Prolog 8 fois, Modula 3 fois, Ada 7 fois, Eiffel 2 fois; Unt été cités une fois; ML, C++, Smalltalk, hypercards, Loops, Scheme, Oblogis, Elixir et Pluss.

-- Matériels: Les matériels cités sont: PC (IBM ou autres) 10 fois, DEC 5 fois, SUN 3 fois, Bull 2 fois, Gould 1 fois, Apollo 1 fois et Apple (Mac) 1 fois.

Beaucoup de réponses mentionnent l'utilisation de versions Turbo sur PC.

-- **Compétences:** certaines compétences particulières semblent se détacher: Eiffel à Nice (ESSI et Mlab), Scheme à Bordeaux (Bordeaux1) et Modula2 à Nice.

La commission attendait des résultats plus exploitables; c'est-à-dire qu'ils auraient pu permettre de fournir à l'ensemble de la communauté des résultats d'expérience d'enseignement sur des langages récents. Il semble que l'enseignement de l'algorithmique nécessite un investissement trop important pour pouvoir coller aux innovations, et que, par ailleurs, les langages classiques apportent suffisamment d'outils.

En revanche, un certain nombre de commentaires devraient pouvoir intéresser nos collègues de la commission enseignement, auxquels nous sommes prêts à communiquer les réponses reçues.

2) DIFFUSION D'INFORMATIONS: La commission s'est penchée une fois de plus sur une organisation permettant de faire bénéficier ceux qui le souhaitent de l'expérience des autres. Plusieurs moyens sont envisagés: lecture régulière et compte-rendu du mini-bulletin du CNRS, constitution d'un groupe spécial de news sur le réseau F-NEI. Malheureusement, ces solutions nécessitent l'emploi d'un personnel qualifié à temps plus ou moins partiel, il faudrait alors envisager d'aller voir du côté des centres de calcul interuniversitaires, mais pour l'instant aucune solution satisfaisante n'apparaît.

3) RECHERCHE et DEVELOPPEMENT: Après avoir observé que certains développements pouvant intéresser nos collègues s'effectuaient dans les Universités, sans toutefois permettre l'obtention d'un produit fini (stages de DEA, projet de fin d'études et autres), nous nous sommes penchés sur les moyens à mettre en oeuvre pour permettre le développement de produits libres de droits (comme pour la FSF). Le ministère sera consulté sur ce point. (Une entrevue de G. de SABLET avec MM. CELANIRE et JACQUEMARD (NEPENTE) a permis d'évoquer ce problème, l'accueil de la proposition n'a pas été défavorable, mais les interlocuteurs entrevoyaient mal la manière dont le Ministère pourrait agir pour soutenir une telle action.

4) FONCTIONNEMENT des MOYENS INFORMATIQUES DANS LES UFR:

Compte tenu du faible nombre de techniciens et ingénieurs mis à la disposition des UFR pour faire fonctionner les matériels informatiques qui y sont installés, ce sont les enseignants-chercheurs qui sont le plus souvent en charge de les faire fonctionner. Cette situation est particulièrement néfaste pour ceux qui assurent cette fonction, même dans les cas où ils sont faiblement indemnisés, car cette fonction est très lourde. La commission décide donc d'organiser un débat autour de cette question, en commençant par faire un état des lieux. Une enquête (encore une!) va donc être lancée pour savoir qui

rait quoi dans nos universités vis-à-vis de cette occupation. Un livre blanc pourrait alors être rédigé et présenté aux autorités de tutelle.

5) FORUM UNIVERSITAIRES-FOURNISSEURS: C. Aperghis propose l'organisation d'un forum pour l'année 90-91 destiné aux enseignants chercheurs qui sont touchés de près ou de loin par les problèmes d'achat et de maintenance de moyens informatiques. Ce forum permettrait la rencontre avec des responsables techniques des fournisseurs sur des thèmes à préciser et qui correspondraient aux préoccupations actuelles des enseignants et des chercheurs.

La prochaine réunion de la commission devrait jeter les bases de ce forum.

La réunion se termine en déplorant une fois de plus le petit nombre de participants effectifs à la commission, qui met en cause l'existence même de la commission.

La prochaine réunion fixée ce jour au jeudi 3 mai est à confirmer.

G. de SABLET

La commission matériels et logiciels de SPECIF a décidé d'étudier la charge que représente pour les enseignants-chercheurs la responsabilité des moyens informatiques mis à la disposition des étudiants. Elle vous prie donc de prendre quelques minutes pour répondre au questionnaire ci-joint.

Après dépouillement, nous comptons rédiger un rapport qui pourra être publié et remis à nos autorités de tutelle.

En effet, les informations partielles et essentiellement orales que nous possédons nous conduisent à penser qu'un certain nombre de nos collègues enseignants-chercheurs des centres universitaires petits et moyens consacrent, par nécessité, une part importante de leur temps à assurer le bon fonctionnement des moyens informatiques nécessaires à l'enseignement.

Cette occupation, qui nécessite souvent une grande disponibilité, ne leur permet pas d'assurer correctement leur mission de chercheur, ce qui nuit à leur avancement et à la qualité générale de leur enseignement.

Nous vous prions donc de bien vouloir transmettre ce questionnaire à toute personne concernée.

TEXTE DU QUESTIONNAIRE:

NOM DE LA PERSONNE A CONTACTER:

UNIVERSITE:

UFR:

ADRESSE (POSTALE TELEPHONIQUE RESEAU):

PAR CATEGORIE DE MATERIEL:

NIVEAU DES ETUDIANTS CONCERNES (1er cycle, 2ème cycle, 3ème cycle):

TYPE DE MATERIEL (micro monoposte, station de travail, mini multiposte):

CARACTERISTIQUES:

nombre:

taille disque:

utilisation (exercices de programmation, génie logiciel, BD, IA, etc.):

connexions (réseau local interne, réseau externe):

nombre d'utilisateurs:

SYSTEME D'EXPLOITATION:

NOMBRE ET QUALIFICATION REELLE DES TECHNICIENS (ATUS):

NOMBRE D'ENSEIGNANTS RESPONSABLES ET CHARGE HERBOMADAIRE PAR PERSONNE:

COMMISSION ENSEIGNEMENT

(D. FAYARD)

Rapport d'activités

COMMISSION ENSEIGNEMENT

D. FAYARD

Contrairement aux habitudes la commission enseignement n'a pas cette année, organisé de journées SPECIF, ceci pour deux raisons essentielles :

- 1°) SPECIF c'est aussi la recherche et il avait semblé intéressant qu'un thème recherche prenne le relais.
- 2°) Une tentative, centrée sur les Ecoles d'Ingénieurs d'Informatique a eu peu d'échos auprès de la communauté.

Pour le reste la commission a fonctionné avec 3 groupes de travail.

- 1) Un groupe "DUEG", dont vous avez pu lire un compte-rendu d'activité dans le bulletin, groupe qui s'occupe de deux sujets :
 - Enseigner l'Informatique en tant que discipline dans les DUEG (journées à Paris 27-28-29 Mars 1990)
 - L'enseignement de l'Informatique dans les premiers cycles scientifiques (journées à Lyon 26-27 Avril 1990)

Pour tout renseignement ou si vous désirez participer activement, prendre contact avec M. LUCAS (Nantes)

- 2) Un groupe "Europe" qui, après quelques contacts avec le club EEA, réalise deux pages recto/verso qui paraîtront dans le bulletin de Mars, avant diffusion plus large à travers l'Europe. Ces deux pages ont pour but de présenter les principales formations universitaires d'Informatique.
- 3) Un groupe qui participe depuis le début (Avril 89) à l'organisation avec essentiellement SYNTEC Informatique, l'AF CET et le CIGREF à un colloque, au mois de Mars - 20 et 21, sur la formation des informaticiens. Colloque mettant en présence "l'offre et la demande" d'informaticiens.

20 Mars matin Différents rapports sur la formation des informaticiens et après!... par les auteurs des rapports.

20 Mars après midi A chacun sa vérité

- la demande (SSI, constructeurs, chercheurs...)
- l'offre
Grandes Ecoles, IUT, Maitrise, MIAGE, DEA, DESS.

21 Mars matin

5 Ateliers

Formation continue, Mission d'observation des emplois et de la formation en Informatique, les formateurs à l'Informatique, la place de l'Informatique dans l'Enseignement Supérieur, les programmes de formation.

21 Mars après midi

Compte rendu des ateliers, réponses des différents Ministères.

Perspectives

Outre la fin des actions engagées.

Des journées Ecoles d'Ingénieurs en Informatique.

Prendre contact avec :

- J. SEGUIN : ENSSAT LANNION 96 46 50 30
- D. FAYARD : ORSAY (1) 69 41 00 40

"Normalisation" des demandes d'admission en DEA/DESS afin d'alléger le travail des responsables de Maîtrises.

Prendre contact avec :

- J. GIANNESINI : AIX MARSEILLE II 91 26 92 74
91 26 90 69
- D. FAYARD : Courrier électronique
BITNET FAYARD AT PRIUTO11
UUCP LRI.LRI.FR

COMMISSION RECHERCHE

(P. LESCANNE)

RAPPORT SNOWBIRD 1988 : UNE DISCIPLINE EN PLEINE MATURATION

Extrait de "Communications of the ACM", volume 32, mars 1989.
Traduit de l'américain par Christiane GUYOT, de l'INRIA-Lorraine.

La grave crise de l'informatique étant passée, la lumière se fait sur les problèmes internes liés à ce secteur. La représentation de la recherche en informatique auprès du public et du monde politique fut l'un des thèmes essentiels du Snowbird 88. Le Computing Research Board (C.R.B.) s'attaque à ce problème non encore résolu.

David GRIES, Terry WALKER et Paul YOUNG.

Les présidents des départements américains et canadiens habilités à délivrer le Ph.D d'Informatique et de Génie étaient réunis à Snowbird (Utah) du 11 au 13 juillet 1988 pour leur conférence bi-annuelle. Les représentants de 105 départements sur 150, toutes disciplines confondues, étaient présents, ainsi que 27 participants appartenant au gouvernement ou à l'industrie. La conférence était organisée par le Computing Research Board, sous-comité de vingt-quatre Directeurs élus par les départements délivrant le Ph.D et, dont le but principal est de représenter les intérêts de l'informatique (c'est-à-dire la science de l'ordinateur et le génie de l'ordinateur) en matière de recherche et d'enseignement.

Il ressort de SNOWBIRD 88 l'impression que l'informatique atteint une certaine maturité et un certain équilibre. La plupart des problèmes qui touchaient cette discipline à la fin des années 1970 et en 1980 ont été résolus ou aplanis. Il était temps que l'informatique mette fin à ses activités nombrilistes et se tourne vers l'extérieur. En tant qu'outil technologique pour d'autres secteurs d'activité, l'informatique doit acquérir un rôle plus actif en faisant valoir ses besoins, ses intérêts et sa recherche fondamentale, auprès des autres disciplines et en collaborant avec celles-ci à l'évolution des applications de l'informatique. Plus encore, du SNOWBIRD 88 un accord s'est dégagé, consistant à dire que le C.R.B. devrait entamer des actions décisives pour exprimer clairement ses besoins et défendre les intérêts de la recherche en informatique.

La fin de la crise

Comme l'a montré le rapport SNOWBIRD 1980, la fin des années 1970 et le début des années 1980 furent, pour l'informatique, une période d'expérimentation. Le rapport de 1980 comparait le doublement du nombre des étudiants inscrits, de 1975 à 1979, avec les 3 % d'augmentation du corps enseignant dans les départements d'informatique délivrant le Ph.D pendant la même période. Il témoignait aussi d'un cruel manque d'équipements de laboratoires adéquates et soulignait les graves compressions de locaux dans la plupart des départements. Il suggérait des solutions à ces problèmes.

Le rapport SNOWBIRD 1982 [7] signalait encore la présence de difficultés mais quelques améliorations se faisaient sentir. Le rapport indiquait les mesures à prendre dans ce sens par les universités, l'Etat et l'industrie.

Quand vint le SNOWBIRD 84, l'informatique était alors sortie de sa période critique mais elle avait encore besoin d'une aide substantielle [6]. Cette conférence, consacrée aux problèmes internes de l'informatique, traitait de sujets tels que le contenu des sciences de l'informatique, la gestion des départements, la jeunesse de cette discipline et les problèmes que cela implique, les besoins en infrastructures des départements et les prévisions à long terme.

Le SNOWBIRD 86 traitait aussi en grande partie de ces questions internes. Le déséquilibre entre l'apparente croissance du personnel - à cette époque - et l'augmentation insuffisante des fonds de recherche fondamentale [4] a sans doute conduit aux discussions les plus importantes.

L'idée principale qui a émergé du SNOWBIRD 88 était que l'informatique avait atteint sa maturité. Quelques-unes des difficultés internes rencontrées à la fin des années 70 et au début des années 80 étaient résolues et d'autres paraissaient moins insurmontables. Dans certaines spécialités, l'informatique avait acquis une envergure comparable à celle d'autres domaines scientifiques. En voici quelques exemples :

- On estime le nombre des nouveaux diplômés de Ph.D en 1987-88 à 500, soit le double de ce qu'il était dix années auparavant [3]. Plus de 50 % de ces diplômés s'orientant vers l'enseignement, la pénurie en enseignants commence à se résorber dans les départements habilités à délivrer le Ph.D.

- Un rapport [1] établi par Richards ADRION et Paul YOUNG lors du SNOWBIRD 88 indiquait que des solutions étaient apportées au problème du manque de locaux dans les départements de Ph.D. d'informatique. 78 départements ayant répondu à l'enquête ont bénéficié - ou vont bénéficier - de 27 nouveaux bâtiments et 50 d'entre eux doivent être entièrement rénovés entre 1985 et 1991.

- La baisse des recrutements dans les disciplines informatiques, au cours des trois dernières années n'est plus guère un sujet de préoccupation. Une crainte a disparu : celle de se voir submergé par un rapport enseignants-étudiants largement défavorable.

- Les professeurs assistants sont plus nombreux que les professeurs titulaires et l'informatique souffre, du point de vue administratif et politique, d'un nombre trop élevé de jeunes. Cependant, cette discipline atteint sa maturité et compte davantage de scientifiques reconnus, respectés et disposés à se consacrer à leurs activités.

- La National Science Foundation (NSF) a créé le Computer and Information Science Directorate of the NSF (CISE) avec, à sa tête, Gordon BELL puis, aujourd'hui, Bill WULF ce qui a permis de mettre l'informatique (à savoir la science de l'ordinateur et le génie des ordinateurs) et les autres branches scientifiques sur un pied d'égalité.

- Le National Research Council a créé un Bureau des Sciences et des Technologies de l'Informatique (CSTB), présidé par Joe TRAUB. Comme l'affirmait TRAUB lors du SNOWBIRD 88, le CSTB exerce une interaction avec le monde politique, il contrôle le milieu informatique et il est responsable des travaux effectués dans différents domaines. Cette évolution contribue aussi à valoriser l'informatique par rapport aux autres disciplines technologiques et scientifiques.

- Les débats issus du SNOWBIRD 88 ont également permis d'affirmer que l'informatique est aujourd'hui suffisamment avancée pour se tourner vers l'extérieur. Puisque nous avons surmonté les graves problèmes internes et que nous avons formé une nouvelle génération de scientifiques expérimentés et responsables, nous pouvons maintenant regarder plus loin. Nous retrouvons cet opinion au travers des discussions sur les orientations de la recherche, les responsabilités professionnelles et sur les perspectives que représente l'informatique pour les milieux non scientifiques.

Expression des besoins et des possibilités de la recherche en informatique

Entre autres conférenciers, Les BELADY (MCC), Mike DERTOUZOS (MIT), Peter FREEMAN (NSF), Joe TRAUB (CSTB) et Bill WULF (NSF) ont mis l'accent sur l'importance, pour l'informatique de s'affirmer en tant que partenaire technologique pour d'autres secteurs. FREEMAN et WULF, en particulier, ont insisté sur la nécessité d'une représentation reconnaissable par les autres disciplines et le monde politique.

De nombreux orateurs ont reproché à nos efforts, en matière de recherche, de trop se tourner vers l'intérieur. DERTOUZOS, par exemple, pense que nous devons collaborer avec les autres : "Un architecte, dit-il, ne jette pas ses briques à son client en lui disant ensuite de construire un immeuble. Il y participe en ajustant son sens esthétique et ses compétences techniques à la demande du client. C'est dans ce sens que nous devrions collaborer avec les chercheurs des autres disciplines".

Dans le passé, l'impact de la recherche fondamentale en informatique sur les travaux de la science et de la technologie n'ont pas toujours été suffisamment mis en lumière. A l'aube des années 90, nous devons mener des actions de recherche plus décisives, dans les directions choisies.

Nous devons aussi donner une image plus positive de notre secteur, à la fois à l'extérieur et à l'intérieur de notre région. Le thème de discussion majeur qui est ressorti du Snowbird 88 fut sans doute celui de l'image de la profession. Voici résumées les impressions générales qui ressortent de l'assemblée à cet égard.

L'image de la recherche en informatique à l'extérieur

Il n'est pas, dans le passé, de groupe qui ait représenté de manière active et constante les intérêts de la recherche en informatique ou qui ait été consulté de façon sérieuse par un membre de l'extérieur pour fournir des informations, des analyses ou des directives. Toutes les autres disciplines scientifiques (par exemple les Mathématiques, la Physique) possède un groupe reconnu et respecté qui parle au nom de ses chercheurs et représente sa spécialité auprès du Congrès, des instances fédérales et des institutions, de la presse et du public.

Ces cercles constituent la première source d'information et d'explications globales sur les résultats techniques : ils permettent de déterminer les actions à mener ou d'entrer en contact avec les spécialistes qui participent aux comités et aux conseils de direction.

Le groupe accomplit sa mission de manière active en véhiculant l'information sans attendre passivement la demande.

Ces activités aident à maintenir et à accroître les crédits de recherche et, plus encore, elles sont, pour les autres, riches d'enseignements. Les décisions qui concernent la science, la recherche, la technologie et l'économie doivent être prises à la lumière d'un maximum d'informations. Une discipline doit s'engager à soutenir sérieusement l'enseignement car ce rôle lui incombe.

Ceux qui mettent en place une politique et déterminent des priorités ne peuvent prendre en compte ceux qui restent absents. Considérons, par exemple, le cas des mathématiques. Dans les années 70, les besoins dans le domaine des mathématiques, resté dans l'ombre, n'ont jamais été formulés et cette discipline en a malheureusement subi les conséquences. Il y a plusieurs années, l'AMS, le MAA et le SIAM se sont regroupés pour parrainer le fameux rapport DAVID et pour former une commission paritaire pour les mathématiques, qui comprend trois membres à temps plein, à WASHINGTON. Le JPBM, instigateur des principales campagnes d'information en faveur des mathématiques, a eu un effet marquant. Maintenant, la classe politique est consciente du rôle joué par les mathématiques et de ses difficultés au cours de la dernière décennie.

Un autre exemple est celui du Comité de Recherche en Informatique (CRB). Durant l'élaboration du rapport TRENDS [4], les auteurs reçurent de nombreux coups de téléphone de particuliers et d'agences demandant des détails sur ses activités et ses objectifs. Par la suite, le NRC apporta les changements tant attendus dans le centre d'informatique et de technologie. Nous pensons que la publicité suscitée par le rapport TRENDS est en partie à l'origine du renouveau des comités NRC en informatique et en technologie, et de l'importance croissante de l'informatique et du génie informatique au sein de la NSF.

Peter FREEMAN, de la NSF, a résumé le plaidoyer en faveur de la recherche en informatique de la façon suivante :

"A mon avis (et en tant que membre de la NSF), il ne s'agit pas de refuser l'aide extérieure. mais, en l'état actuel des choses, nous ne disposons pas de porte-parole pour la recherche comme il en existe en physique, mathématiques et chimie. Il y a deux problèmes : la proximité géographique et la légitimité. Actuellement, personne n'est en mesure de parler pour toute la communauté que constitue la recherche en informatique, aucune organisation n'étant reconnue par les gouvernants comme le porte-parole.

De même, Bill WULF, le nouveau directeur adjoint de la NSF, responsable du CISE, a fait allusion, lors de son discours cérémonial, à la nécessité de former un groupe solide et clairement reconnu, pour définir les besoins et les projets de notre secteur en matière de recherche.

Le secteur vu de l'intérieur

Un groupe professionnel se doit d'informer le public, mais aussi ses propres membres, de ses problèmes, des politiques à adopter, des événements (prix, récompenses, partenariat, plans de financement) et de ses responsabilités. Le groupe doit porter à la connaissance de ses chercheurs ce qu'il fait et ce que l'on fait pour lui. Il doit s'efforcer d'ouvrir des horizons, il doit faire prendre conscience que les travaux de recherche sont effectués au sein d'une communauté dans son ensemble, et nous rappeler que les bases de la recherche s'appuient souvent sur des facteurs politiques. Il en résulte, pour les chercheurs, un sentiment d'appartenance à la communauté et de responsabilité professionnelle.

Ceci, en partie grâce aux importantes rencontres nationales et aux bulletins professionnels. SIAM News et Notices of the American Mathematical Society sont deux exemples de bulletins réussis, qui rassemblent une multitude de nouvelles de Washington, de brefs survols techniques dans certains domaines spécialisés, de résumés, de compte rendus d'événements (bourses, collaborations, récompenses, prix, travaux de comités, etc), de sujets traitant de certaines personnalités et de commentaires sur des questions de pédagogies. Communications of the ACM et Computer font partie de ces canaux utiles à la diffusion de ces informations mais, ayant été peu exploités dans le passé, il ne sont pas considérés comme des véhicules importants de ce matériel.

A cet égard, le domaine de l'informatique, encore récent, nous a rendu la tâche difficile. De par une évolution rapide et un nombre supérieur de professeurs assistants par rapport aux professeurs titulaires (problème propre à l'informatique), les membres appartenant aux grades les plus élevés sont souvent submergés par leurs obligations administratives dans leur propre institution et, il leur reste peu de temps pour les obligations extérieures.

De plus, les jeunes rencontrent très peu de scientifiques attachés à la valorisation des intérêts de l'informatique. Leurs aînés sont davantage impliqués dans la recherche et dans les activités universitaires. Il en résulte qu'en informatique, les jeunes n'apprennent pas leurs responsabilités professionnelles comme cela se fait dans d'autres départements.

Les signes de cette absence de responsabilisation abondent. Les directeurs de publications se plaignent d'une pénurie de bons rapporteurs. Les comités tels qu'ACM Turing Award Committee reçoivent peu de candidats bien préparés. Le recrutement de nouveaux membres pour l'ACM et le Computer Science est extrêmement faible, en partie parce que le milieu n'encourage pas ses membres à participer. Les chercheurs évitent les postes à responsabilités à l'intérieur des groupes ou bien refusent tout simplement d'y entrer. Dans les années 80, les départements d'informatique de la NSF avaient des difficultés à trouver des directeurs de programmes temporaires.

De plus, les départements récents et peu importants, habilités à délivrer le Ph.D. et constitués essentiellement de jeunes, ne connaissent pas bien les procédures budgétaires et ont du mal à trouver des fonds de recherche.

La recherche en informatique non représentée

La recherche en informatique a besoin d'une représentation unique qui défende ses intérêts à la fois auprès du public et de ses propres membres. Le CRB a parfois trouvé des réponses à ces problèmes mais il manque de personnel et de fonds pour s'affirmer de façon durable. En partie à cause de leurs effectifs nettement plus importants, l'ACM et le Computer Society n'ont pas eu ce rôle dans la communauté scientifique. Tous deux possèdent d'excellents journaux techniques et parrainent des conférences scientifiques ; ils remplissent ainsi admirablement bien certaines fonctions d'un groupe de recherche mais pas celles que nous venons d'évoquer.

Le manque de représentation propre à la recherche en informatique conduit à des effets négatifs. Citons quelques exemples :

- Entre 1979 et 1985, les aides à la recherche fondamentale en informatique, tant au niveau fédéral qu'au niveau du NSF (particulièrement stables) n'ont pas augmenté aussi rapidement que les effectifs des universités. Dans tous les autres départements principaux et proches de l'informatique, l'aide s'est accrue plus rapidement. Entre 1979 et 1985, les aides du NSF aux boursiers ont chuté de 8 % bien que le nombre de chercheurs dans cette branche ait nettement augmenté. (C'est la conséquence, entre autres, de la création et de l'essor du programme CER (Coordinated Experimental Research) qui a apporté l'infrastructure et les équipements dont bénéficient déjà de nombreuses autres disciplines). La hausse des effectifs n'a pas joué en faveur d'un meilleur soutien financier et l'argumentation dans ce sens n'a pas été assez vigoureusement défendue.

- Il y a quelques années, les membres du Congrès ainsi que d'autres décideurs politiques, estimaient avoir donné, pour la création des quatre grands centres de recherche en informatique de la NSF, d'énormes sommes d'argent qui furent essentiellement utilisées dans d'autres domaines scientifiques. Comme beaucoup, ils associaient l'aide à l'achat d'ordinateurs avec l'aide à l'informatique.

- Le ministère de l'Education nationale vient de développer un programme d'attribution de bourses dans les domaines de la science et de la technologie qui représentent des priorités nationales. L'informatique n'étant pas une priorité nationale, elle ne fut pas incluse ; la physique, la chimie, les mathématiques et toutes les disciplines des sciences de l'ingénieur furent incluses.

- L'AAAS a un projet appelé "Projet 2061" qui concerne l'éducation pour un nouvel avenir. L'ébauche d'un rapport intitulé "Ce qu'il est le plus important de connaître en matière de science" a été rédigé par quarante-deux scientifiques, dont huit mathématiciens. Le milieu de l'informatique n'était pas représenté.

- Dans le numéro du 20 mai 1988 de "Science", un article appelle à la création d'une table ronde autour des priorités scientifiques car "les chemins de la Science ont fini par croiser ceux des grands déficits nationaux" [5]. Les membres de cette table ronde seraient choisis par les groupes concernés. Si une telle liste devait se constituer aujourd'hui, il y a fort à parier que notre branche serait à nouveau représentée par des personnes venant de l'extérieur.

Changement important dans les activités du CRB

Lors du Snowbird 88, David GRIES a évoqué, au nom du CRB, la création d'une voie unique pour la recherche en informatique dans le contexte actuel qui inclut déjà l'ACM, la Computer Society et le SIAM. Une coalition serait-elle possible entre 150 départements universitaires, 50 laboratoires industriels et ces organisations qui comprennent 160 (XX) membres? Une telle coalition donnerait, à Washington et auprès du public, une image beaucoup plus forte de l'informatique que ne le ferait aucune de ces organisations à elle seule, et la contribution de chacune serait relativement modeste, comparée aux énormes dépenses nécessaires.

GRIES a proposé que le CRB dirige cette coalition. En prévision de ce changement, le CRB s'était constitué en société. Il avait changé son mode d'élection et était maintenant élu par les présidents des départements habilités à délivrer le Ph.D.

Les participants au Snowbird ont estimé que le CRB devrait embaucher du personnel dont les fonctions seraient les suivantes :

1) Créer un bulletin qui serait, au moins, distribué à tous les enseignants des départements d'informatique délivrant le Ph.D. Le but est de mieux sensibiliser les informaticiens et les ingénieurs aux problèmes qui atteignent la recherche en informatique et de leur faire prendre conscience de leurs obligations professionnelles. SIAM News fut cité comme un excellent exemple à suivre pour sa fréquence de parution et son contenu.

2) Mettre en place la représentation de l'informatique à Washington. Le CRB rechercherait l'approbation et le soutien de l'ACM, de la Computer Society et du SIAM pour être reconnu comme représentation unique des intérêts de la recherche en informatique. Les trois organisations s'engageraient, par contrat avec l'ACM, à se mettre au service de la communauté de l'informatique dans son ensemble.

Cette action, qui demanderait des fonds importants, serait un tournant essentiel pour le CRB. Créé en 1973 pour représenter les intérêts des départements d'informatique délivrant le Ph.D., le CRB n'a, depuis lors, fonctionné avec quasiment aucun revenu. Les membres (ou leur organisation) ont payé leurs voyages pour se rendre aux congrès ; l'enquête TAULBEE et la liste FORSYTHE des départements délivrant le Ph.D. sont subventionnés par les membres des institutions et il n'y a pas eu de crédits affectés à la valorisation des intérêts de la recherche.

Les représentants des 105 départements d'informatique présents au Snowbird 88 ont exprimé leur parfaite adhésion à ce projet. Ils pensent que le CRB doit aller jusqu'au bout ou ne rien faire et, en conclusion, ont estimé le coût de l'opération serait de 400 à 500 K\$ par an. La plupart des départements représentés à SNOWBIRD 88 étaient prêts à payer des sommes considérables pour appuyer le projet : de 500 \$ à 5 K\$ par département et par année.

Lors de son congrès qui a suivi SNOWBIRD 88, le CRB a accepté cet engagement dans une action offensive pour représenter les intérêts de l'informatique. Il agit maintenant sur trois fronts : mise en place de travaux permanents, développement d'une structure de crédits et d'autres sources budgétaires et enfin, dialogues avec l'ACM, la Computer Society et le SIAM.

Depuis juillet 1988, le CRB s'attache à représenter la recherche en informatique. Les départements habilités à délivrer le Ph.D payent maintenant des droits et d'autres activités lucratives vont être mises en place. Le Docteur Terry WALKER a été nommé directeur. Des contacts ont eu lieu avec ACM, la Computer Society et le SIAM. La Computer Society a fourni à M. WALKER un bureau à Washington ; le CRB, conjointement avec ACM, a prévu une conférence sur les objectifs de la recherche et, un projet de création d'un bulletin est en cours avec le SIAM.

Références

1. W.R. ADRION et P. YOUNG, "Space in time : A report on trends in space available for departments of computer science and computer engineering". Computing Research Board, Washington, D.C., juin 1988.
2. P. J. DENNING, "The 1980 Snowbird report : A discipline in crisis", Commun, ACM 24, 6 (juin 1981), 370-374.
3. D. GRIES et D. MARSH, "The 1986-87 Taulbee survey". Commun. ACM 31,8 (août 1988). 984-991. Egalement publié dans Computer 21 (août 1988), 53-61.
4. D. GRIES, R. MILLER, R. RITCHIE et P. YOUNG, "Imbalance between growth and funding in academic computing science : two trends colliding". Commun. ACM 29, 9 (sept. 1986), 870-878. Egalement publié dans Computer 19 (sept. 1986), 70-76.
5. D.E. KOSHLAND, "Jr. Setting priorities in Science", Sci., 240,4855 (mai 1988), 965.
6. J. TARTAR, "The 1984 Snowbird report : future issues in computer science". Commun. ACM 28, 5 (mai 1985), 490-493.
7. S.S. YAU, "The 1982 Snowbird report : Meeting the crisis in computer science". Commun. ACM 26, 12 (déc. 1983), 1046-1050.

PRIME D'ENCADREMENT DOCTORALE (PROPOSITIONS DE SPECIF)

La prime d'encadrement doctorale est une reconnaissance pour les universitaires de leur fonction de chercheur et surtout de celle de directeur de recherche. SPECIF, notamment par l'intermédiaire de sa commission recherche, se doit de prendre une position sur ce problème et, en particulier, sur les critères d'évaluation. Voici quelques éléments de réflexion.

Tout d'abord, il a paru important de reconnaître la fonction indirecte d'encadrement de recherche que représente la direction d'une unité de recherche. En conséquence, tous les directeurs de laboratoires de qualité reconnue et de taille significative devraient bénéficier de la prime d'encadrement doctorale.

Les critères qui vont être définis ci-dessous doivent être entendus de la façon suivante:

1. Ils constituent un ensemble de faits convergents et aucun d'eux ne doit être pris comme absolument nécessaire, de même chacun doit être apprécié de manière qualitative plutôt que quantitative.
2. Une autonomie dans l'établissement des critères doit être reconnue à chaque discipline, car chacune a sa spécificité.

Pour des raisons évidentes, les critères ne peuvent pas être les mêmes pour un chercheur avancé (senior) et pour un chercheur de moins d'expérience (junior) qui cependant encadre des recherches. Nous avons donc considéré deux types de critères.

Professeurs Il faudrait privilégier l'encadrement doctoral effectif à la qualité intrinsèque de chercheur. Il faudrait évaluer les succès des thésards plutôt que les résultats du professeur lui-même. En particulier, il semble utile de connaître, à l'avenir, les publications qui pourraient être issues des thèses, ce qui n'est pas, semble-t-il, le cas actuellement. Dans les disciplines des sciences de l'ingénieur, comme l'informatique, il paraît important de prendre en compte l'activité d'encadrement de projets de recherche donnant lieu à la réalisation de prototypes.

Maître de conférence Il faudrait juger le dossier personnel notamment sur les points suivants:

- publication dans des conférences sélectives, c'est-à-dire *internationales, avec comité de lecture préalable sur papier complet et dont les actes sont publiés par un éditeur,*
- publication dans des revues spécialisées dans la publication des articles de recherche,
- réalisation de prototypes innovants en démontrables,
- aptitude au travail en équipe,
- participation à l'encadrement de stages de DEA.

COMMISSION PERSONNEL

(P. CHRETIENNE)

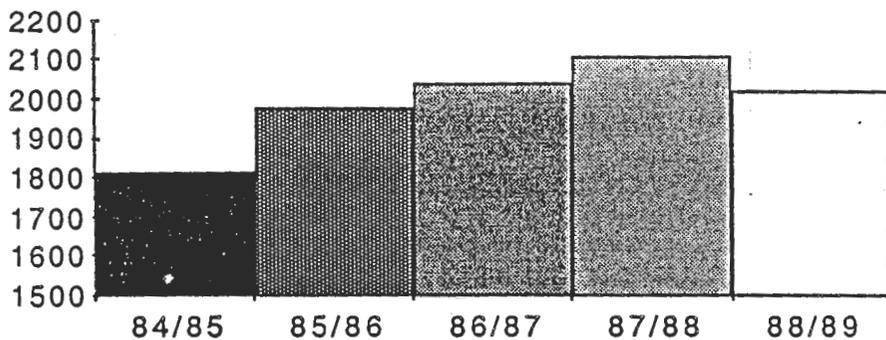
ENQUETE SPECIF

EFFECTIFS ETUDIANTS ENSEIGNANTS-CHERCHEURS EN INFORMATIQUE SUR LES ANNEES 84/85 ----> 88/89

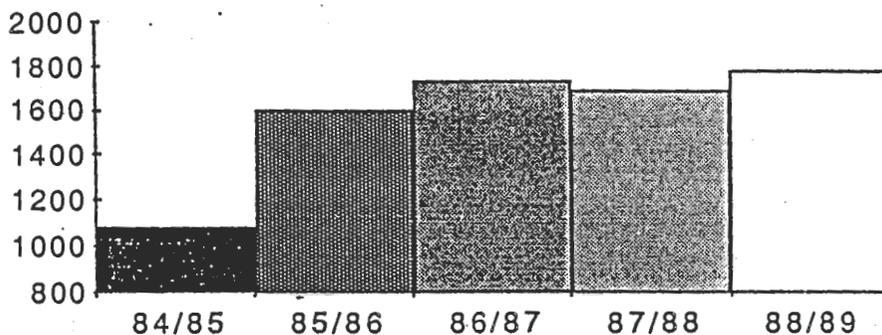
Cette enquête avait pour objectif d'établir des tendances sur les effectifs d'étudiants, enseignants, chercheurs, thésards, à partir des données recueillies sur les cinq années 84/85 88/89. J'ai obtenu des réponses concernant 31 Universités, 19 I.U.T., et 6 écoles. Certaines réponses se sont révélées incomplètes car n'apportant que les informations ponctuelles de l'année 88/89. C'est pourquoi chaque résultat donné ci-dessous a été établi à partir d'un échantillon réduit aux réponses permettant d'établir une tendance. Le problème de la représentativité de cet échantillon n'a pas été étudié. Etant donné le trop faible nombre de réponses des écoles, les résultats présentés ci-dessous ne concerne que les Universités et les I.U.T. Enfin, les données relatives à l'année 84/85 étant souvent soit inexistantes soit sujettes à caution, il ne faut pas dans certains cas tenir compte de ces points "aberrants" dans l'expression de la tendance.

Ph. CHRETIENNE,

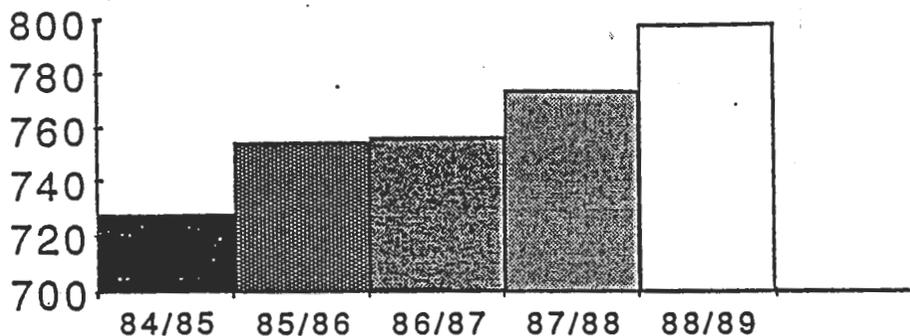
LICENCE
(Echantillon : 20 Universités)
TENDANCE 84-89 : +54/an



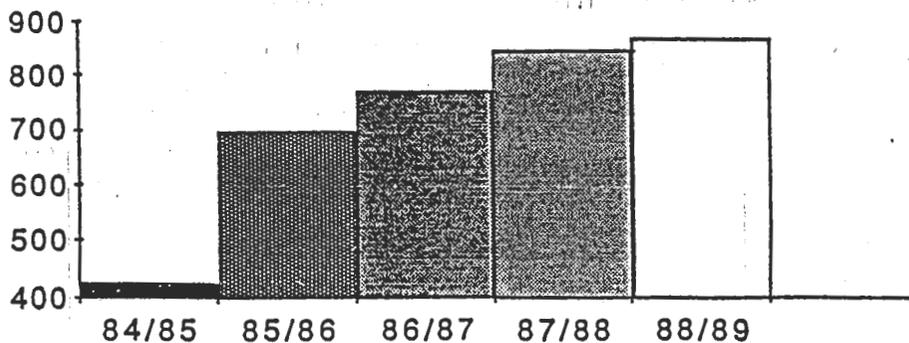
MAITRISE
(Echantillon : 20 Universités)
TENDANCE 84-89 : +48/an



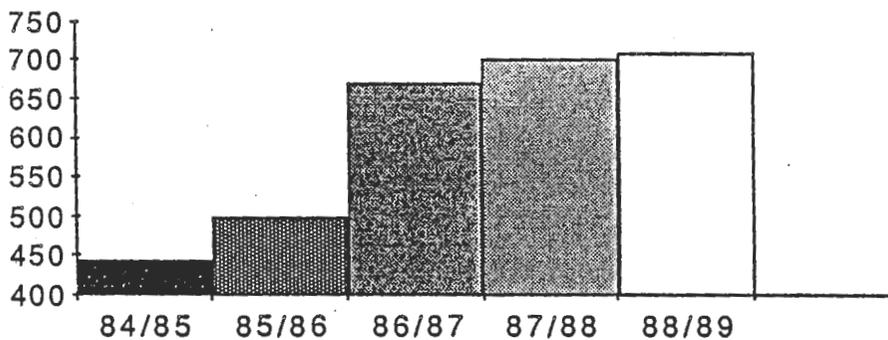
ETUDIANTS DEA
(Echantillon : 31 Universités)
TENDANCE 84-89 : +16/an



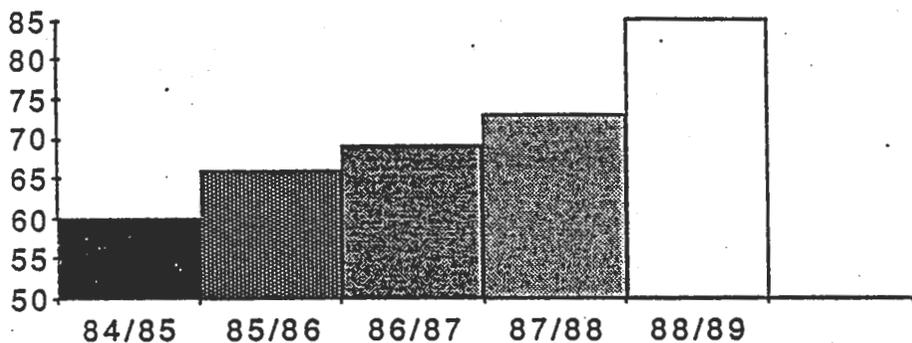
ETUDIANTS DESS
(Echantillon : 31 Universités)
TENDANCE 84-89 : +102/an



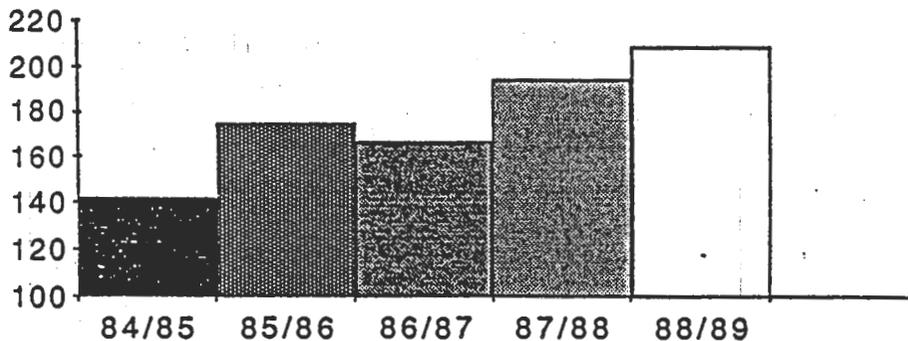
THESARDS
(Echantillon : 30 Universités)
TENDANCE 84-89 : +72/an



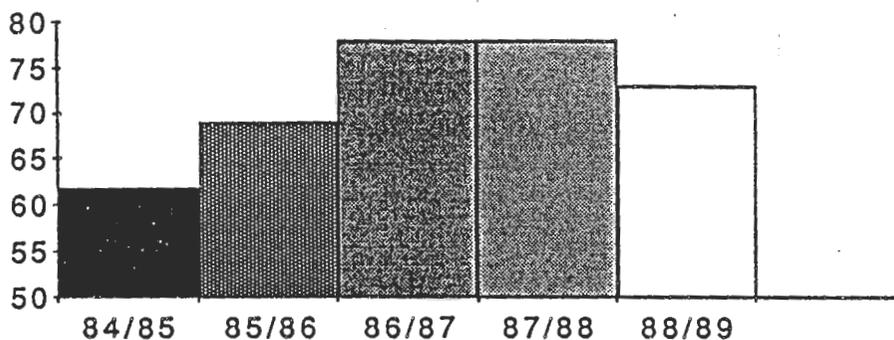
PROFESSEURS
(Echantillon : 20 Universités)
TENDANCE 84-89 : +6/an



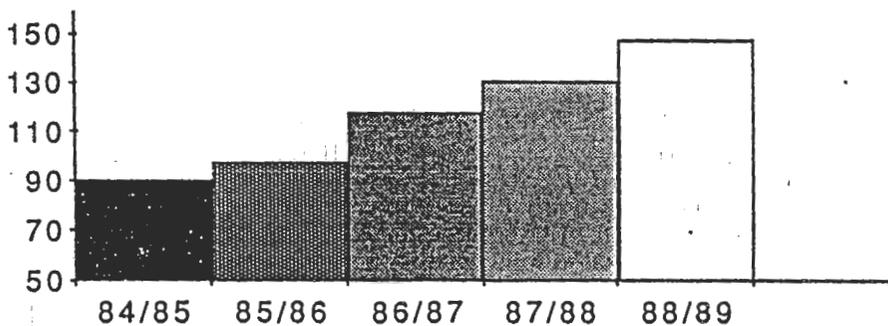
M.A. + M.C.
(Echantillon : 17 Universités)
TENDANCE 84-89 : +15/an



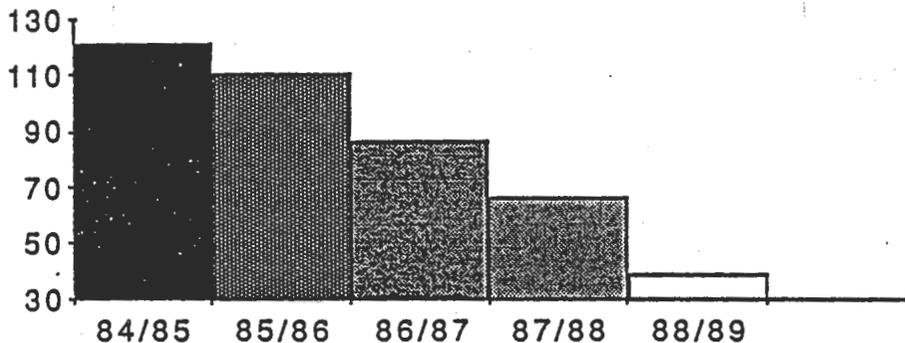
ASS + ALER + ATER
(Echantillon : 17 Universités)
TENDANCE 84-89 : +3/an



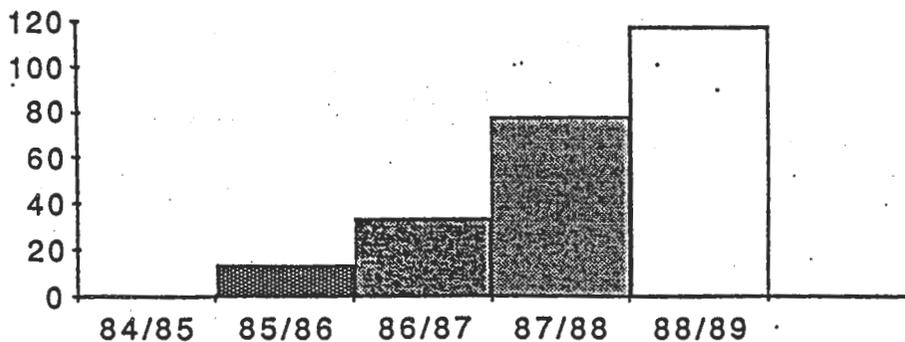
CHERCHEURS CNRS + INRIA
(ECHANTILLON : 17 UNIVERSITES)
TENDANCE 84-89 : +15/AN



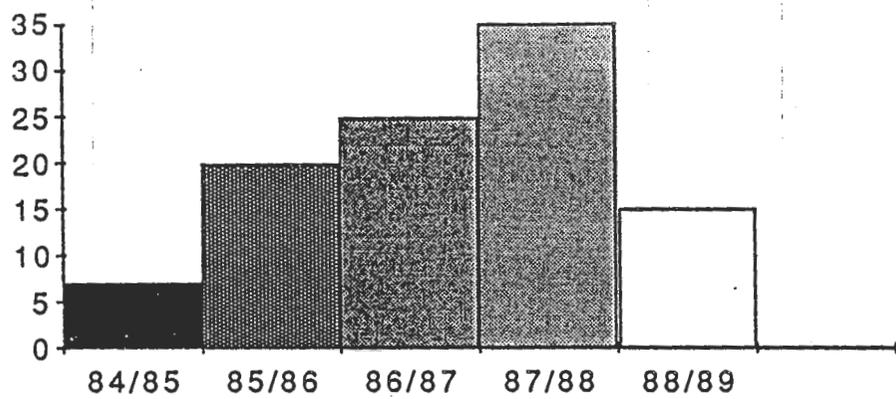
THESES 3ème CYCLE et D.I.
(Echantillon : 30 Universités)
TENDANCE 84-89 : -11/an



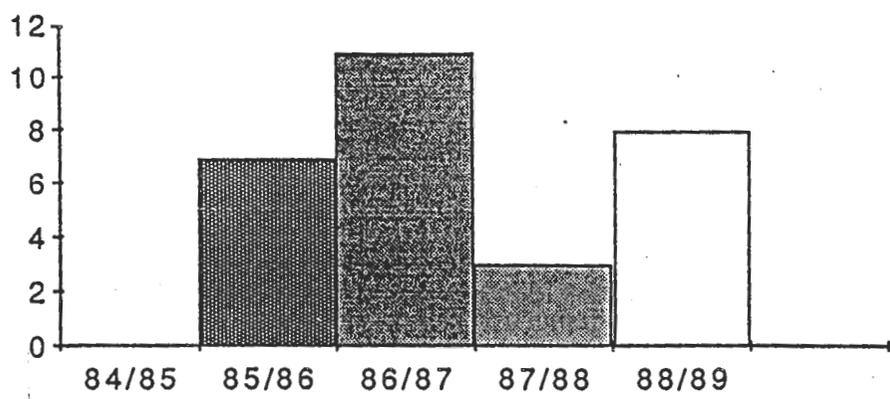
NOUVELLES THESES
(Echantillon : 30 Universités)
TENDANCE 85-89 : +30/an



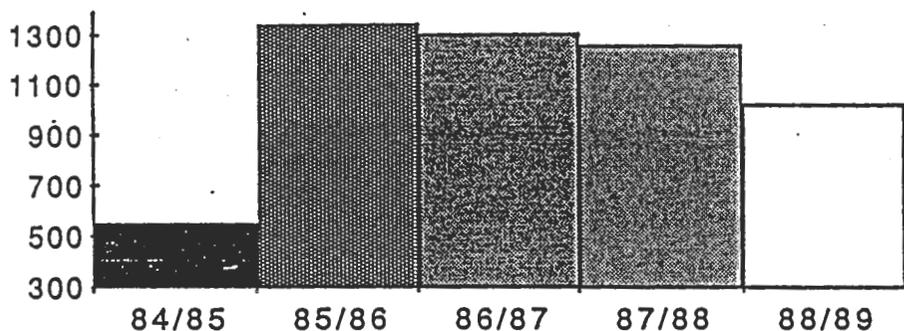
THESES D'ETAT (Echantillon : 30 Universités)



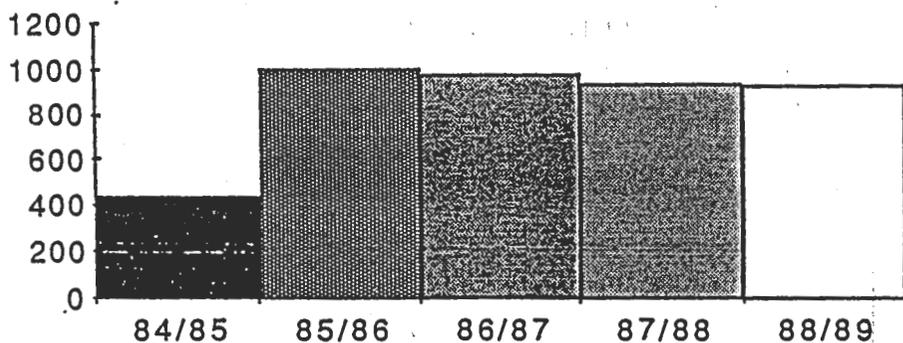
HABILITATIONS (Echantillon : 30 Universités)



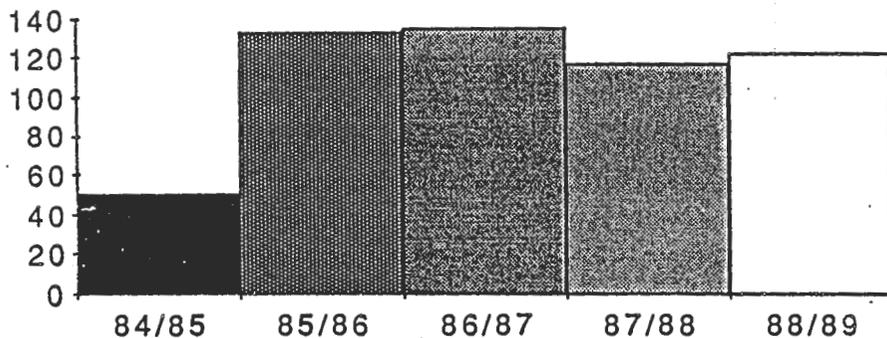
1ère ANNEE IUT
(Echantillon : 10 I.U.T.)
TENDANCE 85-89 : -100/an



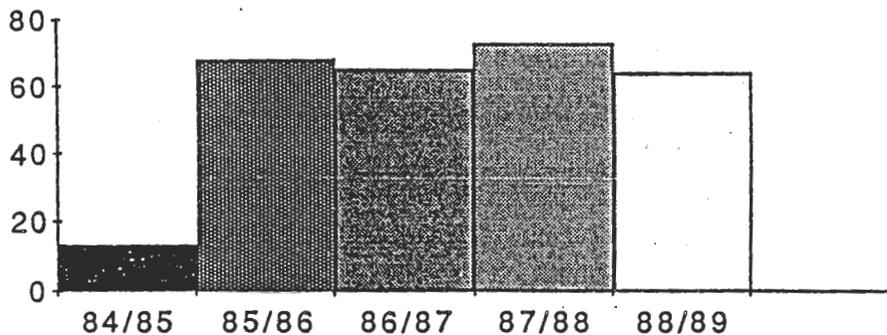
2ème ANNEE IUT
(Echantillon : 10 I.U.T.)
TENDANCE : -28/an



3ème ANNEE IUT
(Echantillon : 10 I.U.T.)
TENDANCE 85-89 : -5/an



ANNEE SPECIALE IUT
(Echantillon : 10 I.U.T.)
TENDANCE 85-89 : -0,5/AN



CNU INFORMATIQUE

(B. LORHO)

- . Répartition des enseignants (24ème)
- . Résultats de la Session MAI 1990

REPARTITION DES ENSEIGNANTS DE LA 24 EME SECTION DU CNU,
PAR DISCIPLINES, CORPS, GRADES, OU FONCTIONS.

disc grades ou fonctions	2401	2402	TOTAL
ALLOC	6	2	8
ALER	93	9	102
ATER	93	8	101
MONITEURS	13		13
CT.ENSAM	2		2
PRAG	11	3	14
PRCE	12	2	14
A E	2		2
AS.NAG	167	38	205
AS.AG	6	1	7
MA.2C	20	1	21
MA.1C	2		2
MCF.2C + MCF STG	289	60	349
MCF.1C	401	66	467
PR.2C	191	29	220
PR.1C	79	14	93
PR.EX	12	2	14
TOTAL	1399	235	1634

CHANGEMENT DE CLASSE : CE 1 --> CE 2	
NOM	UNIVERSITE
CASTAN Serge GROSS Maurice VERJUS Jean-Pierre	Toulouse 3 Paris 7 Grenoble 1

CHANGEMENT DE CLASSE : 1ère Cl. --> CE 1	
NOM	UNIVERSITE
BERSTEL Jean DELOBEL Claude KRAKOWIAK Sacha LORHO Bernard PERRIN Dominique	Paris 6 Paris 11 Grenoble1 Orléans Paris 7

CHANGEMENT DE CLASSE : 2ème Cl. --> 1ère Cl.	
NOM	UNIVERSITE
ALT René ANDRE Françoise BELLISSANT Camille CAMILLE-RAPP Jean CAUSSE Bernard CHIARAMELLA Yves COSNARD Michel COT Norbert CROCHEMORE Max CROWLEY James FAYARD Didier FERRIE Jean FLORIN Gérard GARDAN Yvon GUESSARIAN Irène HABIB Michel PIERREL Jean-Marie PLATEAU Gérard RODRIGUEZ François SALLE Patrick SCHOLL Pierre-Claude SPEHNER Jean-Claude SPYRATOS Nicolas TOLLA Pierre TOMMASINI-QUERE Maryse WOLFMANN Jacques	Caen Rennes 1 Grenoble 2 INSA Rennes Pau Grenoble 1 ENS Lyon Paris 5 Paris 7 INP Grenoble Paris 11 Montpellier 2 CNAM Metz Paris 6 Montpellier 2 Nancy 1 Paris 13 INP Toulouse INP Toulouse Grenoble 1 Mulhouse Paris 11 Paris 9 Nancy 2 Toulon

CHANGEMENT DE CLASSE : 1ère Cl. → Hors Classe	
NOM	UNIVERSITE
ALAMY-VILLARD Janine	Rennes 1
BOTTE Pierre	Nancy 2 IUT A
BOUSQUET Léon	Toulouse 3
DUPUY Maryvonne	Dijon IUT
IGOT Jean-Pierre	Strasbourg 3 IUT
LANDELLE Alain	Grenoble INP
LECCAS Dimitri	Paris 6
PARTOUCHE Jean-Paul	Nice
PECCOUD-CHABRE Monique	Grenoble 1
QUERE Alain	Nancy 1
TERRAT Richard	Montpellier 2
VERGNES Jean	Aix Marseille 3

CHANGEMENT DE CLASSE : 2ème Cl. → 1ère Cl.	
NOM	UNIVERSITE
ACCART-HARDIN Thérèse	CNAM
BERGER Philippe	INP Toulouse
BETAILLE Henri	Montpellier 2 IUT
BROUSSE-GIROIRE Héléne	Paris 6
CART Michèle	Montpellier 2
CHESNEAUX Jean-Marie	Paris 6
CHETTO Houssine	Nantes
CHOQUET Annie	Paris 11
DAVID Jean-Marie	Nancy 1
DEBORD Patrick	Toulouse 3
DOUCET Anne	Paris 11
DUCHAMP Alain	Le Mans
DUPONT Lionel	Grenoble 2
DURAND Jacques	Metz IUT
ENSELME Daniel	CNAM
ESTRAILLER Pascal	Paris 6
EVRRARD Fabrice	INP Toulouse
FDIDA Serge	Paris 6
FOUQUERE Christophe	Paris 13
GARETTA Henri	Aix Marseille 2
GEIB Jean-Marc	Lille 1
GERTOSIO Christine	Lyon 1
GODART Claude	Nancy 1
GOOSENS Daniel	Paris 8
GUYENNET Hervé	Besançon
JOUVE-CSERNEL Mireille	CNAM
KASSEL Gilles	Compiègne
KNIPPEL Jean-Michel	Aix Marseille 2
KREY Charlie	INP Toulouse
LAURENT Dominique	Orléans
LE SAEC Bertrand	ENSER Bordeaux

CHANGEMENT DE CLASSE : 2ème Cl. ---> 1ère Classe (Suite)	
NOM	UNIVERSITE
LELANDAIS Sylvie	Nantes
LELEU-BARON Monique	Paris 6
LEPAGE Daniel	Paris 8
LITOVSKY Igor	ENSER Bordeaux
LOPEZ-KRAHE Jaime	Paris 8
LOUVIERS-ADELAIDE Valentine	Antilles
MAILLARD-VILNAT Anne	Paris 11
MANOUSSAKIS Ioannis	Paris 12
MARTY-MARSOTTE Anne-Marie	Montpellier 2
MASSIE Henri	Toulouse 3
MAZEROLLES-MONDOT Anne-Marie	Toulouse 3
MZOUGHJI Abdelaziz	Toulouse 3
NICAUD Jean-François	ENS Cachan
PALIES Odile	Paris 6
PARUELLE-REVENU Marinette	Caen ENSI
PETIT Antoine	Paris 11
RAPHALEN Michèle	Rennes 2 IUT
REDON René	Lyon 1
ROUX Olivier	Nantes
SAOUDI Ahmed	Paris 13
THIESE Bernard	INP Toulouse
TIMMERMAN Erick	Lille 1 IUT
TOMASIK Jerzy	Clermont 2 IUT
VAUDENE Didier	Paris 6
VILAREM Jean-François	Montpellier 2
VOISIN Frédéric	Paris 11
VRAIN Christel	Orléans

**PROTOCOLE
D'UTILISATION DE LOGICIELS**

—ooo0ooo—

P R O T O C O L E

ENTRE LES EDITEURS DE LOGICIELS,
LE SYNDICAT NATIONAL DES PROFESSIONNELS DU LOGICIEL MICROINFORMATIQUE,
ET LE MINISTERE DE L'EDUCATION NATIONALE, DE LA JEUNESSE ET DES SPORTS
(Direction des Enseignements Supérieurs)

—ooo0ooo—

PREAMBULE

Le protocole d'accord entre les éditeurs de logiciels, le Syndicat National des Professionnels du Logiciel Microinformatique (SNPLM) et le Ministère de l'Education Nationale de la Jeunesse et des Sports (Direction des Enseignements Supérieurs), élaboré par une commission composée d'universitaires et de représentants des professionnels, a pour finalité de trouver un point d'équilibre entre les contraintes pédagogiques des universitaires et les intérêts des éditeurs de logiciels.

Les rédacteurs et les signataires du protocole ont travaillé dans le cadre de la loi du 3 juillet 1985.

Ce protocole comporte trois types de disposition :

-une disposition d'ordre pédagogique par un effort d'information du ministère de l'éducation nationale et des professionnels vers les enseignants et les étudiants ;

-une disposition d'ordre "réglement à l'amiable" en cas de difficultés. A travers une information réciproque, elle doit permettre, si possible, d'aboutir à une solution négociée ;

-une disposition d'ordre technique qui analyse les contraintes techniques des enseignants en vue d'élaborer une liste de solutions possibles.

I - La loi protège les titulaires de droits sur les logiciels et progiciels : à cette fin, elle prohibe la copie illicite des logiciels et progiciels commercialisés en France, quels qu'en soient l'usage et la destination. Soucieux de lutter contre le développement des infractions commises au préjudice des droits des auteurs et éditeurs de logiciels, le ministère de l'éducation nationale, le SNPLM et les Editeurs de logiciels sont convenus d'agir en concertation comme suit :

- . Lorsqu'un éditeur est informé d'un abus, dans la mesure où il n'envisage pas d'employer des mesures conservatoires (saisie, ...), il s'engage à informer au préalable de l'action judiciaire le Ministère de l'Education Nationale (Direction des enseignements supérieurs) étant bien entendu qu'en tout état de cause cette information préalable ne saurait en aucune façon porter atteinte au plein exercice des droits des auteurs, éditeurs ou distributeurs.
- . Par contre dans le cas où l'éditeur envisage d'employer des mesures conservatoires, il s'engage à tenir informé parallèlement à toute action, le Ministère de l'Education Nationale (Direction des enseignements supérieurs) de l'existence des procédures judiciaires en cours impliquant des agents relevant de son autorité et ce, éventuellement, par l'intermédiaire du SNPLM.

Le Ministère de l'Education Nationale (Direction des enseignements supérieurs) s'engage à sensibiliser les responsables d'établissements, les personnels et les étudiants à la nécessité de respecter les droits des auteurs et éditeurs de logiciels et progiciels ; il s'engage en outre à les informer des risques encourus par les auteurs d'infractions aux lois en vigueur.

Le Ministère de l'Education Nationale tiendra le SNPLM informé des initiatives prises en ce sens, en lui communiquant en particulier copie des notes internes relatives à la protection des droits sur les logiciels et progiciels.

En cas d'information par un éditeur d'une présomption de fraude ou copie, le Ministère de l'Education Nationale (Direction des enseignements supérieurs), s'engage à mener une enquête avec information en retour auprès de l'éditeur concerné.

Par ailleurs, le Ministère de l'Education Nationale (Direction des enseignements supérieurs) s'engage à promouvoir des achats de logiciels cohérents avec les acquisitions de matériels, et à vérifier que les budgets dédiés ne sont pas détournés de leur objectif.

Le Ministère de l'Education Nationale et le SNPLM conviennent de se réunir à intervalles réguliers pour conférer des conditions d'application du présent protocole et des éventuels développements auxquels il peut donner lieu.

II - Afin d'envisager s'il est possible de prendre en compte, par des accords ou des coûts différents, les usages très divers des logiciels en milieu pédagogique (usage plein temps, usage simultané ou réitéré dans le temps, usage ponctuel pour démonstration), la liste des problèmes techniques indiqués ci-après, représentant les difficultés le plus souvent mentionnées par les enseignants, a été soumise à chaque éditeur qui lui apportera une réponse individuelle. En effet devant la diversité des problèmes posés et des catégories d'éditeurs concernés, une réponse unique n'a pas pu être trouvée.

Une première synthèse des réponses a été communiquée au ministère de l'éducation nationale (Direction des enseignements supérieurs) ainsi qu'au SNPLM qui tiendra à jour une liste des solutions possibles.

Par ailleurs pour faciliter la réponse aux cas spécifiques, chaque éditeur a désigné un correspondant unique pour les problèmes d'usage pédagogique.

Les problèmes techniques recensés sont les suivants :

- (a) Utilisation simultanée, avec copies, quand l'utilisation réitérée dans le temps est pénalisante du point de vue pédagogique :
 - . pour un même nombre d'heures d'utilisation, il existe plusieurs possibilités d'organisation :

exemple : avec un logiciel, on peut faire travailler successivement 10 étudiants ;
avec un logiciel + 9 copies, on peut les faire travailler simultanément.
- (b) Utilisation ponctuelle par un enseignant d'un logiciel dans le cadre de la préparation d'un cours ou pour une démonstration dans son cours.
- (c) Fiabilité des disquettes dont les chargements maintes fois renouvelés entraînent de fréquentes destructions.
- (d) Sauvegarde, pour l'étudiant, de son environnement de travail (logiciels plus données) qu'il doit retrouver d'une séance de travaux pratiques à la suivante.

La liste des correspondants et les différentes réponses aux problèmes techniques figurent en annexe au présent protocole.

III - Afin de reposer la question de la copie privée et en particulier permettre à un enseignant d'emporter chez lui pour travailler un logiciel (ou une copie de celui-ci) acquis par son unité pédagogique, le SNPLM a engagé une consultation d'avocats. Le résultat de ce travail a été communiqué au ministère de l'éducation nationale, (Direction des enseignements supérieurs), qui pourra de son côté recueillir l'avis de ses experts.

Des recommandations communes seront ensuite formulées.

IV - La fourniture de spécimens ne pouvant donner lieu à un engagement général compte tenu de la structure très diverse des sociétés d'édition, chaque éditeurs se prononcera directement sur ce point en proposant éventuellement la mise à disposition d'exemplaires pour la constitution de bibliothèques au sein des établissements.

V - La Direction des enseignements supérieurs constituera un groupe de travail réunissant professionnels de la microinformatique et universitaires, afin d'améliorer la communication, de favoriser le dialogue et l'échange d'informations entre professionnels et universitaires.

Une brochure rassemblant les explications concernant :

- les mesures de protection,
- les prix en France et à l'étranger,
- les écarts de prix entre solution micro-ordinateur monoposte, micro-ordinateurs en réseau, mini-ordinateur, etc...

sera remise au Ministère de l'Education Nationale pour les 31 décembre 1989 et sera diffusée par la suite dans les établissements

**PREMIERE SYNTHESE DES REPONSES AU POINTS TECHNIQUES EVOQUES PAR
L'EDUCATION NATIONALE DANS LE CADRE DU PROJET DE PROTOCOLE**

ANNEXE AU PROTOCOLE ENTRE LE SNPLM ET L'EDUCATION NATIONALE

1 - NOM DES RESPONSABLES EDUCATION DES EDITEURS

FRAME : Corinne COSTA
MULTILOG : Jean Paul THOMAS
3X : Monsieur USUNIER
MICROPRO : Martine BENHAMOU
WORDPERFECT : Khà TON-THAT
FEEDER : Mr GAUTRON
ISE-CEGOS : Mr MAQUET
ISI : Mr FRAUDEAU
LOTUS : Mr GABELLA
MICROSOFT : Mr Pierre Jean BONALDI
ORD'ASSIST : Mr VINAY
SAARI : Mr DEMNARD
DIGITAL RESEARCH : Mr POUHAER
MICROLAND : Mr CONERARDY
ORACLE : Mr De TOUCHET
PROLOGUE : Mme BONNEFEMME

**2 - REPONSES AUX POINTS 2 DU PROJET DE PROTOCOLE EDUCATION
NATIONALE SNPLM**

Pour chacun des problèmes techniques présentés au chapitre 2 du protocole, l'ensemble des éditeurs ont proposé des solutions adaptées à leurs produits et à leurs stratégies. Elles sont listées ci-dessous. En cas de problèmes d'interprétation ou technique, il est toujours possible de contacter le responsable Education de l'éditeur concerné ou le SNPLM.

**POINT 2A : UTILISATION PEDAGOGIQUE SIMULTANEE ET REITEREE DANS
LE TEMPS**

1 - Mise en place de politique permettant d'offrir des solutions par classe.

2 - Sans objet pour les produits qui occupe une place importante sur les disques durs (plusieurs mega octets)

3 - Des conditions spéciales éducation avec des versions "élève", "enseignant" à des tarifs et dans des packaging étudiés et adaptés aux besoins de l'enseignement.

4 - Possibilité de package pour une pédagogie simultanée et réitérée dans le temps sans réinstallation sur des équipement SI l'accès à ces derniers est contrôlé afin d'éviter les copies illicites.

POINT 2B : UTILISATION PONCTUELLE PAR UN ENSEIGNANT.

1 - Versions spécifiques d'évaluation.

2 - Sans objet pour les produits qui occupent une place importante sur les disques durs (plusieurs mega octets).

3 - Des conditions spéciales éducation avec des versions "élève", "enseignant" à des tarifs et dans des packagings étudiés et adaptés aux besoins de l'enseignement.

4 - Conditions spéciales éducation. Les problèmes spécifiques peuvent être résolus par le correspondant éducation.

5 - L'enseignant peut utiliser le logiciel installé sur des équipements destinés à l'enseignement pour la préparation de son cours ou d'une démonstration dans son cours.

POINT 2C : FIABILITE DES DISQUETTES

1 - Utilisation de la disquette clé.

2 - Sans objet pour les produits qui occupent une place importante sur les disques durs (plusieurs mega octets).

3 - Les produits n'ont pas de protections techniques.

4 - Ouvert à l'approche spécimen dans le cadre de bibliothèque de logiciels dont le fonctionnement à des règles bien définies.

POINT 2D : SAUVEGARDE DE L'ENVIRONNEMENT DE TRAVAIL

1 - Plusieurs milliers de produits utilisent ce type de protection sans problème. Faire connaître les problèmes éventuels au correspondant EDUCATION de l'éditeur.

2 - La protection rend impossible le type d'utilisation définie dans ce point, néanmoins l'éditeur est disposé à étudier avec l'éducation nationale une version spécifique.

3 - La sauvegarde des fichiers de travail est possible et suffisante dans le cas indiqué.

**RECHERCHE FONDAMENTALE
EN INFORMATIQUE**

(NOTE DE M. NIVAT)



LABORATOIRE
INFORMATIQUE
THEORIQUE ET
PROGRAMMATION

Paris, le 12 Mars 1990

Cher collègue,

M. Pierre Aigrain, président du Conseil Scientifique Electronique Informatique m'a demandé d'écrire un rapport sur la recherche fondamentale en Informatique en France. Ce rapport est destiné tout d'abord au Conseil présidé par P. Aigrain, pour aider à la définition d'une politique, mais il doit aussi constituer un argumentaire justifiant l'attribution de crédits incitatifs à cette recherche fondamentale par le Ministère de la Recherche et de la Technologie. A ce titre, il doit clairement faire comprendre ce qu'est l'activité de recherche fondamentale en Informatique, indiquer tout aussi clairement ce que l'on est en droit d'attendre d'une recherche par nature non finalisée et inscrite dans la longue durée. (C'est cette longue durée et le fait que nul ne peut prévoir quels seront les résultats d'une recherche fondamentale qui caractérisent cette dernière : à ce sujet nous tentons d'expliquer ce qu'est pour nous la recherche fondamentale dans le texte annexé à cette lettre).

En fait, nous croyons que la seule justification de la recherche fondamentale est l'existence de problèmes de nature scientifique et technique que l'on ne sait pas ou que l'on ne sait que très mal résoudre en l'état actuel de nos connaissances. L'intérêt que toute la communauté nationale peut prendre à la recherche fondamentale vient alors de ce que plusieurs de ces problèmes ont une importance pratique et économique incontestable et que notre impuissance actuelle à les résoudre de façon satisfaisante est un frein au développement et à l'utilisation de moyens et méthodes informatiques. Un tel problème, ressenti fortement par tous les professionnels de l'informatique ayant à écrire, mettre à jour ou mettre en oeuvre de gros logiciels est celui de la validité (validation) du logiciel : les méthodes actuelles

basées sur des spécifications algébriques ou logiques ne semblent pas être applicables à des logiciels réels de plusieurs dizaines ou centaines de milliers de lignes de code.

Bien peu de problèmes sont susceptibles d'une solution simple, radicale et définitive et par exemple nous n'imaginons pas de "vérificateur automatique universel de propriétés de logiciels". On peut imaginer par contre une lente modification des méthodes d'écriture de logiciels, basée sur une meilleure connaissance des phénomènes de programmation, qui aboutirait à l'écriture de logiciels différents plus faciles à mettre au point, à tester ou valider, et à mettre en oeuvre. Directement, et par le biais de la formation à laquelle elle est intimement liée la recherche fondamentale est la principale responsable de ces lentes évolutions.

M. Aigrain a explicitement souhaité que le rapport qui lui sera remis repose sur une large consultation des acteurs et utilisateurs de l'informatique. Une lettre semblable à celle-ci, accompagnée de la même annexe, est envoyée à un grand nombre de responsables industriels. Celle-ci est destinée à être largement diffusée auprès de chercheurs et d'enseignants-chercheurs, comme vous.

Nous espérons vivement que vous prendrez le temps de lire cette lettre et son annexe et de répondre aux questions suivantes :

- quels problèmes sont, à vos yeux, relevables d'une recherche fondamentale en ce sens que nous ne savons que très mal les résoudre et que nous avons le sentiment (parfois très obscur) qu'un progrès des connaissances est nécessaire à leur solution, ou à une meilleure solution ?

- à quels progrès peut-on s'attendre dans les 10 ans à venir, comme résultat d'une recherche fondamentale active et fructueuse ? Je ne vous demande pas d'étayer vos réponses à cette question, c'est bien ce que vous croyez possible, et qui constituerait un progrès indéniable que je vous demande et qui peut nous éclairer beaucoup. (Tant il est vrai que la croyance dans un certain progrès est un des

moteurs principaux de la recherche, ce qui n'exclut en rien que le progrès puisse venir de là où on ne l'attend pas ou se faire attendre très longtemps).

Des réponses à ces deux questions se dessineront, je l'espère, une image assez précise de ce qu'est et vers quoi tend la recherche fondamentale en informatique. Vous avez dû, comme nous tous, vous rendre compte que de nombreux décideurs (dont dépendent pour une large part le financement de la recherche) et la plupart de nos collègues d'autres disciplines ne savent finalement pas bien ce que cherchent les chercheurs fondamentalistes en informatique et comment leur activité s'articule avec le très rapide progrès technologique qui a multiplié la puissance des ordinateurs par des facteurs stupéfiants. Ce problème d'image de nos activités, d'image que nous donnons de nous-mêmes aux autres et même d'image que nous avons de nous-mêmes me préoccupe essentiellement. Car je crois que c'est bien parce que cette image est brouillée et fluctuante que nous n'arrivons pas à obtenir pour la recherche fondamentale en informatique un financement suffisant et stable.

Une fois dessinée cette image d'autres questions bien évidemment se posent :

- comment expliquer qu'une recherche fondamentale de qualité (la recherche française en informatique est incontestablement de qualité et même de grande qualité si l'on en juge par la présence française dans les colloques, les congrès et les revues internationales) apparaisse si peu en prise avec la réalité quotidienne de l'informatique. Peut-on expliquer ce phénomène par un manque de moyens en hommes et/ou en matériel, des défauts structurels qui empêcheraient la constitution des équipes nécessaires à la valorisation des acquis de la recherche, une incompréhension entre chercheurs et industriels ou toute autre raison que vous pouvez percevoir ?

Nous aimerions que vous vous sentiez très libres dans vos réponses qui ne sortiront pas du petit comité de réflexion formé pour écrire le rapport en question.

D'avance je vous remercie de votre réponse et je vous prie de croire, cher collègue, en mes sentiments très dévoués et les meilleurs.

Maurice NIVAT.

Sur la recherche fondamentale en Informatique

Maurice NIVAT

Ce qui peut distinguer le mieux la recherche fondamentale de la recherche appliquée en Informatique comme dans toutes les autres sciences c'est la *durée*. On pourrait dire que la recherche fondamentale est celle dont les retombées sont imprévisibles ou ne sont pas prévisibles avant une dizaine d'années.

Le progrès très rapide de la technologie informatique fait croire à tort que l'exploitation et la diffusion d'idées neuves y est extrêmement rapide. De très nombreux exemples sont là pour nous rapeler qu'au contraire il s'écoule bien dix ans entre l'émergence d'une idée et sa réalisation ou incorporation dans des architectures matérielles ou logicielles effectivement vendues et utilisées. Le système UNIX en fournit un très bon exemple : sa diffusion, hors d'universités dans lesquelles il restait objet d'étude et d'expérimentation n'est pas intervenue moins de 10 ans après la première maquette. Il en va de même des systèmes de gestion de bases de données relationnelles et en ce qui les concerne on peut constater que l'utilisation de ces systèmes ne s'est pas généralisé dès leur apparition sur le marché, il faut encore le temps que les utilisateurs s'habituent aux nouveaux concepts sur lesquels ils reposent. En ce qui concerne l'architecture d'une machine mettant en oeuvre quelques idées novatrices il est évident que dix ans doivent s'écouler entre la décision de réaliser une maquette à partir de ces idées et la commercialisation éventuelle d'une machine découlant de la maquette : la construction de cette dernière qui peut durer deux ans si l'on s'y prend bien n'est qu'une faible partie du travail. Il faut encore tester cette maquette et mesurer, avec les moyens très rudimentaires dont on dispose, son efficacité : ceci suppose que la machine soit dotée de logiciels de base et que l'on puisse traiter des exemples en vraie grandeur. Et il n'est pas facile de tirer une conclusion de ces exemples, tant les particularités de la machine et de son

logiciel sont interdépendants sans que les liens soient bien explicites. Si, c'est généralement ce que l'on fait, on met en lumière une classe de problèmes à laquelle l'architecture est bien adaptée on sait mal tracer les contours du champ de problèmes que la machine résoud bien. Le cas du calcul parallèle c'est-à-dire de l'emploi de machines comportant plusieurs processeurs pouvant exécuter des morceaux de programmes différents, montre bien que la question est toujours très ouverte et que l'on n'est pas sorti du stade expérimental : toutes les machines ont des vertus, aucune ne s'impose et tous les chercheurs qui les utilisent trouvent des "défauts" à chacune. Malheureusement on ne sait remédier à ces défauts sans remettre l'équilibre d'ensemble en question et l'on ne dispose d'aucun modèle théorique vraiment satisfaisant pour étudier les mécanismes complexes de la communication entre processus ou de la distribution des tâches entre eux.

Une autre différence essentielle entre recherche fondamentale et recherche appliquée réside dans l'*objectif* et dans la *méthode*.

Quand on cherche à résoudre un problème, à améliorer des méthodes, des outils (matériels et logiciels) pour aller plus loin dans la résolution d'un problème ou augmenter l'efficacité de la recherche d'une solution qu'on connaît on peut, dans un souci de rapidité ou d'économie ne chercher que dans la panoplie des concepts des idées et des méthodes déjà assez bien connues et écarter les idées trop éloignées du domaine en question. La recherche appliquée à laquelle on se livre alors peut être extrêmement utile, extrêmement efficace et mérite certainement que les meilleurs esprits s'en occupent et que l'on mobilise des moyens importants. Les contraintes de coût et de temps inhérentes à la recherche appliquée lui sont nécessaires car c'est dans leur existence que la recherche appliquée puise l'essentiel de son intérêt qui est d'une optimisation constante de l'adaptation des moyens aux fins poursuivies.

Clairement différente est la recherche fondamentale qui s'attache à développer une idée originale sans trop savoir vers quoi l'on tend. Cela est si vrai que sans doute les inventeurs d'UNIX ou du Bitmap essayaient au début d'assembler des idées qui ne pouvaient être vraiment exploitées avec la technologie du moment, et il a fallu un formidable progrès technologique pour que ces deux systèmes puissent devenir les très beaux

outils d'usage quotidien que nous connaissons. La contrainte fondamentale, dans la recherche fondamentale, est la contrainte *d'originalité* : il s'agit bien de trouver et d'exploiter des idées neuves et de faire progresser nos connaissances des phénomènes. Nous ne connaissons pas l'inventeur de la machine parallèle qui dans dix ans triomphera et sera adoptée par de très nombreux utilisateurs, nous ne pouvons pas le connaître, même s'il existe, ce qui n'est pas sûr : ou bien il n'existera pas de telle machine dans dix ans, ou bien c'est l'une des machines existantes qui, transformée et rendue utilisable par une meilleure connaissance de son fonctionnement permettant de la doter d'un "bon" logiciel qui finira par triompher, ou bien cet inventeur agite actuellement des idées assez orthogonales à celles qui sous-tendent les machines existantes pour que nous n'en voyions pas aujourd'hui l'intérêt, ou, comme on dit n'y "croyions" pas. L'invention est toutefois le résultat d'un long travail, d'un gigantesque travail collectif, rarement spectaculaire et dont on a trop tendance à minimiser l'importance. Et ce long travail peut être nécessaire sans que de fulgurantes inventions en découlent.

Un des sujets les plus préoccupants pour les utilisateurs d'informatique est la validité du logiciel. Comment s'assurer qu'un volumineux logiciel (c'est désormais en millions de lignes de code qu'il convient de les mesurer) exécute bien ce que l'on désire qu'il exécute ? Disons que pour l'instant on n'a pas de solution, les programmes de validation de logiciel (qui ressemblent beaucoup à des démonstrateurs automatiques de théorèmes dans la mesure où l'assertion "ce programme est correct" est bien un théorème de la logique qui décrit la sémantique du programme), pour intéressants qu'ils soient, ne parviennent guère à valider des programmes de quelques dizaines de lignes. Les responsables de gros logiciels qui doutent de l'applicabilité de ces méthodes formelles à la validation de millions de lignes de code ont certainement toutes les raisons pour ce faire : l'impossibilité est mathématique, elle vient d'une complexité combinatoire exponentielle dont aucun ordinateur ne viendra jamais à bout. Mais ces mêmes responsables ont souvent le tort de minimiser l'importance des acquis théoriques dans les progrès très réels faits par les logiciels : tout l'effort de compréhension de ce qu'est réellement un programme, effort théorique mais aussi très pratique, a donné naissance à de nouveaux concepts, de nouvelles méthodes et des

outils de conception ou de mise au point de logiciels qui fait que l'on ne programme plus du tout aujourd'hui comme il y a vingt ans. L'avenir, face à la complexité du problème de validation ne peut guère s'envisager que sous la forme d'une évolution de toute la méthodologie d'analyse des problèmes, de spécification des solutions et d'écriture de logiciels différents de ceux qu'on connaît par plus de modularité et plus de lisibilité : on identifiera des "fonctions" logicielles auxquelles correspondent des "morceaux" de programme ou de système, d'une taille qui permettra leur certification et qui d'autre part seront écrits de façon à pouvoir être réutilisés dans différents contextes. Mais ne nous leurrions pas, il y a encore beaucoup de travail à faire pour donner un sens précis et suffisamment général aux concepts sous-jacents, pour étudier dans un cadre mathématique ou logique les conséquences de telle ou telle définition et à expérimenter, sur des systèmes assez gros ces mêmes concepts. Et la modification radicale de toute la méthodologie d'écriture de logiciels qui doit s'en suivre ne sera effective que quand une nouvelle génération de programmeurs sera formée à une programmation différente et aura acquis les connaissances d'algèbre, de logique et de sémantique nécessaires, ce qui, à l'évidence, prendra plus de dix ans.

On est amené à considérer qu'il y a au moins trois types de recherches fondamentales :

- le premier est constitué par les problèmes insolubles en l'état actuel des connaissances. par exemple, la reconnaissance de la parole "multi-locuteurs", la traduction automatique. Cette dernière qui a fait couler tant d'encre est peut-être tout à fait impossible, en ce sens qu'aucune machine ne fera jamais aussi bien qu'un traducteur humain (dans la mesure où celui-ci prend en compte tout un contexte de sens que l'on ne sait absolument pas comment inscrire dans la mémoire d'un ordinateur). Sûrement elle est extrêmement difficile en ce sens que les plus petits progrès dans la manipulation de textes en langue naturelle, supposant une analyse syntaxique et sémantique, exigent la constitution de grammaires et de dictionnaires syntaxique sémantique reposant sur une connaissance beaucoup plus intime de la langue que les dictionnaires et grammaires traditionnels et donc une recherche en linguistique dont on ne voit pas le bout ;

- le second est constitué par l'étude de modèles et de structures mathématiques et logiques dont on sait, par expérience, qu'ils sont bons c'est-à-dire reflètent bien les propriétés de systèmes informatiques que l'on cherche à construire ou à améliorer. La difficulté est que ces structures sont complexes et pour l'instant toujours mal connues : nous pensons, par exemple, aux systèmes de réécriture de termes à la base de toute méthode de preuve, aux automates finis que l'on retrouve partout comme moyen de décrire flots de données et structures de contrôle, aux diverses structures d'évènements qui modélisent les systèmes répartis, distribués ou parallèles. Tout progrès, même de nature très théorique et semblant fort éloigné de toute application possible est susceptible d'en recevoir le jour où les problèmes sont mieux compris que l'on songe à la cryptographie et au rôle qu'y joue désormais la théorie des groupes simples ou l'arithmétique sous-jacente au test de primalité ! Nous rattachons à ce type de problème toute l'algorithmique, qui est un combat de tous les jours contre la complexité des phénomènes et des structures, ceci valant aussi bien pour l'algorithmique sur papier que pour celle qui s'écrit sur le silicium ;

- le troisième type de problème est constitué par l'expérimentation de nouveaux systèmes matériels et logiciels. Ce ne sont pas les idées qui manquent pour, par exemple, construire des machines parallèles universelles ou dédiées. Il y a déjà une belle collection de telles machines, mais on est loin d'avoir épuisé toutes les possibilités, aucune architecture de machine parallèle n'a triomphé (au sens qu'un vaste consensus veuille que ce soit la bonne architecture) et aucune n'est dotée d'un logiciel et d'outils de création de logiciels en rien comparable à ceux qui existent pour des machines séquentielles ordinaires ou vectorielles (ces dernières sont des machines SIMD à parallélisme très contraint). La seule méthode là est d'expérimenter machines et logiciels : nous avons déjà dit qu'une telle expérimentation exige des moyens et du temps, beaucoup de temps (le fait que les "connexions machines" existent physiquement et que l'on puisse en installer à l'ETCA, l'INRIA ou l'Institut de Physique du Globe n'implique pas la fin de l'expérience : tout reste à faire pour savoir utiliser cette machine et évaluer le gain qu'elle apporte à la solution de nombreux problèmes).

Que l'on se situe dans l'un ou l'autre de ces trois cas le premier résultat attendu de la recherche fondamentale est l'extension de nos connaissances. On oublie trop que l'informatique est d'abord une physique des signes et des symboles, on ne travaille jamais que sur des données représentées et ce sont les transformations que l'on peut opérer sur ces représentations que l'on étudie. Si une certaine informatique théorique est assez mathématisée pour que ses résultats s'énoncent comme des théorèmes ce qu'est qu'une petite partie de l'informatique fondamentale (notons quand même qu'elle a tendance à croître, un nombre croissant de problèmes étant formalisés dans un cadre mathématique) : des chapitres très importants de la recherche fondamentale en informatique débouchent plutôt sur des méthodologies et des "savoir-faire" où, face à un problème, l'expérience accumulée et un "sens informatique" (au sens où l'on parle de "sens physique") sont peut-être plus utiles qu'un vaste bagage de connaissances formalisées. Si l'on voit fleurir dans les noms de cursus universitaires des termes tels que "génie logiciel" ou "ingénierie informatique" c'est bien parce que la conception d'un circuit, d'une carte, d'une machine ou d'un logiciel ne relève pas d'un processus entièrement descriptible en termes formels ne serait-ce que parce que l'art du compromis y joue un grand rôle, compromis nécessaire entre l'optimisation des performances et la lisibilité, la reproductibilité et la transparence de l'objet conçu et réalisé, par exemple un morceau de logiciel. Chacun sait que l'on peut écrire des programmes de quelques lignes dont la compréhension constitue un véritable casse-tête ou dont on ne sait pas prouver la convergence. L'irruption des sciences cognitives en informatique répond bien à un besoin qui est celui de rapprocher le mode de fonctionnement de certains programmes des modes de raisonnement d'un être humain cherchant à résoudre le même problème : le champ d'application privilégié de ces programmes, baptisés systèmes experts, est assez naturellement élu du contrôle de systèmes complexes, comme une centrale thermique ou nucléaire, où il s'agit de réagir à de très nombreux stimuli extérieurs qui sont des variations normales ou anormales des multiples paramètres caractérisant l'état du système à un moment donné. A défaut d'automatiser entièrement le système on est amené à concevoir des programmes de contrôle qui permettent à un opérateur humain d'intervenir dans certains cas (le plus souvent extrêmes, précédant ce qui

serait une déstabilisation totale du système contrôlé : comme on l'imagine cette double commande par la machine et par l'homme exige qu'en quelque sorte ils agissent de la même façon. Sans doute plus qu'aucune autre technologie, l'informatique doit se soucier de l'adéquation des outils qu'elle fabrique à leur destination future et aux conditions dans lesquelles ils seront utilisés puisque ces outils s'adressent plus à l'esprit qu'à la main et interviennent directement dans les mécanismes intellectuels qui sous-tendent les activités humaines.

On ne peut parler d'extension ou d'amélioration de nos connaissances sans évoquer les problèmes de formation, d'éducation, de transmission des connaissances et d'apprentissage des méthodes. La recherche fondamentale en informatique agit directement sur la formation puisque c'est elle qui engendre et affine les concepts de base, ceux sur lesquels tout l'édifice repose. On oublie trop souvent que fait aussi partie de la recherche fondamentale l'indispensable et constante réflexion sur les concepts de base, leur simplification, la mise au point de formalismes unificateurs qui doit se traduire par une plus grande rapidité et une plus grande sûreté dans l'assimilation des concepts. Pourtant l'auteur d'un bon manuel fait peut-être plus pour l'informatique que celui qui démontre quelques propriétés plus ou moins ésotériques et tout enseignant d'informatique sait bien que nous manquons de bonnes synthèses et que de nombreuses théories sont encore pleines de "trous" (sans doute peut-on expliquer cela par la jeunesse et l'absence de traditions de la discipline).

De toute façon, il est bien difficile de tracer les contours de la recherche fondamentale, de savoir définir et déterminer ce qui est de la recherche et ce qui n'en est pas, ce qui est de la recherche fondamentale par rapport à ce qui est de la recherche plus appliquée. C'est encore l'opinion de toute la communauté des chercheurs qui, là comme ailleurs, est la meilleure mesure de la qualité d'une recherche et il faut admettre là comme ailleurs qu'il y ait des recherches qui n'aboutissent à rien ou que très lentement et par des biais détournés et imprévisibles. Ce qui est le plus coupable c'est d'encourager ou de poursuivre une recherche dont on n'a pas les moyens intellectuels, humains ou financiers, car si l'existence de ces moyens ne garantit pas la réussite, leur absence aboutit sûrement à la faillite.

**L'INFORMATIQUE
DANS LES PREMIERS CYCLES SCIENTIFIQUES**

(M. LUCAS)

- . Activités du groupe SPECIF
- . Journées SPECIF (NANTES, Mars 1990)
- . Document de travail

SPECIF
Groupe de travail

**Enseignement de l'informatique
dans les premiers cycles scientifiques**

rapport d'activité

membres du groupe

Jacques BARRE	Rennes
Jean-Paul BERTRANDIAS	Grenoble
Claude BONNEMOY	Clermont-Ferrand
Jean-Claude BOUSSARD	Nice
Gérard FERRAND	Orléans
Daniel HERMAN	Rennes
Brigitte JARRAY	Nancy
Pierre JULLIEN	Marseille
Michel LUCAS	Nantes
Guy-René PERRIN	Besançon
Régine RAYNAUD	Toulouse
Pierre-Claude SCHOLL	Grenoble
Jean-Pierre STEEN	Lille

objectifs du groupe de travail

Dès la rentrée 90, mettre en place une action cohérente sur les DEUG. La ligne directrice, suite aux travaux des Journées SPECIF 1988 à Besançon, est l'introduction de l'enseignement de l'Informatique en tant que discipline dans les DEUG scientifiques.

Cette mise en place se fera à partir de documents permettant à la communauté informatique de disposer d'un cadre de travail argumenté.

résultats attendus

- rédaction d'un Livre Blanc par un Comité élargi: mise en forme et relevé de conclusions
- détermination de lieux d'expérience: conditions d'expérimentation
- à long terme: écriture de manuels Introduction à l'informatique, à l'usage des Premiers Cycles

modalités

Le travail se divisera en plusieurs actions distinctes:

- dans un premier temps, des Journées, ouvertes aux informaticiens intéressés par l'enseignement de l'informatique dans les DEUG. Trois jours seront consacrés à mettre sur pied une réflexion sur les cursus informatique, la discipline informatique et étudier les possibilités d'intégration de l'informatique en tant que discipline dans les DEUG. Les réflexions résultant de ces trois jours seront diffusées dans notre communauté pour analyse et commentaires.

titre: Enseigner l'Informatique en tant que discipline dans les DEUG scientifiques

nombre de participants: 50 à 60 (un par Université), sur invitation personnelle lancée par le Comité de programme.

- dans un deuxième temps, un Colloque de deux jours, ouvert à toute personne intéressée par l'informatique dans les DEUG, afin de présenter les résultats de la réflexion précédente.

titre: L'enseignement de l'Informatique dans les premiers cycles scientifiques
(deuxième colloque)

nombre de participants: 150 à 200. Il s'agit potentiellement de la même clientèle qu'à Besançon. Les participants à ce premier colloque seront donc invités en premier lieu. Un élargissement des invitations est prévu, afin de donner le plus de retentissement possible à cette action. Les invitations seront lancées par le Comité de programme.

- dans un troisième temps, un Comité élargi rédige le Livre Blanc, en s'aidant des conclusions des Journées, des débats du Colloque et des réactions de la communauté informatique. Ce Livre Blanc sera diffusé très largement.

calendrier

Il s'agit ici des grandes lignes:

- | | |
|-----------------|--|
| 4,5/9/89 | 1ère Réunion du Comité de Programme |
| 9/11/89 | 2ème Réunion du Comité de Programme |
| 30/11/89 | les participants au Colloque de Besançon sont prévenus de la suite de l'action
lancement enquête sur les Licence, Maîtrise, DEA, DESS informatiques |
| 15/12/89 | lancement des invitations pour les Journées |
| 25/1/90 | 3ème Réunion du Comité de Programme |
| 15/2/90 | lancement des invitations pour le Colloque |
| 27, 28, 29/3/90 | Journées: Enseigner l'Informatique en tant que discipline dans les DEUG scientifiques
première rédaction des conclusions des Journées |
| 26, 27/4/90 | Colloque: L'enseignement de l'Informatique dans les premiers cycles scientifiques (deuxième colloque) |
| 31/5/90 | fin de rédaction de la première version du Livre Blanc |
| 30/6/90 | diffusion Livre Blanc, version définitive |

Il s'agit donc d'un travail de longue haleine, à la mesure de l'enjeu: la reconnaissance de l'informatique en tant que discipline à part entière et égale avec la Physique, la Chimie, la Biologie et autres Mathématiques.

Il est tout à fait évident que nous souhaitons que les travaux de ce groupe soient les plus représentatifs possibles. Un certain nombre d'actions d'accompagnement sont donc prévues, et nous vous demandons d'y participer:

- réponses précises aux enquêtes sur les contenus des Licence, Maîtrise, DEA, DESS d'Informatique,
- propositions de maquettes d'enseignement de l'informatique en DEUG (et au delà),
- réflexions sur l'image de la discipline informatique (image actuelle, à donner, ...)

Manifestez votre intérêt (ou votre désaccord !) en écrivant dès maintenant à:

M. Lucas, Ecole Nationale Supérieure de Mécanique de Nantes
1 rue de la Noë
44072 Nantes Cedex 03

Journées SPECIF 1990
Nantes

**Enseigner l'Informatique en tant que discipline dans les DEUG
scientifiques**

programme provisoire

Rappel des conclusions des Journées de Besançon

il est urgent et indispensable d'introduire effectivement l'enseignement de la discipline informatique dans les premiers cycles.

Ce qui est à l'heure actuelle enseigné est plus proche en général de l'enseignement de l'informatique au service de toutes les disciplines. Souhaitable, voire même indispensable, cet enseignement ne suffit plus en ce qui concerne les filières scientifiques devant conduire par exemple aux enseignements professionnels en informatique.

La non reconnaissance de la place nécessaire à une véritable formation à l'informatique a été dénoncée.

Ainsi, il a été relevé que l'informatique est une des rares disciplines (la seule ?) ne bénéficiant pas d'une véritable préparation avant l'entrée en Licence. Si l'on compare le nombre d'heures consacré aux autres disciplines tant dans l'enseignement secondaire que dans les Premiers Cycles universitaires à celui consacré à l'informatique en tant que discipline, il est évident que les étudiants ne peuvent avoir aucune idée véritable de ce qu'est l'informatique. En fait, l'état d'esprit informatique est très différent de ce qu'ils connaissent

Il n'est donc pas étonnant qu'ils se détournent des formations comme les Licences et Maîtrises d'informatique, puisqu'ils n'y sont pas préparés. La plus grande partie des responsables de ces cursus a noté ces dernières années un effondrement sensible du nombre d'étudiants en provenance des DEUG, ce qui est jugé comme très mauvais à long terme.

La prise en compte de la discipline informatique dans les programmes des DEUG doit faire l'objet d'une réflexion approfondie, afin de lui donner sa juste place.

objectif des Journées 1990

L'objectif est de mettre en place, dès la rentrée 90, une action cohérente sur les DEUG. Il s'agit de réfléchir sur les contenus à enseigner, les relations avec les autres cursus, la mise en œuvre possible. Le fil directeur est

enseigner l'informatique en tant que discipline

Les résultats attendus sont les suivants:

- rédaction d'un Livre Blanc par un Comité élargi: mise en forme et relevé de conclusions
- détermination de lieux d'expérience: conditions d'expérimentation
- à long terme: écriture de manuels Introduction à l'informatique, à l'usage des Premiers Cycles

Il est très important qu'un effort de réflexion collectif, engageant la communauté des informaticiens, soit engagé.

C'est pour participer à cet effort que vous êtes invité(e) aujourd'hui.

modalités

L'ensemble de cette action se découpe en plusieurs phases distinctes:

- dans un premier temps, trois jours (appelés Journées, ouverts aux informaticiens intéressés par l'enseignement de l'informatique dans les DEUG. Ces trois jours seront consacrés à mettre sur pied une réflexion sur les cursus informatiques, la discipline informatique, et étudier les possibilités d'intégration de l'informatique en tant que discipline dans les DEUG.

titre: Enseigner l'Informatique en tant que discipline dans les DEUG scientifiques

nombre de participants: 50 à 60, sur invitation du Comité de Programme.

- dans un deuxième temps, deux jours (appelés Colloque), ouverts à toute personne intéressée par l'informatique dans les DEUG, afin de présenter les résultats de la réflexion précédente.

titre: L'enseignement de l'Informatique dans les premiers cycles scientifiques (deuxième colloque)

nombre de participants: 150 à 200, sur invitation largement diffusée.

- dans un troisième temps, un Comité élargi rédige le Livre Blanc, en s'aidant des conclusions des Journées et des débats du Colloque.
- des débats avec la communauté informatique sont organisés, soit localement (par l'intermédiaire des membres du Comité), soit nationalement (par enquêtes, ou assemblée générale SPECIF), afin de faire connaître et mettre en œuvre les propositions.

Les Journées d'Etude

titre Enseigner l'Informatique en tant que discipline dans les DEUG scientifiques

date - les 27, 28 et 29 Mars 1990.

lieu Faculté des Sciences et des Techniques de Nantes

participants - 50 à 60 personnes, sur invitation fermée:

- . membres du Comité de Programme,
- . une personne par Université: informaticien (ne), intéressé(e) par les problèmes de formation en DEUG.

modalités Le travail se déroulera suivant trois modes alternés:

- des exposés, de présentation, de provocation, ...
- des travaux en commissions, sur des thèmes présentés et animés par des membres du Comité de programme. Les travaux de chaque commission sont résumés sur des transparents, qui servent à la présentation en séance plénière et à la préparation de la synthèse finale.
- des synthèses en réunion plénière, où les résultats de chaque commission sont présentés et débattus devant l'ensemble des participants. Les débats permettent d'adopter certains points, ou de mettre en valeur les arguments pour et contre telle ou telle proposition.

Une séance de synthèse finale permet de tracer les grandes lignes du document qui sera le résultat des Journées.

horaire prévu

<u>mardi 27/3/90</u>	10h	ouverture des Journées discours d'accueil
	10h30	exposés (30/45 mn chacun): - résultats du colloque SPECIF/Syntec sur la formation des informaticiens - enseigner l'informatique en tant que discipline en DEUG (pour donner l'esprit des Journées)
		débat
	13h	déjeuner
	14h30	travaux en commission
	16h30	session plènière
	18h30	fin de la première journée
<u>mercredi 28/3/90</u>	9h	exposé (30 mn) - L'enseignement de l'informatique en Europe
		débat
	10h	travaux en commission
	12h	session plènière
	13h	déjeuner
	14h30	travaux en commission
	16h30	session plènière
	18h30	fin de la journée
<u>jeudi 29/3/90</u>	9h	session plènière - premier bilan des Journées
	9h30	travaux en commission
	11h30	session plènière
	13h	déjeuner
	14h30	session plènière synthèse finale
	16h30	clôture des Journées

Pour assurer une bonne continuité et une efficacité maximale, il est souhaité que les participants restent si possible les trois jours.

commissions

Les participants se répartiront entre les commissions suivantes:

- **articulations** C. Bonnemoy, R. Raynaud

- avec les formations précédant le Bac, incomplètes et dispersées,
- avec les enseignements de niveau Bac+2 qui (CNAM excepté) ne concernent que l'informatique appliquée
- avec les formations de niveau Second Cycle exclusivement professionnelles (sauf les cursus universitaires d'Informatique)
- avec les enseignements de niveau correspondant en Europe.

- **méthodes pédagogiques** P.C. Scholl, B. Jafray

- contraintes de l'enseignement de masse
- Enseigner l'informatique sans faire programmer
- qu'est ce qu'un TP en informatique ?
- formes d'échange entre enseignants

- **enseigner l'informatique à égalité avec les autres disciplines**

J.C. Boussard, P.C. Scholl, P. Jullien, G.R. Perrin

- réflexion fondamentale sur l'informatique
- faut-il un DEUG tridisciplinaire ?
- contenus de l'enseignement en DEUG scientifique
- influence des contraintes de temps

- **Bibliographie** J.P. Bertrandias, G. Ferrand

- bibliographie à conseiller pour qui veut découvrir ce qu'est l'informatique
- comment constituer une 'bibliothèque ouverte' ?

- **matériel/logiciel**

J.P. Steen, J. Barre

- implications pédagogiques du choix du matériel et du logiciel
- possibilités actuelles et prospectives matériel/logiciel
- moyens à mettre en oeuvre

D'autres thèmes pourront être mis en œuvre, en fonction des suggestions qui nous parviendront.

Comité de programme

Jacques BARRE	Rennes
Jean-Paul BERTRANDIAS	Grenoble
Claude BONNEMOY	Clermont-Ferrand
Jean-Claude BOUSSARD	Nice
Gérard FERRAND	Orléans
Daniel HERMAN	Rennes
Brigitte JARAY	Nancy
Pierre JULLIEN	Marseille
Michel LUCAS	Nantes
Guy-René PERRIN	Besançon
Régine RAYNAUD	Toulouse
Pierre-Claude SCHOLL	Grenoble
Jean-Pierre STEEN	Lille

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

**Enseigner l'informatique en tant que
discipline dans les premiers cycles
scientifiques**

compte-rendu

INTRODUCTION

1. LA DISCIPLINE INFORMATIQUE

- 1.1 Apports de l'informatique en tant que discipline
- 1.2 Les contours de la discipline informatique
- 1.3 Comment faire reconnaître notre discipline

2. PROFIL D'UN ENSEIGNEMENT D'INFORMATIQUE EN DEUG

- 2.1 Pour un enseignement de l'informatique dans un DEUG bi- ou tri-partite
- 2.2 Principes de l'enseignement de l'Informatique dans un DEUG
- 2.3 Un exemple de découpage de l'enseignement de l'Informatique dans un DEUG
- 2.4 Un exemple d'option Informatique
- 2.5 Articulations avec les autres enseignements

3. METHODES PEDAGOGIQUES

- 3.1 Le contexte
- 3.2 Modalités pratiques
- 3.3 Pour une bibliographie de l'Informatique

4. LES MOYENS NECESSAIRES

- 4.1 L'ordinateur en DEUG
- 4.2 Les fonctionnalités attendues
- 4.3 Les catégories de personnel
- 4.4 Le chiffrage d'un atelier informatique

CONCLUSION

ANNEXES

- A1. Liste des participants
- A2. DEUG scientifiques: pourquoi ? pour qui ? (M. Coste)
- A3. Enseigner l'informatique en DEUG (J.C. Boussard)
- A4. Enseignement de l'Informatique en Europe (J.P. Steen)
- A5. Enseigner l'informatique en tant que discipline, c'est possible ! L'exemple de Metz
- A6. L'informatique, discipline fondamentale en Deug A: un exemple à Grenoble
- A7. Bibliographie

Remerciements

Le Doyen de la Faculté des Sciences et des Techniques de l'Université de Nantes a permis la tenue de ces Journées en mettant à notre disposition des salles. Leur nombre et leur capacité d'accueil ont autorisé la tenue des sessions plénières ainsi que celle de quelques commissions. Chacun connaît le manque d'espace actuel dans les Universités, et pourra apprécier en conséquence l'aide apportée.

Par ailleurs, certaines commissions ont pu se réunir dans des salles aimablement prêtées par le Laboratoire d'Informatique (LIST) et le CFIAP. Que leurs directeurs respectifs en soient chaleureusement remerciés.

Membres du groupe de travail 'Enseignement de l'informatique dans les Premiers Cycles'

Les Journées de Nantes ont eu lieu grâce à l'intense travail de préparation effectué par les personnes suivantes:

BARRE Jacques	Rennes
BERTRANDIAS Jean-Paul	Grenoble
BONNEMOY Claude	Clermont-Ferrand
BOUSSARD Jean-Claude	Nice
FERRAND Gérard	Orléans
HERMAN Daniel	Rennes
JARAY Brigitte	Nancy
JULLIEN Pierre	Aix-Marseille
LUCAS Michel	Nantes
PERRIN Guy-René	Besançon
RAYNAUD Régine	Toulouse
SCHOLL Pierre-Claude	Grenoble
STEEN Jean-Pierre	Lille

Journées SPECIF 1990
Nantes - 27, 28, 29 Mars 1990

Enseigner l'informatique en tant que discipline dans les DEUG scientifiques

compte-rendu

Introduction

En 1988, le Premier Colloque sur l'enseignement de l'informatique dans les Premiers Cycles scientifiques de Besançon s'était terminé sur un constat:

il est urgent et indispensable d'introduire effectivement l'enseignement de la discipline informatique dans les premiers cycles.

Plusieurs lignes force avaient été identifiées:

- continuer à assurer un enseignement de l'outil informatique en DEUG, en attendant sa prise en charge par le secondaire,
- demander un effort de formation de collègues, en réclamant des moyens analogues à ceux qui ont été dégagés pour le secondaire.
- mettre sur pied dans chaque université des comités de réflexion interdisciplinaires, chargés de réfléchir au contenu de l'enseignement de l'Informatique, et à sa mise en place.
- militer pour la création d'un enseignement de la discipline informatique en DEUG, en particulier pour attirer des étudiants en Licence et Maîtrise d'Informatique,

En 1989, un groupe de travail s'est attaqué au dernier point. Son objectif: dès la rentrée 1990, favoriser la mise en place d'une action cohérente sur les DEUG. Pour cela, les actions retenues sont les suivantes:

- favoriser la sensibilisation et la réflexion de la communauté informatique sur ce problème.

C'est le rôle des Journées d'étude qui viennent de se tenir à Nantes.

- disséminer le résultat de ces réflexions au sein de notre communauté, afin d'enregistrer les réactions aux propositions qui sont faites

C'est le rôle de la publication de ce compte-rendu dans le bulletin de SPECIF.

- présenter et confronter nos propositions aux besoins de nos partenaires (autres disciplines, décideurs, profession, ...)

C'est le rôle du Deuxième Colloque sur l'enseignement de l'informatique dans les Premiers Cycles scientifiques, qui aura lieu à Lille les 18 et 19 Septembre 1990.

- toucher un public le plus large possible, pour faire connaître nos souhaits et nos positions quant à l'enseignement de l'informatique en tant que discipline.

C'est le rôle du Livre Blanc qui sera publié en fin d'année. Il contiendra l'ensemble des réflexions issues des travaux que nous venons de mentionner.

Les Journées de Nantes ont réuni 45 participants, représentant 25 Universités délivrant un DEUG scientifique. Une invitation avait été adressée à chacune des 49 Universités délivrant un tel diplôme. L'invitation a été expressément envoyée à des représentants de la 24ème section, à l'exclusion de toute autre section du CNU. Si la tenue des Commissions de Spécialistes a empêché au dernier moment quelques personnes de venir, force est de constater que près de la moitié des Universités n'ont pas délégué de représentants, ni même répondu à l'invitation qui était faite. Comme quoi la sensibilisation de notre communauté à ce problème est encore de mise aujourd'hui !

La méthode de travail utilisée a été la suivante:

- présentation de quelques exposés de réflexion:

- . Michel COSTE (PROMOSCIENCES)
DEUG scientifiques: pourquoi ? pour qui ? (texte en annexe 2)
- . Jean-Claude BOUSSARD (Nice)
L'informatique dans les DEUG scientifiques (texte en annexe 3)
- . Jean-Pierre STEEN (Lille)
L'enseignement de l'informatique en Europe (texte en annexe 4)

Par ailleurs, un certain temps a été consacré à rapporter les conclusions du Colloque sur la formation des informaticiens qui s'était tenu la semaine précédente à Paris.

- travail en commissions:

- | | |
|-------------------------|--|
| . contenu | la discipline informatique; quel contenu enseigner en DEUG ?
(document préparatoire: J.C. Boussard) |
| . méthodes pédagogiques | mise en œuvre de l'enseignement de l'informatique: cours, TD, TP
(document préparatoire: J.P. Bertrandias, P.C. Scholl) |
| . moyens | matériel, logiciel, locaux, personnel
(document préparatoire: J.Barré) |
| . articulations | liens avec le secondaire, le Deuxième Cycle, les autres cursus
(document préparatoire: C. Bonnemoy, R. Raynaud) |
| . bibliographie | comment définir la bibliothèque idéale ?
(document préparatoire: P. Bertrandias, G. Ferrand) |

Le travail s'est fait à partir des différents exposés, ainsi que de notes préparées par les membres du groupe de travail chargés d'animer ces différentes commissions.

- travail en séances plénières:

Il s'est agi d'écouter, d'analyser et de critiquer les rapports des différentes commissions, afin d'arriver soit à des propositions de consensus, soit à élucider le pourquoi de désaccords.

C'est cet ensemble de travaux qui est rapporté ci-après.

1. LA DISCIPLINE INFORMATIQUE

Tous nos travaux ont été fondés sur le postulat suivant:

La discipline informatique existe.

Si cette affirmation nous paraît évidente, elle ne l'est pas autant pour un grand nombre de collègues des autres disciplines. Ils attendent de notre part un discours permettant de cerner les fondements, les moyens d'expression, la complexité, les niveaux d'abstraction, la problématique spécifique de cette discipline. Malheureusement, nous, informaticiens, ne la définissons souvent que à coup de *"l'informatique, ce n'est pas ça; ceci n'est pas de l'informatique, ..."*, ce qui n'est guère enthousiasmant. La discipline informatique doit être définie pour elle-même, et non pas de manière négative. Nous sommes en position de faiblesse, car nous ne montrons rien. L'exemple de *"l'informatique dans les classes préparatoires"*, introduite à grand renfort de manuels et de publicité, montre ce que nous ne faisons pas.

Par ailleurs, l'informatique est définie dans certains dictionnaires. Par exemple, dans le Grand Larousse en 5 volumes, édition 1989, on trouve la définition suivante:

1. *Science du traitement automatique et rationnel de l'information considérée comme le support des connaissances et des communications.*

2. *Ensemble des applications de cette science, mettant en oeuvre des matériels et des logiciels.*

On peut distinguer :

- *Informatique théorique,*
- *Informatique des systèmes,*
- *Informatique technologique,*
- *Informatique méthodologique,*
- *Informatique appliquée.*

Sommes nous d'accord avec cette définition ? Si nous ne le sommes pas, que proposons nous ? Les réflexions qui suivent pourraient servir d'argumentaire si nous voulions définir notre propre discipline.

1.1 Apports de l'informatique en tant que discipline

L'informatique est en mesure de prendre part à la formation générale:

- par son côté fondamental:

- logique,
- rigueur,
- modélisation,
- structuration de la pensée,
- méthodologie.

- par son côté expérimental:

- manipulation concrète:
 - . de fonctions,
 - . de relations,
 - . de constructions de programmes.

Il ne faut pas la réduire à l'informatique de complément, appelée aussi *Informatique outil, informatique au service des autres disciplines*, etc. On trouve sous cette rubrique (entre autres):

- bureautique élémentaire (tableurs, traitements de texte, gestionnaires de fiches, ...),
- rudiments de programmation (pour le calcul scientifique, FORTRAN, ...)
- rudiments de technologie (architecture des micro-ordinateurs en 1 heure, ...).
- ...

Cette informatique est nécessaire, mais ne doit pas remplacer l'enseignement de la discipline. On peut tirer de cette affirmation plusieurs constats:

- il faudrait réserver l'appellation de formation à l'informatique aux seuls cursus visant la discipline. Ce qui est gênant, ce n'est pas que l'on enseigne les tableurs, mais que l'on assimile l'outil à l'informatique (enseigner l'usage des tableurs, ou enseigner quelque chose à l'aide des tableurs ?). A nous de proposer des appellations mettant mieux en évidence cette différence: *introduction à la programmation ; calcul sur ordinateur* (par exemple pour ce qui se fait en Classes Préparatoires ...); *Utilisation de progiciels*; ...
- si le temps imparti le permet, il faut profiter de ces enseignements pour faire comprendre qu'il y a autre chose que des outils. Ceci conduit à une implication des enseignants de 24^{ème} section dans ces enseignements.
- Ne pas cantonner les enseignants dans l'enseignement de ce type d'informatique, de peur qu'ils perdent de vue soit la discipline informatique, soit leur discipline propre (pour les collègues des autres sections).

1.2 Les contours de la discipline informatique

Il faut décrire notre discipline par les contenus, en les organisant, les emboîtant. On trouvera ci-dessous, plus ou moins en vrac, à titre de premier jet, un ensemble de vocables. Il serait bon de travailler ceci, de manière à fournir une description propre et structurée. Ce type d'analyse aurait au moins deux objectifs:

- permettre de répondre à la question: mais qu'est-ce que l'informatique ? en donnant des exemples des champs de connaissances investigués par cette discipline,
- permettre de fixer un seuil minimal de connaissances en dessous duquel nul ne pourrait prétendre avoir des bases sérieuses en informatique (et encore moins se dire informaticien!).

Le découpage ci-dessous est largement repris de la présentation de J.C. Boussard (voir annexe 3). Ce n'est qu'un ballon d'essai, qui demande à être travaillé par la communauté. Tout peut être discuté et amendé, à partir du moment où des arguments solides sont apportés, et non pas seulement des sentiments.

En particulier, un des idées sous-jacentes à la proposition qui suit est de calquer la présentation des disciplines classiques. Cette approche a l'avantage de nous obliger à réfléchir sur différentes rubriques qui existent systématiquement par ailleurs. Un exemple typique est la volonté de trouver une harmonisation entre Informatique fondamentale et Informatique expérimentale. Cette division appelle quelques remarques:

- elle vise à faire ressortir ce qui est du domaine des fondements de notre discipline, par opposition à ce qui est du domaine d'une approche pragmatique de celle-ci.
- les deux aspects sont aussi nobles l'un que l'autre. Il n'y a pas (il ne doit pas y avoir) de hiérarchie entre les deux aspects, qui se nourrissent mutuellement.
- on peut ne pas aimer les mots utilisés. Autant en proposer d'autres, ou ne pas s'attacher à la lettre...

INFORMATIQUE FONDAMENTALE**=> THEORIE GENERALE DES MACHINES ET DES PROGRAMMES**

- Mathématiques de l'informatique
 - . Logiques combinatoires et séquentielles
 - . Analyse combinatoire, algorithmique des graphes
 - . Fonctions calculables
 - . Modélisation mathématique des systèmes
- Architectures d'ordinateurs et de réseaux
 - . Physique des ordinateurs
 - . Architectures parallèles (matériels et logiciels)
 - . réseaux
- Algorithmique et complexité
 - . Algorithmique séquentielle
 - . Algorithmique parallèle et distribuée
 - . Intelligence artificielle, théorie de l'apprentissage
- Théorie de la programmation
 - . Structures de données
 - . Types abstraits
 - . Structures de commande
 - . Pré- et post-conditions, invariants
 - . Styles de programmation
 - . Impératif, fonctionnel, déductif
 - . Procédural ou par objets
 - . multi-style
 - . Spécifications formelles et tests de programmes
 - . Programmation industrielle et Génie logiciel
- Théorie des langages formels
 - . Morphologie, syntaxe et sémantique
 - . Notion de langage régulier
 - . Notion de langage algébrique
 - . Sémantique des langages de programmation
 - . Théorie et pratique de la réécriture

=> THEORIE DES LOGICIELS DE BASE

- analyse syntaxique et compilation
- systèmes d'exploitation
- théorie des systèmes monoprocesseurs
 - Processus et ressources
 - Mémoires partagées et virtuelles
 - Notion de multiprogrammation
 - Temps partagé
 - Notion de multitraitement
- théorie des systèmes multi-processeurs

=> IMPLICATIONS SOCIALES

- Histoire de l'informatique
 - . Les grandes étapes
 - . les grands noms
- Informatique et société
 - . Implications économiques
 - . Implications sociales et politiques

=> GRANDES APPLICATIONS

- les grands domaines d'application
 - . Bureautique et PAO
 - . Calcul scientifique
 - . Grands codes scientifiques
 - . Conduite de processus
 - . Gestion et aide à la décision
 - . Jeux et recherche opérationnelle
 - . Traitement du langage naturel
 - . Réseaux et télématique
 - . Simulation et CAO
 - . Statistiques et analyse de données
 - . Vision et robotique
 - . Traitements graphiques, synthèse d'images
 - . Vision et reconnaissance des formes
 - . Reconnaissance et synthèse vocales
 - ?
- Conception et implémentation (grands projets)
 - . de langages
 - . de systèmes d'exploitation et de réseaux
 - . d'environnements de programmation
 - . d'interfaces graphiques
 - . de jeux
 - . de SGBD
 - . de systèmes experts
 - . de systèmes robotisés
 -

INFORMATIQUE EXPERIMENTALE

=> LE MARCHE DE L'INFORMATIQUE :

- configurations
 - . Micro-ordinateurs personnels
 - Interfaces simples
 - . Stations de travail graphiques
 - Gestionnaire de fenêtres
 - Menus déroulants et arborescents
 - Interfaces souris
 - . Mini-ordinateurs, réseaux locaux
 - . Grosses configurations
 - . Installations industrielles
 - . Grands réseaux
- catalogues, brochures techniques
- = problèmes
 - de coûts
 - de formes d'exploitation
 - de documentation et maintenance

=> MANIPULATION DE PROGICIELS

- Tableurs
- Traitements de textes
- calcul formel
- SGBD's
- DAO
- CAO
- Systèmes experts
- ... ?
- = problèmes
 - de convivialité
 - d'adaptabilité

=> MANIPULATION DE SYSTEMES

- MS/DOS ou équivalents ?
- Mac Intosh, ... ?
- Unix
- VMS, ou PIC ou ... ?
- = problèmes
 - de fiabilité (exactitude + robustesse)
 - de réutilisabilité

=> PROGRAMMATION

- Basic, ou Fortran, ou Cobol, etc. ?
- Pascal, ou Ada, ou Eiffel ?
- LISP, ou ML, ...
- PROLOG, ou ...
- = problèmes
 - de structuration de programmes
 - de langages et de styles de programmation
 - de performances statiques et dynamiques

1.3 Comment faire reconnaître notre discipline

Il semble que plusieurs axes d'action s'ouvrent à nous. Citons entre autres:

- mieux faire connaître ce qu'est la discipline informatique.
 - . préparer une plaquette définissant la problématique de l'informatique. On pourrait trouver une description par le contenu (les champs couverts par notre discipline, les axes de recherche), les grandes lignes de l'histoire de l'informatique (ses grandes évolutions, ses grands noms), une bibliographie succincte (liste ouvrages permettant de comprendre les fondements constitutifs de notre discipline) et une liste des différents types de formation supérieure à l'informatique.
 - . organiser des journées sur la recherche en informatique 'médiatiques' visant à faire connaître les grandes tendances de la recherche en informatique. C'est à nous d'expliquer ce que nous faisons, et non aux autres disciplines ...
- nous investir plus dans les différents organismes qui nous entourent.
 - . sommes nous assez impliqués dans le fonctionnement de l'Université ? la politique de la chaise vide (quelles qu'en soient les 'bonnes' raisons) n'est-elle pas suicidaire à terme ?
 - . sommes nous assez impliqués dans les organisations qui militent pour l'enseignement supérieur ? M. Coste (voir annexe 2) note qu'aucun informaticien n'est présent dans PROMOSCIENCES, association s'intéressant aux problèmes de l'enseignement dans les DEUG.
 - . Quelle participation effective avons nous dans SPECIF, qui devrait être notre organisme représentatif ? Comment SPECIF peut devenir un groupe de pression sans activisme de la part de ses membres ?
- nous impliquer plus profondément dans les Premiers Cycles.
 - . il faut insister sur l'enseignement de l'informatique dans les Premiers Cycles, puisque les nouveaux enseignants passent par les DEUG. Il est impensable qu'ils sortent de ces cursus sans aucune notion d'informatique. Ils seront ensuite mieux armés pour parler de la discipline informatique, plutôt que du simple outil.
 - . il faut motiver les enseignants du Supérieur pour le Premier Cycle, en leur présentant des contenus propres, précis, liés à leur discipline. Pour jouer le jeu du Premier Cycle, il faudrait que des personnalités reconnues s'impliquent, plutôt que de faire assurer ces enseignements par des 'marginiaux'.
 - . il y a un problème de désinformation à propos de l'informatique, en particulier en ce qui concerne le Premier Cycle. A nous d'aller le plus tôt possible prêcher la bonne parole. Des enseignements bien faits et de bon niveau attireraient les étudiants. Par ailleurs, tout ce qui ferait paraître l'informatique comme une discipline au même titre que les autres peut être bon à faire. Par exemple, il est souhaitable d'introduire des tests d'auto-évaluation lors de l'orientation en début de DEUG, comme pour les autres disciplines.
- organiser le dialogue avec les disciplines qui nous entourent.
 - . il s'agit de mettre en valeur les apports réciproques: mathématiques et informatique; linguistique et informatique; ... Il s'agit bien entendu de s'intéresser à la cohérence scientifique induite par les relations entre disciplines, et non pas simplement aux relations de type 'l'une au service de l'autre', même si ce dernier aspect est important en lui-même. Là aussi, des Journées d'étude, ou des ouvrages communs pourraient être précieux. On peut envisager d'ouvrir l'informatique grâce à des enseignements pluri-disciplinaires.

2. PROFIL D'UN ENSEIGNEMENT D'INFORMATIQUE EN DEUG

2.1 Pour un enseignement de l'Informatique dans un DEUG bi- ou tri-partite.

Nous considérons que le DEUG ne doit pas spécialiser immédiatement les étudiants (voir en particulier l'intervention de M. Coste, en annexe 2). Il est au contraire souhaitable de faire acquérir au plus tôt une double ou une triple compétence. Nous défendons l'idée que des études pluri-disciplinaires, équilibrées entre les différentes disciplines, sont formatrices, et permettent d'offrir plusieurs voies à un étudiant. Deux conclusions s'imposent:

- la création de 'DEUG informatique' doit être rejetée. Il est de toute manière antinomique de parler d'études générales à propos ... d'une seule discipline !
- il faut créer des DEUG intégrant la discipline informatique, à égalité avec les autres disciplines. L'équilibre doit être strictement respecté: pour un DEUG à deux disciplines (bi-partite), le temps doit être partagé moitié-moitié; pour un DEUG à trois disciplines (tri-partite), le temps doit être partagé un tiers - un tiers - un tiers.

Le DEUG doit permettre d'acquérir des méthodes, des mécanismes de pensée. Les outils peuvent être vus plus tard. L'intérêt des outils est de montrer ce qui est faisable ou non. Ils permettent de dessiner les contours des applications. Il faut enseigner en DEUG non pas les outils, mais le fond. On peut s'interroger sur ce qui se passe dans les autres disciplines sur ce plan.

Nous avons noté le contraste offert aux étudiants en DEUG:

- . mathématiques = intégrales triples
- . informatique = touches d'un clavier, équivalent des premières manip de physique et chimie au collège!

Pourquoi n'y aurait-il pas de l'Education Manuelle et Technique informatique dans le secondaire ? ce serait au moins aussi utile que d'apprendre à réaliser des recettes de gâteaux

Il faut qu'il y ait de l'informatique dans tous les DEUG, mais:

- nous savons que c'est trop cher,
- nous ne savons pas exactement quoi faire, à cause de la diversité de la demande.

Si on veut parler de véritable formation à l'informatique (acquisition des bases de l'informatique), il faut nécessairement un volume horaire conséquent. Il y a donc un niveau minimal, en dessous duquel le label 'informatique' ne peut être décerné (bien entendu, l'aspect qualitatif de l'enseignement intervient aussi pour décerner le label !).

L'efficacité nous pousse à nous intéresser:

- à un enseignement de l'Informatique dans les DEUG,
- bénéficiant d'une mise en place à court terme,
- avec des retombées à moyen terme sur tout le reste (informatique hors discipline).

On ne peut pas tout faire, il nous faut donc choisir:

- faire là où on peut,
- le faire bien: beaux programmes, livres, environnement d'apprentissage.

Un exemple de niveau attendu d'une formation initiale en informatique serait:

- avoir des bases pour comprendre l'informatique et ses implications,
- avoir compris qu'un programme est un objet d'étude (référence à la notion de méthodologie).
- savoir aider à choisir une configuration,
- savoir lire d'un oeil critique un article de Science et Vie Micro (ou tout équivalent !),

2.2 Principes de l'enseignement de l'Informatique dans un DEUG

Nous avons travaillé avec deux fils conducteurs:

- comment attirer les étudiants dans les filières informatiques de second et troisième cycles ?
- comment concevoir un DEUG ouvrant sur plusieurs filières ?

La première idée est qu'il faut exiger un volume horaire non négligeable, afin de pouvoir parler réellement de formation à l'informatique. Par exemple, nous avons visé un minimum horaire de 400 heures sur deux ans. En dessous de ce seuil, il ne faudrait pas parler d'enseignement de l'informatique.

Le découpage d'un DEUG entre deux ou trois disciplines a fait l'objet d'un débat:

- l'idéal serait un DEUG tri-disciplinaire, afin que l'étudiant puisse bénéficier d'une véritable culture scientifique.
- le réalisme (en particulier à cause des volumes horaires envisagés) peut nous pousser à construire des DEUG bi-disciplinaires. Quelques remarques:
 - . a priori, tout couplage pourrait être retenu: Informatique et Mathématiques; Informatique et Physique; Informatique et Sciences Economiques; Informatique et Biologie; Informatique et Lettres; Informatique et Sciences Sociales; Informatique et Mécanique.
 - . il faudra vérifier qu'un couplage offre bien l'entrée à au moins deux filières différentes. Est-ce qu'un couplage 'Biologie et Informatique' suffira aux Biologistes ? ou aux informaticiens ? On voit apparaître ici l'idée que de tels couplages doivent être négociés entre les partenaires, afin de s'appuyer sur les apports réciproques des deux disciplines, notamment pour offrir une cohérence scientifique globale.

Il faut éviter à tout prix l'équation informatique = programmation. Le mot programmation a été très discuté, plusieurs thèses s'affrontant:

- ne plus parler de programmation:
 - . parce que ce mot a maintenant une connotation péjorative dans l'industrie: le programmeur est en train de disparaître.
 - . parce que le mot de programmation est trop identifié à un style, à savoir la programmation impérative.
 - . parce qu'il ne faut plus programmer ...
- parler de programmation:
 - . parce que c'est une activité inhérente à la fonction d'informaticien (peut-on réellement envisager de dire que les seules personnes qui ne programment pas sont ... les informaticiens!)
 - . parce qu'il existe des méthodes de construction de programmes et de programmation qui sont parfaitement identifiées.
 - . parce qu'il ne faut pas avoir peur des mots, si on sait les placer dans leur contexte.

Bien entendu, le débat n'a pas été tranché. On retiendra cependant plusieurs idées:

- ne pas laisser réduire l'informatique à la seule activité de programmation.
- il faut mettre en valeur les activités méthodologiques. On ne parlera pas seulement de programmation (si on en parle ...), mais aussi de construction de programmes, d'algorithmique. On rejoint le souci déjà exprimé de proposer des dénominations plus précises pour les différents secteurs d'activité en informatique.
- on s'attachera à mettre en évidence au moins deux styles de programmation. Pourquoi ne pas commencer par la programmation impérative (par exemple), à condition d'étudier ensuite la programmation fonctionnelle (ou vice-versa) ? Les étudiants verront ainsi plusieurs approches de résolution de problèmes, avec des méthodes d'analyse différentes. Le passage d'une méthode à l'autre pourra être enseigné: par exemple analysé récursive, programmation par élimination de la récursivité.

Il faut intéresser les étudiants:

- en montrant des applications de l'informatique. Des présentations des travaux de recherche des Laboratoires locaux semblent avoir un fort impact.
- en leur faisant travailler des exemples intéressants et accrocheurs: décryptage, graphique, ... Chacun parmi nous a pu remarquer les résultats remarquables obtenus par des étudiants très motivés.
- en leur montrant comment aborder certaines difficultés inhérentes au calcul sur ordinateur : problèmes des erreurs sur les réels, complexité des algorithmes, ... L'idée est de faire découvrir que, contrairement au discours ambiant, tout n'est pas si simple en informatique, et que les machines ont des limites.
- en introduisant ce qui est bien formalisé (logique, langages formels, ...). On introduit ainsi la théorie sous-jacente aux méthodologies de résolution de problèmes. L'informatique n'est plus l'art de la bidouille, mais une science qui s'appuie sur des théorèmes.

Il faut intéresser les enseignants en informatique:

- en assurant une organisation confortable de cet enseignement : sous prétexte d'enseignement de masse le DEUG ne doit pas être le parent pauvre de l'enseignement supérieur.
- en montrant qu'ils ne perdent pas leur âme (leur discipline) en enseignant l'informatique en Premier Cycle. La définition des bases de l'Informatique est donc déterminante.
- en montrant que leur carrière n'est pas brisée s'ils enseignent en Premier Cycle. En fait, on devrait banaliser l'enseignement dans tous les Cycles: il est sain que tout enseignant enseigne un jour ou l'autre dans les trois cycles du Supérieur. Par ailleurs, il faut valoriser le travail réalisé en Premier Cycle: sans un enseignement de premier niveau (au sens : le meilleur possible) dans les DEUG, il n'y aura plus d'étudiant en 2ème et, a fortiori, en 3ème cycle informatique.
- il faut cependant éviter d'organiser un défilé de spécialistes, sous prétexte que l'on enseigne des 'matières' différentes. Il faut revaloriser la notion d'enseignant généraliste: le danger de n'avoir plus que des super-spécialistes, incapables d'enseigner autre chose que la super-pointe de leur super-recherche nous guette. Par ailleurs, n'est il pas raisonnable de penser que tout informaticien de l'Enseignement Supérieur devrait être capable d'enseigner l'ensemble des bases de l'informatique en Premier Cycle ? Il faut donc intéresser les meilleurs de nos collègues, pour que leur rayonnement s'applique également en Premier Cycle.

Il faut intéresser les collègues des autres disciplines, en leur faisant constater qu'une formation de base profonde en informatique n'est pas une perte de temps, mais au contraire complète une culture scientifique de manière significative et, incidemment, conduit à une meilleure connaissance de l'outil informatique. Ainsi, les étudiants choisissant leurs filières apporteront une compétence certaine en informatique, susceptible de modifier profondément l'utilisation de l'outil dans leur discipline.

C'est en travaillant dans toutes ces directions que nous arriverons non seulement à imposer, mais à faire reconnaître et désirer l'existence de l'enseignement de l'Informatique.

2.3 Un exemple de découpage de l'enseignement de l'Informatique dans un DEUG

Plusieurs exemples ont été proposés et travaillés:

- voir la proposition de J.C. Boussard en Annexe 3,
- voir les exemples d'enseignement existant déjà en Annexe 5 et Annexe 6.

Un autre exemple de découpage se trouve ci-dessous

	INF. FONDAMENTALE	I	INF. EXPERIMENTALE	I	BUREAUTIQUE
1ère année	Architecture des systèmes monoprocesseurs et des réseaux	I	Micro-ordinateurs	I	INITIATION A LA PROGRAMMATION
		I	Systèmes d'exploitation	I	
	50 heures	I	50 heures	I	
		I		I	maintien du statu quo
	Logiques combinatoire et séquentielle	I	Construction des programmes	I	
		I		I	
	Algorithmique, calculabilité et complexité	I		I	
		I		I	pour ceux qui ne se
	Conception des systèmes d'information	I		I	
		I		I	
50 heures	I	100 heures	I		
<hr/>					dirigent pas vers
2ème année	Théorie de la programmation	I	Mini-ordinateurs	I	
		I	Systèmes d'exploitation	I	
		I	Manipulation de logiciels	I	
	50 heures	I	50 heures	I	l'informatique
		I		I	
	Langages formels	I	Programmation	I	
		I		I	
	Informatique et Société	I		I	
		I		I	
	Grandes applications	I		I	
	I		I		
50 heures	I	100 heures	I		
<hr/>					
total	200 heures	I	300 heures	I	

Quelques remarques sur ce tableau:

- les thèmes envisagés.

On remarque qu'il y a quelques thèmes horizontaux:

- . architecture des systèmes <-> systèmes d'exploitation
- . logiques, algorithmique <-> construction des programmes, structures de données
- . théorie de la programmation <-> programmation en 2ème année

Ces thèmes permettent de donner une certaine unité entre l'informatique fondamentale et l'informatique expérimentale.

- les objectifs envisagés:

- . apprendre à analyser, organiser, décomposer.
 - . apprendre à "mécaniser", ou "automatiser" conduit à analyser le problème.
 - . l'analyse conduit à apprendre à décomposer le problème en sous-problèmes (modules).
 - . la décomposition conduit à s'intéresser à la communication entre les modules.

- utiliser des outils de résolution:
 - . représentation et circulation des informations (machine de Von Neuman)
 - . systèmes d'exploitation
 - . compilateurs et langages
- connaître des problèmes et des méthodes
 - . complexité des algorithmes en termes de performance (sur exemples)

Il est clair qu'il s'agit ici:

- de la base minimale qui doit permettre à un étudiant de commencer des études de 2ème et 3ème cycles d'informatique,
- de la base minimale qui permet à un étudiant de prétendre qu'il a des connaissances de base en informatique, quelle que soit la filière (autre qu'informatique) qu'il choisira.

L'approche proposée ici permet de déduire le contenu des différents modules. Par exemple, sous la dénomination de conception de systèmes d'information, on vise à faire connaître les étapes qui permettent d'arriver à un produit informatisé. Un exemple de méthode dans ce cadre consiste à prendre un ou plusieurs projet(s) documenté(s) et étudier ce que l'on sait de ce qui est à faire de l'information dont on dispose à chaque étape, plus ou moins en fonction des modèles retenus. On peut étudier le cahier des charges, la conception, la maquette, et différentes implémentations (programmes 'classiques', LAG, ...).

Il est clair qu'il faut s'interroger sur les rapports avec l'(es) autre(s) discipline(s):

- faut-il prévoir des compléments utiles aux collègues des autres sections (par exemple: Bureautique et programmation). Quelle place ces compléments doivent-ils occuper ?
- ce programme est-il compatible avec autre chose qu'un DEUG Informatique-Mathématiques ? Dans le cadre d'un DEUG Informatique-Physique, ou Informatique-Chimie, ne risque-t-il pas de manquer des mathématiques ?
- on ne peut faire l'impasse sur l'apport potentiel des autres disciplines. Qu'en est-il par exemple de l'enseignement des mathématiques de l'informatique (logique, théorie des langages, ...) et des mathématiques pour l'informatique (analyse de données, probabilités statistiques, ...) ? Pourquoi les mathématiciens devraient ils enseigner les mathématiques pour l'informatique, si nous estimons ne pas devoir enseigner l'informatique pour les mathématiques ?
- Beaucoup d'autres enseignements sont souhaitables: anglais, biologie, ... Le volume horaire d'un DEUG étant fixé, que restera-t-il réellement comme temps pour les disciplines de base ? Que reste-t-il du programme envisagé si l'on ne dispose plus que de 250, 200 ou 150 heures en tout sur deux ans ? Que devient, dans ces conditions, la question du label informatique ?

Les travaux réalisés pendant les Journées ont montré qu'il était possible, au prix de (dures) négociations d'obtenir des volumes horaires importants. On trouvera en annexe deux exemples de tels DEUG existant aujourd'hui: en annexe 5 l'exemple de Metz, en annexe 6 l'exemple de Grenoble. Leur existence même prouve que l'on peut obtenir de bons résultats, et peut servir de base de discussion. D'où le slogan de ces Journées:

Enseigner l'informatique en tant que discipline dans les premiers cycles scientifiques c'est possible !

Bien entendu, il faut se battre pour obtenir la création de tels DEUG. Il semble que la période soit propice, le Ministère soutenant notre point de vue (tout au moins une partie des conseillers !).

2.4 Un exemple d'option informatique.

Nous avons également réfléchi à la possibilité d'introduire un enseignement d'informatique destiné aux DEUG ne souhaitant pas intégrer l'Informatique en tant que discipline fondamentale. Il s'agit quand même d'enseigner quelques rudiments de la discipline informatique. Voici un exemple de proposition, servant à alimenter les débats:

- objectifs:

- . savoir qu'il existe des méthodes pour exprimer un problème,
- . savoir qu'il existe des méthodes de modélisation,
- . savoir qu'il existe des problèmes liés à l'efficacité.

- méthodes:

- . avoir observé des modélisations de problèmes,
- . avoir observé des problèmes de complexité de programmes,
- . avoir mesuré l'efficacité de programmes.

- contenu

- . 12h Introduction historique
- . 24h Architecture de systèmes monoprocesseurs et réseaux
- . 24h Conception de systèmes (modélisation des traitements et données)
- . 36h Informatique et Société - Grandes applications
- . 12h Interface homme-machine
- . 48h Algorithmique, structure des données
- . 48h implémentation (style fonctionnel, problèmes et modèles réalisés).

soit un total d'environ 200 heures.

On remarquera que c'est notablement plus que ce qui existe à l'heure actuelle en général, sans qu'il puisse y avoir cependant confusion avec l'enseignement de l'Informatique décrit plus haut. Ce type d'approche peut être intéressant pour des étudiants qui souhaitent entrer un peu en profondeur quant à la connaissance de l'outil informatique, sans pour autant vouloir se prétendre des informaticiens (des utilisateurs éclairés ?).

Par ailleurs, l'aspect Bureautique et initiation à la programmation n'est pas du tout abordé. Nous n'avons pas travaillé le problème de cet enseignement. Il reste donc entier.

2.5 Articulations avec les autres enseignements

Nous regroupons sous ce thème les relations qui peuvent lier l'enseignement de l'Informatique dans un DEUG avec les autres formations en informatique, les autres disciplines, l'enseignement de l'informatique en Europe et la profession.

- l'informatique en DEUG et les autres formations informatiques

. dans le Second Degré

Le constat déjà fait à Besançon est renouvelé: il est impossible de tenir compte de ce qui est enseigné dans le secondaire, compte-tenu du nombre trop faible d'élèves formés à l'informatique (option Informatique: 5%, Clubs, autodidactes: 10%). Par ailleurs, il peut y avoir un problème de démotivation des enseignants du Second degré, dans la mesure où l'informatique n'est pas encore une discipline.

Par ailleurs, la question reste posée de savoir ce que tout élève ayant obtenu le baccalauréat (80% de la tranche d'âge, à terme !) devrait connaître de l'informatique: des rudiments de la discipline, ou une bonne formation de type bureautique ?

. entre Bac et Bac+2 (formations parallèles au DEUG)

Ce sont essentiellement des formations professionnelles (IUT, BTS) ou para-professionnelles (CNAM), avec 50% d'informatique et 50% d'autres disciplines. On peut donc constater qu'il y a la place dans les DEUG pour une formation généraliste, comprenant un enseignement de l'Informatique.

• après Bac+2

On trouve des formations professionnelles de même type que précédemment (MIAG, CNAM). La formation de chercheurs par les second et troisième Cycles en informatique (ou certaines Ecoles) ne peut qu'être confortée par une formation plus précoce à la discipline. Il faudra revoir les contenus des Licence et Maîtrise d'informatique en fonction de ces DEUG nouvelle manière.

• les passerelles potentielles

Les passerelles les plus classiques sont les suivantes:

- + équivalence du DEUG accordée aux élèves des Classes Préparatoires bi-admissibles, pour une entrée directe en Licence. Il y a donc de gros problèmes en vue pour cette catégorie d'étudiants, puisqu'ils n'auront pas reçu de formation de base en informatique.
- + passage du DUT vers la Licence d'informatique ou les MIAG. Il reste à vérifier que les formations reçues resteront compatibles avec celles qui vont être données en DEUG. En tout état de cause, il n'y aura pas équivalence directe entre un DUT et un DEUG.
- + passage du DEUG vers une année spéciale IUT. Il ne semble pas qu'il y aura des problèmes.
- + passage du DEUG vers les Seconds Cycles, ou les Ecoles d'Ingénieur. Tout dépendra bien entendu du DEUG préparé, et des prérequis exigés par chaque filière de second cycle, ou chaque Ecole d'Ingénieur. On ne voit cependant pas pourquoi le système marcherait moins bien que maintenant, avec un renforcement des connaissances en informatique !

- l'informatique et les autres disciplines

Il s'agit de percevoir les apports réciproques:

• apport de la discipline informatique aux autres disciplines:

- + méthodologie, modélisation, structuration des problèmes, rigueur, logique, analyse
- + outil informatique proprement dit

Dans un contexte où l'étudiant de DEUG n'a pas de temps pour prendre du recul et se perd dans des détails souvent abscons, l'informatique, par la nature même de cette discipline, impose à l'étudiant de savoir prendre de la hauteur par rapport à un problème donné. Les projets développent l'esprit de groupe et de synthèse. Les accès libres permettent l'auto-formation. Les notions de communication et de portabilité sont abordées à travers les règles de lisibilité des programmes.

• apports des autres disciplines à l'informatique

- + Il est facile de trouver des apports directs (les besoins de l'informatique ?): on parle souvent des Mathématiques ou de la Physique pour l'Informatique, de l'anglais technique, etc... Comment dialoguer pour que la parité des échanges soit obtenue ?
- + On mesure moins bien l'apport d'autres disciplines, en tant qu'ouverture de l'informatique à de nouveaux domaines d'application. Peut-être ne sommes nous pas assez sensibilisés (ouverts) à ce que nous pourrions gagner en engageant plus systématiquement le dialogue avec toutes les disciplines: certaines des plus belles avancées de l'Informatique sont dues à la nécessité de résoudre des problèmes particuliers à de grands domaines d'application. Comment favoriser le passage vers l'informatique d'étudiants en provenance de cursus de Sciences Economiques, de Linguistique, de Biologie, etc.

- l'enseignement de l'informatique en Europe

Un remarquable travail de synthèse a été réalisé par J.P. Steen (voir annexe 4). Notons :

- le découpage systématique en semestres, qui n'est vraiment pas dans les mœurs françaises !

- l'étudiant français semble trop materné: c'est lui qui 'bénéficie' du plus grand nombre d'heures d'encadrement. Notons par exemple en Angleterre la notion de cours d'auto-formation (*solo courses*), par exemple pour la programmation.
- l'existence de programmes nationaux, qui semble être un moteur pour le développement et la reconnaissance de la discipline. On parle parfois de programme national pour les filières informatiques autres que les IUT et les MIAG. Qu'en penser ?
- la difficulté évidente de passage entre les diplômes européens. Un problème d'harmonisation se pose peut être.

Par manque d'information (de sensibilisation ?), il n'y a pas eu débat sur ce sujet. Il a été émis l'idée qu'il faudrait établir un bilan des échanges auxquels ont procédé certaines universités, en particulier à travers le programme ERASMUS.

- les relations avec la profession

Ce sujet n'a quasiment pas été abordé au cours de nos travaux. Cependant:

- il nous paraît évident que le DEUG n'a pas à être considéré comme une formation professionnelle.
- il existe des formations à vocation professionnelle: IUT, BTS et MIAG. Elles entretiennent des relations étroites avec le milieu professionnel, et atteignent parfaitement l'objectif qu'elles se sont fixé.
- les Licence et Maîtrise d'Informatique ne sont pas des formations à vocation professionnelle. On remarquera quand même leurs grandes capacités d'adaptation à l'évolution de l'informatique: c'est en leur sein que naissent les premières expériences d'enseignement de tout nouveau champ ouvert à l'informatique: compilation, bases de données, systèmes experts, intelligence artificielle, réseaux et télématique, ... Le milieu professionnel a été le premier à bénéficier de la création de ces enseignements nouveaux !

En conclusion de ces réflexions sur les articulations, on relèvera que, d'une manière générale, le problème qui est revenu constamment sur le tapis sans recevoir de solution est le suivant: faut-il tenir compte des différences de formation des étudiants ? Il est clair que ce problème se pose à tous les niveaux dans les trois cycles de formation (il se pose moins dans le cas des Ecoles d'ingénieur, qui peuvent recruter des promotions relativement homogènes).

La question se pose dans les deux sens:

- que faire pour les étudiants ayant un niveau supérieur à celui attendu, dans tout ou partie d'un cursus ?
 - . élèves de l'option Informatique du Second Degré à l'entrée d'un DEUG,
 - . étudiants titulaires d'un DUT à l'entrée en Licence,

Il faudrait briser le cycle actuel qui veut qu'un étudiant subisse une intitution à l'informatique tous les ans, de la maternelle à la Licence d'Informatique... Les étudiants ayant déjà reçu une partie de la formation à venir devraient pouvoir en tirer parti !

- que faire pour les étudiants ayant un niveau inférieur à celui attendu, dans tout ou partie d'un cursus ?
 - . élèves des Classes Préparatoires voulant intégrer une Licence d'Informatique,
 - . étudiants à l'entrée d'une Licence d'Informatique, en provenance d'une Université où il n'y a pas d'enseignement de l'Informatique en Premier Cycle,

Il semble clair qu'une remise à niveau s'impose. Mais dans quelles conditions ? sous quels délais ? à quelle période ? pouvons nous réellement mettre sur pied des 'cours particuliers' ?

Un des problèmes de l'enseignement de masse (cas des DEUG) est qu'il semble difficile de faire des traitements particuliers. Et pourtant, il faut réfléchir sérieusement à ce problème, car il se posera sans doute de façon accrue (voire aiguë) dans les toutes prochaines années.

3. METHODES PEDAGOGIQUES

3.1 Le contexte

Il a été constaté la diversité de l'existant:

- dans le cadre:
 - . nombre d'étudiants concernés: de 50 à 2 000 étudiants en DEUG A par année,
 - . nombre d'heures de formation: de 0 à 50h en première année; de 50 à 150h en 2ème année.
 - . hétérogénéité des étudiants: acquis, motivation
 - . DEUG A ou DEUG B: s'agit-il de la même informatique ?
 - . enseignement de l'informatique dans les autres disciplines: y-a-t-il un "must" informatique ?
 - . types d'intervenants: titulaires; vacataires; non informaticiens
 - . environnement logistique: salles, matériels, logiciels, personnel.
 - . mode d'organisation: journées bloquées, ou séries étalées dans le temps ?
- dans l'esprit:
 - . savoir ce qu'est l'informatique, l'ordinateur ou savoir utiliser l'informatique, l'ordinateur ?
 - . démythifier, montrer les limites, les possibilités de l'informatique ou informatiser à tout prix ?
 - . offrir un vernis ou préparer à une spécialisation ?
 - . contribuer à l'image de l'informatique discipline fondamentale ou promouvoir la bureautique et l'initiation à la programmation ?
 - . construire les contenus:
 - + par rapport à la discipline informatique,
 - + par rapport aux demandes des autres disciplines (savoir utiliser, exploiter, maintenir)
 - + par rapport aux finalités d'un DEUG.

Quelques remarques supplémentaires:

- la diversité freine l'émergence d'une unité ainsi que les possibilités d'échange. A nous de nous organiser pour rompre cet engrenage.
- la faiblesse des horaires empêche de résoudre le problème de manière ambitieuse. Par ailleurs, nous sommes conduits à une sorte d'auto-censure, faute de moyens pour assurer l'enseignement souhaité.
- les contraintes (non reconnaissance de l'informatique, faiblesse des horaires, ...) conduisent à des impressions d'échec qui démotivent les enseignants. Nous avons pourtant joué un rôle important en définissant les grandes lignes de l'enseignement de l'informatique en DEUG il y a quelques années (Journées de Bordeaux, 1981). Les propositions formulées à cette époque (entre autres: usage de micro-ordinateurs, algorithmique et schémas de traduction, etc.) ont apporté de grandes améliorations par rapport à ce qui se faisait alors. Ce schéma ne donne plus satisfaction aujourd'hui, du fait de l'évolution de notre discipline: à nous d'en proposer un nouveau !

Il est apparu rapidement que:

pour faire reconnaître l'informatique en tant que discipline aussi bien par les étudiants que par les collègues, il faut en organiser l'enseignement à l'image de celui des autres disciplines:

- mêmes volumes horaires,
- même type d'évaluation: partiels, examens, coefficients corrects, ...
- mêmes modalités d'enseignement : Cours, Travaux Dirigés, Travaux Pratiques.

Cet alignement (mimétisme) sur ce qui a fait ses preuves dans les autres disciplines ne doit pas empêcher une démarche originale, fondée sur les particularités de l'informatique.

Par ailleurs, un minimum d'organisation devrait être mis sur pieds, afin de nous permettre d'échanger nos expériences pédagogiques.

3.2 Modalités pratiques

Un long débat a eu lieu quant aux activités pédagogiques. Les principaux mots-clés ont été les suivants:

- cours magistral;
- TP, projets, libre-service
- contrôle, évaluation,
- travail en équipe pédagogique
- documents
- reconnaissance de ces activités.

Si l'on part du point de vue du mimétisme quant aux autres disciplines, il apparaît que:

trois formes d'enseignement sont nécessaires:

- le cours,
- les Travaux Dirigés,
- les Travaux pratiques.

Les raisons tiennent à la nécessité de disposer de ces trois formes d'intervention pour transmettre un savoir, des techniques, ..., comme cela se pratique dans les autres disciplines et non pas à des méthodes de gestion de la pénurie (économies: cours à 800 étudiants, pas de TP, ...). Il faut donc séparer la nécessité d'assurer ces trois formes d'intervention des techniques de réalisation de celles-ci (nombre de groupes, salles, problèmes de service, ...).

Ce point de vue a cependant été fortement critiqué:

- hostilité marquée par certains quant à l'utilité de faire des cours (anonymat des étudiants, faible rendement dès que le nombre est important, absentéisme, ...)
- hostilité marquée par d'autres quant à l'intérêt de faire des Travaux Pratiques: cette notion existe-t-elle en Informatique ?

Nous avons recueilli un certain nombre de réflexions autour de ces notions, sans parvenir à des conclusions définitives.

les cours magistraux:

- le terme de cours magistral est chargé de plusieurs significations:
 - . acte pédagogique
 - + ambiance générale de l'enseignement
 - + transmission de connaissances: terminologie, concepts, ...
 - + redondance nécessaire par rapport à d'autres activités
 - + évaluation des connaissances
 - . outil d'homogénéisation
 - + vis-à-vis de l'hétérogénéité des enseignants
 - . identification de responsabilités
 - + vis-à-vis de l'enseignement
 - + vis-à-vis des acteurs de cet enseignement,
 - + vis-à-vis du reste du DEUG,
 - + vis-à-vis de l'UFR,
 - + ...
 - . outil de gestion de la pénurie
 - + il permet de diminuer le nombre d'heures à effectuer, en regroupant les étudiants.
- le cours est gratifiant pour l'enseignant
 - . il est ressenti comme étant la forme supérieure d'intervention,
 - . il est intéressant en termes de service (quant au nombre d'heures passé devant les étudiants)

- le cours magistral est synonyme de transmission de savoir. Supprimer le cours ne revient-il pas à dire qu'il n'y a pas de savoir à transmettre ?
- le cours est une "performance" physique, voire même d'acteur. Lorsqu'il est réussi, il contribue à l'image de la discipline.
- la question du cours ne se pose vraiment qu'en situation d'un nombre important d'étudiants.

Les activités en Travaux Dirigés et Travaux Pratiques

Un constat: ce qui se fait actuellement relève plus des Travaux Dirigés, que ce soit avec ou sans machine.

Le constat de la quasi inexistence de cette notion dans notre discipline, alors qu'elle est présente dans toutes les autres disciplines ayant une composante expérimentale, nous a conduits à analyser les raisons de cette absence. Voici quelques exemples d'arguments entendus:

- les Travaux Pratiques sont peu gratifiants pour les enseignants (notoriété, intérêt, service), et très coûteux en termes d'encadrement (temps de présence auprès des étudiants).
- cette notion n'a pas de sens en Informatique. La preuve, c'est qu'il n'y a pas de TP !
- l'absence de TP est compensée par le temps consacré au libre-service (quand il y en a).

Il semble bien qu'aucun argument majeur en faveur de l'absence d'activités de Travaux Pratiques n'ait été émis, dès lors que l'on s'est attaché à rechercher les avantages offerts d'un point de vue pédagogique. Donnons quelques exemples:

- rythme de travail
 - . en TD, le rythme est celui de l'enseignant,
 - . en TP, c'est celui des étudiants.
- autonomie
 - . en TD, l'enseignant tire les étudiants,
 - . en TP, les étudiants doivent organiser eux-mêmes leur travail.
- certaines activités sont typiques de Travaux pratiques. En voici une liste, dressée rapidement, pour voir !:
 - . observer la trace d'un programme
 - . étudier la correction d'un programme
 - . réaliser des jeux de test pour un programme
 - . déterminer ce que fait un programme
 - . compléter un programme
 - . échanger des programmes d'étudiants
 - . observer des programmes à l'aide de diagrammes syntaxiques
 - . utiliser des simulateurs de machines
 - . observer les ambiguïtés de la langue naturelle
 - . réaliser l'autopsie de disquettes et de machines
 - . partir d'une expérience vécue (distributeur) pour construire un dialogue homme-machine
 - . faire écrire de petits programmes pour réaliser un projet collectif
 - . comparer des performances de programmes
 - . déterminer les limites d'un programme, d'une machine
 - . utiliser une grammaire pour spécifier des informations
 -

Il est évident que de nombreux autres exemples pourraient être trouvés.

On peut se poser la question d'inventer d'autres types d'activité:

- . des projets, réalisés en groupes, avec éventuellement des étudiants plus avancés,
- . des Travaux Pratiques.
- ... ?

Quelques remarques pour aider à la réflexion:

- il faut pouvoir caractériser, inventer, contrôler, évaluer les différentes activités de l'étudiant. Ceci relève de la problématique de la discipline au niveau considéré. Par exemple:
 - . confrontation à la complexité (sous diverses formes)
 - + des démarches,
 - + des problèmes
 - . découverte de la nécessaire variété des modes d'expression
 - . observation de niveaux de concrétisation logiciels et matériels
 - . proposer l'idée de contrat (objectif, cahier des charges, temps à consacrer, mode d'évaluation).
- il faut inventer des modalités de mise en œuvre:
 - expérimenter en
 - . observant
 - . complétant
 - . produisant
 - . concevant
 - . mesurant
- l'informatique ne se conçoit pas sans un contact étroit avec la machine. La programmation est une activité importante dans ce contexte: on peut espérer l'acquisition de bons réflexes.
- il faut exploiter la motivation des étudiants:
 - . pour l'utilisation des machines,
 - . par les thèmes choisis (graphique, Interface Homme-machine, ...)
 - . par les progressions (de l'observation à la production)

A l'inverse, il faut éviter les situations d'échec (rien ne marche, je ne comprends rien, ...)

La question de la mise en œuvre de Travaux Pratiques a également été abordée: Il faut remarquer que notre manque d'expérience dans le domaine conduit à être prudent quant aux réponses:

- faut-il réaliser des 'TP tournants' ou une 'série de TP indépendants' ? Il existe des arguments d'ordre économique, pour l'un ou l'autre type de montage, pas vraiment d'arguments d'ordre pédagogique.
- pas de consensus sur la qualification pédagogique ou technique nécessaire pour assurer des TP (question sous-jacente: des moniteurs ne suffiraient-ils pas à la tâche ?)
- les Travaux Pratiques ont un aspect répétitif, et nécessitent un investissement important:
 - . de conception,
 - . de réalisation,
 mais moins de travail d'exploitation.
- les travaux pratiques nécessitent l'élaboration de documents d'accompagnement, à l'usage des étudiants et des enseignants.

Le débat n'était pas clos à l'issue des Journées. Il a été convenu qu'il fallait poursuivre la réflexion. Un petit groupe de travail a été formé, chargé de préparer des réflexions et des propositions quant à la définition et la mise en œuvre de Travaux Pratiques en Informatique. Il comprend Y. Ahronovitz (Saint-Etienne), S. Bensalem (Grenoble), J.J. Bourdin (Poitiers), M. Chabre-Peccoud (Grenoble), J. Giannesini (Aix-Marseille III), B. Jaray (Nancy), M. Lucas (Nantes). Toute autre personne intéressée peut le rejoindre ! Contacter M. Lucas.

L'évaluation des connaissances

Le point a simplement été évoqué, faute de temps:

- elle doit être conforme à ce qui se passe dans les autres disciplines,
- il faut sans doute une certaine variété dans les modalités,
- il y a des points de blocage liés au nombre d'étudiants:
 - . coût de la correction,
 - . problèmes d'homogénéisation
 - . notation de seuil critique (nombre d'étudiants) pour une section autonome.

Nous avons relevé une certaine incohérence entre les ambitions pédagogiques affichées (enseigner une discipline à un haut niveau) et les moyens donnés pour l'évaluation.

L'équipe pédagogique

Bien entendu, elle n'a pas été absente de nos débats: de sa qualité dépend la qualité de l'enseignement. Quelques remarques:

- il faut lui assurer une certaine cohérence. Une méthode consiste à produire des documents pédagogiques à l'usage des enseignants.
- dans le même ordre d'idées, un débat sur les photocopiés a été engagé:
 - . un photocopié doit garder un statut 'd'inachevé', de temporaire:
 - + pour rester évolutif,
 - + pour ne pas freiner l'évolution des enseignements.
 - . un photocopié est un moyen d'homogénéiser l'enseignement:
 - + en ce sens, il est complémentaire du cours,
 - + il peut comporter toute sorte de documents (éléments de cours, sujets de travail, documentation technique, ...)
 - . il y a une crainte qu'il détourne les étudiants du cours (absentéisme)
 - . il nécessite beaucoup de travail. Qui va s'en charger ? comment valoriser ce travail ?
- les enseignants d'une équipe pédagogique doivent cependant avoir droit à une certaine liberté d'action. Il faut qu'ils puissent travailler en fonction de ce qu'ils ressentent.
- les expériences pédagogiques réalisées dans nos diverses Universités ne sont pas assez connues. Il est dommage que nous recommandions tous à redécouvrir ce que d'autres ont déjà mis au point, après quelques déboires (exemples: du bon usage des disques durs; les joies de l'utilisation d'un réseau de micro-ordinateurs; les surprises de l'emploi des projecteurs d'images d'écran; ...).

Afin que l'information circule mieux entre nous, nous proposons plusieurs actions:

- un recensement des offres d'outils pédagogiques
 - Il s'agit de "centraliser", répertorier les offres d'outils pédagogiques et en diffuser la liste à ceux qui le souhaitent, en vue de contacts directs ultérieurs.
 - S'adresser à Françoise MADAULE, Laboratoire MASI, Université Pierre et Marie Curie, 4 place Jussieu, 75252 Paris Cedex 05 tel: (1) 43 36 25 25 poste 31 57.
- l'organisation d'une bourse d'échanges
 - Que chaque responsable informatique envoie chaque année à ses homologues (liste que l'on peut se procurer auprès de M. Lucas):
 - . les documents distribués aux étudiants
 - . les sujets d'examen
 - . les documents élaborés à l'usage de l'équipe pédagogique
 - . les textes (et peut-être les corrigés !) des TP-TD

3.3 Pour une bibliographie de l'Informatique

Il nous semble que la constitution d'une bibliographie de référence permettrait de mieux faire comprendre ce qu'est l'informatique. Plusieurs catégories de personnes sont visées:

- les collègues non informaticiens (ou les non informaticiens, sans qu'il soient obligatoirement collègues !) souhaitant comprendre ce qu'est la discipline informatique. Il s'agit ici de proposer des titres accessibles à tout public, donnant une image exacte de notre discipline.
- les collègues désirant prendre contact rapidement (et cependant sérieusement !) avec des questions qui leur sont moins familières. Il peut s'agir de domaines particuliers de l'informatique: compilation, bases de données, Intelligence Artificielle, programmation fonctionnelle, logique, programmation orientée objets, ...)
- les collègues désirant approfondir certaines questions. Il nous faut alors proposer les "bibles" dans chaque domaine (Knuth I, II, III, Aho-Ullman, ...)
- les étudiants, à propos desquels plusieurs difficultés surgissent: niveaux de lecture (DEUG, 2ème Cycle, 3ème Cycle, achalandage des bibliothèques, ...).

Bien entendu, de nombreux problèmes techniques se posent:

- comment collecter et diffuser l'information, afin que cette bibliographie reste vivante, représentative. Etant donné la profusion de titres qui paraissent, la question du délai de mise à jour de cette bibliographie est brûlante !
- faut-il diffuser de simples listes de titres, ou des bibliographies détaillées:
 - . tables de matières
 - . résumé du contenu,
 - . appréciations critiques sur l'intérêt, la lisibilité, le degré d'approfondissement ...
- quelle classification utiliser ?
 - . niveaux de lecture,
 - . thèmes,

L'objectif final n'est pas encore bien défini. Dans un premier temps, nous cherchons à tendre vers une bibliographie détaillée à l'usage de tous les enseignants de l'informatique en DEUG, découpée en quatre catégories:

- à conseiller aux étudiants,
- pour en savoir un bon peu,
- pour tout savoir,
- monographies.

Quelques rubriques sont envisagées (liste non exhaustive, non figée):

- Histoire de l'informatique et rapports divers,
- ouvrages généraux, grands classiques,
- algorithmiques
- programmation et génie logiciel,
- architectures,
- systèmes d'exploitation,
- langages, compilation,
- mathématiques de l'informatique et informatique théorique,
- bases de données
- Intelligence Artificielle,
- réseaux et télématique
- reconnaissances, synthèses et traitements divers - robotique
- N'importe Quoi Assisté par Ordinateur
- calculs scientifiques
- conception de systèmes d'information
- informatique industrielle, temps réel

Un début de liste est proposé en annexe 7. Envoyez vos propositions, suggestions et critiques à Jean-Paul Bertrandias, IMAG, BP 53X, 38041 Grenoble Cedex

4. LES MOYENS NECESSAIRES

4.1 L'ordinateur en DEUG

Il peut jouer plusieurs rôles:

- outil de formation générale
 - . outil personnel de l'étudiant (et du futur cadre)
 - + le triptyque bureautique (éditeur de texte, tableur, SGBD) est indispensable (même si nous pensons que sa place serait plutôt dans l'enseignement secondaire)
 - . outil pour les autres disciplines
 - + outils généraux (exemple Mathematica - DOS)
 - + utilisation de logiciels spécifiques (par exemple Macsyma, ...)
 - (doit être enseigné par les utilisateurs)
- outil pour l'enseignement de l'informatique expérimentale
 - . TP de démonstration (exemple: vitesse des algorithmes de tri)
 - . expérimentation en programmation

L'utilisation des ordinateurs en DEUG peut donc répondre à deux objectifs:

- connaissance de l'outil,
- connaissance des objets étudiés en Informatique.

L'équipement matériel aura donc une importance considérable quant à la qualité de l'enseignement . Un paragraphe sera consacré plus loin à cette question. Relevons ici quelques problèmes:

- le problème des ordinateurs personnels des étudiants.
 - . l'égalité des chances entre les étudiants n'est pas préservée,
 - . ils permettent de pallier l'insuffisance des moyens de l'Université,
 - . il est nécessaire de disposer de procédures de conversion de disquettes
 - . le problème de copie des logiciels est difficile (disposer de logiciels recopiables, même s'ils sont bridés ?)
- le problème du temps passé par les étudiants devant les ordinateurs.
 - . les résultats ne sont pas en rapport (trop de temps passé devant les machines, souvent sans réflexion),
 - . difficulté de contrôler ce temps.
- le problème de l'accès libre (ou libre service).
 - . difficulté de contrôler ce qui se passe dans la salle (jeux, piratage, ...)
 - . semble remplacer la notion de TP: activités équivalentes, mais sans encadrement.
 - indifférent : les étudiants sont assez grands pour gérer leur temps,
 - opposé: pas trop de temps à passer devant les machines en initiation par exemple
 - pour: utile pour l'auto-formation sur des logiciels simples accompagnés d'une documentation adéquate (tableurs, traitement de textes...) ==> surveillance du matériel
- le problème de la disparité des niveaux de connaissance
 - . la difficulté est d'occuper les étudiants qui, pour une raison ou une autre, savent déjà quelque chose
 - . proposer des sujets, qui permettent:
 - + aux non initiés de se former,
 - + aux initiés de synthétiser

Toutes ces difficultés sont bien connues. Des solutions ont été mises en place, au coup par coup, en fonction des installations propres à chaque Université. Cependant, tout n'est pas résolu pour le mieux.

4.2 Les fonctionnalités attendues

4.2.1 Fonctionnalités du matériel

On trouvera ci-dessous une liste de quelques avantages et inconvénients attachés aux différents types de matériel susceptibles d'être utilisés.

Calculatrices programmables

- avantages:
- . faible prix (1 000 F), entretien minimum, salle banalisée,
 - . peuvent favoriser l'analyse des problèmes en ramenant la programmation à sa juste place
 - . peuvent servir dans certaines autres disciplines (calculs en TP de physique, de maths)
 - . peuvent servir à étudier le fonctionnement des machines.

- inconvénients:
- . machines très limitées,
 - . vol

Micro-ordinateurs indépendants

Pas de partage de fichier mais imprimante commune possible. On est donc confronté ici au douloureux problème des copies de logiciels, même avec les disques durs qui entraînent en plus de gros problèmes de gestion et de virus. Pas de contrôle des consommables.

. les portables (pouvant être stockés dans une armoire - style Amstrad à 4 500 F)

- avantages:
- . salles banalisées,
 - . peu chers

- inconvénients:
- . fiabilité ?
 - . durée ?,
 - . vol

. les non portables (Compatibles PC, Mac, Atari...)

- avantages:
- . on a un grand choix et tous les prix (de 5 000 F à 20 000 F et plus)

- contraintes:
- . salle adaptée (normal pour des TP),
 - . surveillance et entretien

Micro-ordinateurs en réseau

- avantages:
- . partage de ressources, de fichiers, gestion des logiciels simplifiée (mise à jour, copies)
 - . possibilité de courrier électronique
 - . poste maître superviseur possible

- inconvénients:
- . limitation géographique des salles sauf dispositif de câblage spécifique (fibres optiques, répéteurs)
 - . augmentation du coût par micro

. serveur local de type micro -

- avantages:
- . Desserte suivant le type de serveur jusqu'à 50 postes
 - . gestion assez simple, ne nécessite pas de grande qualification

- inconvénients:
- . si plusieurs serveurs: gestion des sauvegardes si stockage fichiers utilisateurs,
 - . mise à jour des logiciels
 - . ne pas multiplier les serveurs micros: 1 mini à la place de 4 serveurs micros (1 mini coûte environ 200 000F)

. serveur local de type mini

- avantages: . possibilités serveur de logiciels et de stockage accrue
 . gestion centralisée
 . nombre de micros connectés important (plus de 100)
 . le système du serveur peut être différent de celui du micro: il peut y avoir un mini servant de machine d'exécution, le micro devenant un terminal d'accès
- inconvenients: . coût du serveur,
 . maintenance,
 . personnel qualifié (nécessité d'un ingénieur)

Terminaux reliés à un mini

- avantages: . les mêmes que pour la solution réseau et serveur local de type mini sauf pour l'unicité du système
- inconvenients: . idem ci-dessus mais en plus blocage en cas de panne (délicat dans un enseignement de masse)
 . logiciels moins variés que sur les micros,
 . limitations liées au terminal

Terminaux reliés à un gros ordinateur

On retrouve les mêmes avantages et inconvenients qu'avec la solution mini, mais la gestion est déportée à un service communautaire (Centre de calcul). Les logiciels systèmes permettent alors une meilleure gestion et le suivi des étudiants.

Connexion à distance

- modem dans la salle sur le serveur
- 1 poste pour des démonstrations d'accès à des machines lointaines (documentation scientifique par exemple).

4.2.2 Fonctionnalités du système d'exploitation

Principales fonctions attendues:

- gestion des fichiers : protection (problème des copies de logiciels), partage, copie
- partage de ressources communes (imprimantes, disques durs..)
- accès à des "gros serveurs"
- courrier électronique
- système bien répandu sur le marché ou non
- système très convivial (style Mac) ou moins (MS/DOS)
- poste du maître vidéo superviseur (style nano réseau Leanord)

4.2.3 Fonctionnalités des logiciels

Logiciels de développement

Evidemment suivant le type de langage utilisé (impératif, fonctionnel ...) nous avons des outils spécifiques.

- Editeur de texte:
 - . banalisé, pleine page, simple
 - . indépendant du logiciel ==> gain de temps à l'apprentissage
 - . intégré ==> facilite la mise au point mais peut favoriser le bricolage
- Edition de liens: inutile ou utile en initiation? indispensable ensuite
- contrôle de l'exécution: travail méthodique
- gestion de l'environnement
 - . gestionnaire de projet intégré (style "make")
 - . bibliothécaire
 - . interface langage d'assemblage

Logiciels de culture générale

Ces logiciels rentrent dans l'aspect formation générale "applications pédagogiques de l'informatique" et comportent donc un aspect pluridisciplinaire.

- logiciels de calcul symbolique, outils de simulation
- logiciels de bureautique : traitement de texte, tableur, gestionnaire de données
- systèmes experts ...

4.3 Les catégories de personnel

le personnel enseignant

- personnel enseignant habituel pour les cours, les TD, les TP:
 - . professeurs, maîtres de conférences de la 24ème section, en priorité
 - . collègues d'autres disciplines, formés, pour assurer l'utilisation de l'outil informatique dans leur domaine.
- moniteurs style IPT (étudiants qualifiés en maîtrise)
 - . pour l'encadrement en libre-service: renseignements, aides diverses
 - . initiation à la bureautique (aide à l'auto-formation, avec documents adaptés)
- moniteurs d'enseignement
 - . encadrement des TP

le personnel ATOS

- garçon de laboratoire
 - . gestion des consommables
 - . maintenance premier niveau (claviers, disquettes, écrans, ...)
- technicien de maintenance
 - . maintenance du matériel
 - . installation des logiciels
 - . gestion des disques durs (nettoyage; virus, ...)
 - . gestion du serveur micro et du réseau

- ingénieur
 - . aide à l'enseignement (ce que ne font plus les centres de calcul)
 - . gestion du réseau, enregistrement des étudiants
 - . développement de logiciels pour les TP, démonstrations
 - . gestion et connaissance des logiciels utilisés
 - . aide à la mise en place des TP d'observation
 - . suivi de l'évolution du marché (matériel et logiciel)
 - . gestion du parc matériel et logiciel

Citons pour mémoire le personnel qui n'est pas particulièrement lié à l'informatique, mais joue un rôle important: secrétariat, nettoyage.

4.4 Le chiffrage d'un atelier informatique

4.4.1 La notion d'atelier informatique

Les idées de base sous-jacente à ce chiffrage sont les suivantes:

- nécessité d'un environnement adapté:
 - . matériel élémentaire (tables, chaises, connectique)
 - . armoires (stockage documentation, disquettes),
 - . local de stockage du papier, des fournitures
 - . visualisation sur grand écran de la console pour démonstration de logiciels ou de techniques de travail (Data Show (8 000 F), vidéo projecteur (50 000 F))
 - . salle de réflexion (TD) contiguë
 - . atelier de maintenance
- notion de poste de travail:
 - . outil pour la discipline informatique,
 - . outil ouvert aux autres disciplines,
 - . suffisamment adaptés pour faire autre chose que du Pascal
- personnel:
 - . existence d'une équipe pédagogique,
 - . existence d'une équipe responsable de la gestion et la maintenance (technicien, ingénieur, ...)
- mode de travail:
 - . outils d'observation,
 - . Travaux Pratiques ne se limitant pas à l'usage de la machine
- coûts:
 - . pas de complexes sur le nombre de m^2 , les coûts d'investissement ou de fonctionnement
 - . parfaitement comparables à ceux des autres disciplines (Physique, Chimie, Biologie)

Cet ensemble d'idées trouve sa réalisation dans la notion d'**atelier informatique** pour l'enseignement de l'informatique en DEUG, qui va être développée ci-dessous.

Structure de l'enseignement

total atelier	24 sem. de 24h			576h par an
informatique	cours	1h30		
(par semaine)	TD	1h30	sur le cours	
		1h30	sur le TP	
	TP	4h30	3h encadrées au début	
		<hr/>	1h30 à la fin	
	total	9h	par semaine, par étudiant	216h par an

Découpage en groupes

cours	1 section	= 128 étudiants	
TD	1 groupe	= 32 étudiants	4 groupes
TP	1 sous-groupe	= 16 étudiants	8 sous-groupes

Surfaces pour un atelier

1 salle de TD	50 m ²
2 salles de TP	2*40 m ²
total	<u>130 m²</u>

Occupation d'un atelier par une section

- cours en amphi
- TD 4 gr. * 3h 12h par semaine
- TP 8 ssgr. * 4h30 36h
- total 48h par semaine

- accès nécessaire à la salle de TD pendant les TP, pour les phases de réflexion

- accès libre en dehors des heures encadrées

- il faut prévoir deux salles de TP

. si le groupe éclate en deux sous-groupes simultanés pour les TP

TD 4 gr. * 3h 12h par semaine

TP 4 ssgr. * 4h30 18h

. une section dans un atelier 30h par semaine

. sur 40h de disponibilité: taux d'occupation: 75%; temps d'accès libre: 10h

Nombre de postes de travail

- 1 sous-groupe de TP = 16 étudiants = 8 binômes
- il faut 10 postes de travail (8 binômes, 1 enseignant, 1 secours).
- les deux postes supplémentaires peuvent servir pour des étudiants isolés, ou en surnombre.

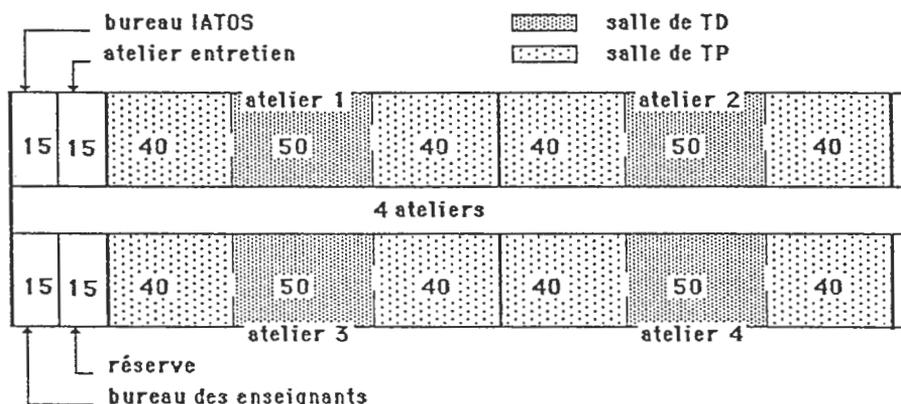
4.4.2 Chiffrage pour quatre sections (512 étudiants)

Surfaces nécessaires

Il faut:

- quatre ateliers (580 m²),
- en étage, pour des problèmes de sécurité,
- avec des bureaux à proximité: les ATOS, les enseignants,
- l'atelier de maintenance,
- la réserve (stockage du papier, de la documentation, des fournitures, ...)

Exemple de disposition:

Personnel enseignant

- | | | | |
|--|---|----------------|---|
| - 4 sections | 4*1h30 cours | 144 h/an | -> 1 enseignant |
| - 16 groupes | 16*3h TD | 1 152 | -> 6 enseignants |
| - 32 ss-groupes | 32*4h30 TP
mais encadrées, seulement | 3 456
2 016 | -> 7 enseignants
ou 3 enseignants
+ 9 moniteurs |
| - total: pour 512 étudiants: 10 enseignants, 9 moniteurs | | | |

Personnel IATOS

- 1 ingénieur
- 1 garçon de laboratoire
- 1/2 femme de ménage

Matériel

nombre	type	coût unitaire	coût total
- 80	postes de travail *	20 000	1 600 000
- 20	imprimantes	2 300	46 000
- 160	calculettes	600	96 000
- 4	Data-Show+rétro	15 000	60 000
- 80	cartes réseau	5 500	400 000
- 2	cablerie serveurs	25 000	50 000
- 80	valises logiciel **	4 200	336 000
-	meubles *** enseignement		186 240
-	meubles bureaux		46 560
-	équipement salles ****		100 000
		total	2 920 800

* exemple 1 poste: 80386 - 12MHz - 2 Mo - DD 20 Mo - VGA - 4 ans de maintenance

** logiciel: Pascal 5.5 500F; Lisp 1 200F; Prolog 1 200F; Works 1 100F

*** TP: 88 tables à 1 000F, 176 chaises à 120 F, 4 armoires à 3 000 F = 121 120 F
 TD: 44 tables à 1 000F, 176 chaises à 120 F = 65 120 F

**** tableaux, rideaux, sécurité (serrures, alarmes, ...), fixation ordinateurs, cablerie, ...

Coût d'investissement pour un étudiant/an

- amortissement sur 4 ans
- 512 étudiants/an pour 216h de formation: 2 048 étudiants formés sur 4 ans.
- investissement pour un étudiant = $2\,920\,800/2048 =$ environ 1 500F

Ces chiffres sont tout à fait comparables à ceux qui sont donnés pour les autres disciplines. A titre de comparaison, voici le coût prévisionnel d'une délocalisation de DEUG SSM et SNV en 1990 (exemple réel, donc nous protégeons l'anonymat !). C'est un DEUG 'classique', avec les séries MPI, PM, PC et BC. Pour les calculs, il a été considéré que:

- la physique est enseignée dans toutes les séries,

- la chimie également, sauf en MPI,

- la biologie uniquement en BC.

L'effectif global étudiant est de 500:

- 300 en première année,

- 200 en 2ème année.

	Physique	Chimie	Biologie
locaux	830 m ²	595 m ²	519 m ²
matériel pédagogique tables, chaises, armoires	1 305 KF 70 KF	1 500 KF 42 KF	3 097 KF 155 KF
total	<u>1 375 KF</u>	<u>1 542 KF</u>	<u>3 252 KF</u>
étudiants concernés	500	370	160
nombre m ² par étudiant	1,66	1,60	3,2
coût par étudiant (sans l'amortissement)	2,75 KF	4,16 KF	20,3 KF

remarque : la biologie pourrait accueillir 3 fois plus d'étudiants avec le matériel mis en place (heureusement...)

Il n'y a donc pas lieu de faire des complexes ! Au contraire, nous avons maintenant des bases pour exiger l'équivalence de traitement.

Coût de fonctionnement

Ce coût n'a pas été estimé. Les différents éléments sont là pour mémoire.

- salaires des IATOS et des enseignants
- rémunération moniteurs IPT
- fluides 12 heures par jour
 - . électricité
 - . chauffage
- informatique
 - . disquettes: 3 par étudiant
 - . papier (1 boîte par jour sur 2 ateliers)
 - . ruban imprimante (1/semaine/imprimante)
- entretien
 - . fournitures de maintenance
 - . pièces détachées
 - . documentation technique
- documentation
 - . livres techniques et pédagogiques sur les logiciels
- formation
 - . IATOS, aux logiciels, à la maintenance

CONCLUSION

A l'issue de ce compte-rendu, il apparaît clairement que les Journées de Nantes n'ont pas permis de traiter entièrement le problème de l'enseignement de l'informatique en tant que discipline dans les DEUG scientifiques. Cependant, il semble bien que nous puissions faire notre slogan:

**Enseigner l'informatique en tant que discipline dans les DEUG scientifiques,
c'est possible !**

Quelles actions devons nous mener dès maintenant pour que ce slogan devienne réalité:

- au sein de la communauté informatique (SPECIF)

- reprendre les idées proposées dans ce compte-rendu, les critiquer, les améliorer, de manière à aboutir à un accord général. L'idéal serait que ce compte-rendu fasse l'objet d'un large débat au sein de chaque établissement. A vous de l'organiser, et de faire remonter vos conclusions à SPECIF.
- promouvoir les actions permettant de conforter l'image de la discipline informatique. Nous attendons vos réactions quant à la définition de l'informatique, quant à la bibliographie de base, ...

- au sein de nos établissements

- il nous faut militer pour que nos idées soient acceptées. Impliquons nous plus dans leur fonctionnement, dans les conseils divers et variés.
- impliquons nous dans la définition de la politique de l'établissement, avec sérieux et réalisme. Quelques exemples où notre intervention pourrait être importante:
 - pour assurer l'homogénéité du matériel, éviter son éparpillement (mieux vaut une salle correctement équipée que 2 ou 3 micros à droite et à gauche)
 - pour mettre en place un atelier de maintenance interdisciplinaire quand un technicien à temps complet ne se justifie pas
 - pour faire donner sa juste place à l'informatique souvent reléguée dans des locaux "bricolés", mal adaptés car c'est " le petit dernier " et la maison est pleine.

- avec le Ministère

- demander le maintien des opérations spécifiques pour obtenir des réductions tant soit sur le matériel que le logiciel
- discuter des relations avec les éditeurs de logiciels: assouplissement des conditions d'utilisation de logiciels, licence site ...
- demander la mise en place d'une base de données de matériel et logiciel accessible via minitel: nouveautés, prix, conseils

Dans l'immédiat, envoyez vos réactions, vos propositions, vos critiques à

Michel LUCAS,
ENSM-Informatique,
1 rue de la Noë,
44072 Nantes Cedex 03.

Elles serviront à étayer notre dossier.

Préparez vous également à venir au Colloque de Lille, les 18 et 19 Septembre 1990. Il nous faut montrer un front uni vis-à-vis des autres disciplines qui seront présentés. Nous démontrerons ainsi que notre discipline existe, et qu'elle est adulte.

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 1

Liste des participants

Aix-Marseille I	JULLIEN Pierre
Aix-Marseille II	GIANNESINI Jacqueline
Angers	BOYER Jacques
Antilles-Guyanne	ADELAIDE Bertille
Besançon	LASALLE Marie-France, TREHEL Michel
Bordeaux	SENIZERGUES Géraud
Brest	MARCE Lionel
Clermont-Ferrand II	BONNEMOY Claude, SCHNEIDER Michel
Grenoble I	BENSALEM Sadek, BERTRANDIAS Jean-Paul, CHABRE-PECCOUD Monique, LEVY Michel, SCHOLL Pierre-Claude
Le Mans	PARCHEMAL Yannick
Lille I	LEGUY Bernard, LEGUY Jeannine, STEEN Jean-Pierre
Metz	CANSELL Dominique
Mulhouse	DESCHIZEAUX Pierre
Nancy I	JARAY Brigitte, LEVY Nicole
Nantes Nantes ENSM	ALLIAUME François, GUILBAULT François, MARTIN Philippe LUCAS Michel
Nice	BOUSSARD Jean-Claude, HERVIER Yves
Orléans	FERRAND Gérard, LE BERRE François, LORHO Bernard
Paris VI	MADAULE Françoise
Pau	RAFIQ Omar
Poitiers	BOURDIN Jean-Jacques
Rennes I Rennes INSA	BARRE Jacques, COSTE Michel NICOLAS Pierrick
Saint-Etienne	AHRONOVITZ Yolande
Toulon	RABIZZONI Patrice
Toulouse III	MASSIE Henri, MASSIP-PAILHES Louis, RAYNAUD Régine

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 2

DEUG scientifiques: pourquoi ? pour qui ?

Michel COSTE (PROMOSCIENCES)

L'association PROMOSCIENCES est née de la rénovation des DEUG, lancée en 1985. Une des préoccupations majeures était de valoriser les DEUG scientifiques (notamment par rapport aux formations "concurrentes" du réseau bac+2), d'affirmer leur rôle (pourquoi) et leur public (pour qui).

Le rôle:

Les DEUG scientifiques sont le premier pas d'une formation supérieure longue. La principale fonction du DEUG est d'amener les étudiants aux deuxièmes cycles des Universités. Les DEUG scientifiques font partie intégrante de l'acte formateur global de l'Université.

En corollaire, les DEUG ne doivent pas être coupés des formations universitaires commençant à bac+2 (pas de "collège universitaire"). La qualité des formations de second et de troisième cycle dépend de la qualité des DEUG. Il doit y avoir cohérence. Les enseignants intervenant en DEUG ne doivent pas former une catégorie séparée. La spécificité de l'enseignement universitaire (osmose avec la recherche) doit jouer aussi en DEUG - la mise en œuvre de ce principe n'est pas évidente.

Le public:

L'accès aux DEUG scientifiques ne peut pas faire l'impasse sur la formation secondaire antérieure. Le public des DEUG scientifiques est naturellement formé des bacheliers scientifiques (C, D, E). Les bacheliers technologiques sont à accueillir préférentiellement dans les formations technologiques (STS, IUT), qui ne doivent pas servir de voie de détournement du DEUG pour des études universitaires longues.

La phase de rénovation des DEUG scientifiques n'a pas vu une implication en masse des informaticiens (je sais, par expérience locale, qu'une part de la responsabilité de cette absence est portée par les collègues des autres disciplines). Comme le montrent ces Journées, la situation change. Ce changement survient à un moment où les DEUG se trouvent confrontés au problème de l'afflux des bacheliers (alors que la "rénovation" s'était déroulée dans une période de croissance modérée, voire de stagnation, des effectifs). Cet afflux fait ressurgir la question du rôle et du public des DEUG, en particulier scientifiques. Il faut prendre en compte dans la réflexion sur l'avenir des DEUG la volonté politique du gouvernement de ne pas instaurer de sélection à l'entrée, et aussi le fait que, malgré les efforts, les moyens supplémentaires ne suivent pas l'augmentation des charges.

Plusieurs réponses peuvent être proposées:

- Faire du premier cycle un prolongement de l'enseignement secondaire, et instaurer une coupure avec le second cycle. Ceci n'est sûrement pas souhaitable, et en tout cas la démarche de SPECIF se situe résolument en dehors de ce schéma.
- développer de nouvelles sorties à bac+2. On peut penser à des formations technologiques, mais alors pourquoi ce rôle ne pourrait-il pas être rempli par des formations existantes (STS, IUT), qui doivent évoluer au même titre que les DEUG ? Y a-t-il alors place pour une formation scientifique générale orientée vers une sortie à bac+2, et si oui, quelle doit être son articulation avec un DEUG conduisant aux seconds cycles et en quoi devrait elle avoir une spécificité universitaire ?
- il est de toute façon inévitable de se poser le problème du devenir des étudiants qui échouent dans le DEUG. Faciliter et organiser les réorientations en concertation avec l'ensemble du réseau bac+2 apporterait déjà une nette amélioration.

On répète que notre pays manque de gens ayant une formation supérieure longue, et que les écoles d'ingénieurs ne pourront pas répondre à ce besoin. Dans ces conditions, on peut estimer que le nombre de diplômés de DEUG scientifiques poursuivant leurs études en second cycle (professionnalisé ou "classique") doit suivre la progression du nombre des bacheliers scientifiques. Et donc, que cet accès au second cycle est plus que jamais la finalité première des DEUG (sans négliger, comme dit plus haut, les problèmes de réorientation). Il me semble que c'est dans cette optique que SPECIF travaille sur l'enseignement de l'informatique en DEUG, en considérant à la fois la formation des informaticiens et celle des autres scientifiques ayant à utiliser l'outil informatique. Mais si la finalité reste la même, il n'empêche que l'évolution irrésistible de l'enseignement universitaire vers un enseignement de masse va le modifier profondément (*), et nous poser des problèmes auxquels il ne sera pas facile de répondre, surtout avec des contraintes de moyens. Il est par exemple irréaliste de s'orienter vers un encadrement lourd, et il faut plutôt réfléchir sur les moyens de susciter et de favoriser l'autonomie et la démarche personnelle de l'étudiant.

Ces journées rassemblent des informaticiens. Il n'y a pas et il est clair qu'il n'y aura pas de DEUG d'informatique. Les DEUG scientifiques sont pluridisciplinaires. Avec sans doute une spécialisation qui peut s'accroître au fil des deux années de la formation, mais qui ne se réduit jamais à une seule discipline. Il faut reconnaître que la pluridisciplinarité n'est souvent qu'une simple juxtaposition. Or, pour développer un véritable esprit scientifique chez l'ensemble des étudiants, il est indispensable d'arriver à donner un contenu réel à cette pluridisciplinarité. C'est une tâche difficile et qui bouscule bien des habitudes. Mais est ce que nous ne revendiquons pas, avec notre spécificité d'universitaire, la capacité à l'innovation et au changement ?

(*) PROMOSCIENCES organise les 31 Mai et 1er Juin, à Nancy, un colloque sur le thème "Vers une Université de masse ?".

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 3

L'INFORMATIQUE DANS LES D.E.U.G. SCIENTIFIQUES

Jean-Claude BOUSSARD (Nice)

Le lecteur trouvera dans cette annexe une reproduction des transparents utilisés par Jean-Claude Boussard lors de son exposé d'introduction.

Le contenu est suffisamment explicite pour que les thèses défendues par l'orateur puissent être facilement reconstituées. Il faut prendre ce texte comme un ensemble de réflexions, peut-être un peu provocatrices, destinées à engendrer quelques réactions d'approbation ou de rejet !

Il a servi de toile de fond à nos travaux au cours des Journées et, même s'il n'a pas été adopté formellement par les participants, on peut considérer qu'il a reçu un accord tacite de la part de l'assemblée.

Pour gagner quelques pages, nous avons diminué largement la taille des caractères utilisés, et parfois regroupé plusieurs transparents sur une même page. Que Jean-Claude Boussard veuille bien nous excuser d'avoir ainsi défiguré son travail!

Ca va pas...

Du lycée à l'université

Ne s'en prendre qu'à nous

DEMAIN...

Ca peut aller mieux !

L'utopie

L'idéal

Le réaliste

APRES - DEMAIN ?

Ca coûtera très cher...

Ce qui dépend peut-être de nous

Ce qui ne dépend que du citoyen

CE QUI NE VA PAS

Du lycée à l'université

- Depuis 50 ans
Une science naît sous nos yeux

ET NOS ENFANTS L'IGNORENT

- => Ineptie des programmes
- => Inadéquation des méthodes

- de la seconde à la licence

TOUTES LES "GRANDES" DISCIPLINES
TRICHENT

sous couvert d'un enseignement général
qui peut parfaitement être acquis autrement,
les cursus actuels ne visent que leur propre intérêt

- => METTRE TOUTES LES DISCIPLINES
SUR LE MEME PIED

- des exercices extraordinairement pointus
en français
en mathématiques
en sciences physiques

PEUVENT FAIRE PLACE AVEC PROFIT

à la découverte d'un algorithme de tri
à l'analyse d'un parcours de graphe
à la programmation d'une simulation
...

Ne nous en prendre qu'à nous

(1)

- Convaincre nos interlocuteurs que **NOUS AVONS QUELQUE CHOSE A DIRE**
 - => **BIBLIOGRAPHIE**
exhaustive
structurée
commentée
évolutive
- **ARTICULATION AVEC LE SECONDAIRE**
Tout le monde semble convaincu
 - => **MAIS QUE FAUT-IL FAIRE ?**
- **ARTICULATION AVEC LES UTILISATEURS DE NOS "PRODUITS"**
 - => étudiants
parents
employeurs
politiciens

Sinon, prévoir divorce IUT / DEUG
analogue à celui Ecoles / Facultés
- **ARTICULATION AVEC L'EUROPE**
 - => **ACTIONS COMMUNES**
à toutes occasions

Ne nous en prendre qu'à nous

(2)

- et surtout...

ARTICULATION AVEC NOUS-MEMES

- => **NI ENFLURE de la recherche**
NI SCLEROSE de l'enseignement
- => **NI CONDESCENDANCE du théoricien**
NI INCREDULITE du praticien
- => **NI CONFLIT INFORMATICO - INFORMATIQUE**
 - en présence
des autres disciplines
ou des autorités de tutelle
- => **NI TIMIDITE OU AUTO-CENSURE**
pour obtenir notre digne place
et promouvoir
la discipline **INFORMATIQUE**
- = **ne pas disparaître dans l'opinion publique comme mécanique, cosmographie
ou sciences "naturelles"**

COMMENT ALLER MIEUX

L'utopie

S'il s'agit de préparer les jeunes "élites" à la vie qui les attend :

- ESPRIT D'OBSERVATION ET D'ANALYSE,
D'EXPERIENCE ET DE SYNTHÈSE
- MEMOIRE, SENS DE LA DEDUCTION
PRATIQUE DE L'EXEMPLE
- MAITRISE DE SOI
ET INSERTION DANS LE MONDE REEL
- COMMUNICATION
ECRITE, ORALE ET CORPORELLE
- UNE CERTAINE FORME D'ESTHETIQUE

=> Alors contentons-nous d'un DEUG unique équitablement réparti entre

- CONNAISSANCE DU MONDE
 - . Géographie physique et comparée
 - . Histoire des peuples et géopolitique
 - . Histoire des arts et des croyances
 - . Histoire des sciences
- CONNAISSANCE DE L'INDIVIDU
 - . Sciences de l'homme
 - . Education physique et sportive

L'idéal

D.E.U.G. Scientifique tri-disciplinaire

NOUS DEVONS ASSUMER

- LES OBJECTIFS DE LA MISE EN PLACE
DE VERITABLES MODULES INFORMATIQUES

- => culture générale
- => ou préparant à une culture informatique pointue
- => ou au service des autres disciplines

cette trichotomie n'est peut-être pas fatale
MAIS IL FAUT EN PARLER

- L'EQUILIBRE PARFAIT DE CES MODULES
avec par exemple deux autres disciplines
 - . l'une obligatoirement scientifique
 - . l'autre scientifique ou
 - en Sciences humaines
 - en Sciences Economiques
 - en Education Physique et Sportive

- LE CONTENU ASSEZ PRECIS
de ce que nous estimons souhaitable d'enseigner

=> D'OU LA PROPOSITION QUI SUIT EN TANT QUE BALLON D'ESSAI

L'INFORMATIQUE EN TANT QUE DISCIPLINE

APPRENDRE A RECONNAITRE ET FAIRE RECONNAITRE

- Informatique "fondamentale"
- Informatique "expérimentale"

TOUTES DEUX AUSSI RESPECTABLES

- = Banalisation des enseignants
- = Respect réciproque des méthodes et du travail
- = Uniformité des carrières
- => SUPPOSER PAR EXEMPLE 260 HEURES
 - par étudiant
 - par année

dans l'une des hypothèses

- culture scientifique "générale"
- préparation à une culture informatique pointue

INFORMATIQUE FONDAMENTALE (100, 100)

- COURS MAGISTRAUX (séances d'1h ?)
- TD TRADITIONNELS (séances de 1h 30 ?)
 - . exercices papier/tableau
 - . exposés d'étudiants ("applications")

=> THEORIE GENERALE
DES MACHINES ET DES PROGRAMMES

=> THEORIE DES LOGICIELS DE BASE

- langages
- systèmes d'exploitation

=> IMPLICATIONS SOCIALES

=> GRANDES APPLICATIONS

 I. Machines et programmes

- (15, -) LOGIQUES COMBINATOIRES
ET SEQUENTIELLES
- (15, 15) ARCHITECTURES D'ORDINATEURS
ET DE RESEAUX
- (20, 20) FONCTIONS CALCULABLES
ALGORITHMIQUE ET COMPLEXITE
- (25, 25) THEORIE DE LA PROGRAMMATION
- Structures de données
 - Types abstraits
 - Structures de commande
 - Pré- et post-conditions, invariants
 - Style de programmation
 - . Impératif, fonctionnel, déductif
 - . Procédural ou par objets
-

II. Outils et applications

- (-, 20) THEORIE DES LANGAGES FORMELS
Morphologie, syntaxe et sémantique
Notion de langage régulier
Notion de langage algébrique
- (-, 20) THEORIE DES SYSTEMES
MONOPROCESSEURS
Processus et ressources
Mémoires partagées et virtuelles
Notion de multiprogrammation
Temps partagé
Notion de multitraitement
- (9, -) INFORMATIQUE ET SOCIETE
Implications économiques
Implications sociales et politiques
- (16, -) GRANDES APPLICATIONS EN ...
Bureautique et PAO
Calcul scientifique
Conduite de processus
Gestion et aide à la décision
Jeux et recherche opérationnelle
Simulation et CAO
Statistiques et analyse de données
Vision et robotique
... ?

**INFORMATIQUE
EXPERIMENTALE
(160, 160)**

- TRAVAUX PRATIQUES (séances de 3h ?)
- TD D'INTRODUCTION (séances d'1h ?)
 - . spécification des problèmes
 - . description des outils
- => LE MARCHE DE L'INFORMATIQUE :
 - configurations, visites,
 - catalogues, brochures techniques
 - = problèmes
 - de coûts
 - de formes d'exploitation
 - de documentation et maintenance
- => MANIPULATION DE PROGICIELS
 - = problèmes
 - de convivialité
 - d'adaptabilité
- => MANIPULATION DE SYSTEMES
 - = problèmes
 - de fiabilité (exactitude + robustesse)
 - de réutilisabilité
- => PROGRAMMATION EFFECTIVE
 - = problèmes
 - de structuration de programmes
 - de langages et de styles de programmation
 - de performances statiques et dynamiques

Première année

(20) MARCHE DE L'INFORMATIQUE

- Microordinateurs personnels
 Interfaces simples
- Stations de travail graphiques
 Gestionnaire de fenêtres
 Menus déroulants et arborescents
 Interfaces souris

(20) PROGICIELS

- Tableurs
- Traitements de textes
- ... ?

(20) SYSTEMES ET ENVIRONNEMENTS

- MS/DOS ou équivalents ?
- Mac Intosh, ... ?

(100) LANGAGES ET PROGRAMMATION

- Basic, ou Fortran, ou Cobol, etc. ?
- Pascal, ou Ada, ou Eiffel ? (I)

Deuxième année

(20) MARCHE DE L'INFORMATIQUE

- Miniordinateurs, réseaux locaux
- Grosses configurations
- Installations industrielles
- Grands réseaux

(20) PROGICIELS

- SGBD's
- DAO
- CAO
- Systèmes experts
- ... ?

(20) SYSTEMES ET ENVIRONNEMENTS

- Unix
- VMS, ou PIC ou ... ?

(100) LANGAGES ET PROGRAMMATION

- Pascal, ou Ada, ou Eiffel ? (II)
- Lisp, ou ML, ... ?
- Prolog III, ou ... ?

 Deuxièmes
 et troisièmes cycles

NE PAS ACCEPTER L'IDEE

plus répandue qu'il n'y paraît

DE NE PAS TROP EN FAIRE EN PREMIER CYCLE

sous peine ne n'avoir plus rien à enseigner
dans les cycles ultérieurs !

NOTRE DISCIPLINE
EST ETONNAMMENT RICHE

- EN PROFONDEUR

- . Théories
- . Méthodes
- . Techniques

- EN SURFACE

- . Domaines connexes
- . Applications

=> La proposition qui précède
ou toute autre analogue

N'EST QUE LE VERNIS
UN PEU SOLIDE MAIS MINIMUM

suffisant pour qu'un "scientifique" puisse se dire "INFORMATICIEN" ...

**AU TITRE
DE L'APPROFONDISSEMENT**

PHYSIQUE DES ORDINATEURS

Optronique

... ?

ARCHITECTURES AVANCEES**THEORIE DE LA CALCULABILITE**

Algorithmique avancée

**ANALYSE SYNTAXIQUE
ET COMPILATION****PROGRAMMATION MULTI-STYLES****MANIPULATION**

de langages

et de systèmes de toutes sortes

TOUTES LES GRANDES APPLICATIONS

...

**AU TITRE
DES NOUVEAUTES**

Architectures parallèles (matériels et logiciels)
Analyse combinatoire, algorithmique des graphes
Algorithmique parallèle et distribuée

Sémantique des langages de programmation
Théorie et pratique de la réécriture
Spécifications formelles et tests de programmes
Programmation industrielle et Génie logiciel

Modélisation mathématique des systèmes
Grands codes scientifiques

Théorie des SGBD

Intelligence artificielle, théorie de l'apprentissage

Traitements graphiques, synthèse d'images
Vision et reconnaissance des formes
Reconnaissance et synthèse vocale

Conception et implémentation (grands projets)
de langages
de systèmes d'exploitation et de réseaux
d'environnements de programmation
d'interfaces graphiques
de jeux
de SGBD
de systèmes experts
de systèmes robotisés

...

Quel boulot !

Le réaliste

- **POUR QUAND**
de tels D.E.U.G. tri-disciplinaires ???
- **Y-A-T-IL**
des solutions de repli ???
- **FAUT-IL**
les mettre en oeuvre ???
- **DEVONS-NOUS NOUS-MEMES**
investir dans de telles solutions imparfaites ???

Le présent colloque
doit répondre à ces questions

=> faute de quoi
D'AUTRES REpondront POUR NOUS

Bibliographie

UN TRAVAIL ESSENTIEL DU PRESENT COLLOQUE ET DES SUIVANTS...

- Pas essentiel qu'elle soit pléthorique
mais fondamental qu'elle soit STRUCTUREE
- => **PAR EXEMPLE**, à titre purement indicatif :
- **OUVRAGES DE LIBRAIRIE**
 - . Histoire et insertion de l'Informatique
 - . Introduction à l'Informatique
(pas exclusivement logicielle)
 - . Informatique "deuxième niveau"
(sans langage particulier)
 - . Langages particuliers
Ada, Basic, C, Eiffel, Fortran,
Lisp, Modula 2, Pascal, Prolog, etc.
- **TOUS MATERIELS PEDAGOGIQUES
RELEVANT D'UNE BOURSE D'ECHANGE**
 - . Polycopiés et séquences audio-visuelles
 - . Exercices corrigés
 - . Séries de travaux pratiques
 - . Sujets de mini-projets commentés

POURQUOI SI CHER ?

Ce qui dépend peut-être
de l'enseignant-chercheur
informaticien

S'INFORMER

INFORMER ET CONSEILLER

ORGANISER

INNOVER

=> PAR EXEMPLE :

- LA FORME DES SEANCES

- . de cours
- . d'exercices et d'exposés
- . de travaux pratiques

- LA PEDAGOGIE DE GROUPE

- . en nature
- . en taille
- . en type et en volume de matériel
- . etc.

MAIS ...

Ce qui ne dépend
que du citoyen

DECIDER

- DU NOMBRE D'ETUDIANTS

- DES GRANDS TYPES D'OBJECTIFS

- DES NIVEAUX A ATTEINDRE

ET S'EN DONNER LES MOYENS

= BULLETIN DE VOTE

= POLITIQUE AU SENS NOBLE

=> L'ENSEIGNANT-CHERCHEUR
EN TANT QUE TEL

=> L'INFORMATICIEN EN PARTICULIER

N'A PAS A ASSUMER
ET ENCORE MOINS A PRONER

LA DEGRADATION EVENTUELLE
DE CE TYPE D'ENSEIGNEMENT

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 4

ENSEIGNEMENT DE L'INFORMATIQUE EN EUROPE

Jean-Pierre STEEN (LIFL)

L'enquête qui est résumée ici a été faite en écrivant à différents organismes des pays de la CEE pour obtenir des documents. Les résultats de cette enquête proviennent d'une compilation des différents documents reçus.

ECHANTILLON

<i>Pays</i>	<i>Documents obtenus</i>	<i>Universités</i>
Allemagne	L P	Dortmund - Karlsruhe - München - Nürnberg - Paderborn - Stuttgart
Belgique flamande	-	Leuven (Louvain)
française	L	
Danemark	L	-
Espagne	-	-
Grèce	-	-
Grande-Bretagne Angleterre	-	Thames (London) - Kent (Canterbury) - Salford (Manchester)
Ecosse	-	-
Irlande	-	Cork - Limerick
Italie	L P	Milano - Pisa - Trento
Luxembourg	-	pas d'enseignement supérieur
Pays-Bas	L	Utrecht
Portugal	-	Lisboa

L : liste des formations supérieures en informatique

P : programme officiel

remarques

- trois universités proposent des formations en informatique au Danemark: KobenHavn, Arhus et Odense. Pas d'autre information actuellement.
- aucune information reçue à ce jour d'Espagne ou de Grèce
- le Luxembourg n'a pas d'enseignement supérieur.
Il passe des conventions avec les pays voisins. Environ 500 de ses ressortissants s'engagent, tous les ans, dans des études supérieures.

ALLEMAGNE (RFA)**Généralités**

Deux sortes d'établissements:

- Universités et Universités techniques (4 à 5 ans)
- FachHochSchulen (Ecoles d'ingénieurs en 3 ans - dans chaque province, créées en 1968)

Les HochSchulen seraient comparables aux IUT

Liste des établissements :

En gras, les établissements ayant des cursus informatique et des restrictions d'admission (au semestre d'été 1987)

Universités Techniques

Aachen
 Augsburg
 Bamberg
 Bayreuth
 Bochum
 Bonn
 Bremen
 Dortmund
 Duisburg
 Düsseldorf
 Erlangen-Nürnberg
 Frankfurt
 Giessen
 Hagen
 Hamburg
 Hannover
 Kaiserslautern
 Karlsruhe
 Kassel
 Kiel
 Mainz
 Mannheim
 München
 Münster
 Oldenburg
 Osnabrück
 Paderborn
 Passau
 Saarbrücken
 Stuttgart
 Tübingen
 Wuppertal
 Würzburg

FachHochSchulen

Aachen
 Bielefeld
 Darmstadt
 Dortmund
 Giessen
 Hamburg
 Hannover
 Hildesheim
 Karlsruhe
 Kassel
 Marburg
 Neubiberg-München
 Siegen
 Speyer

Universités

Berlin
 Braunschweig
 Clausthal
 München

Universités sans filière informatique

Eichstätt
 Essen
 Freiburg
 Göttingen
 Hohenheim
 Lübeck
 Lüneburg
 Rheinland-Pfalz
 Witten-Herdecke

Divers (Centres de calcul, organismes, ...)

Bergish Gladbach
 Heidelberg
 Konstanz
 Köhl
 Oldenburg
 Resenbourg
 Rossdorf
 St. Augustin, Sieg
 Trier
 Ulm, Donau
 Wessling, Oberbay

ALLEMAGNE (RFA)

Les examens:

- avant l'entrée en Université:
 - . certificat de fin d'études secondaires (comparable à Bac+1)
- à l'Université, deux diplômes:
 - . Pré-diplôme (*Diplomvorprüfung*) (4 semestres - comparable à la Licence)
 - . diplôme principal (*Diplomhauptprüfung*) (5 semestres - comparable au DEA ou DESS)
- après l'Université:
 - . examen d'Etat (*Staatsexamen*) (pour entrer dans la fonction publique)
 - . Doctorat (*Promotion*) (de 2 à 6 ans)

Organisation des études

Deux semestres par an:

- Hiver: octobre - mars avec cours de mi-octobre à mi-février.
- Eté: avril - septembre, avec cours de mi-avril à mi-juillet
(travail individuel pendant les 5 mois sans cours).

Forme des enseignements

V: Cours (*Vorlesung*)

Ü: TD (*Übungen*)

P: TP (*Praktischen Übung an Rechnersystem*)

S: Séminaire (*Seminar*) 12 étudiants

HS: Séminaire supérieur (*Hauptseminar*)

Tutorat

Projets en groupe (*Projektgruppen*)

Stages (*Praktikum*) 26 semaines (1/2 année) à réaliser en partie (la moitié) avant le pré-diplôme

Restrictions d'admission

Publiées régulièrement par la Conférence des Recteurs

Associations:

GAMM : *Gesellschaft für Angewandte Mathematik und Mechanik*

NTG : *Nachrichtentechnische Gesellschaft*

GI: *Gesellschaft für Informatik* (créée en 1969)

Programme national

Premiers enseignements d'informatique en 1966 à Dortmund, Gummersbach, Furtwangen et Paderborn.

GAMM et NTG ont créé en 1968-1969 le premier programme officiel pour l'informatique (*Empfehlung zur Ausbildung auf dem Gebiet der Datenverarbeitung*)

Y ont participé des Mathématiciens, des Ingénieurs en Electronique et des Physiciens. Beaucoup de départements d'Informatique sont attachés à des facultés de Physique. L'analyse numérique reste dans les enseignements de base de l'informatique.

Ce programme est révisé régulièrement.

Grandes lignes du contenu:

- Informatique (25 à 50% selon le semestre)
- Mathématiques
- Electrotechnique
- discipline non informatique: 25%)

ALLEMAGNE (RFA)**Exemple de contenu du programme d'Informatique**

München, Université Technique

Titres des chapitres (traduction approchée)

- Informatique 1
 1. l'information et ses représentations
 2. algorithmique
 3. application des langages de programmation
 4. structuration par les procédures
 5. structures spéciales de calcul (fonctionnement théorique du micro-processeur)
 6. éléments d'assembleur (étiquettes et sauts, passage de paramètres)
 7. calculs récursifs
- Informatique 2
 1. base du codage et de l'informatique théorique
 2. binaire, calculs et réseaux
 3. conception des ordinateurs
 4. aspects machines de la programmation
- Informatique 3
 1. processus, communication et coordination dans les systèmes distribués
 2. systèmes d'exploitation et d'aide à la programmation
 3. interprétation et traduction des programmes
 4. langages formels
- Informatique 4
 1. calculabilité
 2. complexité
 3. style et formalisme en programmation
 4. banque de données et système d'information

Cours d'enseignement supérieur avec l'Informatique comme sujet principal

- Ingénieurs
 - T.H. Darmstadt
 - U. Stuttgart
- Economie
 - U.G. Duisbourg
 - T.H. Karlsruhe
 - U. Köln
- Médecine
 - U. Heidelberg
 - F.H. Heilbronn
- Informatique pédagogique
 - G.U. Paderborn
- Législation
 - Diverses *Hochschulen*

BELGIQUE

Généralités

Deux langues (seul programme obtenu: Louvain, en flamand)
Trois Cycles d'études, selon le modèle français

Liste des établissements

en gras, les établissements ayant des cursus d'Informatique

Communauté française

- U. Arlon (Fondation Universitaire Luxembourgeoise)
- U.L. Bruxelles**
- F.U. Bruxelles (Saint-Louis)
- F.Bruxelles (Théologie protestante)
- E.Bruxelles (Ecole Royale Militaire)
- F.E. Gembloux (Agronomie)
- U.C. Leuven (Louvain-la-Neuve)**
- U.E. Liège
- U.E. Mons
- F.U.C. Mons
- F.P. Mons
- F.U. Namur (Notre-Dame de la Paix)

Communauté flamande

- U.L. Leuven (Louvain)**

Pas d'information sur les autres formations.

Katholieke Universiteit Leuven

Formations

- Ingénieurs
 - . formation initiale d'ingénieurs en
 - + informatique
 - + mathématiques appliquées
 - + mécatronique
 - . formation complémentaire pour
 - + ingénieurs généralistes
 - + licenciés de mathématiques
- licence de Mathématiques Industrielles (la Licence belge correspond à notre Maîtrise)
- licence d'informatique (4 ans)
 - deux parties:
 - + *Kandidatuur Informatica* (2 ans)
 - + *Licentie Informatica* (2ans)
- complément de Licence (d'une autre discipline) en Informatique (2 ans)
 - Aanvullende Licentie Informatica*

remarque

Il y a donc à Louvain l'équivalent d'un DEUG orienté vers l'informatique. Il comprend, entre autres:

- . des mathématiques classiques (algèbre, analyse, ...)
- . des mathématiques orientées vers l'Informatique (logique, calcul numérique, statistiques, ...)
- . d'autres disciplines (sciences naturelles, économie, électronique, ...)

GRANDE-BRETAGNE et IRLANDE

Organisation des études

Liberté: les étudiants choisissent parmi les modules proposés et ceci dès l'école primaire.

Des filières sont conseillées, en particulier pour les études d'informatique dans le supérieur.

Ainsi, à l'Université du Kent (Canterbury), on peut, par le jeu de ces choix, obtenir une formation dans les spécialités suivantes:

- *Technical and Scientific Computing*
- *Computing in Humanities*
- *Computing in the Social Sciences*

La première année est un tronc commun; la spécialisation se fait en 2^{ème} et 3^{ème} années.

Diplômes

Le secondaire est sanctionné par deux examens. Les résultats sont notés avec des lettres de A à E.

- *O-level*
 - . enseignement général, 7 à 9 matières
 - . passé vers 16 ans
 - . les futurs étudiants d'informatique doivent y obtenir au moins C en Anglais et en Maths.
- *A-level (Advanced)*
 - . enseignement spécialisé, 3 à 4 disciplines
 - . passé vers 18 ans.
 - . il faut y obtenir A en math, physique et une matière technique, ainsi que français ou allemand pour engager certaines études d'informatique à l'Université du Kent.
 - . des conditions équivalentes sont appliquées dans les autres universités. Ainsi, Londres et Manchester exigent des A dans les disciplines en rapport avec les études choisies, et acceptent de moins bons résultats dans les autres disciplines. Pour entrer à Oxford ou Cambridge, il faut des A dans toutes les matières du *A-level*.

Ceci permet de contrôler les flux d'entrée et de sélectionner les étudiants.

Les diplômes du Supérieur sont:

- BSc. (BA) *Bachelor of Science (of Art)* préparé en 3 ou 4 ans.
- MSc. (MA) *Master of Science (of Art)* préparé en 1 an après le BSc.
- PhD.

Forme des enseignements

- cours magistraux (*Lectures*)
- conférences et séminaires (*short courses*)
- auto-formation (*solo courses*)
- tous les enseignants utilisent des supports de cours.
- projets industriels
- stage en entreprise
- celui-ci peut durer un an avec rémunération, entre la 3^{ème} et la 4^{ème} année, comme à la Thames University (Londres). C'est le principe du *Thick Sandwich*. Chaque étudiant est suivi par un tuteur dont l'avis est important dans la note finale.

Participation intense des industriels. Beaucoup d'étudiants se font parrainer pour leurs études.

Matériel

Exemple de l'Université de Limerick, Irlande

- Vax 11/780 et Vax 11/750
- 40 DEC Professional/350 computer
- 30 Amstrad
- des IBM

A l'Université de Salford, chaque étudiant dispose d'un micro-ordinateur pendant le cycle de ses études.

GRANDE-BRETAGNE

Contenu des cursus

Exemple: *Technical and Scientific Computing* à l'Université du Kent

filières

- *Computer Science*
- *Computer Systems Engineering*
- *Computer Communication*
- *Mathematics and Computer Science*
- *Computing, Operational Research and Statistics,*
- *Physics with Computing.*

Têtes des chapitres

- Première partie (première année)
 - . *Structured programming I. Pascal programming*
 - . *Structured programming II. Data structures and computer algorithm*
 - . *Computer system I. Introduction to operating systems and software tools*
 - . *Computer system II. Assembly language programming and machine architecture*
 - . *Computer system III. Occam and concurrent systems*
 - . *Computer peripherals*
 - . *Circuits and signals*
 - . *Digital systems I*
 - . *Electronics CAD (Computer Aided Design)*
 - . *Human communication and business skills*
 - . *Pascal programming and application*
- Deuxième partie (deuxième et troisième années)
 - . *Operating systems*
 - . *Functional programming and formal languages*
 - . *Software engineering*
 - . *Compiling techniques*
 - . *Scientific computing*
 - . *Computer networks and communication*
 - . *Digital systems II*
 - . *Information engineering*
 - . *Transmission techniques*
 - . *Control and communications theory*
 - . *Engineering practice*
 - . *Computer graphics*
 - . *Databases and Information systems*
 - . *Formal treatment of programming and programming languages*
 - . *Expert systems*
 - . *Text processing and document manipulation*
 - . *Theoretical computer science*
 - . *Networked systems*
 - . *Protocol engineering*
 - . *Language and systems*
 - . *group project*

Note: à l'Université de Salford, pour ces projets (*group project*) les étudiants sont regroupés à 4 ou 5, en provenance d'années d'étude différentes.

ITALIE

Organisation des études

Comme en France, des Universités et des Ecoles de spécialité, équivalentes des IUT.

Il existe des programmes nationaux.

Liste des établissements

Universités (en gras, celles qui délivrent les *Corsi di Laurea in Scienze dell'Informazione*)

Alessandria (Torino/Statale)
 Bari
 Bologna
 Genova
 l'Aquila
Milano/Statale
 Pisa
 Roma/La Sapienza
 Salerno
 Torino/Statale
 Udine

Ecoles (*Scuole dirette a fini speciali di informatica* qui délivrent le *Diploma (universitari) in Informatica*)

Acona
 Cagliari
 Camerino
 Messina
 Milano/Politecnico
 Pavia
 Perugia
 Pisa
 Trieste
 Trento

Durée des études: 2 ans

Diplôme

Corso di Laurea in Scienza dell'Informazione

Durée des études: 4 ans, en deux *biennio* de 2 ans

Trois *indirizzi*:

- *generale*
- *applicativo (gestione)*
- *tecnico (sistemi per l'elaborazione)*

Contenu des *biennio*

1. *Matematica*
Fisica
Informatica di basi
2. *Informatica di basi (Avanzata)*
Insegnamenti specialistici

ITALIE

Exemple de contenu

Université de Milano

- *Teoria e applicazione delle machine calcolatrici (1ère année)*
 - ... 1. Algoritmi
 - ... 2. Il linguaggio Pascal
- *Sistemi per elaborazione dell'informazione I (2ème année)*
 - ... parte I: introduzione ai sistemi operativi: UNIX.
 - ... il linguaggio di programmazione della Shell sh.
 - ... parte II: Strutture dati e algoritmi.
 - ... parte III: ... Introduzione al Linguaggio C ...
- *Elaborazione dell'informazione non numerica*
Linguaggi funzionali: struttura di un sistema LISP. ...
- *Linguaggi speciali di programmazione*
 - ... la struttura dei linguaggi Algol-like ...
 - ... la struttura dei tipi di dati in Algol68, Pascal e ADA
 - ... Cenni a Smaltalk, C++, Flavour, CLOS, ...
- *Documentation Automatica*
... SQL, un linguaggio "standard" per sistemi relazionali

Matériel

Exemple de l'Université de Milan

- première année
 - . HP9000/840
 - . 53 terminaux, Pascal et assembleur
- deuxième année
 - . 4 HP9000/500
 - . 16 terminaux, Unix, Langage C

PAYS-BAS

Organisation

comparable au système allemand

Liste des établissements

en gras, les *Universiteit* qui enseignent l'informatique (en général ce sont des universités scientifiques)

Amsterdam
 Amsterdam (Vrije Universiteit)
 Delft (T.U.)
 Eindhoven (T.U.)
 Twente
 Groningen
 Leiden
 Limburg
 Nijmegen
 Rotterdam
 Tilburg
 Utrecht
 Wageningen

Les 7 *Hogeshool* de l'Enseignement Professionnel Supérieur Technique qui forment des Ingénieurs en Informatique en 4 ans:

Eindhoven
 Emmen
 Enschede
 Gravenhage
 Groningen
 Heerlen
 Utrecht

organisation

Exemple de l'Université d'Utrecht (*Rijksuniversiteit te Utrecht*)

- Cycle d'études en 4 ans, décomposés en semestres
 - . première année: *Propedeuse*
 - . deuxième à quatrième années: *Doctoraal (basistudie, afstudeerdase* au dernier semestre)
- troisième cycle: *Tweede fase*

forme des enseignements

- *Colleges*
- *Werkcolleges*
- *Practicum*
- *Projecten*
- *Seminaria*
- *Stages*
- *Afstudeerfase* (environ 900 *werkuren*)

Les modules d'enseignement vont de 100 à 130 heures

matériel

- Harris HCX-9
- réseau Ethernet
- HP 9000/370 (serveur)
- 22 HP9000/340
- des SUN 3/60, Atari 1040T et Apple
- une grappe de 16 Transputer T800

PORTUGAL

Organisation

Diplômes délivrés par l'*Universidade Tecnica de Lisboa*

Licenciatura em Engenharia Electrotécnica e Computadores

- numerus clausus: 190
- formation à *profissao de Engenheiro Electrotécnico*
- organisation sur 5 ans, en semestres:
 - . avec trois branches (*ramo*) à partir de la troisième année:
 - + *Energia e electronica*
 - + *Telecomunicações e Electronica*
 - + *Sistemas e Computadores*
 - . et trois séries d'options (*opção*) enseignées, selon les branches, au deuxième semestre de la quatrième année et en cinquième année.
- les deux premières années comportent des enseignements de:
 - . *matematica, probabilidades e estatística*
 - . *mecanica general*
 - . *teoria dos circuitos, electronicos, electromagnetismo*
 - . *química general*
 - . *física experimental, termodinamica*

Licenciatura em Matematica Aplicada e Computação

- numerus clausus: 30
- formation en 4 ans, organisée en semestres,
- apparition de disciplines non informatiques au cours des premières années
 - . *mecanica*
 - . *electromagnetismo*
 - . *termodinamica*

CONCLUSION

1. Partout en Europe:

- organisation en semestres
- Cycle d'études de 4 ans et plus

2. Langages

- Pascal d'abord (Tous les informaticiens d'Europe connaissent ce langage), puis Modula II, et FORTRAN pour les Mathématiques Appliquées.
- Prolog (dans la spécialisation) et C (autour d'UNIX)
- Ada (dans les grandes universités scientifiques: Leuven, Milano)
- pas de BASIC (n'existe plus, on s'y attendait !)

3. Systèmes

Beaucoup de micros sous MS/DOS pour l'initiation, puis le système de l'ordinateur de l'établissement.

Partout des stations de travail sous Unix, mais ce système n'est apparu qu'une fois explicitement dans les programmes.

4. Disciplines annexes

- Mathématiques et Physique
- fréquente obligation d'étudier une discipline non Informatique

5. Limitations d'effectifs

- partout en Europe
- la pénurie généralisée d'inforamticiens est-elle liée à une pénurie de moyens ?

6. Programme national

Puissant moteur pour le développement de la discipline

7. Pas de bureautique

N'entre pas dans la formation des informaticiens.

Un langage utilisé dans toute l'Europe :

PASCAL

L'informatique fera-t-elle l'Europe ?

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 5

L'INFORMATIQUE EN TANT QUE DISCIPLINE DANS LES DEUG SCIENTIFIQUES CELA EXISTE.

UN EXEMPLE A L'UNIVERSITE DE METZ

Dominique CANSSELL (Metz)

1 PRESENTATION

1.1 DEUG rénové

Les DEUG A et B de l'université de Metz, sont gérés par les UFR scientifiques qui sont l'UFR MIM (Mathématique, Informatique, Mécanique et Automatismes) et l'UFR SciFA (Sciences Fondamentales Appliquées). Ce sont des DEUG rénovés dont les avantages sont les suivants:

- * huit semaines d'orientation (d'adaptation à la vie universitaire) durant lesquelles tous les étudiants suivent à peu près les mêmes cours. Ils peuvent donc au terme de ces semaines conforter ou changer l'orientation de leurs études.
- * 150 étudiants par séances de cours et 32 par groupes de travaux dirigés (TD), ce qui n'est pas excessif par rapport au nombre d'élèves dans une classe de lycée.
- * des contrôles fréquents, pour que l'étudiant puisse organiser et répartir ses efforts tout au long de l'année.

1.2 Le DEUG A

Il y a trois sections dont les thèmes principaux apparaissent dans leur sigle. Ces trois sections sont:

- M.I.E. (Mathématique, Informatique, Electronique)
- T.M. (Technologie et Matériaux)
- P.I.E. (Physique, Informatique, Electronique)

1.3 Le DEUG B

Une seule section:

- C.B.B. (Chimie, Biologie, Biochimie)

1.4 L'informatique en DEUG

Tous les étudiants des DEUG scientifiques ont un module d'informatique. Vu le nombre d'heures (370 pour les MIE et plus de 160 pour les autres) et le contenu de ces modules, chaque étudiant peut prétendre avoir eu une bonne formation de base en informatique.

2 MODALITES PRATIQUES

2.1 Nombre d'heures

	cours	travaux dirigés	salle machine
MIE	83	130	170
TM	59	79	80
PIE	57	79	70
CBB	56	56	50

2.2 Matériel

Les étudiants ont à leur disposition une salle machine de 16 PC du plan informatique pour tous.

Remarque: Les étudiants de TM utilisent pour certaines heures les PC de la filière technologique.

2.3 Logiciel

Besoins: un éditeur de texte,
un compilateur Pascal

Choix: un TURBO-Pascal

2.4 Documentation

A part leurs notes de cours et leurs documents personnels, les étudiants ont à leur disposition:

- * Le manuel du TURBO-Pascal
- * Le manuel d'utilisation du système

2.5 Nombre d'étudiants (1989-90)

MIE1	187	MIE2	67
TM1	74	TM2	53
PIE1	186	PIE2	94
CBB1	268	CBB2	133

3 FONCTIONNEMENT

3.1 Les cours

Ils sont assurés en totalité par des informaticiens de l'UFR MIM (24-01).

3.2 Les Travaux Dirigés

Ils sont assurés en totalité par des informaticiens de l'UFR MIM (24-01) et des vacataires informaticiens. La règle adoptée est que le responsable du cours prend en charge un des groupes de TD et prépare toutes les séances (seul ou avec les autres intervenants) pour que tous les étudiants d'une même section abordent les mêmes exercices.

3.3 Les heures machine

Ces heures machine servent surtout à conforter les notions apprises en cours et utilisées en TD, par des exercices de programmation (algorithmes écrits en TD) et elles servent également à développer un projet bien "dégressif" en TD.

Pour que l'étudiant puisse utiliser l'éditeur de texte et le compilateur Pascal, les premières séances (4 heures) sont encadrées. Le reste du temps n'est pas encadré, mais l'étudiant n'est pas livré à lui-même. Les travaux de programmation sont demandés ou développés en TD et il peut dialoguer avec les enseignants, les personnels IATOS qui s'occupent de tout le matériel du service informatique, et surtout avec des étudiants plus compétents que lui dans ce domaine. La qualité de certains travaux rendus et le taux d'occupation des salles montrent que cela est largement suffisant.

4 PROGRESSION

Le fil directeur de tous les cours d'informatique de DEUG est l'algorithmique. Nous considérons que c'est la matière de base. Par son biais nous abordons des notions:

- de système
- de langage de programmation impérative
- d'architectures, en décrivant les mécanismes d'allocation de mémoires, d'appel de sous-programme et en écrivant les algorithmes de codages et des opérations sur les entiers,
- de type abstrait et de programmation fonctionnelle en utilisant une approche fonctionnelle pour les listes linéaires

4.1 DEUG A et B première année

Même si le nombre d'heures diffère, la progression et le message à faire passer sont les mêmes. L'année est découpée en deux parties.

4.1.1 Les 8 premières semaines

Cours	16 h
TD	24 h (16 pour les CBB)
Machine	8

L'objectif de ce cours est de présenter la discipline informatique et d'aborder les points suivants:

- * algorithmique de base (affectation, conditionnelles, boucles, tableau)
- * notion de correction d'algorithme (assertions)
- * notion de coût d'algorithme (comparaison d'algorithmes)
- * historique, présentation, description et utilisation de la machine (système MS/DOS, éditeur de texte, compilateur, exécution de programme).

4.1.2 Les 20 semaines suivantes

	MIE	TM+PIE+CBB
cours	24 h	20 h
TD	36 h	20 h
Machine	60 h	20 h

L'objectif de ce cours est de présenter la programmation structurée et de l'utiliser dans la décomposition de problème. Les points suivants sont abordés:

- * programmation structurée (fonction et procédure)
- * décomposition d'un problème (programmation arborescente descendante)
- * calcul de propriétés (obtention d'un algorithme classique sur les tableaux à l'aide d'une propriété, boucle while avec variable booléenne)
- * type et algorithme de base sur les enregistrements
- * type et algorithme de base sur les fichiers (fichier standard et TURBO)
- * graphisme

Remarque: Les étudiants de MIE développent en TD une partie d'un gros projet. Toutes ces parties sont regroupées à la fin, et donne lieu à une exécution d'un gros programme.

4.2 Le DEUG A Deuxième année

En deuxième année les objectifs et donc les contenus diffèrent entre le DEUG A et B.

4.2.1 Les 14 premières semaines

	MIE	TM+PIE
cours	21	21
TD	35	35
Machine	42	28 (sur l'année)

L'objectif de ce cours est d'expliquer les vertus de la programmation récursive. La récursivité est présentée comme un outil puissant aussi bien pour la conception que pour la compréhension des algorithmes, et cela quel que soit le type de problème abordé (calcul, comportement, tri, liste). L'élimination de la récursivité est enseignée pour que les étudiants puissent faire le lien avec ce qu'ils ont appris en première année, mais surtout pour qu'ils puissent, plus tard dans leur vie professionnelle, programmer leurs algorithmes récursifs (proprement et correctement écrits) en utilisant des langages non récursifs. L'approche fonctionnelle des listes linéaires leur présente la notion de type abstrait et un autre style de programmation, le style fonctionnel. Les points abordés sont les suivants:

- * récursivité (théorème de récurrence, arrêt et exécution d'un algorithme récursif, élimination de la récursivité)
- * tri (les principes et les algorithmes récursifs de la plupart des tris, notion de coût)
- * les listes linéaires (approche fonctionnelle, notion de tas, approche procédurale par transformation des fonctions pour une optimisation des algorithmes en place et en temps)

4.2.2 Les 14 semaines suivantes

Seules les sections MIE et TM ont un cours d'informatique durant cette période.

4.2.2.1 Les MIE

cours	21 h
TD	35 h
Machine	42 h

L'objectif de ce cours est de compléter la connaissance de l'étudiant en définissant et en utilisant les structures de donnée plus complexes telles que les automates, les arbres et les graphes. C'est également d'essayer de trouver la structure de donnée la plus appropriée à un problème donné. Les points suivants sont abordés:

- * automate
- * grammaire (spécification syntaxique)
- * arbres (représentation par table, par liste, dynamique; arbres binaires de recherche; parcours; arbres à critère d'équilibre).

- * graphes (exemples; définitions; représentation par matrice, par table, par liste; algorithmes de parcours, de recouvrement, du chemin le plus court ..)

4.2.2.2 Les TM

Cours	2 h
Machine	10 h

L'objectif de ce cours est de comprendre les concepts de base en utilisant le logiciel de C.A.O, SACADO (Système Adaptatif de Conception Assistée et de Développement par Ordinateur) développé par l'équipe du professeur Y. Gardan du LRIM (Laboratoire de Recherche en Informatique de Metz).

Les mécaniciens ont créé une MST CFMAO où l'informatique prend une part non négligeable (Algorithmique, Architecture et système, Base de donnée et C.A.O)

4.3 Le DEUG B deuxième année

cours	20 h
TD	20 h
Machine	20 h

L'objectif de ce cours est de donner des compléments d'algorithmique sur les tableaux à la demande des responsables des deuxièmes cycles de chimie et d'initier les étudiants à l'utilisation des tableaux.

5 EVALUATION

Chaque cours d'informatique est évalué de la manière suivante:

- 2 contrôles écrits (de 2 heures)
- 1 note de projet

La règle de calcul de la moyenne est propre à chaque enseignant (commune pour chaque section). Il en existe deux:

- 1) la note de projet compte comme une note d'écrit
- 2) la note de projet compte comme une note d'écrit si le projet n'est pas rendu ou comme des points ajoutés (ou enlevés) à la moyenne de l'écrit si le projet est rendu.

Remarque: Les petits projets sont corrigés par simple lecture du dossier de programmation rendu. Les plus conséquents nécessitent une vérification sur machine.

6 OBJECTIFS

Les objectifs particuliers à chaque enseignement ont déjà été donnés. L'objectif général est que chaque étudiant de DEUG puisse dire qu'il a abordé des notions plus ou moins importantes d'informatique théorique (système, architecture, algorithmique et structuré de donnée) et qu'il a mis en oeuvre ces notions, au niveau de la machine, par le biais de la programmation Pascal. Il pourra donc exploiter ses connaissances soit dans sa vie professionnelle, soit pour aborder des cours d'informatique plus conséquents d'un deuxième cycle. L'informatique lui apporte également une certaine rigueur, une forme de démonstration constructive (il ne suffit pas de savoir qu'une solution existe encore faut-il pouvoir l'exhiber, la construire), et de plus dans leur travaux de programmation ils doivent aller jusqu'au bout de leurs idées.

7 CONCLUSION

L'informatique, telle qu'elle est enseignée dans les DEUG scientifiques de l'Université de Metz, participe à la formation générale de tous les étudiants. Elle n'est pas reléguée au rang de prestataire de service du syle "informatique=programmer". Elle peut donc sans complexe, à l'Université de Metz, prétendre au titre de DISCIPLINE à part entière.

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 6

L'informatique, discipline fondamentale en Deug A: un exemple à Grenoble

J.-P. BERTRANDIAS, P.-C. SCHOLL
(Univ. Joseph Fourier Grenoble)
Document de travail - Avril 1990

Pour contribuer à la réflexion organisée par SPECIF sur l'informatique en Deug Scientifique, nous illustrons ici l'action entreprise à l'Université Joseph Fourier depuis quelques années. Ce document de travail ne reflète qu'une partie des multiples expériences qui ont été conduites par divers enseignants de l'Université, en Deug A ou en Deug B. Il a été élaboré notamment à partir de textes produits par F. Bertrandias, M Chabre-Peccoud et A.-M. Pastel, et à la suite de réunions de travail avec les responsables des enseignements d'informatique en Deug A.

L'Université Joseph Fourier a reconnu une place significative à l'informatique dans sa maquette de Deug A:

- en première année: 4h annuelles pour tous les étudiants (en 1989-90, 24 groupes de 32 à 36 étudiants, répartis en 6 sections. Les prévisions pour 1990-91 sont de 30 groupes). Nous n'en assurons qu'une partie, faute de moyens humains: 4h pendant un semestre dans 5 des 6 sections, et 3h annuelles dans la 6^{ème}.
- en deuxième année: un enseignement de base de 6h annuelles dans une section où l'informatique est une discipline majeure (en 1989-90, section Mathématiques-Physique-Informatique, 75 étudiants. Prévision pour 1990-91, 100 à 120 étudiants) et un enseignement optionnel de 4h annuelles dans les autres sections (en 1989-90, trois groupes. Prévision pour 1990-91, 4 groupes). En 1989-90, environ 45% des étudiants de deuxième année auront ainsi suivi un enseignement d'informatique.

Les 35 enseignants qui sont intervenus en 1989-90 en Deug A et en Deug B sont en majorité des informaticiens (18 dont 5 moniteurs) ou des mathématiciens (8 dont 1 moniteur) de l'UFR Informatique et Mathématiques Appliquées, mais on doit noter la participation de collègues des UFR de Biologie (2), Chimie (2), Géologie (1) et Physique (4).

1. Eléments de réflexion

Promotion de l'informatique en premier cycle

L'un des facteurs de l'évolution de l'informatique est l'amélioration de son image en tant que discipline fondamentale, que ce soit auprès des industriels (qui sous-estiment l'importance des aspects fondamentaux par rapport aux technologies à évolution rapide), auprès des scientifiques, universitaires notamment (qui, en général, ne perçoivent de l'informatique que ses aspects technologiques si avancés soient-ils), auprès des jeunes (mal informés du fait de la faiblesse de la formation dans le second degré) et auprès des pouvoirs publics (qui s'ils multiplient les actions incitatives du fait des enjeux économiques et sociaux, s'obstinent à ne pas en tirer les conséquences dans les structures de décision).

Face à cette question, et pour agir sur le long terme, il nous semble essentiel de favoriser l'insertion de l'informatique fondamentale dans toutes les formations universitaires. L'introduction progressive d'enseignements d'informatique en second cycle, la mise en place de DESS Informatique Double Compétence ou de formations pour les professeurs du second degré relèvent de cette démarche, mais le premier cycle est le maillon central de cette ligne d'action.

Notre UFR s'est attaquée très activement à ce problème dans les cinq dernières années. Il a d'abord fallu obtenir un nombre d'heures suffisant dans les programmes de premier cycle, valoriser ce type de formation par rapport aux formations spécialisées de 2^e et 3^e cycle, organiser des équipes pédagogiques importantes (du fait du nombre d'étudiants), mais surtout inventer des contenus (en dépassant les idées standard liées à la programmation ou à l'utilisation de logiciels) et les mises en œuvre pédagogiques appropriées.

Insertion de la discipline

Pour pouvoir enseigner l'informatique en Deug A comme discipline fondamentale, il faut d'abord assurer une insertion favorable dans l'environnement universitaire:

Un cadre administratif analogue à celui des autres disciplines

- reconnaissance par les différents conseils (du Premier Cycle, Pédagogique, Scientifique, d'Université, ...),
- mise en place d'un budget suffisant (Université, informatique pour tous, ...)
- obtention de locaux adaptés (salles de TD et salles machines proches les unes des autres, salles de maintenance, de permanence, de réunion),
- horaire global convenable (seuil qu'on peut fixer à 50 heures en 1^{ère} année),
- coefficient suffisant (en principe, correspondant au volume horaire),
- plan concerté sur les deux années (formation générale en 1^{ère} année, formation plus spécialisée en 2^{ème} année dans le cadre de la réforme des Deug).

Une conformité du cadre pédagogique avec celui des autres disciplines fondamentales

- exercices progressifs et bien cadrés,
- épreuves de contrôle en temps limité, comportant des parties théoriques et pratiques, amenant à une moyenne analogue à celle des autres disciplines,
- examen de T.P. individuel.

Une concentration suffisamment forte de l'enseignement

Par exemple, en première année, nous répartissons les 2h annuelles assurées de la manière suivante: 4 heures par semaine sur une période favorable, entre Novembre et Mars, ce qui permet deux rendez-vous différenciés par semaine: 2 heures de TD + 2 heures de programmation devant machine.

Une bonne coordination entre enseignement théorique et expérimentation

Par exemple, en première année, nous articulons l'enseignement autour de l'expérimentation pratique proposée: les notions et techniques introduites en TD sont directement illustrées par les sujets de travaux pratiques et sont immédiatement renforcées en TD après les séances pratiques.

Une reconnaissance par rapport aux autres enseignements d'informatique

Dans notre cas, l'UFR d'Informatique et Mathématiques Appliquées a donné une forte priorité à cette question, notamment en favorisant les affectations de service en premier cycle, la formation de collèges d'autres disciplines et la constitution de groupes de travail.

A propos d'objectifs

L'enseignement de l'informatique pose de multiples défis, par la variété des approches qu'elle permet en matière d'objectifs, de contenus ou de mise en oeuvre pédagogique.

Peut-on dégager des objectifs pour la pédagogie d'une discipline fondamentale, dans le cadre d'un enseignement de masse? C'est une question difficile du fait de la jeunesse de l'informatique, qu'il faut aborder progressivement au fur et à mesure des expériences. C'est dans cet esprit que nous présentons les quelques éléments suivants.

Notre hypothèse est que l'informatique peut apporter des bases culturelles utiles à l'étude des disciplines scientifiques, et que l'enseignement en Deug A doit en développer les aspects fondamentaux, déclenchant éventuellement des vocations en direction de l'informatique.

Dans notre contexte, la cinquantaine d'heures de première année sont une sensibilisation à l'informatique dans laquelle on veut habituer les étudiants à aborder l'informatique de manière "académique" (au sens noble du terme). Ce démarrage permet d'homogénéiser les acquis des étudiants, notamment en prévision de la seconde année. On peut proposer:

- une expérimentation sur l'environnement informatique: matériel, système d'exploitation, langage, ...
- l'élaboration d'une problématique de l'informatique: formulation de problèmes-types, méthodes de réflexion et d'analyse, situations où intervient l'informatique, ...
- une réflexion sur les modes d'expression utilisés en informatique: problèmes, algorithmes, programmes, résultats, ...

En ce qui concerne la seconde année, nous essayons de dégager les principes de base permettant de situer les branches assez différentes que recouvre le terme "informatique", en évitant de tomber dans certains écueils connus:

- un enseignement purement technique (apprentissage d'un langage);
- un enseignement purement théorique (bases mathématiques de l'informatique);
- un enseignement discipline de service (utilisation de logiciels).

Au delà d'une présentation d'éléments techniques, nous pensons qu'un enseignement d'informatique doit être organisé autour des aspects méthodologiques inhérents aux problèmes de conception, d'expression et de validation. Il prépare alors aussi bien

- à une pratique expérimentale qui sera utile dans les enseignements où l'informatique est vue par ses aspects technologiques
- qu'à l'acquisition de fondements dans divers domaines: mathématiques (récurrence, algèbre, logique), langages, modélisation des informations, preuve, complexité, génie logiciel.

Un objectif global est de transmettre une attitude scientifique vis à vis des programmes, des algorithmes, des descriptions d'analyse et de problèmes, ... objets susceptibles d'être manipulés par l'homme, plutôt qu'objets d'exécution par une machine. La motivation première des étudiants est le plus souvent liée à la machine. Il faut la réorienter vers l'activité humaine. Il faut faire comprendre aux étudiants le besoin d'une discipline. Il faut équilibrer leurs goûts et leurs énergies entre les problèmes de conception, la réutilisation adéquate de techniques répertoriées et la mise en oeuvre concrète.

Un élément essentiel dans la structuration de l'apprentissage est une claire perception des divers niveaux de complexité qui caractérisent l'informatique: complexité factuelle, inhérente aux problèmes et aux limites des machines, complexité technique liée à la variété des outils et à leur sophistication, et complexité conceptuelle, imposée par la distance entre l'énoncé d'un problème et une solution informatique. Il faut repérer ces divers types de complexité et placer des lignes directrices simples permettant de leur faire face: classification des types de problèmes, organisation méthodique et hiérarchique des connaissances techniques, mise en évidence de méthodes d'analyse, approche pratique expérimentale.

Les multiples savoir-faire informatiques s'appuient sur la maîtrise des modes d'expression employés, aux divers niveaux d'abstraction considérés, et sur des attitudes rigoureuses et précises, même si elles ne sont pas formelles. On trouve là une difficulté majeure rencontrée par les étudiants qui ont une tendance naturelle à ne s'intéresser au problème d'expression qu'au moment de rédiger un programme dans un langage de programmation particulier (voire devant la machine!). Nous pensons qu'un cours d'informatique doit mettre un accent important sur la notion même de langage, en insistant sur les aspects pragmatiques, au plan conceptuel pour l'expression de la pensée, et au plan technique pour la mise en oeuvre d'une solution.

A propos de mise en oeuvre

La mise en oeuvre pédagogique d'un enseignement d'informatique est affaire délicate notamment en matière d'équilibre entre aspects fondamentaux et pratiques, et de contrôle des activités proposées dans des directions nécessairement multiples: résolution de problèmes, rigueur d'expression, maîtrise des techniques, pratique des machines au travers de la programmation, ...

Pour animer une progression pédagogique qui intègre les divers objectifs ci-dessus, il faut éviter l'énumération fastidieuse des techniques, savoir choisir parmi la diversité des sujets à aborder et renouveler les thèmes d'application. Enfin, il faut d'année en année intégrer les progrès conceptuels et technologiques de l'informatique. C'est ici qu'intervient la notion d'équipe. Les résultats sont d'autant plus riches si les participants peuvent apporter les contributions de domaines variés de l'informatique.

Dans ce qui suit, nous proposons quelques réflexions sur la base de l'expérience menée en première année avec 5 sections de 4 groupes (environ 720 étudiants).

Unité pédagogique

Le nombre d'étudiants intervient sur le nombre de groupes à gérer, le nombre d'enseignants concernés, la diffusion des documents s'ils sont communs et le retentissement de tout problème...

Ce nombre fait ressortir fortement la question de l'unité pédagogique: comment mettre en oeuvre une politique commune à tous les étudiants du Deug A ? Faut-il unifier les groupes ? les sections ? Quel degré de proximité adopte-t-on entre les groupes ?

Si l'on choisit d'organiser l'enseignement par section, par exemple sur la base d'un programme commun, on réduit la lourdeur de la coordination, mais au détriment d'un consensus réel sur l'enseignement. Il devient difficile de comparer les acquis et de faire évoluer les enseignements de manière cohérente d'année en année. On risque de propager ainsi une hétérogénéité de connaissances inévitable du fait de l'indépendance des sections. On peut enfin estimer que l'investissement pédagogique est finalement plus important, par la duplication de certaines tâches et la difficulté de mise en commun de matériel pédagogique.

Nous avons choisi d'une part d'organiser l'enseignement uniquement sur la base de TD et de TP (pas de cours magistral) et d'autre part de coordonner fortement les 20 groupes concernés. Ceci nécessite un grand consensus de l'équipe pédagogique sur les objectifs et les méthodes d'enseignement. Il est notamment fondé sur des documents de référence, pour la coordination entre les enseignants ou destinés aux étudiants. Ces documents doivent concerner la matière, mais aussi chacune des notions ou techniques présentées, la progression des TD et leur conduite, le matériel et son utilisation...

Dans notre contexte expérimental, ces choix ont permis une meilleure appréciation de la démarche envisagée en favorisant une concertation sur la nature des exemples à présenter et en permettant une synthèse immédiate après les expériences de chacun. Cette mise en oeuvre commune évite les déviations produites par la division en plusieurs sections et facilite très sensiblement l'intégration d'enseignants nouveaux et notamment des jeunes (moniteurs).

Le principal écueil réside dans la lourdeur de coordination et de gestion. Les groupes doivent progresser en parallèle avec des points d'appui soigneusement choisis. Malgré cela, les écarts pédagogiques persistent mais peuvent être discutés explicitement.

On doit s'appuyer sur une structure de coordination aussi légère et efficace que possible. On est ainsi conduit à une organisation hiérarchisée car on ne peut coordonner 20 à 25 personnes à la fois en direct. Il faut se donner les moyens de transmettre d'une année à l'autre l'investissement ainsi réalisé, par exemple en définissant un cahier de compte rendu chronologique, une batterie d'exercices ciblés et commentés, une mémoire d'exemples et de sujets, une bibliothèque de référence, ..., le tout étant enrichi progressivement. L'investissement pédagogique correspondant, le travail matériel important de rédaction et de relecture peuvent être répartis sur l'équipe.

Il faut assurer une correction adéquate des TP. On peut s'interroger sur l'opportunité de "corrigés types": ils sont toujours réducteurs de la réalité, car ils ne donnent qu'une vision partielle des multiples approches possibles d'un problème et ne peuvent, pour des raisons matérielles, expliciter suffisamment les raisonnements qui sous-tendent l'élaboration de la (ou des quelques) solution(s) présentée(s).

On peut déjà percevoir que ces conditions de travail peuvent difficilement être considérées comme une norme car il n'est pas dans la tradition de nos universités de définir avec une telle précision les objectifs et les méthodes pédagogiques qu'un groupe s'impose et impose aux différents enseignants, même si le consensus est souvent recherché.

L'hétérogénéité des acquis en informatique

Quelques étudiants (1 à 2%) ont suivi une option informatique au lycée. Cette préformation ne développe pas automatiquement un désintérêt. On pourrait imaginer un aménagement de leur travail. Les autodidactes du BASIC (plus de 20% des étudiants) peuvent introduire le trouble ou au contraire être à l'origine de bonnes questions. L'organisation que nous pratiquons permet de les intégrer rapidement à notre progression pédagogique, de leur faire percevoir la nouveauté de la démarche, et même de récupérer certaines de leurs compétences au profit de tous.

A propos de matériel

L'enseignement de l'informatique en premier cycle s'appuie actuellement sur trois salles équipées en matériel ATARI et EPSON:

- 2 salles de 18 ATARI 520 avec 5 imprimantes ATARI ou EPSON
- 1 salle de 16 ATARI 1040 avec 4 imprimantes.

Le choix de ce type de matériel a été effectué en 1986 sur les bases suivantes :

- matériels de bonnes caractéristiques:
 - mémoire centrale de 512 KO (ou 1024 KO),
 - écran graphique 640x400 d'excellente qualité,
 - système d'exploitation souris - menus déroulants
(de type Macintosh mais 4 fois moins cher !),...
- logiciels classiques disponibles et peu coûteux:
 - langages, tableurs, gestionnaires de fichiers, traitements de texte...
 - logiciels graphiques 2 et 3 dimensions...
- prix du matériel et du logiciel remarquablement bas:
 - (un poste 520 coûtait environ 4000 F).

Ce matériel a été acquis progressivement en remplacement de la dotation en Goupil 3 et Micral et la remplace maintenant complètement. L'homogénéité du parc a été très appréciée par les enseignants lorsqu'elle a été réalisée: simplification de l'utilisation, normalisation de la documentation et des prises de contact, uniformisation des thèmes pouvant servir dans plusieurs enseignements, ...

Les salles d'ordinateurs servent essentiellement aux travaux pratiques et sont d'autre part ouvertes en libre service dans la mesure des possibilités de l'équipe de surveillance. Actuellement, en 1989-1990, la charge en séances encadrées est :

- 24 groupes en A1 (séances de 2 heures sur un semestre),
- 5 groupes en A2 (séances de 2 heures annuelles),
- 2 groupes en A0 (séances de 2 heures annuelles),
- 11 groupes en B2 (séances de 2 heures annuelles).

L'équipe de surveillance assure aussi le chargement des systèmes sur chaque poste au début des séances de Travaux Pratiques, ainsi qu'une aide pour l'organisation des examens de TP.

La sécurité du matériel est assurée par un câble anti-vol (pas de disparition en 4 ans, hormis quelques boules de souris!).

Les langages utilisés actuellement pour le DEUG A sont:

- Pascal PROSPERO : Pascal professionnel très (trop pour le DEUG ?) complet (1 licence par poste de travail).
- Pascal ALICE : Pascal interprété avec un éditeur syntaxique (1 licence de site)
- CL-Plus : LISP proche de Common LISP (1 licence de site).
- sont aussi disponibles : Modula II (domaine public), BASIC ST (fourni avec le matériel), C, Prolog, Assembleurs, Little Smalltalk ...

La dotation accordée récemment (Décembre 1989 : 36 postes) par le Ministère permettra d'une part une homogénéisation sur des machines de bonnes performances (1 MO) permettant de remédier aux difficultés actuelles d'implémentation et d'autre part de mettre à la disposition des enseignants les machines atteintes par la limite d'âge (après révision et gonflement !).

Le raccord récent (Janvier 1990) d'une salle sur le réseau grenoblois GRENET permet l'accès immédiat aux services du Centre de Calcul (grands logiciels de calcul scientifique ou de statistique) ainsi que la connection aux différents réseaux locaux ou nationaux.

2. Une progression en première année axée sur les TP

Modalités pratiques

Durée

En 1989-90, l'enseignement de l'Informatique en Deug A1 dure 14 semaines, à raison de 2 heures de Travaux Dirigés et 2 heures de Travaux Pratiques par semaine.

Matériel

Les trois salles de TP sont équipées en ATARI 520 ou 1040 (18 postes par salle); les étudiants travaillent par binômes. On dispose de 4 imprimantes par salle. Pendant les trois premières séances, on utilise des magnétophones (un par binôme, avec écouteurs). Une des salles est accessible aux étudiants en libre service pendant certains créneaux horaires.

Logiciel

Le langage enseigné est Pascal ; on utilise le système GEM et le Pascal PROSPERO.

Documentation

Chaque étudiant dispose d'un "aide-mémoire Pascal" d'une trentaine de pages, contenant en particulier une carte syntaxique de Pascal; lui est jointe une brève documentation sur GEM et PROSPERO.

Nombre d'étudiants concernés

Cet enseignement est donné à cinq des six sections du Deug A1 de l'Université J. Fourier; chaque section comprend quatre groupes de 32 à 36 étudiants. La 6ème section a un fonctionnement différent, non détaillé ici.

Fonctionnement

Principes

- Les travaux sur machine sont prévus en liaison très forte avec les travaux dirigés:
- ils sont présentés (ou préparés) en TD,
 - leurs thèmes exploitent les notions qui viennent d'être introduites en TD,
 - un retour est fait en TD, après lecture des listings de TP, pour préciser ou rectifier la compréhension de ces notions (retour le plus rapide possible);

ils jouent donc un rôle essentiel dans le déroulement de l'enseignement.

Déroulement

Les premiers sujets proposés sont courts et traités chacun en une séance; par la suite, les sujets tournent autour d'un thème qui demande deux ou trois séances de travail.

Rédaction

On attache de l'importance à la présentation qui doit refléter la structure du programme par une utilisation judicieuse de commentaires, puis de procédures ; on est très exigeant sur l'usage d'une bonne indentation. On demande de choisir des identificateurs descriptifs et d'inclure un lexique explicite.

Contrôle

Le contrôle commun est nécessairement limité: on ne peut pas mettre en épreuve 700 personnes à la fois trop souvent. Outre l'examen final, nous procédons à un examen de TP en fin d'enseignement.

Progression

Les quatre premières semaines

Leur rôle est l'apprentissage simultané:

- de rudiments du langage (notions de syntaxe, type, variable, affectation, entrée, sortie, ...),
- de l'utilisation du matériel et des possibilités données par le logiciel utilisé : l'utilisation pendant les trois premières séances des cassettes d'autoformation (système Telima développé par G. Vivier) et une rédaction très précise des textes de TP permettent à la plupart des étudiants une assez bonne autonomie devant l'ordinateur au bout de ces quatre semaines (frappe d'un texte, enregistrement, duplication, suppression d'un fichier, compilation et exécution d'un programme Pascal, impression d'un fichier, ...).

Textes

Ils montrent de "petits" programmes (minimum, somme, moyenne, utilisation du générateur aléatoire, ...); on demande frappe, compilation et exécution de ces programmes, puis quelques modifications ou extensions.

Les quatre semaines suivantes

Elles sont utilisées pour approfondir les structures algorithmiques fondamentales introduites en TD (instructions conditionnelles, instructions itératives, traitement séquentiel).

Textes

- Calcul de X^n ,
- Modélisation d'une situation concrète (prêt de livres),
- Calcul d'une limite de suite récurrente (la suite : $u_n = (u_{n-1} + A/u_{n-1}) * 0.5$),
- Moyenne des entiers d'une suite tapée au clavier et terminée par une marque,
- Moyenne "olympique" : problème analogue, mais on ne tient pas compte du plus grand et du plus petit entier,
- Positionnement d'un prix : mise en place d'un programme "meneur de jeu", pour faire deviner à un utilisateur un entier "tiré au sort" par le générateur aléatoire.

Commentaires

- Calcul de la "moyenne olympique" des entiers d'une suite : il permet, à partir des cas simples de calcul d'une moyenne, d'un maximum, d'un minimum déjà observés, de franchir un premier niveau de complexité par la nécessité d'une analyse préalable du problème.
- Positionnement d'un prix : il fait franchir une nouvelle abstraction, car il comporte deux traitements séquentiels imbriqués et les éléments des séquences sont plus abstraits (essais et jeux successifs); on a donné le principe détaillé du programme.

Les six dernières semaines

Les TP sont beaucoup moins guidés et laissent une large part à l'initiative personnelle. Le but est l'assimilation des notions de tableau, procédure, fonction. La période se finit par l'examen de TP.

Deux thèmes :

- le "pendu" : construire un programme jouant le rôle de "meneur de jeu" dans le jeu classique qui consiste à deviner successivement les lettres d'un mot,
- la "chenille" : simuler les déplacements d'une chenille (15 cercles ...) sur l'écran, en utilisant les procédures de la bibliothèque graphique pour tracer et effacer les cercles, et le générateur aléatoire pour conduire les déplacements.

Commentaires

- Le "pendu" nécessite la mise en place de traitements séquentiels emboîtés, l'utilisation de tableaux de caractères, de booléens. Le niveau de difficulté est bien supérieur à celui du TP précédent (positionnement d'un prix).
- La "chenille" permet de travailler sur
 - . les structures de données, qui sont imposées (structure de tableau de tableaux),
 - . l'écriture de procédures et de fonctions.

Une grande part d'initiative est laissée aux étudiants pour le scénario du dessin animé. Le niveau d'abstraction nécessaire est bien supérieur à celui des autres TP.

Ces deux derniers thèmes ont montré aux étudiants (même les plus avancés ...) que, sans une analyse sérieuse du problème, ils ne pouvaient parvenir à un programme correct.

Evaluation

Le parcours de cette année bénéficie de l'expérience analogue effectuée en 1988-89. Pour cette année, l'évaluation ne sera possible qu'après le déroulement et la correction des examens; on ne peut donner pour le moment que des impressions individuelles d'enseignants ou d'étudiants.

Dans l'ensemble, l'impression est positive:

- les étudiants ont trouvé que les thèmes proposés étaient à leur portée, même si certains ont eu des difficultés pour mener à terme les deux derniers TP;
- la manière dont ils ont traité les sujets proposés (messages, scénarios originaux, ...) montrent une bonne intégration des thèmes à leur univers culturel;
- la participation aux TP a été bonne.

Il faudra cependant "affiner" cette impression en regardant de manière précise certaines phases de la progression, ainsi que les différences de réactions d'un groupe à l'autre.

Les difficultés des sujets proposés pendant les sept premières semaines ont été bien contrôlées et très progressives; le "pendu" a marqué une discontinuité que certains enseignants ont jugée excessive et d'autres bénéfique ("déstabilisation didactique")... Dans certains groupes, la "chenille" n'a pu "démarrer" sans un apport important de l'enseignant, alors que, dans d'autres, les étudiants souhaitaient manifestement parvenir seuls à "leur" solution.

3. Un exemple en seconde année

Il s'agit de l'enseignement assuré dans une section où l'informatique est l'une des matières dominantes: en 1988-89 (1h30 de cours, 2h de TD, 2h de TP) pour une section de 56 étudiants répartis en 2 groupes, en 1989-90 (2h de cours, 4h de TD) pour une section de 74 étudiants répartis en deux groupes.

Nous présentons d'une part un plan général de ce qui a été expérimenté en 1988-89, et qui peut servir de programme de référence, et d'autre part une mise en oeuvre un peu plus détaillée correspondant à l'année suivante. Les variations d'une année sur l'autre tiennent d'une part à un changement de l'équipe enseignante et d'autre part au souci d'expérimenter diverses formes d'enseignement et de contenus. L'esprit général reste le même.

La démarche consiste à intégrer plusieurs thèmes fondamentaux dans un enseignement unifié centré sur

- les aspects méthodologiques: analyse, variété des niveaux d'abstraction et des formes d'expression, rigueur dans la manipulation des spécifications ou des programmes, liens avec la matérialisation, ...
- et des formes d'expérimentation adaptées: activités concrètes (différentes des sujets classiques de stricte programmation) mettant les étudiants en présence de réalités informatiques (complexité, mode d'expression, matériel).

Programme 1988-89

I. INFORMATIONS

- 1) Désignation - Signification
- 2) Variables: noms - valeur - référence
- 3) Informations élémentaires
 - mémorisation: mémoires centrales - mémoires de masse
 - transmission: série - parallèle
 - traitement: circuits élémentaires et, ou, non - additionneur
- 4) Codage des informations: types élémentaires

II. ORGANISATION DES ORDINATEURS

- 1) Organisation physique - Matériel
 - Mémorisation - Transmission - Traitement
 - Microprocesseurs
 - Codage des instructions - Cycle d'exécution
- 2) Organisation logique
 - Couches logicielles - Métiers de l'informatique
 - Niveaux - Primitives
 - Interfaçage --> langages
 - Histoire des langages de programmation

TD - TP

- révision de programmation Pascal: complexité combinatoire (tables de nombres premiers, tests de primalité).
- auscultation de la mémoire SHARP 1745 (programmation BASIC)

III. LANGAGES

Langages courants / langages de programmation

- 1) Pragmatique
 - fonctions et usage d'un langage: commande, description, spécification
 - niveaux d'abstraction
 - machines virtuelles ou réelles
- 2) Sémantique
 - signification ?
 - Dédution - Réécriture
 - Sémantiques opérationnelle, dénotationnelle, axiomatique
- 3) Syntaxe
 - Notation de Backus-Naur - Diagrammes
 - Exemples: expressions - mini-Pascal
 - Analyse lexicale - Analyse syntaxique
 - Compilation - Interprétation
 - Automates: vérification - traduction

TD - TP

- Editeur avec caractère d'effacement
- Evaluation d'expression arithmétiques

DS: langage de formules chimiques

IV. LANGAGE DES EXPRESSIONS ET DES FONCTIONS

- 1) Définition des fonctions
 - généralisations successives de la notion d'expression
 - variables - contexte
 - évaluation des expressions
 - lambda expressions - Fonctions
 - définitions récursives
 - dénombrabilité - calculabilité
- 2) Programmation fonctionnelle
 - machine: évaluateur
 - composition - choix - récursion
 - langage type: Lisp (pur)
 - S-expressions - fonctions de base
 - Evaluation - fonctions - récursion - ...
- 3) Ambiances de programmation
 - analyse récurrente de problèmes de traitement
 - utilisation fonctionnelle (récursion) / actionnelle (itération)
 - exemples: parcours, recherches, tris

TD - TP : apprentissage de la programmation en Lisp

DS: morpion à 9 cases

V. STRUCTURES DE DONNEES

- 1) Méthodes de construction des types
 - produits: tableaux, enregistrements
 - unions
 - fonctions
 - récursion
- 2) Notations algorithmiques
implémentation en Pascal et en Lisp
- 3) Exemples: séquences - arbres

TD - TP: listes chaînées - arbres en Pascal

VI. PROGRAMMATION ACTIONNELLE

- 1) Affectation - Variables en informatique
- 2) Pointeurs - Structures dynamiques
- 3) Procédures - Passages des paramètres
- 4) Schémas de traitement - Invariants

*TD - TP: programmation en Pascal: complexité logicielle
(ADN, formules chimiques, traitement de textes)*

Examen: langage de description de "patchwork" - réalisation en Lisp et en Pascal

VII. PROGRAMMATION LOGIQUE (prévu mais non fait)

- 1) Logique formelle
 - Calcul des propositions
 - Calcul des prédicats
- 2) Introduction à Prolog

Détails de mise en oeuvre 1989-90

Temps pédagogiques

L'enseignement a été divisé en 4 thèmes:

1. Structure d'un ordinateur, représentation des informations, structures algorithmiques de base (4 semaines).
2. Expression fonctionnelle de l'analyse d'un problème, des algorithmes (12 semaines).
3. Ouverture sur d'autres formes d'expression (5 semaines)
4. Langages, automates et grammaires (5 semaines).

L'algorithmique est utilisée comme fil directeur du cours. En effet de nombreux domaines de l'informatique peuvent être abordés sous forme de conception et de mise en oeuvre d'algorithmes: spécification des problèmes, mise en évidence de divers niveaux d'abstraction au cours de la résolution, choix de formes d'expression adéquates, exploitation de techniques répertoriées pour le passage d'un niveau à un autre.

L'accent est mis sur les méthodes de résolution de problèmes, sur la variété des formes d'expression disponibles, sur la maîtrise des niveaux d'abstraction. Diverses techniques de l'informatique sont ainsi abordées dans un but de sensibilisation et de concrétisation.

Méthodes

L'enseignement comporte 2h de cours et 2 fois 2h de travaux dirigés par semaine.

Une première séance de travaux dirigés (TD d'algorithmique) est destinée à concrétiser les divers éléments du cours au fur et à mesure de son déroulement. Il s'agit essentiellement de maîtriser des méthodes d'analyse et les formes d'expression associées: expression actionnelle, expression fonctionnelle, description d'automates et de grammaires, modélisation d'un système d'information et expression de requêtes relationnelles, choix d'une technique de représentation.

Une deuxième séance de travaux dirigés (TD d'application) est destinée à des études de cas permettant aux étudiants de mettre en pratique les éléments du cours dans le contexte de manipulations concrètes (notamment programmation en Lisp et en Pascal).

Thème 1: Structures algorithmiques de base

Ce thème est développé en cours et en TD d'algorithmique durant les quatre premières semaines. En parallèle, les TD d'applications sont centrés sur l'architecture d'un ordinateur et la représentation des informations autour de manipulations sur un calculateur de poche programmable en Basic.

On s'appuie sur la sensibilisation acquise en première année autour de Pascal. Les diverses notions sont rappelées et formalisées au travers d'une notation algorithmique qui est progressivement présentée autour d'exemples. L'accent est mis sur la différence à tous niveaux entre spécification et réalisation. On sensibilise ici d'une part à la notion de type abstrait, d'autre part à une expression fonctionnelle de l'analyse d'un problème.

1. Informations, actions et opérations primitives

1.1. Exemple: "jours, heures, minutes, secondes"

1.2. Action nommée

1.3. Informations simples, types de base

2. Analyse par cas

2.1. Exemple: "maximum de trois valeurs"

2.2. Schéma d'analyse par cas

3. Abstraction, paramétrisation: actions

3.1. Exemple: "classement de 3 valeurs"

3.2. Actions paramétrées, paramètre donné

3.3. Actions paramétrées, paramètre donnée-résultat

4. Abstraction: fonctions

4.1. Exemple: "maximum de 2 valeurs"

4.2. Expression conditionnelle

4.3. Fonctions à plusieurs résultats

4.4. Exercices

4.5. Fonctions et Actions - Paramètre résultat

5. Abstraction: types nommés

5.1. Exemples: manipulation de "dates", de "segments"

5.2. Un premier constructeur de types: produit de types

5.3. Exemple: "arithmétique sur les fractions"

5.4. Etude de cas: "calendrier grégorien"

5.5. Un deuxième constructeur de type: tableau

TD d'application: organisation physique d'une machine, prise de contact avec la SHARP 1245, quelques mots de BASIC, lecture du contenu de la mémoire, codage.

Référence: [ScP88] P.-C. Scholl, J.-P. Peyrin, Schémas algorithmiques fondamentaux: séquences et itération - Masson 1988 pp 23-77

Thème 2: expression fonctionnelle

On travaille sur le rôle de l'expression fonctionnelle dans l'analyse d'un problème, sur la structuration de l'analyse autour des types d'information rencontrés, et sur la notion de type abstrait (constructeurs, sélecteurs). On introduit la composition récursive et on expérimente autour des types inductifs de base: entiers, séquences, arbres. On termine en montrant que les fonctions peuvent être manipulées comme les autres objets.

Les TD d'algorithmique ont pour but essentiel de développer un comportement systématique dans la définition récurrente de fonctions définies sur des domaines inductifs. On renforce les mécanismes d'analyse descendante.

Les TD d'application ont pour but d'expérimenter la démarche fonctionnelle en utilisant le langage Lisp. On renforce aussi l'idée de niveaux d'abstractions et de type abstrait: Lisp n'est pas typé et ne comporte qu'un seul constructeur de type.

Par ailleurs, on sensibilise les étudiants à une démarche ascendante dans le codage et le test d'un programme, on introduit quelques éléments à propos de langage (étude de la syntaxe de Lisp, notion d'évaluation,...) et on aborde l'idée de coût.

6. Récurrence

- 6.1. Preuve par récurrence sur N
- 6.2. Définition récurrente d'un ensemble
- 6.3. Définition récurrente d'une fonction sur un domaine inductif
- 6.4. Expression récursive d'une fonction
- 6.5. Etapes d'une analyse récurrente

7. Traitement de séquences: schémas d'analyse

- 7.1. Principes de base, découpage de l'information
- 7.2. Modes de découpage classiques d'une séquence
- 7.3. Classes de problèmes
- 7.4. Analyse du traitement d'une séquence en termes d'une séquence intermédiaire
- 7.5. Evaluation séquentielle: fonctions d'accumulation

8. Le type arbre

- 8.1. Définition récurrente des arbres
- 8.2. Analyse guidée par la structure d'arbre

9. Ordre supérieur

- 9.1. Fonctions comme paramètres
- 9.2. Exemple d'application: schémas fonctionnels de traitement de séquence
- 9.3. Notation lambda, valeur d'une fonction
- 9.4. Localité des noms, liaison des noms

TD d'application

- découverte de Lisp à travers d'un exemple complet: calendrier grégorien
- fonctions manipulant les séquences, constructeurs et sélecteurs pour les listes
- mécanisme d'évaluation, problématique de test, sur de petits exemples: échangeur de monnaie, fonction de fibonacci, sous-suites, carré magique.
- représentation des arbres en Lisp: tri en arbre, expressions arithmétiques.
- ordre supérieur: manipulation de schémas fonctionnels sur les séquences: somme d'une suite d'entiers, à partir d'une séquence de caractères.

Références

- [AbS89] H. Abelson, G.J. Sussman, *Structure et Interprétation des programmes informatiques* - InterEditions 89
- [FIH88] A.J. Field, P.G. Harrison, *Functional Programming (chapitres 1 et 2)* - Addison Wesley 1988
- [Knu68] D.E. Knuth, *The Art of computer programming, Voll Fundamental algorithms* - Addison Wesley 1968 (pp11-20)
- [Sch84] P.-C. Scholl, *Algorithmique et représentation des données: Tome 3 Récursivité et arbres* - Masson 1984

Thème 3: ouverture sur d'autres formes d'expression

On travaille avec d'autres formes d'expression (déclarative, impérative) autour du thème *gestion d'un ensemble d'informations*. On veut essentiellement montrer le lien avec l'expression fonctionnelle du thème 2.

En ce qui concerne l'expression impérative, on s'appuie sur les acquis de la première année et on donne des éléments méthodologiques:

- construction systématique d'itérations sur le base de schémas correspondants aux schémas fonctionnels vus dans le thème 2. On montre le lien entre récurrence et itération.
- modélisation d'un ensemble d'informations et description d'un lexique: on s'appuie essentiellement sur la structure de séquence et sur des représentations contiguës à l'aide de tableaux.

En ce qui concerne l'expression déclarative, on procède à une sensibilisation autour du support concret des primitives de l'algèbre relationnelle et du langage SQL.

Les TD sont centrés sur des exemples de gestion d'ensemble d'informations. Ils introduisent quelques éléments de modélisation et sensibilisent à des questions telles que hiérarchie des informations, redondance d'informations, sécurité vis à vis des modifications, problèmes d'intégrité,... Ces exemples sont traités concrètement soit en Pascal, soit en SQL, de manière à illustrer des formes d'expression de niveaux différents.

10. Analyse itérative

- 10.1. Formes de composition itérative
- 10.2. Expression itérative du traitement de séquences
- 10.3. Représentation d'une séquence à l'aide d'un tableau
- 10.4. Représentation d'une séquence à l'aide d'un fichier séquentiel
- 10.5. Récurrence et itération

11. Gestion d'un ensemble d'information: introduction aux bases de données

- 11.1. Objectifs d'une base de donnée et définitions
- 11.2. Le modèle relationnel
- 11.3. Systèmes de gestion de fichiers, bases de données et systèmes d'information

TD d'application:

programmation en Pascal et en SQL d'un exemple de gestion d'ensemble d'informations: cinémascope

Références

- [Dat77] Date, *An introduction to Database Systems* - Addison Wesley 1977
- [DeA82] C. Delobel, M. Adiba, *Bases de données et systèmes relationnels* - Dunod 1982
- [Gar82] Gardarin, *Bases de données: les systèmes et leurs langages* - Eyrolles 1982
- [ScP88] P.-C. Scholl, J.-P. Peyrin, *Schémas algorithmiques fondamentaux: séquences et itération* - Masson 1988 pp 78-171

Thème 4: Langages et traduction: une introduction

On s'appuie sur l'expérience linguistique de l'année (notation algorithmique, basic, lisp, pascal, relationnel, sql) pour renforcer la notion de langage. On place quelques idées sur les outils et méthodes élaborés pour la définition et la traduction de langages informatiques. Sur le plan des méthodes, on illustre une démarche de résolution de problèmes consistant à définir un langage pour exprimer le problème considéré et à utiliser les techniques issues de la compilation pour construire une solution.

12. Divers aspects d'un langage

- 12.1. Définitions: pragmatique, sémantique et syntaxe
- 12.2. Fonction et usages d'un langage
- 12.3. A propos de la définition d'un langage
- 12.4. A propos de traduction: le processus de compilation

13. Grammaires: spécification d'une syntaxe et analyse syntaxique

- 13.1. Définition d'une grammaire
- 13.2. Stratégies d'analyse

14. Automates: analyse lexicographique

- 14.1. Langages réguliers
- 14.2. Automates d'états finis

TD d'application:

définition d'un petit langage de description de figures géométriques et expérimentation autour du processus d'analyse des phrases et de production des figures.

Références

- [Wat89] D. Watson, *High Level Languages and their compilers* - Addison Wesley 1989
- [ASU89] A. Aho, R. Sethi, J. Ullman, *Compilateurs: principes, techniques et outils* - Interéditions 89 (ch 1 et 2)

4. Générique 1989-1990

Le générique qui suit permet de se rendre compte de l'énergie humaine mise en oeuvre en Deug A à l'Université Joseph Fourier.

Enseignement 1^o Année

Responsable: Monique Chabre-Peccoud

Section A11, responsable: Philippe Genoud
Ph. Genoud (2 groupes), A. Labiad, J. Lemordant

Section A12, responsable: Anne-Marie Pastel
J. Chalemont, Ph. Chol (moniteur), D. Fort (moniteur), X. Nicollin (moniteur), A.M. Pastel

Section A13, responsable: Michel Lévy
L. Biard, M. Eberhard, M. Guillerault, M. Lévy

Section A14, responsable: Monique Chabre-Peccoud
J. Bouteillon (chimiste), M. Chabre-Peccoud, A. Labiad, P. Raymond (moniteur)

Section A15, responsable: F. Bertrandias
F. Bertrandias, J.-L. Monin (physicien), S. Pinchinat (moniteur), F. Tcheou (physicien)

Section A16, responsable: Jean Chalemont
J. Chalemont (2 groupes), Anne-Marie Pastel (2 groupes)

Méthodes audio-actives (Télisma): Gérard Vivier

Enseignement 2^o Année

Section Maths-Physique-Informatique, responsable P.-C. Scholl
M.-C. Fauvet, F. Lagnier, J. Mailfert, P.-C. Scholl

Autres Sections (3): modules optionnels
O. Baudon, S. Bensalem, G. Kuntz, G. Vivier

Salles de TP - Mise en place des horaires et préparation des salles
Responsable : Jean-Pierre Bony
Equipe des stagiaires TUC : A. Durrani, A. Munier, V. Pral, L. Bellahmar, F. Gentil

Libre Service - Aide aux étudiants : Jean-Pierre Bony

Entretien du matériel : Jacques Meyer

Budget - Matériel - Logiciel - Locaux : Jean-Paul Bertrandias

sans oublier les étudiants (dont on ne peut évidemment pas donner la liste ici!)

Journées SPECIF 1990
Nantes - 27, 28 et 29 Mars 1990

Annexe 7

Bibliographie

J.P. BERTRANDIAS (Univ. Joseph Fourier Grenoble)
Projet - Avril 1990

OUVRAGES GENERAUX

L. GOLDSCHLAGER - A. LISTER
Informatique et algorithmique InterEditions (1986)

ALGORITHMIQUE

H. ABELSON - G. SUSSMAN
Structure et interprétation des programmes informatiques InterEditions (1989)

A. AHO - J. HOPCROFT - J. ULLMAN
Structures de données et algorithmes InterEditions (1987)

J. ARSAC
Les bases de la programmation Dunod (1983)

R. BACKHOUSE
Construction et vérification de programmes Masson (1989)

P. BERLIOUX - P. BIZARD
Algorithmique (2 tomes) Dunod (1989)

J. BIONDI - G. CLAVEL
Introduction à la programmation (3 tomes) Masson (87 - 89 - 85)

J.C. BOUSSARD - R. MAHL
Algorithmique et structures de données Eyrolles (1983)

G. BRASSARD - P. BRATLEY
Algorithmique - Conception et analyse Masson (1987)

J. COUKTIN - I. KOWARSKI
Initiation à l'algorithmique et aux structures de données (2 tomes) Dunod (87)

A. DUCRIN
Les bases de la programmation (2 tomes) Masson (1984)

C. FROIDEVAUX - M.-C. GAUDEL - M. SORIA
Types de données et algorithmes McGraw-Hill (1989)

A. GRAM
Raisonnement pour programmer Dunod (1986)

GREGOIRE
Informatique - Programmation (3 tomes) Masson (85-88-90)

J.F. LAURENT
Initiation à l'analyse et à la programmation Dunod (1982)

C. PAIR - R. MOHR - R. SCHOTT
Construire les algorithmes

Dunod (1988)

P.C. SCHOLL - J.P. PEYRIN
Schémas algorithmiques fondamentaux

Masson (1989)

R. SEDGEWICK
Algorithmes

InterEditions (1990)

PROGRAMMATION

W. ARMSBURY
Data Structures : from arrays to priority queues

Wadsworth (1985)

J. BARNES
Programmer en Ada

InterEditions (1988)

M. de CHAMPLAIN
Standards, style et exercices en C

Dunod (1986)

P. COUSOT
*Introduction à l'algorithmique numérique et
à la programmation en Pascal*

McGraw-Hill (1988)

C. DELANNOY
Exercices en Turbo Pascal

Eyrolles (1989)

F. GIANNESINI - H. KENOUI - R. PASERO -
M. VAN CANEGHEM
Prolog

InterEditions (1985)

P. GROGONO
La programmation en Pascal

InterEditions (1984)

S. HARBISON - G. STEELE
Langage C

Masson (1989)

T. HASEMER
LISP

InterEditions (1985)

B. KERNIGHAN - D. RICHTIE
Le langage C

Masson (1989)

O. LECARME - J.L. NEBUT
Pascal pour programmeurs

MacGraw-Hill (1985)

F. MADAULE - L. PAGET - M.-M. POC - O. URTASUN
Programmes commentés en Turbo-Pascal

InterEditions (1988)

M. MARGENSTERN
Langage Pascal et logique du premier ordre (2 tomes)

Masson (1989 - 1990)

G. MASINI - A. NAPOLI - D. COLNET - D. LEONARD -
K. TOMBRE
Les langages à objets

InterEditions (1989)

B. MEYER
Conception et programmation par objets

InterEditions (1990)

B. MEYER - C. BEAUDOIN
Méthodes de programmation

Eyrolles (1978)

- J. RIVIERE
Turbo-Pascal et son environnement Dunod ()
- B. STROUSTRUP
Le langage C++ InterEditions (1982)
- L. STERLING - E. SHAPIRO
L'art de Prolog Masson (1989)
- D.F. STUBBS - N.W. WEBER
Data Structures with Abstract data types and Pascal Brooks/Cole (1985)

- P. WINSTON - B. HORN
LISP InterEditions (1990)

ARCHITECTURE - SYSTEMES D'EXPLOITATION

- M.J. BACH
The design of the Unix operating System
(existe en traduction) Prentice Hall (1986)
- J. BEAUQUIER - B. BERARD
Systèmes d'exploitations - Concepts et algorithmes McGraw-Hill (1989)
- S. BOURNE
Le système UNIX InterEditions (1985)
- CORNAFION
Systèmes informatiques répartis Dunod (1981)
- M. GRIFFITHS - M. VAYSSADE
Architecture des systèmes d'exploitation Hermès (1988)
- V. HAMACHER - Z. VRANESIC - S. ZAKY
Structure des ordinateurs McGraw-Hill (1985)
- B. KERINIGHAN - R. PIKE
L'environnement de programmation UNIX InterEditions (1986)
- S. KRAKOWIAK
Principes des systèmes d'exploitation des ordinateurs Dunod (1985)
- J.-A. MONTAGNON - E. PICHAT
Architecture des ordinateurs (3 tomes) Masson (86-87-89)
- L. NOLIN
Matériel et logiciel (2 tomes) Masson (1988 - 1988)
- J.M. RIFFLET
La programmation sous UNIX McGraw-Hill (89, 2° éd)
- A. TANENBAUM
Architecture de l'ordinateur InterEditions (1987)
Les réseaux. Architectures, protocoles, applications InterEditions (1990)
Les systèmes d'exploitation InterEditions (1989)

LANGAGES - COMPILATION

- A. AHO - R. SETHI - J. ULLMAN
Compilateurs InterEditions (1989)
- R. SETHI
Programming languages - Concepts and constructs Addison-Wesley (1989)
- D. WATSON
High-level Languages and their Compilers Addison-Wesley (1989)

BASES DE DONNEES

- M. ADIBA - C. DELOBEL
Bases de données et systèmes relationnels Dunod (1982)
- C.J. DATE
An introduction to database systems Addison-Wesley (1977)
- GALACSI
Les systèmes d'informations Dunod (1984)
Comprendre les systèmes d'informations Dunod (1984)
Conception des bases de données Dunod (1984)
- G. GARDARIN
Bases de données : les systèmes et leurs langages Eyrolles (1982)
SGDB relationnels : analyses et comparaison de bases de données Eyrolles (1989)
- S. MIRANDA - J.M. BUSTA
L'art des bases de données (3 tomes) Eyrolles (1990)

GENIE LOGICIEL

- O. LECARME - M. PELLISSIER
La transportabilité du logiciel Masson (1989)
- D. PARTRIDGE
Apports de l'intelligence artificielle au génie logiciel Masson (1989)
- M. THORIN
Génie logiciel Masson (1984)

MATHEMATIQUES de l'INFORMATIQUE

- J.G. BROOKSHEAR
Theory of Computation : Formal Languages, Automata, and Complexity Benjamin-Cummings (89)
- M. GAREY - D. JOHNSON
Computers and Intractability : A Guide to the theory of NP-Completeness Freeman (1969)
- A. SALOMAA
Computation and Automata (existe en traduction) Cambridge U. Press (85)
- S.A. WIITALA
Discrete Mathematics : a unified Approach MacGraw-Hill (1987)

INTELLIGENCE ARTIFICIELLE

- J.P. DELAHAYE
Outils logiques pour l'intelligence artificielle
Eyrolles (1986)
- H. FARRENY - M. GHALLAB
Éléments d'intelligence artificielle
Hermès (1987)
- M. GENESERETH - N. NILSSON
Fondements logiques de l'intelligence artificielle
InterEditions (1990)
- M. GRIFFITHS
Techniques algorithmiques pour l'intelligence artificielle
Hermès (1986)
- J.L. LAURIEPE
Résolution de problèmes par l'homme et la machine
(2 tomes)
Eyrolles (1987 -1988)
- G.F. LUGER - W.A. STUBBLEFIELD
Artificial Intelligence and the design of Expert Systems
Benjamin-Cummings (89)
- P. WINSTON
Intelligence artificielle
InterEditions (1988)

DIVERS

- D. HAREL
The Science of Computing
Addison-Wesley (1989)
- D. HOFSTADTER
Gödel, Escher, Bach
InterEditions (1985)
- R. MOREAU
Ainsi naquit l'informatique
Dunod (1981)

DIVERS

- . Université d'Eté de CIRILLE
- . Couverture du Bulletin de l'Informatique n° 23
(diffusion gratuite)
- . Appel de cotisations.

CIRILLE regroupe des enseignants-chercheurs et chercheurs en Informatique, Logique et Mathématiques discrètes des établissements suivants :

Université Claude-Bernard - Lyon

INSA de Lyon

Université Lyon III

Ecole Centrale de Lyon

Ecole Nationale Supérieure des Mines de Saint-Etienne

Le centre comprend 150 chercheurs dont 22 sont des Professeurs d'Université ou assimilés. Les équipes de CIRILLE travaillent sur 26 opérations de recherche réparties en 7 thèmes :

- Ingénierie de la Connaissance
- Informatique de Production
- Génie logiciel
- Infographie et Imagerie
- Informatique Théorique
- Logique
- Mathématiques discrètes

La recherche développée par CIRILLE est à la fois théorique et pratique puisque de nombreuses opérations de recherche sont soutenues par des contrats publics ou privés.

OBJECTIF de l'Université d'Eté

En Intelligence Artificielle, les applications doivent traiter des objets et des processus de plus en plus complexes, passant de la simple évaluation de règles à de véritables mécanismes de raisonnement. En effet, la gestion des grosses bases de connaissances, les applications avancées de bureautique, de robotique et de CAO nécessitent non pas seulement le traitement, l'activation et l'apprentissage de quelques dizaines de règles, mais l'usage de mécanismes puissants de déduction et de raisonnement logique, flou et géométrique de plus en plus poussés.

L'objectif de cette 4ème Université d'été sera de dresser un panorama des différentes méthodes de raisonnements utilisées en Intelligence Artificielle ainsi que leurs applications.

Celle-ci fait suite aux précédentes Universités d'Eté de Lyon, en IA, sur les thèmes des "Logiques de l'IA" en Juillet 1987, de l'"Organisation et du Traitement des Connaissances en IA" en Juillet 1988 et des "Réseaux Connexionnistes en Informatique : Méthodes et Applications" en Juillet 1989.

PUBLIC CONCERNE

Cette Université d'Eté s'adresse tout particulièrement aux Ingénieurs, Chercheurs et Enseignants concernés par les nouvelles méthodes en Informatique et en Intelligence Artificielle.

RENSEIGNEMENTS PRATIQUES

LIEU : IUSA de Lyon - Amphithéâtre G. BERGER
20, avenue Albert-Einstein
69621 - VILLEURBANNE CEDEX

DROITS D'INSCRIPTION :

Tarif : 4000 frs *

Universitaire : 1000 frs *

Membre de CIRILLE : 400 frs

* incluant les repas de midi et dîner officiels

HEBERGEMENT :

Un hébergement de type " Cité Universitaire " est possible pour une somme de 110 frs par jour
à verser à l'inscription
(chambre + petit déjeuner).

RENSEIGNEMENTS : CIRILLE

R. LAURINI ou Y. PAGNOTTE Dépt. Informatique IUT - A
43, bd du 11 Novembre 1918
69622 - VILLEURBANNE - CEDEX
Tél : 78.94.88.52 ou 78.94.88.47
Télécopie : 78.93.51.95
EARN : LAURINI AT FRMOP11.BITNET

METHODES DE RAISONNEMENT EN INTELLIGENCE ARTIFICIELLE

Lundi 9 Juillet 1990

Matin

- Ouverture et introduction. Robert LAURINI (IUT A/INSA)
- Rappels sur les systèmes experts et les moteurs d'inférence. Etienne PICHAT (UCBL)

Après-Midi

- Psychologie cognitive en raisonnement. Jean Michel HOC (Paris)
- Mécanisme de déduction. Jacques KOULOUMDJIAN (INSA)

Mardi 10 Juillet 1990

Matin

- L'usage des métaconnaissances dans le raisonnement. Jean Marc FOUET (UCBL)
- Raisonnement par logiques non-monotones. Iufsa ITURRIOZ (UCBL)

Après-Midi

- Raisonnement avec d'autres Logiques. Laurence CHOLLY (Groupe LEA SOMBE CERT Toulouse)
- Méthodes de raisonnement par la modélisation en biologie. Alain PAYE (UCBL)

En soirée

- Réception à la Maîtrise de Lyon.

Mercredi 11 Juillet 1990

Matin

- Introduction au raisonnement géométrique. Robert LAURINI (IUT A/INSA)
- Raisonnement spatial pour la modélisation et la planification en robotique. Pierre TOURMASSOUD (Laboratoire CGE de Marcoussis)

Après-Midi

- Planification des mouvements en robotique. Quand les robots ne sont pas des pianos. Jean Paul LAUMOND (LAAS de Toulouse)
- Raisonnement spatial en géométrie. Laurent BUISSON (Laboratoire Artémis IMAG)

En soirée

- Repas

Jeudi 12 Juillet 1990

Matin

- Raisonnements flous et possibilistes. Didier DUBOIS (IRIT Université Paul SABATIER Toulouse)
- Raisonnement dans le domaine médical. Jean-Christophe BUISSON (IRIT II Université Paul SABATIER Toulouse)

Après-Midi

- Raisonnement numérique symbolique. Henri PALAMBONDRAINY (INRIA)
- Bilan et Clôture. Yves PAGNOTTE (IUT A Lyon I)

ISSN 0291-5413

BULLETIN

N°23

of INFORMATION
Approche & Applications
TECHNOLOGIQUE



© 1995 by the publisher
All rights reserved

APPEL DE COTISATION SPECIF 1990

L'Assemblée Générale de l'Association a décidé de maintenir la cotisation SPECIF à 100 Francs pour l'année 1990.

Tous les adhérents de 1989 (et les autres) sont donc invités à transmettre leur règlement, soit par l'intermédiaire de leur correspondant, soit directement au trésorier à l'adresse suivante

SCHNEIDER Michel
Laboratoire d'Informatique
Complexe des Cézeaux
63177 - AUBIERE CEDEX

Prrière de bien vouloir retourner la fiche ci-dessous en cas de nouvelle adhésion ou de changement d'adresse.

NOM :Prénom :

FONCTION(*): Grade :

Adresse complète pour recevoir le courrier de SPECIF (professionnelle de préférence) :

.....
.....

Téléphone :

Etablissement administratif de rattachement (libellé uniquement) :

.....

Laboratoire (libellé uniquement) :

Zone de rattachement :

(*) enseignant, chercheur,....

12

7